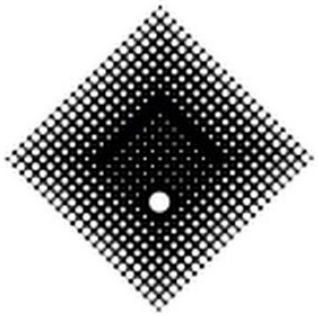
Vegetation High-Impedance Fault Detection and Characterization using Machine Learning



Douglas Pinto Sampaio Gomes

Institute for Sustainable Industries & Liveable Cities

Victoria University

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

College of Engineering & Science

July 2020

This thesis is dedicated to all who altruistically devoted time to the dissemination of free information online, propagating knowledge that greatly inspired this work.

Declaration

I, Douglas Pinto Sampaio Gomes, declare that the PhD thesis entitled 'Vegetation High-Impedance Fault Detection and Characterization using Machine Learning' is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

20th February, 2020

Acknowledgements

The Victorian Government, through its Department of Economic Development, Jobs, Transport and Resources (abolished in 2018), should be first acknowledged as responsible for this thesis. It not only partially funded the candidature as part of its Victorian Latin American Doctoral (VLAD) scholarship program, but was also responsible for funding the massive vegetation ignition project that resulted in the data supporting this thesis. Victoria University deserves acknowledgements as it also partially funded the candidate scholarship and provided all the infrastructure and research training needed for the completion of this work.

This research project would not exist without the efforts of the principal supervisor: Dr Cagil Ozansoy. He was the one first proposing the idea to use the data from vegetation ignition experiments, resulting in all the findings presented here. His guidance and supervision allowed the candidate to freely work and pursue methodological approaches that considerably benefited this research. Thanks also should be given to the project's associate supervisor, Dr Anwaar Ulhaq, which provided motivation and guidance through his expertise. Moreover, the participation of Dr José Carlos de Melo Vieira Júnior deserves acknowledgement for his valuable mentoring and first introducing the candidate to this PhD opportunity and supervisor.

The candidate would like to acknowledge two particular people who provided everlasting support and love throughout the whole candidature process throughout their roles as mother and partner: Dulce Pinto Sampaio and Elise Oliveira Schweig. Close friends made in Melbourne also acted as incredible support, keeping the candidate sanity in check with care and love. In particular, Bojana Klepač, which was always present with a tremendous level of care and support in our many lunch and coffee meetings. For their supporting friendship, thanks are also given to Johnny Ko, Susan Jankovic, Lavern Nyamutswa, and Nuchrapon Liangruenrom.

Finally, all supporting staff at Victoria University, including teaching, administrative, and facilities staff, should be acknowledged as contributing to this thesis. Thanks to all the colleagues for inspiring discussions, in particular, Dr Seyed Morteza Alizadeh, Jiahua Du, Bassam Saleh, Mahamudul Hasan, and Sheikha Variz.

Abstract

Vegetation High-Impedance Faults (VHIFs) are relevant and under-addressed power distribution system disturbances. They are low-energy events, represented by the contact between power lines and nearby vegetation, that are not detected by traditional protection devices. Despite not harmful to power equipment, they can ignite fires in vegetation with great potential to life and property damage. After devastating HIF-related fires in 2009, the Victorian Government found the lack of technical solutions to prevent similar disasters and funded a vegetation ignition testing program to foster further research. It staged hundreds of VHIFs that generated the data pertained to this thesis.

In the related literature, High-Impedance Faults (HIFs) comprise an extensive research field, but few works are solely dedicated to studying VHIFs. Although generally treated as a single problem, different high-impedance conducting surfaces introduce significant variance in faults' characteristics and behaviours. For these reasons, the staged VHIFs recordings represent a niche type of faults having specific behaviours with significant potential for insights regarding phenomenon characterization.

The main contributions from this thesis result from using the staged VHIF data to address the knowledge gaps related to its characterization and detection method. Initial investigations presented the likely presence of discriminative features in the signals' high-frequency (HF) spectrum. The results gave confidence for the production of a machine learning-based VHIF classifier, conceptualized and discussed as part of a potential detection method. Subsequently, the existence of discriminative information and invariance in the HF signals was proved with the application of renowned signal representation techniques and machine learning algorithms. A study regarding the importance of using HF signals was also performed to support the chosen approach when conceptualizing the classifier. It led to the finding that although the accessibility of such signals might be not optimal, they may be imperative for an effective VHIF detection method. To deflate some of the potential implementation concerns, a low-cost, proof-of-concept prototype was produced, attesting the capabilities of real-time classification. Lastly, an unsupervised learning technique was used to capture some of the convoluted and complex fault signatures in the time domain. The found patterns led to insights about VHIF behaviour and signatures signals that resulted in more detailed phenomena characterization.

Publications from this thesis

1. D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "High-frequency spectral analysis of high impedance vegetation faults on a three-wire system," in 2017 Australasian Universities Power Engineering Conference (AUPEC), 19-22 Nov. 2017 2017, pp. 1-5, doi: 10.1109/AUPEC.2017.8282456.

2. D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "High-Sensitivity Vegetation High-Impedance Fault Detection Based on Signal's High-Frequency Contents," (in English), IEEE Transactions on Power Delivery, vol. 33, no. 3, pp. 1398-1407, Jun 2018, doi: 10.1109/Tpwrd.2018.2791986.

3. D. P. S. Gomes, C. Ozansoy, A. Ulhaq, and J. C. de Melo Vieira Júnior, "The effectiveness of different sampling rates in vegetation high-impedance fault classification," Electric Power Systems Research, vol. 174, p. 105872, 2019/09/01/ 2019, doi: 10.1016/j.epsr.2019.105872.

4. D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "Vegetation High-Impedance Faults' High-Frequency Signatures via Sparse Coding," IEEE Transactions on Instrumentation and Measurement, pp. 1-1, 2019, doi: 10.1109/TIM.2019.2950822.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

ADC	Analog-to-Digital Converter
AER	Australian Energy Regulator
ANN	Artificial Neural Network
API	Application Programming Interface
CCVT	Coupling Capacitor Voltage Transform
CNN	Convolution Neural Network
DAC	Digital-to-Analog Converter
DT	Decision Tree
EMD	Empirical Mode Decomposition
EMI	Electromagnetic Interference
EMTP	Electromagnetic Transient Program
FPGA	Field-Programmable Gate Array
FT	Fourier Transform
GI	Gini Impurity
HF	High-Frequency
HIF	High-Impedance Fault
HV	High-Voltage
IP	Intellectual Property

IQR	Interquartile Range
KNN	k-Nearest Neighbour
LF	Low-Frequency
MF	Magnetic Field
ML	Machine Learning
MM	Mathematical Morphology
MRA	Multi-Resolution Analysis
NER	Neutral Earthing Resistor
PBSP	Powerline Bushfire Safety Program
PCA	Principal Component Analysis
PD	Partial Discharge
PLC	Powerline Communication
PSD	Power Spectrum Density
R&D	Research & Development
REFCL	Rapid Earth Fault Current Limiter
RMS	Root Mean Square
SISC	Shift-Invariant Sparse Coding
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
SWER	Single-Wire Earth Return
TACS	Transient Analysis of Control Systems
VHIF	Vegetation High-Impedance Fault
WPA	Wavelet Packet Analysis
WT	Wavelet Transform

Chapter 1

Introduction

Faults on power distribution systems are irregular operating conditions involving a power system equipment failure. The fault events are traditionally classified into two categories: short- and open-circuit faults. Short-circuit faults can result from degradation of insulation or an overvoltage event where current flows through a non-desired, usually low-impedance, alternative path. Open-circuit faults are characterised by the interruption of the load current and lack of fault current. Conductors that break but remain insulated, creating service interruption, are the best example of open-circuit faults.

Short-circuit faults usually receive more attention since they are much more likely to damage equipment. If not addressed, their fault current can exceed the rating of power equipment such as busbars, transformers, and cables, potentially inflicting extreme thermal damage. Short-circuits are also more likely due to the sheer number of possible scenarios. They may occur between conductor and earth, between phases, and between phases and earth; the possible cause may be lighting strikes, accumulation of snow, strong winds, floods, equipment failure as in transformers, machines, reactors, or human failure. On overhead systems, in particular, 80-90% of short-circuit faults occur on the power lines while the rest tend to take place on substation equipment and busbars [1]. This figure is not much surprising since the conductors in most overhead lines are naked bare wires supported by insulators. The lack of insulation makes conductors vulnerable to any current-conducting surface that might come into contact with a conductor. Common scenarios of overhead lines suffering from short-circuits include conductors breaking and falling to low-resistance surfaces and pole insulation failure.

Between the short- and open-circuit fault categories exists a disturbance that blurs the line between these classifications — High-Impedance Faults (HIFs). Short-circuit faults are characterised by hazardous large fault currents generated by the alternative low impedance path. They are easily detected due to their substantial effects on the currents of the system by protection devices such as overcurrent or zero sequence relays. If such a

conducting path has a high impedance, alternatively, the resulting fault current magnitude may be relatively small in comparison to the system nominal current. Consequently, as the current magnitude may remain under the maximal nominal values, HIFs diverge from the perceived category of short-circuit faults because they do not pose the same thermal hazardous stresses. However, as they might not interfere with the functionality of the system, HIFs also do not fit the open-circuit fault category. In HIF occurrences where there is no conductor breakage, the load current is not interrupted, and no trivial changes can be perceived in the signals at the substation level. These seemly non-threatening characteristics of HIFs was probably the reason why they were neglected until the 70s [2], where evidence of their problems started to stack up.

High-impedance faults are the phenomena studied in this thesis, but specifically, the ones created by the contact between powerlines and vegetation. As it will be apparent in the next chapters, an attempt to address all possible HIF types at once is likely to be a misguided task. A considerable amount of evidence is presented to make a case for further dividing HIFs into more specific categories like the ones involving vegetation. The summarised point for this specialisation approach is that addressing all types of HIFs comprise a much more complex problem requiring large amounts of data. In fact, addressing only vegetation HIFs (VHIFs) is already a complex enough problem that demands sizeable data and possibly the use of machine learning algorithms. The following chapters discuss how this complexity may actually be the reason why there is no strong consensus in the literature on the optimum way to address HIFs.

Beyond problem definition, this chapter also describes vegetation HIFs in the Australian local context, the significance of addressing these faults, and the goals this thesis set to achieve to fulfil the potential contribution to knowledge.

1.1 High-impedance faults and vegetation

Despite not representing a threat to power equipment, HIFs are still dangerous disturbances due to their elusive behaviour. They became critical disturbances in power distribution systems when their potential to create safety risks and fire hazards was fully realised. For example, a HIF given by an energised conductor that breaks and falls to the ground can be sustained for an extended period as it goes undetected by protection devices. An interesting work [3] attested this fact by describing interviews with power line crews. They stated that around one-third of broken conductor faults were still energised when the crews reached the fault location. Such an alarming figure, however, only considers broken conductor occurrences that often leads to service discontinuity and consequent reports from customers. For HIFs given by contact with vegetation, service discontinuity may

never happen as they can form without conductor breakage, thus likely never being noticed. Vegetation HIFs are indeed likely to develop into short-circuits with time [4]. Nevertheless, the period when a VHIF develops into a low-impedance fault (before trivial detection) is where the process of combustion and charring of the vegetation happens. This process can result in the ignition of fires, posing a threat to human life and potentially resulting in significant financial damages [5, 6]. Therefore, if VHIFs are not detected and addressed before they develop into short-circuits, it is unlikely that their associated fire risk can be mitigated.

Since the seminal works that established the field in the 80s [7–9], HIFs have been loosely defined as faults with current magnitudes lower than the pickup threshold of traditional overcurrent devices. Their investigation started with the study of broken conductors under the field of ground/earth faults. As a fair share of HIFs will eventually develop into short-circuits, ground overcurrent protection was the primary mechanism responsible for addressing them. However, as most three-phase systems are not perfectly balanced, ground overcurrent devices are adjusted to tolerate minor unbalancing; few amperes of tolerance are already enough for HIFs to occur without being detected [10]. The evidence that a considerable percentage of these faults would go undetected by protection devices was only published in 1982 [7]. The following publications and growing interest on the topic led to the realisation that simple measurements such as current amplitude and system unbalancing would not suffice for HIF detection (especially for solidly or low-impedance grounded systems). Hence, researchers started looking for sources of reliable predictive information in the fault signals that could point to HIF occurrences.

Due to their hazardous nature, HIFs became widely researched power systems disturbances [2, 11, 12]. Aiming at understanding HIFs, the findings from the related investigations led to more clarity on the nature of the disturbance but often disagreed with each other. Although most proposed methods have particular characteristics due to the constant need of claiming novelty, some HIF classic features are generally accepted:

- HIF currents have an impulsive nature that results in increases on signals high-frequency components in large bandwidths [4, 7, 8].
- It is common to observe a higher density of impulses near voltage zero-crossings. The voltage needs to meet a threshold value to break the surface dielectric barrier, generating current discontinuities [4, 13, 14].
- The relatively small current magnitude can significantly vary in between power frequency cycles, pointing them to have a certain level of randomness [7, 8, 15].
- Electric arcs are commonly formed between the conductor and high-impedance surface due to air gaps that separate them [7, 9].

- There is a period where the fault current grows to a maximum local value (build-up), lasting for tens of cycles, which is followed by a few cycles where it ceases to grow (shoulder) [16–18].
- The fault current and behaviour will heavily depend on the conducting surface [14, 15, 19–26].

The most relevant observation for this thesis comes from the last noted point regarding the relationship between fault behaviour and conducting surface. Between all the works discussed in chapter 2, few try to discriminate between fault conducting surfaces such as grass, tree branches, gravel, asphalt, and sand. Detection methods are often proposed with the alleged ability to detect all HIFs, despite the conducting surface. Nevertheless, since seminal works in this field, authors have already stated that it is unlikely that a single method would be able to detect all types on HIFs [10]. After analysing numerous publications from the field of HIF detection, one could argue that it is likely that the scarcity of HIF type specificity is one of the reasons for the lack of consensus regarding a definitive fault detection solution. If true, such assumption would mean that the aforementioned traditional definition — HIF as faults with current below protective devices sensitivity — is a condensed definition for a more intricate problem. Based on the consensus of fault variability, therefore, it is reasonable to say that HIFs should also be investigated in sub-classes given by parameters such as the type of the fault, contact surface, and network type.

Researches have recently started to publish more on tree/vegetation faults as a specific type of fault to be studied [20, 27–30]. They have gathered evidence of essential and distinct HIFs characteristics that diverge from other investigated surfaces. One of these is the current magnitude in the first moments of the fault occurrence; initial values are often in the range of a few amperes [4, 19, 30]. This characteristic is conflictual to a large part of the works in the literature since, as discussed in chapter 2, most methods do not have their current sensitivity defined but are instead presented as a generic solution to the traditional definition of HIFs. The second interesting characteristic relates to the first in that, on those initial seconds of the fault, the current seems to have an almost linear relationship with the voltage [4, 19, 31]. That behaviour also represents a conflict with part of the methods in the literature since they heavily rely on the harmonic content of current signals as the predictive information. This linearity assertion is part of the experiments proposed in chapter 3 in this thesis. The relevance of these two particular characteristics — low current amplitude and low harmonic content — further increases when aspects like detection speed are considered. According to a work testing hundreds of vegetation species [4], VHIFs have to be detected and addressed in the first five seconds of fault inception if a considerable fire risk reduction is desired.

As described by recent research [4, 32], the process of sustained vegetation ignition by powerline faults have specific characteristics. One could interpret these works as describing two possible causes of vegetation ignition: high-energy electric arcs, and falling embers from charring vegetation. The first begins with an electric arc created in the air gaps between the conductor and conducting surface (vegetation). The high temperature of the arc ignites the vegetation, which acts as a reductant combustion fuel. If the heating is sufficient, the vegetation organic material goes through pyrolysis, leading to the production of gases that form flames. The flames meet nearby vegetation, which also enters pyrolysis, forming a cycle of sustained ignition. Such a cycle is not always formed as many factors can interrupt any of its steps; the fact that there are many chances for interruption is the reason why not all vegetation faults result in fires. Electric arcs not having energy enough to ignite a critical mass of vegetation, heating being insufficient to start pyrolysis, and inadequate air movements that do not effectively disperse combustion gases are some of the many interrupting factors of sustained ignition. The second scenario resulting in sustained ignition can be exemplified by a tree branch that comes in contact with a conductor at the pole level and begins to char. The charring process can release embers that ignite vegetation at ground level with the same ignition cycle described in the electric arc scenario. The vegetation touching conductor loses its moisture and starts charring due to the ohmic increase in temperature from the fault current. Therefore, the cross-sectional area and moisture content of the branch can play a significant role in the speed at which the branch starts to release embers. The bigger the cross-sectional area, and the higher the moisture content, the longer the conductor may take to start charring [4]. This latter scenario is discussed and expanded in phases of ignition in further sections as it is the one represented by most of the experiments used in this thesis.

The evidence of fault specificity makes VHIFs particularly relevant since they can start fires with devastating damages [5, 6, 33]. Powerlines breaking and falling to vegetation at ground level, vegetation brought by heavy winds bridging two phase conductors, or tall trees reaching powerlines are examples of such scenarios. Countries such as Australia, United States, Spain, and Brazil experienced HIF-related fires created by power distribution lines [5, 18, 29, 33]. Australia, in particular, has long suffered from many large-scale fires related to faults caused by distribution lines. The fires of February 1977 [34], 'Ash Wednesday' in 1983 [35], and the 'Black Saturday Fires' in 2009 [36] are examples of devastating consequences that non-detected faults can cause.

The state of Victoria in Australia is no stranger to bushfires associated with electric distribution systems. The most extreme example was the fires of Black Saturday in February 2009. It was a series of fifteen fires that collectively burnt over 270 000 ha, caused more than 150 fatalities and $4 billion in damage, and destroyed more than 1800 houses [5]. The fires destroyed buildings and infrastructures that supported communities,

causing severe, lasting environmental impact. A combination of extreme environmental factors amplified their reach, making them greater than most fires: heatwaves, temperatures above 40°C, dry weather, and strong winds. Regarding the fire-starting causes, failure of electricity assets were associated with five of the fifteen fires, including the ones creating the most damage.

The consequent damages of Black Saturday were so drastic that the '2009 Victorian Bushfires Royal Commission' was established to address and investigate its causes. The outcome of their work was published in four different volumes approaching different aspects of the investigation [36]: (1) the description of the fires and the related deaths; (2) the recommendations for preparation, response and recovery from fires; (3) the establishment and operation of the commission; and (4) presentation of lay witnesses statements. Most of the insights and conclusion were then presented in the final report [5], which also proposed recommendations regarding safety policy, emergency management, fireground response, and electrical system practices.

1.2 Powerline Bushfire Safety Program and the Vegetation Ignition Testing

The Royal Commission recommendations related to electrical system upgrades were so onerous that a separate initiative named 'Powerline Bushfire Safety Program (PBSP)' was created to address them. The two primary recommendations, #27 and #32, were mainly related to delivering improved electricity assets to the grid and directing practices changes in their protection philosophy. The recommendation #27 was the most arduous and expensive task to address: progressive replacement of single-wire earth return (SWER) power lines and three-phase distribution feeders with aerial bundled cable or underground cabling. The recommendation #32 was less demanding but still challenging; it basically mandated changes in the companies' automatic circuit recloser policy. Companies should disable the reclose function in SWER lines on the weeks of highest fire risk and limit the 22-kV feeders to one reclose attempt before lockout. The PBSP initiative was announced in 2011 as a $750 million project, with a duration of 10 years. The largest part of the budget came from private electricity business (67%) while government contributions complemented the rest. The $500 million invested by private business were directed to investments on new protection asset and control equipment. The remaining $250 million government contribution was unevenly allocated in other areas: $200 million was applied in the replacement of power lines in the areas of high fire risk, $40 million was invested in mitigating power reliability impacts on customers, and $10 million was directed to research and development projects to take place over the following five years.

Despite being the area receiving the smallest portion of investments, the PBSP research and development outcomes are the most relevant to the work presented in this thesis. The initial $10 million was distributed to three priority areas: (1) bushfire mapping and modelling, (2) power line faults and fire ignition, and (3) power line protection technology. Mapping and modelling tasks were performed by assessing the bushfire risk throughout the whole distribution system. Its goal was to have enough information to support the efficient distribution of resources to the areas with the highest fire risk or better risk benefits towards particular classes of electrical assets. With the combination of risk analysis and modelling, expert knowledge, and geospatial images, the project returned valuable decision-supporting information: estimations of fire outcomes and their likelihood throughout the network, cost-benefit analysis of classes of equipment to be deployed to particular locations, and geographical areas with the highest fire risk.

The second R&D area, 'power line protection technology,' resulted in two other child projects: 'Covered Conductor' and 'Rapid Earth Fault Current Limiter (REFCL) Technology'. The former was a grant program set to accelerate the development of novel or improved conductor coverage. It was inspired by the fact that Victorian distribution lines have more than 80,000 km in extension, which are mainly overhead lines with bare-wire conductors vulnerable to any touching conducting surfaces. The current alternatives for naked conductors are aerial bundle cables or moving the overhead lines underground. None of these is particularly economically efficient solutions compared to the less expensive, already existent, bare-wire overhead lines. The grant, therefore, intended to incentivise innovative solutions in cable technology that could be more cost-effective and help mitigate bushfires. The Rapid Earth Fault Current Limiter was the other power protection technology investigated to detect and prevent fire ignition from HIFs. They have shown to be effective at limiting even very small-current HIFs [37]. However, there are issues with its implementation, which are going to be detailed discussed in the next section.

The last R&D project, and bedrock for the work in this thesis, is the 'Vegetation Conduction Ignition Testing'. This program was conceptualised around two main goals: to identify which of the native species are most and least likely to start a fire, and to deliver a reference database of fault recordings to foster the development of fault detection technology. The program's methodology comprised of sampling many local vegetation species and testing them in staged VHIFs experiments. The tests were straightforward in the sense that they were performed by merely subjecting the samples to the network voltage. The staged faults recordings of the current and voltage signals were then compiled in a reference database delivered as one of the program's outcomes.

Around twenty plant species were tested in hundreds of experiments on a real three-wire 22-kV feeder. Since there are no standards experiments for such investigations or dedicated labs to perform them, the project team had to conceptualise original methodologies and

experimental set-ups. The positive aspect of having to develop the experiments from scratch is that equipment and sampling devices can be tailored to the specific experiment; the negative is that chances of making questionable and constraining decisions are much higher. These are going to be extensively discussed in this thesis, especially in chapter 3, since they guided and demanded modifications in the developed methodology. The experimental test rig was produced in two shipping containers that were located at a substation at the Springvale suburb in Melbourne. As the substation allowed direct access to the network voltage, a considerable part of the work was dedicated to producing a safe ignition test space and recording set-up.

The experiments were performed as three categories of possible real fault scenarios: Branch touching wires (phase to earth), Branch across wires (phase to phase), and Wire into vegetation (phase to earth). The first fault type refers to a tree branch laid across two conductors, one earthed and one with the nominal phase voltage (12.7 kV). The second followed the same method but with both conductors energised (22 kV). The third was conducted by dropping the HV conductor into vegetation, either grass or bush, also under phase voltage. It is worth remembering that HIFs are traditionally approached as a scenario represented by a conductor breaking and falling to the ground. These are phase to earth tests that consequently involve the neutral current. 'Branch between wires' (phase-to-phase), nevertheless, are tests only involving phase currents, which are harder to detect in three-wire (resonant grounding or ungrounded) systems. The large number of experiments performed in this under-discussed scenario represents yet another pioneering aspect of this program. The data from these tests can not only support relevant insights regarding phase-to-phase VHIFs behaviour but also lead to higher generalisation ability when used to produce a fault classification method.

The tests resulted in an extensive data set of vegetation HIF fault recordings with high-resolution sampling and wideband signals, having a high potential for insights. All the work and methods described in this thesis were then guided and conceptualised as experiments performed in this data set of much particular type of faults. The expectation when adopting this data set was that it could inform a deeper understanding of VHIF behaviours and support the development of novel detection methods never before explored in the literature.

The specificity aspect of the data set is not only given by the conducting surface (vegetation) but also due to particularities worth mentioning. One of those was that the staged VHIFs had their currents limited by arbitrary thresholds set between 0.5 and 4 A in most tests. The reason for doing so was mainly related to the program's objective to compare fire risk between different species. The experiments were terminated whenever the thresholds were met. As different species took distinct times to reach a particular current magnitude, and they were also in different conditions once the test was terminated,

the adoption of a threshold allowed for an interesting comparison regarding their relative fire risk. Although some tests only lasted for a brief moment (under 1 second), having a set threshold did not necessarily limit most staged faults. A relevant part of the tests took tens of seconds to reach a fault current of a few amperes, producing evidence of their dangerous potential and challenging detection.

One may look at the particularity of adopting thresholds as a drawback but actually introduces pertinent novelty to the work. When compared to fault current values discussed in HIF models in the literature [16, 26, 38], these thresholds translate to much smaller current values. Often neglected, such low fault currents are part of a critical scenario to be considered as they can ignite fires despite having small amplitudes. Therefore, if fire risk reduction is a goal of any given approach, attesting accuracy for even very small fault currents becomes an important validation metric.

Other particularities involved one of the tests' most distinct and relevant characteristics — high-resolution sampling. In particular fashion, the project team decided that it was important that the current and voltage signals were simultaneously sampled in two channels to ensure wideband, low-noise sampling. The bandwidth recorded was non-linearly divided into two channels with different band-pass characteristics. The low-frequency (LF) channel was responsible for continuously sampling the electric signals at 100 kSa/s with suitable anti-aliasing filters, resulting in a 0 to 50 kHz bandwidth. The high-frequency (HF) channel sampled signals at 2 MSa/s with anti-aliasing and high-pass filters that limited its bandwidth from 10 kHz to 1 MHz. The recorded bands represented a severe diversion from sampling rates usually adopted in the literature (discussed in greater detail in Section 2). Consequently, such data not only can introduce novelty by the type of fault which they represent but also by their characterising frequency bandwidth, which is much higher than previous investigations. Lastly, as another worth-mentioning characteristic, the project decided to perform the connection of the test rig to the network via high-voltage current-limiting resistors. In this case, safety was the reason for adopting the use of resistors as the non-occurrence of internal flashovers had to be ensured.

As most claims made here about fire risk mitigation are based on the findings by this program, it is important to note how fire ignition risk was defined throughout the tests. Unambiguous definitions are significant because tests generated a plenitude of responses, from tiny sparks to sizeable embers of burned bark. The definition was based on the assumption that most fire-ignition scenarios are given by embers formed at conductor level with significant size and fuel to ignite adjacent dry grass at ground level. To evaluate the thermal capacity of the embers, a thermal camera, capable of automatically identify the embers and their temperature, recorded every test. Three constraints were then set to label tests as resulting in fire: embers that fell to the floor and remained glowing for at least a

second, small embers (leaves, for example) with temperatures exceeding 350 °C, or large embers (burned twig) with temperatures exceeding 250 °C.

1.2.1 Findings from the Vegetation Conduction Ignition Testing

The 'Vegetation Conduction Ignition Test' Final Report [4] reported all the findings resulted from the analysis of the staged fault, some of which were essential guides to the methodology presented here. In respect to the adopted fault current limit versus fire risk, 'branch touching wire' faults showed that current values greater than 0.5 A rapidly increased the probability of fire. A 1 A threshold translated to 33%, and 2 A represented a 53% chance of fire risk. This finding is critically important given that traditional earth-fault protection systems used in Victoria have a detection sensitivity in the range of 5-10 A [4]. As many tests resulted in flashovers that bridged the HV conductors, it was concluded that traditional protection systems would eventually detect most faults. However, getting to flashover state means that the vegetation sample went through the phases of expulsion of moisture and progressive charring of the bark, which are the phases when fire ignition takes place. Therefore, when conventional protection responds, most of the fire-creating phases would already occur.

The characterization of distinct phases of vegetation conduction was one of the primary findings from the project. It was confidently stated that, by analyzing the tests, four consistent phases of vegetation conduction could be identified: (1) development of full contact between conductor and vegetation; (2) expulsion of vegetation moisture; (3) progressive charring of the bark; and (4) flashover bridging conductors [4]. After initial contact with the conductor, the current magnitude will almost monotonically increase up to a maximum local value where phase 1 ends. When entering phase 2, whistling noises can be heard, and visual moisture is seen falling from the vegetation branches. Phase 2 is characterized by a high-volatility current that alternates until reaching a new maximum value, which indicates the beginning of phase 3. The branch charring of Phase 3 expands the carbonized area of the material, increasing its conductivity and fault current. The increased current with high thermal capacity starts forming embers from one side of the branch, which eventually bridges with the other side, generating a flashover and starting phase 4. The flashover has high conductivity, and the current rapidly increases to values large enough to sensitize conventional protection. The findings point to fire risk primarily arising in phase 3 when the outer layer of the branch is burning and charring. Moisture content, however, was one of the factors that most correlated with ignited samples. Vegetation samples with moisture contents under 10% did not result in fire ignition.

When assessing the fire risk of different vegetation species, the most dangerous were shown to be the ones that took the longest for fault current to fully develop. It was

observed that if the fault current rapidly grew to the detection rate, ember formation was less probable than cases of sustained conduction. In the same manner, faults with higher initial current value such as 'Branch between wires' showed a reduced fire probability as they develop faster than other configurations. Considerations regarding the speed of fault development led to an important recommendation for time detection delays. In 'branch between wires' faults, it was determined that detection times longer than 20 s would be unlikely to reduce fire risk dramatically. However, a significant decrease in such risk could be achieved by responding in 5 s or less. In the case of 'branch touching wires', event detection in 2 s with 0.5 A sensitivity could reduce fire risk in tenfold.

1.3 Significance and motivation

It is worth clearly stating the arguments that motivated the work presented in this thesis. Previous sections directly alluded to the problems resulted from VHIFs, but a hierarchical justification can clarify directions and decisions taken when conceptualising the methodology. The reasons and significance of this work occupy a range of relevancies; from just adding pieces of evidence to the problem importance to possible substantial implications in power line-related fire mitigation.

1.3.1 Fire mitigation and VHIF detection technology

There is plenty of unambiguous evidence of the need for improved VHIF detection technology. The Black Saturday fires are just one example of the possible consequences of unaddressed HIFs, and they are likely to happen again. At present of writing (January 2019), Australia has been facing terrible fires, resulting in the death of 33 people and 500 million animals, 11 million hectares burned, and loss of more than two thousand houses [39]. New South Wales and Victoria have been the states suffering the most significant damages, with more than 50 fires are still burning. As stated in the final report from the Royal commission [5], although electrical faults are not associated with a large proportion of fire ignition causes, their risks dramatically increase on days of extreme fire danger. The report points to the fact that Australia has an ageing electricity system with deteriorating assets that contributed to three fires on Black Saturday. It concludes by stating that given the current economic regime, any substantial reform would be difficult, and as assets continue to age, there probably will be an increased number of fires caused by electrical failures. In regards to the damage done by electrical system-related fires, evidence point that although they are as less likely to occur, they have the propensity to become larger than fires caused by other factors [6]. The dimensions of power line-related fires, in particular, appears to average ten times larger than others [33]. There is no reason to expect that the frequency

of fires is going to reduce in the near future. A study on the effect of climate change on global fire activity pointed that dry regions in the world, such as the Australian continent, are projected to experience a consistent and expressive increase in fire occurrences [40].

Strong evidence point to the insufficient ability of current protection devices to prevent power line-related fires with timely detection. Results from a project testing many broken conductor scenarios in Australia [41] to evaluate existing protection technologies are an example. In the paper summarising the findings [42], the author reported the current protection devices in Victoria's rural systems produced 100% probability of sustained ignition from broken conductor faults. Although not having the testing of protection technologies as the primary goal, the discussed Vegetation Conduction Ignition Testing also presented evidence of their insufficiency. A well-known commercial protection relay developed for the North-American market with an embedded HIF detection function was tested throughout all the tests in the project. As stated in its final report [4], the device did not detect any of the 1038 faults. The fact that there are products in the market claiming the ability to detect all broken conductors and HIFs are, at least, worrisome. It is arguable that the ever occurrence of power line-ignited fires, and the existence of an active, focused research field (explored in the next chapter), are also evidence of insufficient technology. If there is any validity to such an argument, the burden of proof is on researchers and power companies to sufficiently validate their solutions before making claims about its capabilities.

There is current protection technology capable of mitigating the fire-risk created by VHIFs, but its applicability is limited. The previously mentioned work set to evaluate current protection capabilities in Victoria [41] was actually part of one of the PBSP R&D projects — 'Rapid Earth Fault Current Limiter Technology'. The REFCL technology can be summarised as an active residual ground current compensation to be implemented as part of a resonant grounded system. Explored in more details in the next chapter, a compensated or resonant grounded system is basically an high-impedance grounding scheme with adjustable reactance that compensates stray system capacitances. One of the advantages of having the high-impedance connected between the neutral and ground point of a three-phase system is the resulting smaller earth-fault current. The operation of the system benefits from the fact that, since the loads have all to be connected between phases, zero-sequence measurements are *free* to be used for earth fault detection. The REFCL, labelled 'Ground fault neutralizer' [43] by its pioneering company, *Swedish neutral*, presents itself as an enhanced suppression coil. Its main advantage lies on speed, being able to compensate neutral residual currents in under three power cycles (150 ms). Such capabilities make the REFCLs an effective solution for earth faults in terms of fire safety. However, as most engineering solutions, there are trade-offs and limitations in their application in Australian systems.

The first REFCL applicability constraint is the fact that it can only be implemented in resonant-grounded systems. As their functionality is primarily dependent on the residual earth current, REFCLs are not suitable to systems that intentionally rely on the earth path for their standard operation like most Australian networks. This incompatibility represents a considerable constraint since, from the approximately 88,000 km of distribution overhead power lines in Victoria, 28,000 km are SWER lines [44]. These lines are intentionally configured to use the earth as the single path for current return and are thus unable to be ungrounded. Even though they do not represent half of the extension of power lines, SWER lines are arguably more important because they are extensively applied in rural and remote areas, which are rich in vegetation. They are typically composed of one galvanised bare steel wire extended through long distances, often more than 500 m, significantly reducing its associated costs [45]. One SWER line is usually responsible for supplying 10-50 farms or houses that are up to 20 km apart [45]. It is worth noting that, besides REFCLs, the only solutions considered by the government was either to underground the lines or to insulate them as aerial bundle or single insulated unscreened conductors. These insulation varieties can be more challenging to maintain because the insulation can hide damages in the conductors. Moreover, fire ignition is still possible with aerial bundled conductors, and it has happened before; it is possible that in fault occurrences, they even release more energy than leaning trees on bare wires [45].

All the solutions formally considered by the government, including REFCLs, suffer from cost constraints. The Nous group — a consulting company for the Victorian government — estimated the costs associated with each solution scenario. The estimation for insulating all SWER bare-wire lines and transforming all three-phase lines to aerial bundle cable would cost around 11.8 billion AUD [45]. The scenario of moving all SWER and three-phase lines underground could cost more than 20 billion AUD. After seeing this as a preposterous amount, the government decided to move some of the lines in critical areas to underground and mandate the installation of REFCLs in selected substations throughout Victoria. The mandate became legislation with the 'Electricity Safety (Bushfire Mitigation) Amendment Regulations 2016' [46], which made provision as requirements for power companies to increase safety standards of their operation. As determined by the legislation, the network modifications should be performed in a term given by the following seven years. In this period, the companies need to regularly report their progress to the Australia Energy Regulator (AER), which is the wholesale electricity regulator that enforces the rules. To make sure that such mandates would be carried over, the Victorian government also introduced the 'Bushfire Mitigation Civil Penalties Scheme' via an amendment to the Electricity Safety Act 1998 [47]. It included incremental penalties of up to 2 million AUD per substation that did not address the installation mandate. Such combination of power line replacement and REFCLs as a plan of action did not come at a low cost. In one

of its reports to the AER, a large power company stated that the installation of REFCLs at six zone substations needed an adjusted investment of 95.4 million AUD [48]. It also stated that the government probably underestimated the cost of installing the necessary equipment. The government plan for REFCL installations aims to force its deployment to 45 zone substations throughout regional and rural Victoria areas identified as having a high risk of bushfires.

The installation of REFCLs in Australia has exceptional added costs, making it more expensive. These costs emerge from the intrinsic behaviour of ungrounded systems in fault occurrences. As in all other configurations, if a conductor comes in contact with the earth, they both assume the same electrical potential. However, phase-to-phase voltages magnitudes will not change in ungrounded systems. Instead, a shifting of the neutral point happens, changing its value to a point where all the voltages between phases remain the same as before the fault. This intrinsic characteristic of ungrounded systems gives it the ability to ride-through earth faults without any service disruptions (loads are all connected between phases). However, if the fault happens in the phase C, for example, the voltages of phase A and B to ground will increase to values higher than nominal, assuming previous phase-to-phase values. This overvoltage event on the remaining phases is one of the sides from the trade-off associated with the adoption of ungrounded systems; it adds more stress to power equipment insulation. Therefore, power companies have to take precautions with vulnerable equipment in the network before installing REFCLs. These actions come in the form of replacing equipment in the network and having to install isolating transformers to protect the high-voltage customers' equipment [48].

Lastly, REFCLs are also ineffective at detecting VHIFs between phases. As they are installed in the substations neutral-to-ground connection, a HIF between phases not involving the ground would go completely undetected. Examples of these faults are tree branches that fall over two conductors or are carried out by the wind. In fairness, such faults are not as common, but they do happen and can lead to vegetation ignition [4].

In summary, there is no definitive solution for the VHIFs problem; the ones being considered are extensively costly, and none reduces the fire-ignition risk to zero. While reducing the fire risk to zero is probably impossible, producing novel cost-effective solutions is certainly not. These novel solutions can also be much less invasive than having to change the grounding configuration of large systems or changing their extensive conductors. In a sense, most works in the literature attempts at creating such a solution; this thesis is no different. However, there are reasons why the author believes this work, in particular, is distinct from most and deserves attention from R&D managers. These reasons are described in the following subsection.

1.3.2 Contribution to knowledge

Differently from the high-level non-quantifiable goal of inspiring field technology, the more realistic and quantitative goal of discrete contribution to knowledge is easier to demonstrate. Most claims of original contribution are rooted in the data set of staged VHIFs adopted here. It is worth remembering that the methodology and results presented in the following chapters are all derived from the 'Vegetation Conduction Ignition Testing' program recordings [49]. The data from these experiments have many interesting characteristics, such as high specificity, resolution, and particular bias, making it a potential source of original insights. The original contribution to knowledge from this thesis, in this sense, is primarily related to the conceptualisation of the methodology, which comprises the signal analysis and VHIF detection methods.

The aspect that notably differentiates this work from others in the literature is fault type specificity. The experiments are based on a data set with hundreds of vegetation faults from local species, which seems to share behaviours and characteristics according to the resulting evidence. Specificity is essential since vegetation faults are fundamentally different from other types of HIFs. This point is a persistent argument throughout this thesis, evidenced by works in the literature discussed in Chapter 2. Although there are plenty of works that misguidedly generalize their results to all type of HIFs, there is a general consensus that the fault surface will greatly influence the fault behaviour [14, 15, 19–26]. . Vegetation HIFs are particularly relevant to the literature, in this sense. From the almost five decades of research in this field, few works sparsely focus on this type of fault [20, 28–30], most comes from the last two decades, and none have approached the topic in the same way as in this thesis. It is arguable that the findings here would generalise for other vegetation species given the high invariance found in their behaviour, but such claim could only be made with more evidence and practical testing.

Having hundreds of tests recorded in a particularly high resolution is another significant discriminative aspect of this work. The staged tests were recorded with sampling rates up to 2 MSa/s, allowing investigations on bands seldom investigated in the literature. When higher frequencies are mentioned in other works, they are mostly related to bands smaller than 10 kHz (see section 2.5). The probable reasons for this limitation are higher costs associated with high-resolution sampling and the desire to target existing hardware. High-speed digitisers are most expensive if precision is equated, and they do require more resources in terms of data management efforts, which also translate to higher costs. Nevertheless, the fact that the proposed methodologies seem to actively target existing hardware is probably the most influential factor. This understandable focus comes from the desire to propose low-cost solutions by using the existing transducers and digitisers on the field. Therefore, since digital relays commonly have sampling rates in the order

of a few thousand samples per second, most works also tend to adopt similar values as their digitisation rate. Example such as the General Electric feeder protection *F60* relay sampling signals at 64 samples per power cycle (3.2 kHz) [50] can be cited, or the ABB's Feeder protection and control *REF620*, which digitizes 32 samples per cycle at 1.6 kHz [51]. Regarding VHIFs, in particular, the evidence presented in this thesis point to the high probability that these sampling rates are too low to effectively represent this specific HIF type discriminative behaviour in the fault signals. Regarding the sampling rates usually used in the literature, section 2.5 presents an analysis of the frequency bands analysed in the surveyed works.

Some aspects of the work in this thesis are not entirely divergent from the literature but are particular enough to be labelled as part of the original contribution. Characteristics such as the range of fault current amplitudes or configuration of the fault seem to be under-discussed in the literature even though they are essential aspects of any fault detection method. Some particular characteristics of this work should then be addressed to properly bound the scope of this work.

Specifications of fault current magnitudes or sensitivity of the method are particularly crucial aspects of fire ignition risk that should be discussed. Its importance is further highlighted if some of the differences between types of grounding are considered. HIFs in solidly grounded systems can result in relatively large earth-fault currents since they are only limited by the fault and system impedance; in ungrounded systems, however, they are limited by the system stray capacitances. This difference is crucial for the protection system, which may have ground relays adjusted to hundreds of amperes in solidly grounded systems depending on the load [52]. Without being detected by traditional protection, HIFs can assume currents of tens of amperes in grounded systems, while resulting in much smaller values in ungrounded systems. Therefore, works that do not discriminate between current levels, or make their sensitivity precise, present ambiguous results when considering the classical definition for HIFs: faults with currents below protection thresholds. The issue is further aggravated when considering evidence showing consistent fire ignition in vegetation with low-current faults such as one or two amperes [4]. As previously mentioned, mainly all the fault currents represented in the data set used here were well limited to values up to four amperes. A detection method addressing restricted faults like these could be said to have a higher sensitivity with more precise boundaries regarding its fire risk mitigation capabilities.

If the HIF detection field is considered as a whole and the arguments for fault specificity presented here are accepted, any sound analysis of novel data from real faults can be seen as an original contribution to knowledge. Staging HIFs in real networks can be onerous and expensive, hence whoever does it have minimal incentive to share the resulting data. To alleviate this barrier to entry, researchers started proposing HIF models, which soon

became a HIF sub-field. The HIF models can be used in power systems simulations in many fault scenarios and are tentatively employed to test detection methods. However, the nuances of fault specificity can make all the difference in the simulations results. If HIFs or VHIFs are as high-variance as they appear in the literature and here, any detection method based on HIF models is going to be heavily biased towards the fault types staged in the model creation. If carefully investigated, one can observe that the works proposing models are heavily representative of a few types of faults or/and do not stage a vast number of faults. This scenario is detailed explored in Chapter 2, but the important conclusion is that being heavily biased towards one type of fault is not necessarily detrimental; it becomes a problem when other works use such models to generalise conclusions about all HIFs. Moreover, when used to attest the effectiveness of a newly proposed detection method, the models are usually included in noiseless or non-representative simulations. The neglect of network noise in these works has been shown to diminish detection security in a work made to test this issue [53]. These issues are not necessarily signs of faulty research but are only the result of researchers working with the data they had. The work presented in this thesis overcomes some of these challenges by having an extensive data set of fault recordings heavily biased towards vegetation faults, which were all staged in a real network and never published on before. It is nowhere here claimed that the presented results could be fully generalised to other types of HIF, although similarities are discussed.

In this sense, this work has the pioneering aspect of being the first to formally publish results obtained while working with the mentioned data set. This thesis is not the first discussing it as, in 2017, the Victorian Government set out a challenge in which research groups would use the data set to create detection methodologies [54]. Four groups of entrants were short-listed, and their solutions were made public. However, none of the results presented was quantitatively objective to represent an effective detection method. The method presented in this thesis was not submitted as one of the entrants as it was not conceptualised at the time. In regards to formal scientific publications, to the best of the author knowledge, there is no work proposing detection methods based on this data or any detailed analysis on how it could be achieved at the time of writing this thesis. Nevertheless, from this pioneering investigation and specificity emerges a complication given by the impossibility of comparing it to other existing methods. All the particularities and singular characteristics of the data set previously discussed makes the results unique and biased towards VHIFs, which can be said to be an under-discussed type of fault in the literature. This comparing issue was present in all publications resulting from this work [31, 55–57] since comparisons with existing methods are standard practice. However, the specificity and novelty argument were eventually accepted by all peer reviews as part of this work originality.

Lastly, experiments developed with the intent to attest the value of adopted approaches in methodology are also claimed as original contribution. The author understands that, as directions in methodology diverge from most in the related literature, they might be received with scepticism by peers and technicians. The experiments developed in an attempt to mitigate concerns are composed of known techniques, but the methodology in which they are used was explicitly conceptualised for this thesis. The evidence of their originality is their acceptance as publications in renowned peer-review journals (further discussed) [56, 57]. The main goal of these experiments is to highlight that considering higher frequency bands for VHIF detection can have real benefits in terms of sensitivity and performance; the challenges of doing so, however, are also real. As it is as essential to discuss the possible constraints and consequences from adopting such an approach, Chapter 5 have comprehensive sections dedicated for this purpose.

In short, the main contribution to knowledge can be summarized by the following points:

- Development of a methodology for detecting HIFs from data pre-processing to classifier validation.
  - Focus on a specific type of HIFs given by Australian local vegetation species.
  - Based on a dataset of hundreds of staged faults, not previously published on.
  - Faults currents limited to small currents from 0.5 to 4 A.
  - Signals sampled with high resolution, up to 2 MSa/s.
  - Presentation of original signal representation techniques (features) as fault predictors.
- Conceptualization and execution of experiments to evidence the importance of the methodology.
  - Presentation of evidence of the importance of having high-resolution signals to detect VHIFs.
  - Presentation of evidence of fault signatures, attesting their existence, extraction methodology, and use to identify the source of signal disturbances.

1.4 Thesis statement and goals

Having discussed the context and potential for contribution to knowledge in previous sections, the remaining of this chapter briefs the reader on the work described in this thesis and what it aims to achieve. Although most of this project is rooted on the data from the

'Vegetation Conduction Ignition Testing', the author was not aware of its potential when first encountering it, neither there was a clear intention to dedicate most of the candidature effort to it. A familiarization and analysis of the potential conducive information in the data set were required.

The initial investigations on the data set resulted in surprisingly remarkable findings deserving of publication [31]. As the main intention was to improve HIF detection technology, the investigation on the data set was directed to find useful information at discriminating fault occurrences from nominal states. The first steps were based on the comprehensive final project report [4], where the author alluded to changes in the signals HF spectrum when a fault was occurring. With the use of signal processing tools, such information was indeed identified, and it showed to be quite consistent. Section 3.2 is dedicated to informing the reader on the signal processing techniques employed in the analysis and how they were used. The initial findings resulting from these investigations are described in section 4.1. It not only describes the consistent potential discriminative information found in the HF spectrum but also presents insights about fault behaviour. They represented the first evidence pointing to the importance of the concept of specificity. The findings regarding VHIF behaviour contradicted expectations inspired by how the HIF phenomenon is described in the literature while agreeing with a few works studying only VHIFs.

The initial findings gave confidence to pursue the development of a VHIF detection method based on the vegetation ignition data set. However, initial attempts at conceptualizing such a classifier showed the VHIF behaviour to be more complex than first imagined. The task complexity inspired attempts to employ machine learning algorithms to discriminate fault signals from non-fault ones. Such attempts were immediately challenged with the problem of not having clear labelled data, but a sensible alternative was soon found, followed by the successful application of classification algorithms. The problems and challenges with the data and developed solutions are detailed described in section 3.1 of the Methodology chapter. The theory base and discussion on the machine learning algorithms considered are presented in section 3.3, while the outcomes from learning a fault classifier are presented in section 4.2 of the Results chapter. The expressive results were also documented in a publication in the journal *IEEE Transactions on Power Delivery* [55].

Although presenting promising results, the developed classifier was an offline algorithm that could benefit from being implemented as an online fault classification prototype. Having the fault classifier learned from HF signals recorded from the staged tests is an important aspect of the relevance of this prototype. The use of HF signals is rarely adopted in the literature and dealing with higher frequencies often result in higher costs, which can lead to scepticism from experts in the field. Therefore, a prototype was developed

with the intention to be a low-cost solution that would play the role of a fault classification module from a broad detection device. If successful, the results could help substantiate feasibility claims made when first proposing the method. The prototype development, including hardware and software set-up, are detailed described in section 3.4, which also describes its experimentation and testing.

Given the novelty associated with using HF signals and the potential hesitation it can inspire, it was desired to acquire further supporting evidence to substantiate the adopted approach. The experiments performed to obtain such evidence comprised in applying the signal processing techniques to LF and HF signals and learning classifiers as in the same manner as originally proposed. The resulting accuracy comparison from both domains would then serve as evidence of the content of predicting information present in the LF and HF signals. The results showing the substantial importance of sampling HF signals for detecting the specific VHIFs are presented in section 4.3. They were also further documented as a paper in the journal *Electric Power Systems Research* [56].

Lastly, an investigation to define how the fault signatures are expressed in the signals time domain was performed. It intended to clearly show how the fault signatures are manifested while attesting their existence. It was not an easy task since the recordings are the result of the convolution of many signal sources such as electromagnetic interference from radio and other broadcasting means, load, and system noises. Recent signal processing techniques had to be used to code and deconvolute such patterns from different noises, and a methodology was created to find which ones were highly correlated to fault occurrences. The signal processing technique is described in section 3.2.3, where it is discussed how the technique can work as an unsupervised learning method, finding recurring deconvoluted patterns in a data set. The results from this fault signature representation methodology are presented in section 4.4 from the Results chapter. They were also documented in a publication in the journal *IEEE Transactions on Instrumentation and Measurement* [57].

All the mentioned investigations took place during the author's candidature, resulting in recurrent, insightful findings. As they came during different moments in the project, they inspired constant changes and updates on the parameters and tools used in the initially proposed detection method. To clarify the aspects of its final stage, the last section (4.6) of the Results chapter is dedicated to describing all the characteristics of the last working version of the detection method. It contemplates the best practices found so far, after many trial and error attempts.

This thesis is composed of five chapters with specific aims that can be briefly pointed out. The HIF Literature chapter (2) aims at familiarizing the reader with state of the HIF detection technology, mainly focusing on how it had evolved throughout the last decades of research and highlighting its cornerstone findings. It also serves to substantiate the discussion of previous works and to support the claims of original contribution to

knowledge from this thesis. The Methodology chapter (3) aims at informing the reader on the use and practices adopted in data pre-processing, signal processing, machine learning, and prototype development. The Results chapter (4) has the goal of objectively presenting the outcomes of the performed experiments, tests, and trials, without judgement or discussion on their relevance. Regarding their contextualization and implications to real implementation, the Discussions chapter (5) is dedicated to contextualizing the findings and results by discussing the method advantages and constraints. The goals of the work described in this thesis are summarized as they were chronologically achieved:

- Perform an investigation on the Powerline Bushfire Safety Program data set of staged vegetation fault recordings to find potential predicting information of fault occurrences.
- Produce signal processing techniques to extract correlational fault features to be used for fault detection.
- Conceptualize a fault detection method based on machine learning algorithms to leverage the existence of extensive labelled data.
- Compare the amount of predicting information on the LF and HF signals, gathering evidence of the importance of HF signals in VHIF detection.
- Develop a proof-of-concept prototype to support the potential feasibility of the conceptualized detection method.
- Extract fault signatures from the HF recordings to attest their existence and generate insights regarding VHIFs behaviour.

Chapter 2

High-impedance fault literature

Attempts to address High Impedance Faults (HIFs) in the literature goes back to at least the mid-70s [58]. A comprehensive chronological literature review [2] actually points to early related work dating back to 1960. Many approaches leading to important contributions were proposed since then. The HIF research field has presently matured into a distinct discipline, which continues to be active to this day. A simple IEEE *Xplore* database search with 'high-impedance fault' as key-words on their scope reveals that most of the work has been published in the last ten years. The search results certainly do not account for actual contributions made to the field, but it certainly is a proxy of the field popularity. Some works will not appear in the search result since they indirectly studied this problem or do not use the same exact term [6, 59–61].

Despite being well-established as a field of research, it is hard to find consensus on the best way to address HIFs in the literature. A root cause connecting most arguments for this fact, and a predominant theme throughout this chapter, is that HIFs have high intrinsic complexity. To clarify, consider the broader HIF definition as a fault with current magnitude lower than overcurrent devices' sensitivity (adopted by most the works cited herein). This definition consequently means that an ideal HIF detector should be sensitized by all possible scenarios fitting its description such as a conductor that breaks and falls to the ground, trees/vegetation leaning in nearby conductors, and faulty isolation leading to leaking currents. Many factors impact each of these fault types such as vegetation moisture content, surface condition, voltage level, type and size of conductor, weather conditions, and network grounding type. Particular fault characteristics like current amplitude, non-linearity, disturbance, and intermittency in conduction, could then be generated. Therefore, one should not expect that a HIF resulted from a conductor that falls to the ground in sandy soil to have the same characteristics as another created from the contact between conductor and grounded vegetation, for example [14, 15, 19–26].

The findings and developments on HIF detection and their consequent implications to this thesis are presented in the next sections. Only the works where an evident contribution to knowledge was observed have been considered. Referenced works that were not comprehensively discussed were only used to support statical inferences such as the state of the literature or the popularity of a particular approach or technique. The ones chosen to be discussed were organized in chronological order, narrative structure, as to better portray the development of the field through time. Divided by relevant decades, the next sections discuss the development of relevant HIF detection solutions and the evolution of the technical conversation by many research groups around the world.

Before starting such a discussion, however, it is useful to define some of the qualitative terms used when describing the surveyed approaches. The terms *a priori* or *bottom-up* refers to approaches that utilize *deducted* knowledge about a phenomena when conceptualizing solutions. These are often described in a clear mathematical way representing sub-systems which, when pieced together, form more complex emergent systems. Works describing mechanical- or circuit theory-based solutions are an example of this classification. Conversely, *a posteriori* or *top-down* approaches utilize knowledge learned from the studied phenomena by observational data. Works with this quality are heavily present in the recent literature with the ubiquitous use of machine learning, which learns from existing observational data. The advantages of bottom-up approaches are their predictability and reliability given by their often accompanied mathematical descriptions and unambiguous behaviour with clear decision boundaries. Their main disadvantage — which conversely is the advantage of top-down approaches — is its lack of adaptability to different environmental conditions. Top-down approaches disadvantages are their lack of a theoretical causal framework and the possibility of bad generalization if the task is learned from a biased data set. Therefore, one can define these two approaches as complementary with opposite advantages and constraints. Not surprisingly, elaborate and appealing methods on the edge of the recent literature contemplate a mixture of *a priori* and *a posteriori* approaches. *A priori* knowledge-based approaches are mostly *deterministic* (another term used herein), meaning that their output always represents total certainty in regards to a certain input. *Probabilistic* methods, however, output the probability of an input to be drawn from a certain probability distribution of a particular class of data. An example of a deterministic method is one where a HIF detection decision boundary is given by an arbitrary threshold on a measured parameter. A probabilistic method example is one where an Artificial Neural Network (ANN) does such a task by outputting the probability of input data being from a fault occurrence or not.

2.1 The 70s and 80s — Genesis of a research field

Pioneer researchers at Texas A&M University [7, 58] (TAMU), Pennsylvania Power & Light Company [62, 63] and Amicus Engineering Company [9, 64] laid the path for following researchers with innovative approaches to HIF detection. Their ideas still profoundly inspire the works in the recent literature, such as the use of frequency components higher than the fundamental [7], power frequency of harmonics and voltage sequences [9], and the ratio between sequence currents [62, 63]. Another honourable mention as preliminary research on this topic is the work presented by researchers at the National Chen Kung University in Taiwan at concurrent time [65].

Before the seminal works, the HIF problem was a small subset of a research field responsible for addressing earth faults in power distribution systems. When being established as a field, the HIF problem was mainly described by the concerning scenario where a broken conductor falls to the ground. By then, power engineers relied on ground overcurrent protection to address these faults. Nevertheless, it was quickly realized that the sensitivity of these relays, which had to tolerate some load unbalancing, was not high enough to adequately address the problem. One early work [7], based on technical reports by the Pennsylvania Power & Light Company, drew attention to the underestimation of undetected broken conductor faults. This work, conducted in 1974-75, found that overcurrent devices failed to operate in 32% of the 390 staged faults. When faults were staged 2-3 miles from the substation, only one of the twenty cases was cleared. They also conducted surveys with many utility personnel about the clearance of such faults. From the 83 surveys, 61% said they had experience with broken conductors problems.

It may be useful to note that there is a parallel field of research which addresses a subset of HIFs named *ground arcing faults* [25, 66–68]. As the authors explained [7], the return path of the current is often not fully established in a broken conductor that falls into a solid surface such as asphalt. The air gap between surfaces usually results in arcs created when the voltage reaches a *breakdown value*. Therefore, due to the sinusoidal nature of the voltage waveform, a burst of current conduction can happen near zero-crossings where the voltage crosses this reigniting-arc breakdown value. Some works also may use the terms ‘ground arcing faults’ and ‘high-impedance faults’ interchangeably.

The first attempt to detect real staged HIF with hardware prototypes in the accessible literature were also made by the mentioned pioneers from TAMU [7]. It comprised of a method utilizing the frequency components from 2-10 kHz, being the first to investigate higher frequencies in HIF detection. The authors felt that frequency components higher than ones close to the fundamental frequency represented an appropriate approach for detecting ‘ground arcing faults’ from broken conductors. However, the authors raised caution for networks containing grounded wye capacitor banks which could increase the

attenuation of these frequency components. The method was implemented in a computer capable of detecting some of the staged faults, attesting for the sophistication of the adopted approach. They had conceptualized a top-down approach, that despite not having absolute accuracy and security, already used observational data from staged faults to conceptualize HIF detection methods. Researchers at the Pennsylvania Power & Light Company also did ground-breaking work [62] by proposing a bottom-up modelling of a 12.47-kV, four-wire, multi-grounded system followed by an electro-mechanical ground relay to detect HIF occurrences. The method consisted of measuring the ratio between the phase positive-sequence current and the three-phase zero-sequence current. Such a protection philosophy was based on the premise that the ratio of these currents remains relatively constant for a given feeder in the absence of a HIF. Further work [63], published in the same year, presented a digital computer implementation to compare the proposed ground ratio relay with the existing protection schemes. This approach was further validated and proven [65] to be effective when detecting currents higher than 15 A in a well-balanced system.

By the late 80s, TAMU had more than a decade in broken conductor research. This experience gave them the insight which is often missing in contemporary works: HIF is too complex to depend on one single detection method. In their work [15, 25], the authors discussed a detection approach where many different algorithms calculated distinct *features* from the electric signals. They were later fed to a class of primal learning algorithms called *expert systems*. It intended to leverage the existence of observational data to emulate the decision-making of a human expert. It is not surprising that the authors opted for such an approach since the use of expert systems exploded in popularity in the 80s [69]. Due to its intention, i.e. replication of human intelligence to some extent, expert systems were considered to be the first successful Artificial Intelligence agent *expert systems*.

2.1.1 Highlights

The 70s and 80s found many valuable findings and insights on HIF behaviour. The most relevant insights can be summarized:

- Due to the impulsive nature of the fault current, HIFs can increase the energy of a wide band of harmonic, and non-harmonic, frequency components [7, 8].
- HIF current magnitude can vary greatly in between cycles having an intrinsic *random* nature [7, 8, 15].
- Arcing often happens in broken conductor faults due to the air gaps between the conductor and high-impedance surface [7, 9].

- The high-frequency bursts in the fault current often happens near zero-crossings of the voltage due to arc reignition phenomenon called *voltage breakdown* effect [7, 8].

2.2 The 90s — Experts systems, commercial HIF detection, and modelling

Research at TAMU continued to dominate the HIF field in the 90s, despite the appearance of new players from other universities and countries such as South Korea [70], Canada [71], Singapore [13], and Brazil [72]. At the beginning of the decade, TAMU researchers had fully embraced the idea that no single method could cover all HIFs. They started first to publish their ideas on how to address this fact, discussing how environmental parameters such as system unbalance, feeder configuration, load type, and surface conditions could affect the fault behaviour [24]. Based on the assumption that each technique will have strengths and weakness giving different conditions, they proposed a heuristic to select a detection method based on environmental parameters.

2.2.1 HIF detection goes commercial

The heuristics path by TAMU would become more evident in a further revealing paper [73] where a product, resulted from a collaboration with General Electric (GE) company, was disclosed. The paper described the first substation equipment having HIF detection as its primary goal, embodied in a case that fitted the panel cut-off for a GE overcurrent relay. Following what was previously mentioned as a possibility, this work presented an expert system approach consisting of many independent algorithms to detect HIFs. These algorithms dealt with numerous signal's features: energies on the frequency spectrum, randomness, arcing (24 algorithms), load analysis and event, and burst patterns. The expert system heuristic consisted of receiving the algorithm's outputs which were then weighted to arbitrary values and aggregated in a detection result. An undertaking so sophisticated at the time that GE had initiated an advisory committee of experts to facilitate the device acceptance in the market [73]. As yet another significant aspect of this work, researchers validated their method using tests staged in a dedicated location called the 'downed conductor test facility', constructed and designed by TAMU just for the sake of the experiments.

The announced product [73] was not a relay but a 'Digital Feeder Monitor' with broken and arcing conductor detection function. The distinction between relay and monitor represents an important point here. The authors firmly held the opinion that one should not trip a whole feeder for all HIF positive detections. They were aware that the device was

not capable of correctly detecting all HIFs occurrences without producing false-positives, meaning that there was a trade-off between *dependability* and *security* to be considered. The 90s was the starting point for discussions around these two critical concepts. Dependability refers to the frequency of true positives when testing a classifier; it expresses how good the classifier is at detecting faults. In the HIF detection field, it is represented by the ratio between the number of fault observations correctly labelled, to the total amount of fault observations tested. Security, conversely, refers to the frequency of true negatives when testing a classifier; it expresses how good the classifier is at not detecting faults when they are not occurring. It can be calculated by the number of non-fault observations correctly labelled as non-fault, divided by the total number of non-fault observations. With the increasing proposal of different detection methods, the consequent question of addressing imperfect security and what to do with the detection result had to be addressed. The authors thus made use of the product disclosure to discuss possible liabilities assigned to utilities which decide not to install such equipment in regards to damages created by undetected HIFs [73]. Comments on how such a method could be used in different regions were also made. The authors sensibly argued that tripping a relay in arid areas where a downed conductor could ignite wildfires in vegetation would make more sense than in a city environment.

Researches at TAMU continued to build on their work by proposing a detection method based on current RMS fractal analysis [74] and publishing on practical experiences gained from the use of their device in real feeders [68]. The authors focused on the security discussion defending that service continuity of clients is far too valuable to be sacrificed for a 'trigger happy' algorithm. The paper [68] analysed the equivalent of forty-seven unit-months of device operation in five feeders where additional faults were also staged. It showed an optimistic view of the device application by stating that faults that were not cleared by overcurrent devices were mostly all detected by the feeder monitor (88% of staged faults). Furthermore, two more similar papers were published discussing the device implementation [52] and its practicality [75]. In the first, many issues arising from the use of such technology were discussed. From legal to emotional issues, the paper included a framework for testing and evaluating a HIF detection method, the result of surveys, and expert opinions. However, none of those subjects was discussed in too broad detail but rather in brief discussions. The second paper [75] disclosed details on the IP licensing of the technology to GE, the balancing of dependability and security for service continuity, and the functionality of the device.

The research undertaken throughout more than a decade at TAMU has evolved and culminated on the present GE's *Multilin F60 Feeder Protection System* [76] which, to present day, still uses expert systems to detect HIFs. Given current knowledge about learning systems, and how expert systems were made obsolete by machine learning

techniques, there is an argument that such a device could be substantially improved. This claim can be made based solely on the fact that the features' weights used in the expert system method when making a decision were actually based on *a priori* knowledge from human experts.

In 1990, researchers had already realized possible issues with arbitrary decision boundaries when proposing the first application of neural networks to detect HIFs [77]. The proposed method calculated large amounts of signal features such as the peak of transient current, RMS value, the magnitude of positive sequence, and energy of harmonics, which were used to learn a multi-layer perceptron feed-forward neural network. It was also one of the first methods to use simulated HIFs in their validation, instead of real, staged fault signals (a ubiquitous practice in the following decades). Despite being novel, there were many outstanding issues with this approach. The neural network had a large number of inputs and nodes compared with the amount of generated data (*overfitting* further discussed). There were no validated HIFs models at the time, making it an excessive unrealistic simulation, and features were still hand-engineered. By hand-engineered features is meant that the predictors are created from human knowledge, calculated from signal representation techniques; the converse would be the feature extraction process done by deep learning methods, which captures the predictors from the data latent space. Further work with neural networks [71] addressed some of these problems. The methodology, instead of using hand-engineered features as inputs, consisted of feeding a fundamental frequency cycle of raw current samples to a feed-forward neural network. It was the first relevant full top-down, supervised learning approach, used to learn patterns in the current time-domain signals.

2.2.2 Modelling HIFs

Although powerful, top-down approaches still present the apparent constraint of requiring data from real, staged faults. Such experiments can be onerous and expensive, only to be performed by prominent universities and large companies. Consequently, due to the commercial value of the resulting IP, organizations performing such experiments had many incentives to keep the data private. This restriction led researchers to invest efforts in creating HIF models to be used in simulations, circumventing the problem of requiring experimental data. The complexity of the HIF phenomena and non-converging opinions in the literature clearly make such an attempt ambitious and highly challenging.

Historically, the first relevant and influential HIF model was proposed in 1990 [38]. The authors represented a HIF as a fault impedance in series with an anti-parallel diode and DC voltage sources branch, as illustrated in Fig. 2.1. The anti-parallel branch helped to model the previously explained breakdown voltage where conduction only started after

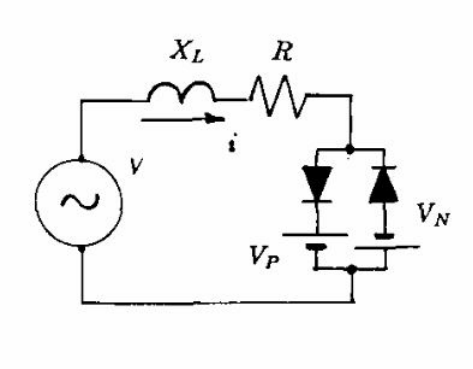


Figure 2.1. Emanuel's et al. HIF Model (1990).
Adapted from [38]

it was high enough to break the dielectric nature of the high-impedance surface. Its main parameters were the fault impedance, which dictated most of the fault current amplitude, and the DC voltage sources values. The DC sources could assume different values to account for the asymmetric nature of HIFs, which can have different breakdown values for positive and negative half-cycles. Notwithstanding, one should mention that this work attempted to model the HIF behaviour of a broken conductor falling to sandy soil. The authors had the main goal of analysing to what extent frequencies from 120 to 180 Hz could be used to detect such HIFs. Any strong claims present in works directly influenced by this work which choose to ignore or dismiss these environmental conditions should be prefaced as hypotheses, speculations, or are just fallacies.

2.2.3 Wavelets and Fourier

Advancements in HIF detection with top-down approaches, however, do not provide a phenomena understanding or casual framework that bottom-up approaches do. The urge for finding better HIF features, together with developments in the signal and image processing fields [78], was responsible for the introduction of a technique that revolutionized signal representation: wavelets. When used to decompose signals, the wavelet Transform (WT) serves as an efficient time-frequency signal representation with fair time localization [79]. Different from the dominant and prevalent Fourier Transform (FT), the WT's decomposition basis functions are dilated and translated versions of finite oscillations with constant shape. This difference makes the transformation outputs to be well localized in time, conversely to the stationary (infinite in time) sinusoid basis functions from the FT. The oscillations used in the decomposition are allowed to have arbitrary shapes, although one would prefer differentiable, compact, zero-mean, and square-integrable functions for practical reasons. Examples of commonly used wavelets with a specific shape, named 'wavelet families', are illustrated in Fig. 2.2. In general, wavelets are more

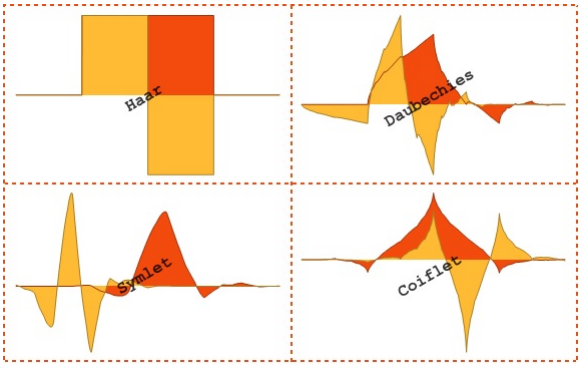


Figure 2.2. Examples of commonly used wavelet families.
Adapted from [80]

efficient at representing signals' discontinuities and transients since they have more *sparse* representations in the wavelet domain with consequent localized energy (discussed in further details in the 3.2.2). This effect becomes especially evident if the mother-wavelet used in the decomposition resembles the shape of the represented discontinuity. An FT of the same signal, contrarily, will result in a wideband representation in the frequency domain with *spreaded* energy.

The use of wavelets for HIF feature extraction that could represent fault occurrences started at the end of the decade. The first influential work [13] used wavelets to detect the transients generated by the faults, followed by a heuristic to differentiate them from other disturbances. It was not only novel in its wavelet application but also built on the previously mentioned model [38] to propose a more intricate HIF model based on arc theory. Claiming that its predecessor did not represent the universal behaviour of HIFs, the authors proposed a model that was supposedly better at representing the non-linearity of a HIF impedance. A snapshot of the proposed model [13] is illustrated in Fig. 2.3. Differently from the inspired work, the voltage sources in the anti-parallel branch are time-varying, not restricted to DC values. Switch 3 is connected to a Transient Analysis Control System (TACS) model, which is set to represent the arc reignition and extinction. The TACS model inputs are the arc re-ignition voltage, peak value of the applied voltage, and arc voltage. The variable resistance is mainly responsible for the value of the fault current magnitude, which was in the order of tens of amps in the paper [13]. It was also one of the first papers to present a full simulation-based methodology instead of relying on experimental data. A second wavelet-based influential work [81] soon followed it by proposing the use of a different wavelet family as its only main contribution. As yet a novel idea, wavelet-based works became widely popular in the next decades, fiercely competing with Fourier-based approaches.

Nevertheless, Fourier/Harmonic analysis was still the prevalent feature extraction method in the majority of detection methodologies. Most methods were based on findings

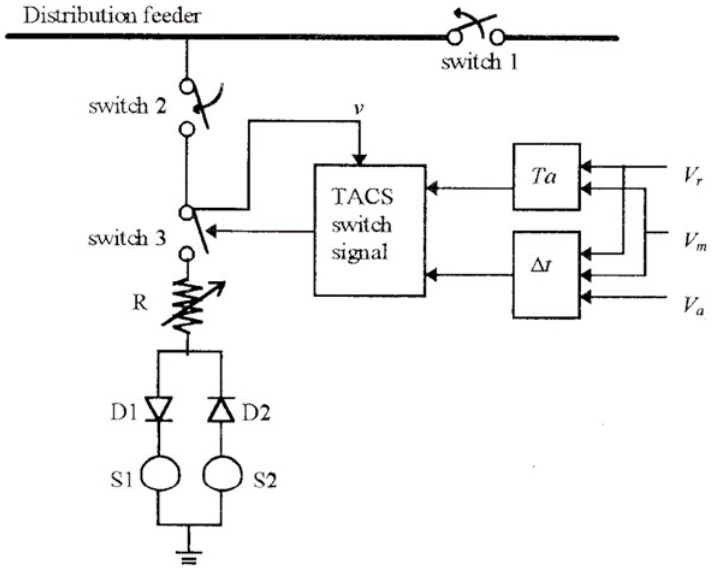


Figure 2.3. David and Xia's HIF Model (1998).
Adapted from [13]

made in the 80s around the HIFs effects on the harmonic signature of phase and neutral currents. One pioneering work [82] presented a Kalman filtering approach as an on-line recursive estimator of harmonics that accounted for time-varying nature of the signals. The advantage of this approach was that it did not need to assume the signal to be stationary (like the conventional FT). The method was centred on calculating a randomness value from estimated harmonics, in four to six fundamental cycles, and detecting a fault if it increased higher than normally observed values. A following work [83] argued for the unique harmonic characteristics of HIFs. It stated that such faults would have high third harmonic components with characteristic phase shifts that could be used for detection. Similarly, a creative approach using the ratio of odd and even harmonics to detect HIF occurrences was proposed [70]. It was based on the hypotheses that HIFs have a particular harmonic signature due to their half-cycle asymmetry, which could introduce even harmonics to the system. The unbalanced current and energy of harmonics were also used to conceptualize another method [84] that was validated with staged tests. Such an approach seemed to work reasonably well for fault currents higher than 5 A.

These cited works have the advantage of being validated with data from real, staged tests, usually in multi-grounded systems, mimicking a downed conductor scenario. Their weakness, however, relates to the consensus that harmonic content alone is not a reliable parameter to attest for a fault occurrences [14, 72, 85–88]. It is often stated that similar harmonic conditions could appear in normal operation states of the system, mostly from switching events and the contemporary diversity of non-linear loads. Experiments presented in this thesis also asserts that not all HIFs generate a significant amount of

harmonics. A significant number of VHIFs do not produce harmonics in the first seconds of conduction in the fault location, meaning that they would be even more attenuated for measurements made away from the fault point. Nonetheless, it was not a widely shared opinion at the time these methods were proposed but became a more prevalent opinion with the ever-increasing penetration of non-linear loads and novel signal representation tools.

Two other influential and valuable works from this decade are also worth commenting. The first is the highly innovative method [72] presenting the first active method to detect HIFs. It consisted of injecting periodical impulses to the network and measuring its response, which would presumably change in the presence of a fault. The methodology was tailored to Brazilian networks, which have many single- and two-phases branches, thus being heavily unbalanced. Its most substantial constraints were the need for network data, full knowledge of its topology, and high-speed data acquisition systems. After injection, the electric signals were sampled, processed via FFT, and used as inputs in a fuzzy reasoning system. The fuzzy rules, as in other approaches in this era, also used *a priori* knowledge and rules of thumb from human experts. In the second work, a paper by a GE engineer [89] reviewed and discussed some of the current technologies to detect HIF. This paper was the first influential work in the literature to discuss mechanical methods to detect HIFs. The author explains that such a device would be mounted, in a cross arm or pole, under each phase wire and connected to the ground. In this manner, a conductor breakage would realize contact between phase and device, creating a ground fault that could be easily detected by existing overcurrent devices. Wester then quickly dismisses the usage of such a method claiming that installation and maintenance costs would be too high to be feasible. Nevertheless, he still recommended its installation close to critical areas such as churches, schools, or hospitals. The rest of the paper could be seen as a call to utilities to install the previously mentioned feeder monitor conceptualized by TAMU. Comments on the use of the detection result and the need for HIF detection devices were made based on the possible million-dollar liabilities companies could face for damages created by HIFs.

2.2.4 Highlights

One can see the 90s as the golden era of HIF detection engagement and innovation, where the most influential ideas were proposed. The core of insights and innovations of the decade can be summarized:

- First commercial protection device targeting HIFs [73].
- First full top-down application of neural networks in HIF detection [71].
- First HIF model to circumvent the need for experimental data [38].

- First use of wavelets in HIF detection [13].
- First active detection method based on the network impulse response [72].
- Emphasis on HIF detection based on the harmonic content of current signals [70, 82–84].
- Deeper discussions on the trade-off between dependability and security given by possible actions to a HIF detection and utility liabilities [52, 68, 73, 75, 89].
- Despite exceptions, methods still heavily relied on *a priori* human expert knowledge.

2.3 The 00s — Wavelets, machine learning, and specializations

Although falling short in terms of innovation, the 00s were responsible for important incremental contributions to previously proposed techniques and sub-field specializations. Many feature extraction variations were proposed as novel contributions, forming a competitive scenario in signal representation approaches such as wavelet and Fourier. Being mostly hand-engineered and relying on *a priori* knowledge, the new proposed features can be mainly classified as bottom-up approaches to HIF representation. Detection methods using such features, however, started to increasingly have their decision boundaries defined by machine learning techniques (*a posteriori*), instead of arbitrary thresholds. For the most part, most valuable contributions from this era probably came from sub-field specializations such as in specific conduction surface, network grounding types, and sensor-based sampling technologies.

2.3.1 Wavelets and Fourier

Wavelet-based methods surged in numbers and popularity with many feature extraction variations proposed as novel contributions. One of the first works of this decade [90] is an example of such an approach. It proposes a different WT decomposition where the frequency bands are linearly spaced in the frequency domain, named wavelet Packet Analysis (WPA). In addition to being the first to apply WPA to HIF detection, the authors also took into consideration the wavelet components phase distribution w.r.t. the current fundamental cycle. Such phase distribution becomes relevant when considering that the voltage breakdown phenomena will generate discontinuities near zero crossings of the fundamental voltage signals. Its downside, not surprisingly, is that not all HIFs present evident breakdown phenomena (further explained). Another feature variation on the

coefficients of the WT, now in its traditional Multi-Resolution Analysis (MRA), is also presented in [91]. The work calculates the RMS conversion on the detail coefficients extracted from current signals as features used to establish decision boundaries for fault detection. The features are fed to the classical machine learning technique named Nearest Neighbours Rule (NNR). In a similar way to its more known version, K-Nearest Neighbors (KNN), this technique is a simple non-parametric classification method that established decision boundaries based on votes from the nearest data points in the feature space. This work's [91] specialization factor, however, is the consideration of distributed generation in the power system simulations. As in many other works further discussed, its solely modelling approach is somewhat simplistic in regards to the complexity of the phenomena, which makes it hard to defend it as a conclusive finding. Further papers [92, 93] proposed a different heuristic for classification and an important specialization consideration: network neutral earthing type. The feature extraction is performed by the application of the WT on the neutral/residual voltage and current signals. The detection is done by a heuristic on a measure of phase displacement between current and voltage coefficients [92] and a perceptron-based classifier [93]. In its purely simulation-based analysis, the authors conclude that the most challenging faults to be detected were the ones simulated in a compensated network (resonant grounding type) due to the significantly reduced ground fault current. Still in wavelets, the authors in [94, 95] proposed variations in two very similar papers. Both use WT for feature extraction but different approaches in fault detection: (1) Genetic algorithm for feature selection and a Naive Bayes classifier for classification, and (2) Principal Component Analysis (PCA) for feature selection and ANN for classification. The works are relevant due to their use of real, staged tests, intricate machine learning algorithms, and optimization methods. However, the relatively small number of HIF staged test opens the work for criticism when ANN is applied for classification. The complexity of the method in relation to the number of used features makes the approach prone to *overfit* the dataset, and possibly less generalizable to new data. Adding this to the fact that non-fault examples came from simulations makes it hard to evaluate the real effectiveness of the approach, even with the application of intricate, powerful tools. Other worthwhile mentioning works presenting wavelet variations are papers proposing adaptive neural fuzzy inference [96] and Support Vector Machines (SVM) [97] as classifiers. In the first [96], the authors used six features: four derived from the wavelet coefficients energies from current signals and two from third harmonic and signal mean value. In the second [97], the energy of the details was extracted as features, having their dimension reduced by PCA, and then fed to an SVM classifier.

Various Fourier-based approaches shared the theme of feature extraction variation and specialization. Early in the decade, a paper [86] describing the use of harmonics energies from residual currents and voltages was presented. Again, its strength was mainly

correlated to its specialization approach. In this case, it was related to a hardware prototype implementation of an ANN-based classifier, which was indeed novel for the time. A following interesting approach [98] using harmonics and its phases to draw 'Harmonic Patterns/Phase Portraits' was soon after introduced. By making use of the third and fifth harmonic, phase portraits were drawn and used as patterns that could discriminate a fault occurrence from nominal system operation. Although interesting, it was a convoluted heuristic that did not get much attention after proposed. Later, a simple and effective approach using FT [99] became very influential, including for the work undertaken in this thesis. It consisted of merely doing the FT of the current signals and using the components to learn a machine learning algorithm, named 'Decision Trees' (DT). As yet another purely simulation-based work, the authors granted in their conclusions that wavelets could have helped enhance the performance of the classification results (performed in this thesis). If considered as Fourier-based approaches, proposed Kalman filtering techniques [100, 101] for features extraction should also be mentioned. The authors of these two similar papers studied the use of the estimated harmonics with an ANN [100] and an SVM classifier [101].

After discussing some of the most influential wavelets- and Fourier-based works, a couple of important remarks regarding their adoption and HIF modelling should be made. The first concerns the increasing use of wavelets to represent fault signals. Since the first impactful work in 1998, wavelet-based methods went from a fringe idea to a prevalent approach in the surveyed works. One could argue that such an effect, and the ever-present use of wavelets, is evidence of its superiority in representing HIF signals over Fourier-based approaches. The second remark concerns the overwhelming number of proposed methods based solely on HIF model simulations. All the Fourier-based methods mentioned in this section exclusively rely on models rather than data from real experiments. From wavelet-based methods, more than half rely on models when conceptualizing their approach. On the one hand, this practice is not surprising given the fact that experiments are so onerous and simulations are so accessible. On the other hand, it is also not surprising that commercial methods do not use most of these variations, and they are not regarded as robust and defensible evidence. This assertion is especially true if one accepts the presupposition that HIF behaviours are not fully represented in current HIF models.

2.3.2 Model improvements

Despite the challenging nature of the problem, essential works regarding HIF modelling were presented in this period. The authors of the first paper [16] argued that the models previously presented did not represent important HIF behaviours such as *Build-up* and *Shoulder*. Build-up is defined by the period where fault current grows to its maximum local

value, which last approximately tens of cycles, according to the authors. The shoulder is the period where the build-up temporarily cease for a few cycles until it starts growing again towards its maximum global value. The authors intended to propose a simpler and more realistic model incorporating such features that did not have as many components as previous ones. It was conceptualized by just two series time-varying resistances controlled by TACS in Electromagnetic Transients Program (EMTP). Illustrated in Fig. 2.4, this model was inspired by thirty-two tests performed by a Korean power company, sampled at 10 kHz. It attempted to account for the proposed features by modelling one of the time-varying resistances with periodical changes (shoulder), while the other monotonically decreases from its large initial value (build-up). Despite impactful, it is not hard to find issues with this approach. Some examples are the number of tests used, the absence of information around the fault surface, and more importantly, how it was validated. The authors seemed satisfied with the fact that the current curves from simulated and real data closely overlapped visually and had close harmonic content. A further work [102] also used the same two-resistor scheme to propose a HIF model based on arc theory. The authors attempted to propose, in a full bottom-up approach, a model that would mimic surveyed HIF features described in the literature and their arc-like characteristics. Notwithstanding, it is hard to defend such a methodology since there was no verification with real data.

2.3.3 Earthing specialisation

Another important contribution proposed in works from this decade was the specialization on different types of neutral grounding. In 'Multi-grounded' or 'Solidly grounded' systems, a ground-fault current is limited by the series line impedance. In an 'Ungrounded' system, the limiting factor is the stray system capacitance. The difference in the circuit diagrams is

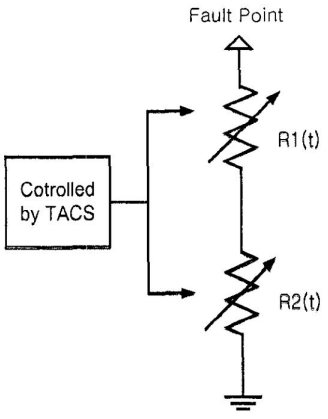
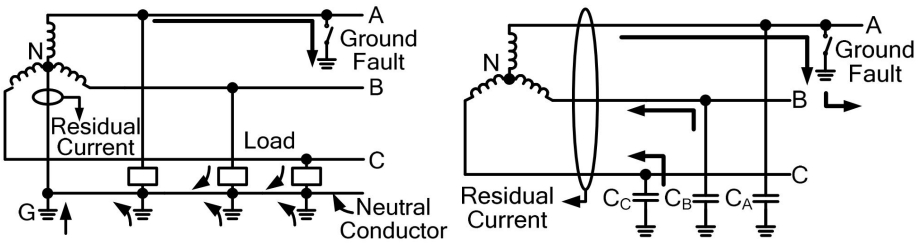


Figure 2.4. Nam et al. HIF Model (2001).
Adapted from [16]

illustrated in Fig. 2.5. As these capacitances are the only current return paths in ground faults, they are the sole responsible for the existence of fault currents in ungrounded systems. This fact makes such a configuration to have major benefits: lower ground-fault current, better personnel safety, and more reliable service [21]. The latter is achieved by the fact that a ground fault will not alter voltage between phases. Two- and three-phase loads can then simply ride through them. Its consequent downside, however, is the increased voltage level in the other two phases created by the shift of the neutral point. The phasor diagram of such a scenario, illustrating how phase voltage can increase to two-phase voltage levels, is depicted in Fig. 2.6. The practical application of neutral grounding types in real systems is, nevertheless, not so polarizing. A Neutral Earth Resistance (NER) can be placed between the neutral and earth, making it a high-impedance neutral earthing system. Even more sophisticated, alternatively, a variable high-impedance reactor connected to the neutral point can be adjusted to compensate the stray system capacitance. Such a reactor, illustrated in Fig. 2.7, is known as 'Peterson Coil', arc-suppression coil, or ground-fault neutralizer. Three-phase networks with this type of grounding are often referred to as resonant-grounded or compensated systems. Ground fault currents in this type of systems can be reduced to about 3 to 10 per cent of that for an ungrounded system. They can be found in Northern and Eastern Europe, especially in Nordic countries [61, 103], China [104], and Israel [105]. The differences between these grounding-type systems are much relevant to the Australian scenario since, as discussed in section 1.3.1 of the Introduction chapter, legislation has been enforcing and mandating utilities to install resonant grounding with Rapid Earth Fault Current Limiters (REFCLs). More information on ground fault protection methods for different neutral grounding systems can be found in a detailed and professional description by *Schweitzer Engineering Laboratories* (SEL) [105].

Discussions of HIF detection in non-effectively grounded systems in the literature started in the middle of the decade with the aforementioned wavelet-based detection work [92]. The authors propose a detection method on the phase displacement between the zero-sequence voltage and current, arguing that it would be more challenging to detect



(a) MultiGroundedSystem (b) UngroundedSystem

Figure 2.5. Types of neutral earthing.
Adapted from [14]

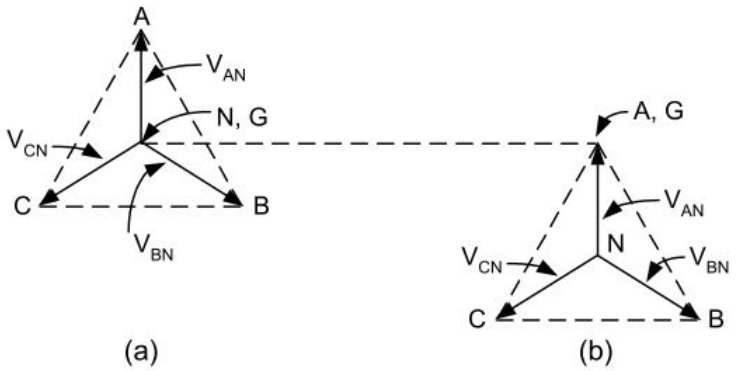


Figure 2.6. Voltage phasor diagrams for ungrounded systems.
a) Unfaulted system. b) Faulted System.
Adapted from [105]

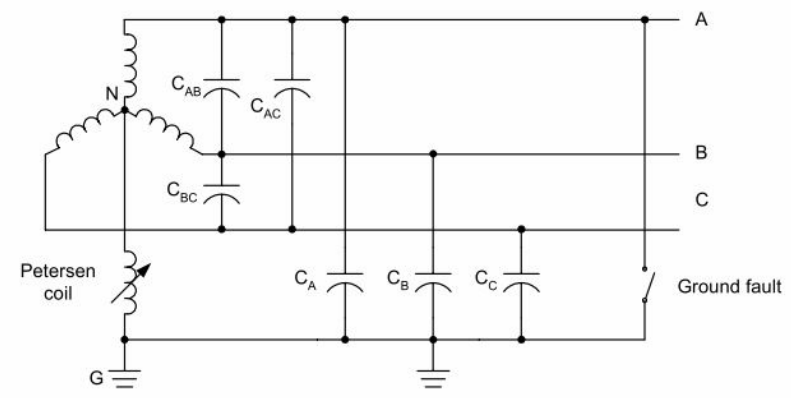


Figure 2.7. Diagrom of a compensated system.
Adapted from [105]

HIFs in such systems due to their smaller fault current. However, a later analysis [21] by a SEL engineer made the case that, in such systems, HIF could be more deterministic and accurate by just having highly-sensitive measurements on the ground residual current (see Fig. 2.5b for residual current representation). With detailed circuit theory, the authors proposed faulted feeder and faulted phase identification with residual voltage and current phasor analysis. Nevertheless, researches from Finland and Egypt were convinced that current resonant systems' technology to detect tree-related HIFs needed improvement. In a series of three publications [103, 106, 107], the authors proposed the most relevant novel specialization for this thesis: tree/vegetation HIFs. In their first paper [103], laboratory experiments were set-up to conceptualize a leaning-tree HIF model. The outcomes included an arc theory-based model and a wavelet-based HIF detection method. Features were extracted from residual voltage and current signals, and HIF occurrence was detected by a simple wavelet coefficients summation. The authors were satisfied with the resulted model, which closely approximated the V-I characteristic curve of a small number of experiments.

The method scheme, with validating simulations, was further presented in their second paper [106]. One of their conclusion was that the fault behaviour was also dependent on the network characteristics, making the transients used in the detection less sensible for faults distant from the measuring point. Therefore, they also had to include and advocate for wireless sensors that would be distributed throughout the network. Such wide-area monitoring through sensors was one of the innovative ideas that got traction in the 00s; it was also presented in their third paper [107], which used new data from a few tree HIF tests staged on a real feeder to validate their previously proposed method. Despite different from the methodology proposed in [21], both authors pointed to the consensus of using residual current and voltages to detect HIFs in non-effectively grounded systems.

2.3.4 Sensor-based and commercial approaches

Regarding the use of wireless technologies, an increasing number of works started to propose and discuss the idea of using distributed sensors to aid in HIF detection. In one of the first influential works from this decade, researchers from Brazil [108] proposed an innovative sensor-based approach. It described a sensor to be placed in power poles that would be sensitized by the electric fields produced by the conductors on the primary feeder. With strategic placement, this single sensor was able to detect three-phase voltages unbalances that would indicate the occurrence of a broken conductor. In addition to the presentation of the detecting device, the authors also suggested a methodology based on powerline carrier communication for signal transmission. The cost could be pointed out as a constraining aspect of the presented scheme. The authors explained that although the rest of the device was relatively low-cost, the capacitor used in the transmitter coupling costed around thirty-five thousand dollars. Moreover, a different sensor approach targeting covered conductors in Finland's systems was proposed [109]. The sensor based on the 'Rogowski coil' — an air-cored coil around the feeder conductor separated by polyethylene isolation — was set to measure leaning-tree fault 'Partial discharges' (PD) pulses. The authors argued that such coils have the advantage of possessing a high signal to noise ratio in wideband frequency response. Hence, they would be effective at detecting short and high-frequency PDs generated by leaning trees. Its strengths are the certain possibility of being a less expensive coupling and non-invasive installation. Its possible main disadvantage, however, is the consequent need for communication methods and data processing. Invested on the approach, nonetheless, the authors continued to publish on the use the device in a further review [110] and in an attempt of validated their method [59]. It is important to note that, these publications mainly targeted the PD detection research field. The devices were not described as a HIF but as a PD detector. However, as a leaning tree that touches a faulty covered conductor in a non-effectively grounded system is most probably a HIF, this

work represented a new literature intersection of the PD and HIF detection fields. A further work proposing PD sensing in overhead distribution lines [60] using frequencies as high as 25 MHz with Rogowski coils can also be cited as an example of such an intersection. Ultimately, it is hard to deny the possible benefits and accessibility that such sensor-based approaches could bring. They probably represent the most likely candidates for future HIF sensing and detection. Recent works and patents granted with very similar ideas [111, 112] can also be counted as evidence for the interest in sensor-based sampling technologies.

On the professional/commercial field, development and discussions of solutions from key industry players continued to evolve. Competitor companies such as *ABB* and SEL felt the need to propose their HIF solutions following the pioneering efforts of GE. In 2004, ABB published an announcement of its HIF detection philosophy to be included in their feeder protection devices [113]. Sampling signals in 32 samples per cycle (1.6 kHz at 50 Hz fundamental) the method comprised of a combination of wavelet analysis and ANN. High order statistics were applied to the wavelet coefficients while a two-layer network was fed five cycles of raw current samples. Although mentioned in the document as a 'voting system', the decision logic on the wavelet features and ANN outputs was unclear. Such a methodology, however, attested for the interest of the company in using trending and modern bottom-up and top-down approaches. The same could not be said for the algorithms proposed by SEL [87]. In the paper, the author argued that a deterministic methodology using traditional relay logic would be easier to understand and simpler to implement. In a somewhat contradictory stance, favouring deterministic approaches and dismissing 'black-box methods' such as neural networks to detect the faults, the author proposed a method consisting of a heuristic applied to the current signals. It starts with the extraction of its single main feature: a running average of a quantity named 'Sum of Difference Current' (SDI). Defined as a one-cycle differentiator set to represent all the cumulative effect of non-harmonic frequencies, the SDI can allegedly assert the occurrence of arcing HIFs. The feature is compared to a trend, which could be adaptively tuned for different environmental conditions, and sent to the decision logic algorithm. The classification label is then given by a counter responsible for tracking how many times the SDI exceeded the threshold within the previous seconds. Validated in apparently four tests in the disclosing paper, this methodology would then evolve to the current commercial 'Arc Sense ^{TM} Technology' present in many feeder protection devices by SEL [114]. The same author also published a paper reviewing HIF detection in systems with different grounding types and describing the performance of the algorithm in staged faults [14]. In this analysis, strong arguments were presented for the specialization hypothesis by criticizing methods that claim specific detection rates without specifying distinct conducting surfaces. From GE and TAMU, however, only one paper describing the field experience with their previously proposed HIF detection algorithms was published [3]. The discussion revolved around a

power company's experiences from installing relays with HIF detection in 280 feeders over a period of two years. Being the first to do it on such a widespread basis, interesting findings were presented. For example, interviews with line crews indicated that around one-third of downed conductors were still energized when they got to the pointed location. In the company's log, forty-eight out of the seventy-one confirmed faults had data available. Forty-six of them armed the relay (96%) but only twenty-eight were detected as downed conductors (58%). The authors explain such difference was due to the bias towards security purposely programmed in the algorithms to have the smallest number of false-positive as possible. Overall, the commercial works from these key industry players asserted, with their individual and competing solutions, the relevancy and importance of HIF detection problem.

Lastly, brief comments on novel approaches regarding HIF location are worth making. Its relevancy becomes apparent when considering tripping a long and branched faulted feeder where finding such a HIF by visual inspection may take a considerable amount of time. This decision will also depend on the importance of the supplied loads and environmental conditions. Quickly locating HIFs would thus be always advantageous, possibly increasing service continuity and chances of preventing damaging fires. Such an important idea was discussed in a simple but influential work [115]. In a fully simulation-based approach, the authors learned an ANN with sequence currents and their harmonics to estimate the fault location. It is not hard to find many issues with this methodology, despite its novel idea. It neglects the fact that the current amplitude, network topology and harmonic content created by the load can significantly vary. It used an apparently small amount of data when training the network and made stretching considerations when validating its results.

2.3.5 Highlights

Overall, most noticeable developments and contributing ideas from the 00s can be summarized:

- Wavelet transform as a signal representation technique surging in popularity, equating to Fourier-based transforms [90–92, 94, 96, 97, 106].
- The ubiquitous use of HIF models, leading to an overwhelming amount of simulation-based works [16, 86, 91, 92, 96, 97, 99, 100, 106, 116].
- Consideration of different types of neutral grounding as a HIF detection specialization [21, 92, 103, 103, 106, 107].

- Discussion on the use of sensor-based approaches as a possible non-invasive, low-cost, sampling technique [59, 60, 108–110].
- Presentation of commercial solutions from other key industry players [87, 113, 114].

2.4 Contemporary literature — More specializations, sensors, and fault location

The research published after the 10s, considered herein as contemporary literature, represents the largest number of publications when compared with other decades. It is not clear that the same could be said in terms of innovation in the field. Most contributions probably came from more specializations in the HIFs surface types, continuing investigations on sensor-based sampling approaches, and more in-depth discussions on HIF location. The latter showed a slight change in field direction, suggesting some transcendence on the problem of detection. It is possible that some researchers were satisfied or saturated with the large number of approaches targeting fault detection and wanted to progress the field to more significant challenges. Such possibility becomes more evident if one investigates the number of replication works, i.e. works that do not present more contributions but instead apply a slight variation of an already proposed technique, published in this period. Notwithstanding, discussions and findings presented in specialization works assert the fact that much still needs to be understood about the HIF phenomenon behaviour.

2.4.1 Modelling specialisation

Unsatisfied with the current state of HIF modelling, researchers continued to propose new ideas for improving phenomena representation. A model proposed at the beginning of this decade [117] aimed at building on the first described model [38] to increase the represented frequency band to components up to 12 kHz. By using 40 tests staged on many types of surfaces such as asphalt, cement, soil, and tree, the proposed model consists of six branches of the previous model (see Fig. 2.1) in parallel. When fitting the model parameters, the authors used FFT for extracting features from the current signals, PCA for dimension reduction, and an iterative minimization on the Bonferroni interval as cost function. The work tried to fit different types of HIFs in a single model, which would make it deficient if high-variance between surface types, as defended here, is indeed present. A further work [26] addressed this hypothesis and presented some evidence in its model proposition. The authors, decided to build on another previously discussed work [16], proposed a model with the same characteristics (see Fig. 2.4) but with parameters individually fitted for each surface. On their results, one can see a significant variance in the impedance parameters

and current waveforms for each of the studied surfaces. With approximately ten tests each, a variety of surface types were considered: grass, cobblestones, gravel, asphalt, sand, and local soil. The paper [16] probably presents one of the most consistent methodologies in the current literature but yet not the most pertinent to this thesis.

A recent modelling approach [28] presents itself as the most relevant to this thesis by proposing an important model specialization: tree-related HIFs. This type of faults are especially crucial due to their implications in HIF-related fire ignition and that they can have significant, particular characteristics. Two different behaviours described in the paper, from the initial phases of conduction (that can last for several seconds), are extremely relevant for this thesis: (1) the fault current is smaller than other surface types, in the range of milliamperes; (2) its impedance behaves closely to a linear resistor, meaning that there is no arcing and no significant levels of harmonic injection. Results from this thesis and another HIF-features work [20] corroborate these robust findings, attesting for the recurrent mA-range fault currents and smoother conduction near zero-crossings. The data used to conceptualize the model [28], despite being collected from a small number of real, staged tests, was collected with a novel approach by sampling signals at 1 MSa/s. Such sampling rate is much higher than the vast majority of works discussed herein and allows for a better representation of the fault HF components. Its parameters were calculated using the Hammerstein-Wiener model to fit low-frequency components of the fault impedance, while the high-frequencies were approximated by a sum of sinusoids determined via the least-squares method. Moreover, in the same paper, the authors also proposed an approach for detecting the faults based on the innovative idea of using the Magnetic Field (MF) strength instead of voltage or current signals.

Recent tree/vegetation HIF specialization works highlight the relevance of particularly addressing these faults. The authors of the model [28] published two more works on tree-related HIFs features and detection [29, 118]. One used the empirical mode decomposition to extract features and the linear regression slope on resulted components quantiles to classify the faults [118]. The other [29] presented the same idea but discussing the use of the magnetic field strength as the domain of the extracted features. The authors argued that using MF sensors would be a more accessible way of sampling the signals and that the MF strength signal is independent of the sensor location on the feeder. The latter, if true, would mean that one sensor would be enough to monitor the whole feeder. However, the authors did not present compelling evidence for this strong claim besides finite element model simulations from part of the studied system. Further questions can then be raised given that no modelling details on the system stray capacitance were detailed, which will attenuate the small fault current (magneto-motive force) in considerable distances. Concerning further tree/vegetation works, the papers published from the works described in this thesis can be cited as part of this specialization [31, 55, 56].

2.4.2 Novel domains and signal representations

Magnetic field sensing is one of the novel ideas discussed in contemporary literature. An innovative work [119] discuss such an approach to replace the three current sensors mounted under each phase with one MF strength sensor. In its feature extraction part, the additional novel idea of *mathematical morphology* for signal representation is used. A SVM classifier performs the classification with feature selection done by a genetic algorithm. The results proposed, in a partly simulation-based approach, show considerable detection accuracy with a fresh, attractive, solution. Similarly, a following recent work [120] proposed another non-invasive MF strength sensor as a continuation of a previous detection method [121]. With the goal of presenting a low-cost method, a coil-shaped sensor to be mounted under (not around) the primary feeder conductors was proposed. The classification of the faults was based on the inter-harmonic current level, as previous work [121], estimated by an accessible Arduino microprocessor. Despite being undoubtedly novel, it is hard to assess the real effectiveness of these methods in the field. They are partially or fully simulation-based methods that still aimed at detecting all HIFs types with a single approach. Nevertheless, they are indeed evidence for the direction of preference in using non-invasive, sensor-based, approaches to sample signals able to indicate HIF occurrences.

Wavelet-based feature extractors maintained their popularity, but other possible signal representation candidates were also discussed. With the assertion of the field relevancy, accessible HIF models, and popularity surge, the idea of using wavelets to extract features was quickly explored in the past decade. This saturation led to an increasing number of papers presenting replicated methodologies with very few novel aspects. A fraction of the surveyed works herein was selected to be discussed in this section. From the chosen ones, none presented the use of wavelet as its main contribution, with a few using it as support for their primary specialization.

Mathematical Morphology (MM), used to analyse spatial structures in the field of image processing, was one of the novel signal representation ideas introduced. It uses a structuring element and two most important concepts, Dilation and Erosion, to encode information regarding the form, shape, and size of structures. The application in HIF detection is mainly made by using it to represent the particular irregular shapes of HIF-generated waveforms [85, 119, 122]. Such application, however, was based on two assumptions with thin evidence: (1) the arbitrarily chosen structuring element will be adequate to represent the transients created by the fault, and (2) the consequent representation will exclusively represent HIFs. In the first paper [85], the authors used a variation of MM to propose a feature named multi-resolution morphological gradient. It was used in an ANN-based classifier learned from a data set composed of a mixture of a small number of real, staged

tests and a large data set from simulations. In their results, the authors argued that the experiments showed MM to be more effective than WT and Fourier-based transforms at representing HIFs. The previously discussed MM-based work [119] is a translation of this approach to the domain of MF strength signals. In the following years, another fully simulation-based work [122] also made a case for using MM to represent HIF signals with just a change in the structuring element and a threshold-based heuristic.

Other two relevant works using different signal representation are also worth discussing. The first relates a work, quickly recognized in the literature, that used a time-frequency analysis based on the Choi-Williams distribution [123] to represent signals. Intending to propose a simple and effective method, staged tests with tree branches, grass, and concrete were performed in a laboratory as part of the methodology. Its feature extraction, going in a different direction as most works in the field, was performed by time-frequency decomposition followed by a joint time-frequency moment calculation. After the application of PCA for dimensionality reduction, the results from learning a SVM classifier showed perfect dependability but deficient security. Its relevant influence on the field was not only resultant from its consistent methodology but also due to its discussion on establishing evaluation criteria for future proposed methods such as cost, objectivity, speed, and completeness. These criteria, despite previously discussed in past methods, are not always present in method-proposing works. They are usually discussed individually when the proposed method wants to highlight a particular advantage in related criteria. In this critical discussion, the authors also drew attention to the need for a systematic presentation of future methods. These standards would include reporting concepts inspired by the machine learning literature: confusion matrix, accuracy, dependability, security, safety, and sensibility. The authors soon followed this work with an important analysis [53] where a similar detection method performance was compared considering data sets with different origins: simulations (using a HIF model) and real data. The expressive results were a proxy of the effectiveness of simulation-based works at representing real-world scenarios. The method's security reduced from 100% in the simulations to 38.4% in the real data scenario, while dependability went from 100% to 88.2%.

The second work addressing signal representation presented an innovative mathematical method for analysing the fault signals [124]. The authors proposed a whole novel orthogonal decomposition where the basis functions were derived from the actual fault signals sampled in staged tests. The main advantage of this approach is non-reliance on a predefined set of basis functions like the ones present in the Fourier and wavelet transforms (further discussed in the sparse coding application section). The authors argued that such decomposition was highly effective due to its sensitivity to phase unbalances present in a HIF occurrence (phase-to-ground fault). Moreover, they tried to make a case that the resulted components highly correlate with the fault distance and thus could be used to

guide fault location. The method was validated with the use of real data, and promising results were presented for relatively moderate current amplitudes (<20 A).

It is hard to ultimately evaluate the discussed novel signal representation works as better or more effective at representing a fault occurrence than previous approaches. The validation performed in different data sets masks any possible objective comparative measurement. Nevertheless, they are valuable and strong candidates which much inspired this thesis's methodology.

2.4.3 Focus on HIF location

Altogether, no subject presented more novel discussions than HIF location. Most fault location methods categorized as *exact* approaches can be divided in *travelling wave* or *parameter estimation* technologies. Other proposed techniques can be categorized as fault location *estimators*, as they are used as support in search of the fault location, usually reducing the fault search space.

From travelling wave-based methods, a work presented in a series of three papers [125–127] targeting HIFs can be regarded as one of the most influential. It was conceptualized to detect and locate a HIF using Power Line Communication (PLC) devices. The detection was performed by a PLC device that continually monitors the feeder impedance and detects the fault when an abrupt change in the HF impedance is asserted. After detection, one PLC device (transmitter) starts to inject impulses into the network, which are to be received by another PLC device (receiver). Based on the travelling wave phenomena, the received impulse by the receiver, and the reflected impulse on the transmitter, the exact fault location can be theoretically calculated. In this manner, the fault location always occurs between two successive communication devices. The methodology, first conceptualized for rural single-phase rural networks with earth return [125, 126], was further generalized to multi-phase systems [127]. The results presented in their simulation analysis for fault location are promising but also related to severe constraints. For example, it requires knowledge of the topology, impedance, and resonant frequencies of the system for the best narrow frequency range selection. Since the detection and location system merely indicates changes in the system topology (HIF as a small topology change), it has to consider such a topology to be stationary. The faults were simulated as constant impedance, as are the loads, far from the complex behaviour comprehensively described herein. Moreover, as it considers the end of the line as open-ended, band-stop filters may be required on the primary side of low-voltage transformers.

One of the most relevant parameters estimation-based works [128] also suffered from strict constraints. It proposed the calculation of the fault distance with a time differential filter on the feeder current. In the methodology, the time-domain subtraction on the current

signals is performed from subsequent cycles observed in the feeder so the fault parameters can be estimated. If an abrupt change in the parameter calculation is asserted, a fault detected. Then, the fault distance is given by a polynomial estimation of the parameters with Newton's gradient descent method. Another recently published work also uses a similar differential current approach but with parameter estimation done via ANN [17]. The authors claimed that such ANN would not need prior training as it was conceptualized to train on-line as it acquires data. The results were compared to the method previously discussed [128], resulting in less estimated distance error. Many issues could be pointed out with these works, but probably the most relevant is the fact that both built their methods on the presupposition that the fault parameters and non-linearity are known. The models used by the authors may reflect the non-linearity of certain HIF types but will, at best, closely represent a single type of fault. These methods, moreover, are conceptualized with currents much bigger than the ones studied herein, in the range of 60-100 A.

Other fault location search supports are also recently proposed. As the ones formerly discussed, they require full knowledge of the system topology, loads, and system accurate modelling. A work using wavelets in MRA [129] proposed HIF detection and location by comparing the values of signals coefficients with a pre-developed data set. Such data should come from system simulations for many faults in all the branches, so calculated values can be compared and ranked in regards to distance. Another work [18] that proposed HIF detection with the use of distributed measurements throughout the network described fault location estimation by how intensely these were sensitized. Thus, with a reasonable number of allocated monitors, the fault search space could be significantly reduced.

For the most part, fault location solutions are still in the early stages and have many obstacles to deal with until attesting generalization. Some intrinsic characteristics of power distribution systems pose critical constraints to the effectiveness of these theoretical solutions: the conductors' size change, making impedance calculations non-linear; possible existence of multiple feeder taps and laterals; non-linear and non-effectively modelled HIF behaviour; phase imbalances; and inaccurate load representation/aggregation. In fairness, however, one should not expect that works from an emerging sub-field as HIF location to have all obstacles solved. Future development in this field is probably going to use different strategies to be more generalizable and have considerable potential to improve with better HIF models and sensor-based technologies.

2.4.4 Highlights

With the largest number of works proposed, contemporary literature was responsible for relevant contributions and many validations studies. The highlights from this research period can be summarized:

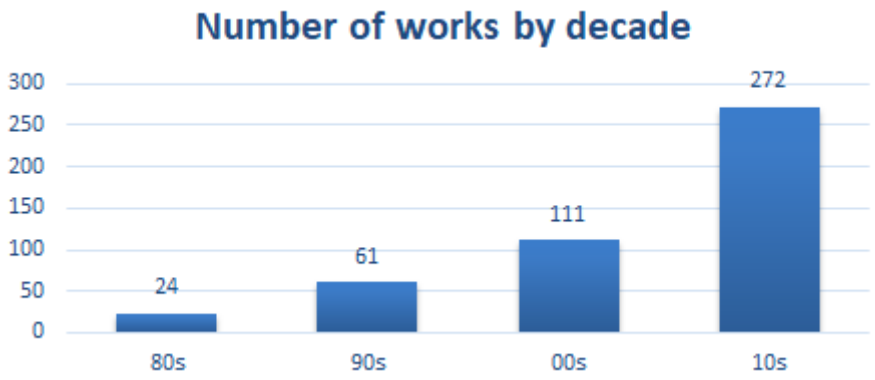
- Modelling and detection specialization of tree/vegetation HIFs [20, 28, 29, 118].
- Increasing number of sensor-based solutions, pointing to increasing interest and consensus on their advantages [111, 112, 119, 120].
- Magnetic field strength sensors as the sampling domain for HIF detection [119, 120].
- Deeper discussions on HIF location asserting it as a legitimate HIF sub-field of research [17, 18, 127–129].

2.5 Commentary and targeted knowledge gap

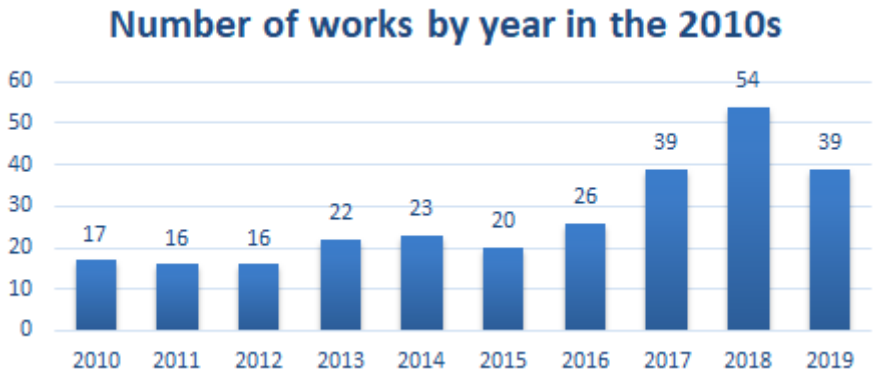
Although used throughout this chapter, the use of qualitative terms to describe the HIF detection field and the distribution of proposed methods was tentatively avoided. When such statements were done, however, they were based on a literature survey made throughout the author's candidature. To bring such a survey closer to a quantitative analysis, this section illustrates and discusses some statistical figures calculated from the analysed works. Before beginning such discussions, it is worth to note that there is a difference between works analysed in detail and those considered just for the quantitative evaluation. The amount of papers and technical documents analysed in detail are more limited: 117 total works, including 88 HIF detection papers, 9 HIF modelling methodologies, 7 HIF location papers, and 3 literature reviews. The remaining of the 117 papers include method validation works, field test documentation, tutorials, related technology, and patents. Although being less than the total amount of papers published in the topic, it is hoped that the set of works analysed in detail will serve as a representative sample of the direction and distribution of techniques used in the field.

The first quantitative figures are presented to support previously made statements about the history and growth of the field. The number of papers published by decade and by year in the IEEExplore repository is illustrated in Fig. 2.8. As seen in Fig. 2.8a, the number of publications consistently increases throughout the decades. It more than doubled twice, and more papers were published in the 10s than all previous decades added together. In the 2010s, papers were consistently published every year with an overall positive trend from the beginning to the end of the decade, as shown by Fig. 2.8b. The number of papers published in 2018 was certainly a record and outlier; the reasons for this is not known. The number of works analysed in detail, mentioned in the last paragraph, is a set of the works presented in Fig. 2.8b as it includes all the papers listed in IEEExplore with the words 'high-impedance fault'.

It is also useful to illustrate how the techniques to extract HIF features from fault signals are distributed in the previously presented methods. As repeatedly discussed in



(a) Papers published in IEEExplore by decade.



(b) Papers published in IEEExplore in the 2010s by year.

Figure 2.8. HIF field popularity and growth.

previous sections, signal representation and feature extraction techniques are often the primary contribution to knowledge attempted by HIF detection works. The pie chart shown in Fig. 2.9, produced from the papers evaluated in detail, shows the distribution of techniques used to extract the discriminative information from fault signals. In the figure, 'MM' represents Mathematical Morphology and 'HH/EMD' signifies the Hilbert-Huang transform or Empirical Mode Decomposition. 'Wavelet' and 'Fourier' represent methods based on their respective transforms while 'Sequence' depicts the analysed methods based on zero and negative sequence currents and voltages. The 'Impedance' works are a few of the papers found where the detection is based on the apparent impedance measured from sampled voltage and current signals. The 'Others' category include alternative time-frequency analysis techniques, impulse response and travelling wave-base methods, time series analysis, and mechanical methods. From all the mentioned techniques, it is remarkable that wavelet-based methods clearly dominate the practice considering it was only popularized in the 00s. It is easier to find wavelet-based methods than Fourier-based ones, even though the latter were introduced in the beginning of the field. Likewise, it is

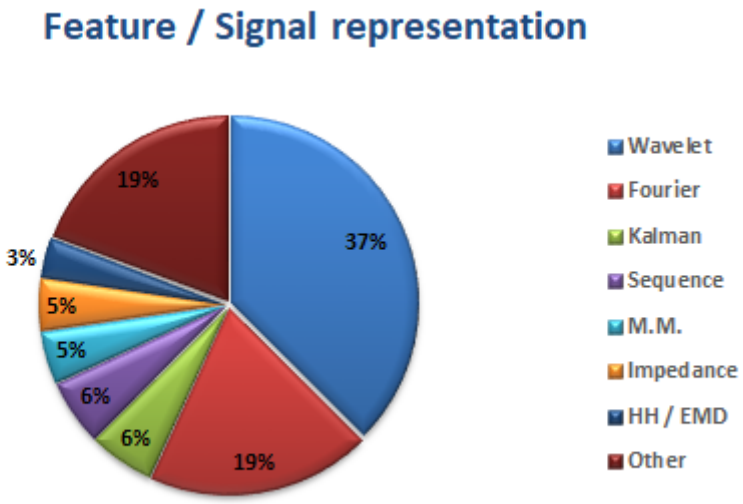


Figure 2.9. Signal representation techniques in HIF detection works.

also remarkable that the 'Other' category is well represented, matching the same proportion as Fourier-based methods. One can hypothesise that the reason for such a representation is due to feature extraction and signal representation techniques being often part of the proposed contribution to knowledge. As authors need to introduce novelty in their work, many publish on experiments on the use of not-previously tested techniques so they can have a claim of original work. These works are often not replicated, ending up in the 'Other' classification. Moreover, one last notable fact attested by Fig. 2.9 is that the majority of the works still use hand-engineered features. As explained in the first section of this chapter, hand-engineered features have the advantage of being closer to explaining causality but are also dependent on the human knowledge that creates them. One can induct two reasons for the dominance of these symbolic approaches: (1) researches are still not proficient with the use of techniques such as deep learning and encoding techniques, or (2) researchers do not believe that exploring such techniques are worth. The latter could be due to the lack of causality or trust in the methods uncertainty, or they could be disincentivized by belief that such approaches will not be well received by the community.

Demonstrating the distribution of choices in the classification techniques is as important as illustrating feature extraction approaches. They are responsible for the crucial task of classifying the signals as originating from a fault or not, which takes place after the discriminative information has been extracted. The practice mainly consists of establishing decision boundaries which will result in the classification of the observed data or feature. The technique used to classify the signals can take many forms and assume different levels of complexity; from a simple threshold value established on a calculated feature to complex decision boundaries defined by neural networks trained on labelled data. From

the works analysed in detail, Fig. 2.10 shows the distribution of classification techniques labelled as 'Deterministic' and 'Probabilistic'. The chart illustrates the interesting fact that probabilistic decision boundary techniques relying on observations are almost as represented as deterministic ones. The fact probabilistic techniques started to be adopted later than thresholds and arbitrary decision boundaries ones is the main reason for their representation being remarkable. This effect is present in the distribution of probabilistic techniques as well, also shown in Fig. 2.10. Artificial Neural Network-based approaches (ANN-based in the chart) surprisingly dominates the probabilistic techniques attesting for the high interest from the research community. Other machine learning techniques such as support vector machines, random forest, and k-nearest neighbours occupy 39% of probabilistic techniques.

It is important to make a note on the sampling rates used by the HIF detection works since statements were also previously made about their range. From the analysed detection papers, only 49 made their data acquisition sampling rate clear. Close to half of them (24) adopted sampling rates lower than 5 kSa/s while 22 adopted values between 5 and 50 kSa/s. Only 3 fitted the exception of sampling signals at rates higher than 50 kSa/s. Nevertheless, one of them was a purely simulation-based work while another was as active method based on travelling wave theory where a pulse gets injected into the line and its response is measured. The third was the only having an approach similar to the one in this thesis but still sampled signal at 64 kSa/s. The fact that higher sampling rates are not often adopted is not necessarily detrimental to the field. Having accurate detection with smaller sampling rates is desired since it is less demanding and closer to existing digitizers allocated in the field. However, if one accepts that HIF detection is a lasting problem to be still definitely solved, investigating the effects of faults at higher sampling rates becomes more interesting and perhaps necessary.

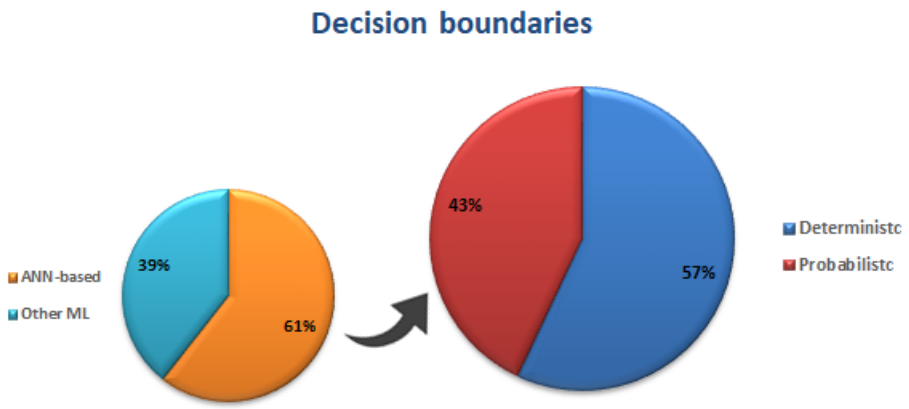


Figure 2.10. Techniques used to define decision boundaries in HIF works.

The proportion of works that were purely simulation-based or focused on vegetation as the primary fault surface type are also two aspects worth commenting. The introduction of HIF modelling in the 90s, as an attempt to circumvent the need of real data by the use of simulations, was effective at becoming a common practice. From the 88 investigated papers, 42 presented pure simulation-based works that made use of previously proposed HIF models to make claims about accurate fault detection by their methods. Nevertheless, given the variance between fault types presented in the literature and here, one could argue that methods based purely on modelling are still to prove their capability of generalizing to real faults. There is no consensus, as seen from the present survey, on the best way to model HIFs or that can they can adequately represent all types of fault surfaces. In regards to works that focus on vegetation as the particular fault surface, conversely, one can confidently state that they are rare. From the analysed works, 8 include some type of vegetation on their tests, but only 4 focuses solely on it as fault surface. From the research that is continuing, two group of researchers presented original contribution regarding vegetation HIFs; one of them produced the work basing this thesis and the other is located in Iran [19, 28, 29, 118, 130].

2.5.1 Knowledge gap hierarchy

Since the main aspects of the field have now been presented and illustrated, it is possible to localise the place this thesis occupies, with its potential knowledge gaps. However, it might be useful to first state the similarities this work shares with previous ones. To that, the most relevant aspect that should be mentioned are the choices in signal processing techniques. Throughout all the experiments performed in the signals, established and renowned signal processing techniques were chosen; the short-time Fourier transform, wavelet transform, and sparse coding are prominent examples. Nevertheless, their use should not be mistaken by how their results are used. Features or combinations of features derived from the application of these techniques are counted as original work but not the techniques themselves. The second pertinent aspect relates to the classification techniques. Despite the much trial and error and parameter tuning attempts, the techniques used to create decision boundaries for classification are known and acclaimed machine learning algorithms.

The knowledge gap that this work helps to address exist in a hierarchy of domains. Its root starts with the use of a data set with hundreds of staged faults with local vegetation species. No work surveyed or described in literature-review papers presents such comprehensive testing of local vegetation species. Local in this context compose another niche, making the findings and insights presented here more relevant to Australia than any other country. Moreover, the experiments were conceptualised with particular limitations

to the fault currents which are rarely cared for in the literature. When proposing detection methods, most papers do not dedicate effort to stablish thresholds or standards for the fault current. This fact is only consequential since it is not possible to know how sensitive are the results if the current is not carefully limited. The limitations on the experiments of this data set include the constraint the current values to 0.5 to 4 A, which are relatively small values considering the distribution level voltage that produce them. Having positive results in this niche current level also represents an unaddressed gap in this level of the hierarchy.

The data acquisition method adopted in the staged faults occupies the next level of the gap hierarchy. As mentioned in previous paragraphs, the vast majority of works do not cross the rates of 50 kSa/s when acquiring data from their experiments. Therefore, having signals sampled in such a large bandwidth, representing frequencies from 10 kHz to 1 MHz, represents the complement of a significant gap. The large bandwidth also allowed distinct experiments, for example, in which the predicting information content from two different bands could be compared and analysed. As shown in the Results chapter (4), having this sampling characteristic was crucial for the positive results further obtained. It is unfortunate, nevertheless, that the data did not contemplate the load current as the experiments were executed in a dedicated feeder. However, having to pivot the focus of the investigation to the voltage signals and obtaining favourable results nevertheless, also represents a contribution to a non-addressed domain. This contribution exists due to the simple fact that previously proposed methods focus heavily on the current signals as it is the domain that is most directly affected by the fault. Lastly, in this level, there is a characteristic of the data acquisition that represents another unexplored territory: the sweep sampling method. It was not discussed yet but soon to be comprehensively commented in the next chapters. It merely is the sampling of small, regular snapshots of the signals, instead of continuing sampling them. The noteworthy aspect, however, is that evidence presented here points that the sweep samples are enough data to produce an accurate detection method. This advantage is remarkable because being able to detect a fault with less data means less computational effort and lower-cost solutions for real implementation.

The results from the experiments discussed in this thesis relate to the high-level and more niche knowledge gaps in the next level of the hierarchy. The first aspect is the main result presented here, which is evidence of the possible detection of VHIFs at higher frequencies of voltage signals. The prototype results play a role of attesting that it could be done in real-time, with the added advantage of possibly being embedded in low-cost hardware. The results, moreover, not only present the possibility of accurate detection but also the representations of the fault signatures in the higher frequencies. Such unique representations, together with the experiments comparing the effectiveness between low- and high-frequency signals, add evidence and fill the gap of VHIFs phenomena

understanding at higher frequencies. To illustrate this discussed hierarchy, these aspects of contribution have been summarised in Fig. 2.11.

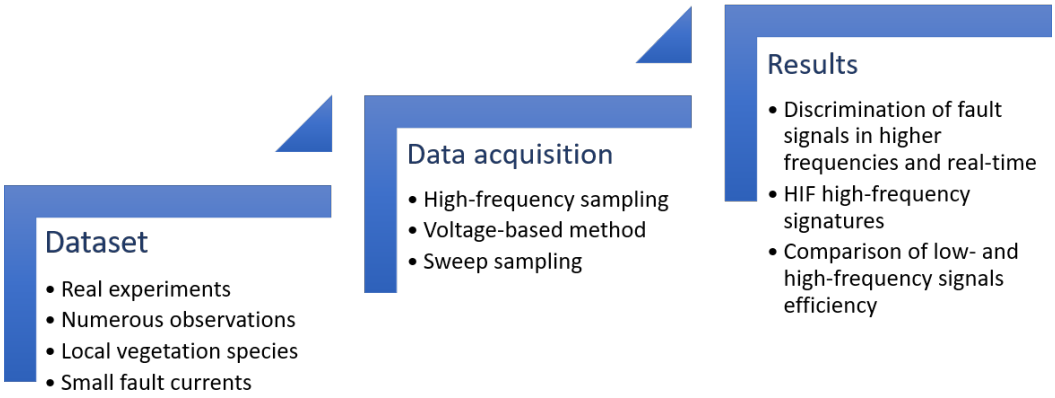


Figure 2.11. Representation of the knowledge gap hierarchy.

Chapter 3

Processing, classification, and experimentation methods

This chapter is dedicated to describe and discuss the techniques and methods that are central to this thesis methodology. Some of these techniques are renowned and established methods such as the Fourier and wavelet transforms, while others are proposed methods intended as original contributions. This chapter aims to inform the reader on the assumptions and choices made throughout the research period and to support the results described in Chapter 4. The latter aim inspired the decision of individually describing the used techniques in their separate respective sections. The author believes that, as the reader might be interested in a specific technique while reading the results chapter, such individual organization of techniques might allow for a better reading experience.

It is important to note that due to the plenitude of techniques available and time required for proficiency, there were considerable changes in the methodology direction throughout the research period. Different techniques were applied in the production of evidence of the relevance of the proposed solution. A choice was also made to describe the experiments as they were conceptualized. The reader may benefit from understanding the reasons that drove such changes and their possible impact on the results as their one of the many approaches to the studied problem.

It is worth mentioning that all the steps, including data management, feature extraction, classification algorithms, and validation were performed in the MATLAB environment with the respective toolbooxes: Signal Processing Toolbox™[131], Statistics and Machine Learning Toolbox™[132], and Wavelet Toolbox™[133].

Examples of the main MATLAB codes used in the research can be found in Appendix B. They were not written in the same organizational manner as this chapter, but contain all the functions and scripts discussed. The code for the shift-invariant sparse coding technique is not included but can be found in the author's website [134].

3.1 Data characteristcs and pre-processing

A core idea when proposing the present methodology was to leverage existing experimental data to propose a VHIF detection methodology. Such specific data was chosen due to its relevance to the problem and the fact that no formal work using it to create a fault detection method had been published at the time of this research inception. As explained in Chapter 1, the data was made public by the Victorian Government [49]; the description and analysis of the tests were given in the project's final report [4]. The original data set contain logs, video and signal recordings, graphs, and analysis of the tests. However, only the tests recordings and metadata were used in the following proposed experiments. From this point onwards, references to the *data* or *data set* adopted in the methodology are synonyms to the set of fault recordings and metadata.

The voltage and current signals were made available in a proprietary format given by the hardware used in their recordings. The data acquisition hardware was the HBM Gen3i high-speed data recorder, fed by four HBM GN110-2 optoisolated digitisers. Such a scheme meant that test recordings were sampled by four different channels. The proprietary format of the test files had an extension named '.pnfr' for Perception Native Recording File. They were designed to be opened with proprietary visualisation software (Perception), thus not directly accessible for manipulation. Each file was labelled with their respective test number and contained all four-channels recordings. Data importing to MATLAB environment was possible mainly by an API interface fortunately offered by HBM.

Two sampling channels were used for each current and voltage signals. Attempting at recording high-fidelity signals with an increased signal-to-noise ratio (SNR), the project team decided that would be beneficial to have a channel dedicated to higher frequencies, without the influence of the main power frequency (50 Hz). All the four channels were connected to an analogue 6-pole Bessel low-pass filter with 10 MHz corner frequency for anti-aliasing. A second low-pass anti-aliasing filter (Bessel IIR digital) with proper data decimation was added to the LF and HF channel; It had a cut-off frequency of 50 kHz in LF channels and 1 MHz in HF channels. HF channels were also filtered by a 10 kHz high-pass filter, characteristic from the capacitive voltage divider design. The resulting effective band of LF channels was approximately 0 Hz to 50 kHz, while the HF channels effectively comprised of frequencies from 10 kHz to 1 MHz. Please refer to Appendix A for a more comprehensive description of the experiment and sampling set ups.

3.1.1 Sweep sampling

The high-speed recorder continuously sampled all channels at 100 kSa/s, plus the additional measurements in the HF channels. Adding HF measurements were needed since the

experiment was designed to characterize frequencies up to 1 MHz. Such characterisation would allow the investigation of discriminative information in higher frequencies, i.e., fault signatures. According to the Nyquist–Shannon sampling theorem, a minimum sampling rate of 2 MSa/s is required to characterise frequencies up to 1 MHz. Nevertheless, adopting higher sampling rates meant sampling more data and generating higher data storage and management demands. These requirements inspired a decision to perform periodic snapshots of HF signals recordings at 2 MSa/s. Perhaps inspired by the period of a power frequency cycle, the duration of these small sampling periods called sweeps was set to 20 ms. The sweeps were triggered by a non-synchronous signal with 1 s period, meaning that HF recordings were composed of one sweep per second. To put in perspective, such sampling duration is enough to comprise 200 cycles at the lower band of the HF channel (10 kHz) in a single sweep. It also implies that sweep recordings duration represented 2% of the whole continuous time when compared to LF recordings.

In the recording files, each sweep resulted in a signal with 40k samples (2 MSa/s in 20 ms). The many sweeps per test were extracted and sequentially concatenated in a single vector for further analysis. The plot in Fig. 4.1 illustrate the LF and HF channel recordings of test #36. Despite presented in the same form as the LF channel, the HF signals appear continuous only due to the sequential concatenation of sweeps.

Having snapshots (sweeps) of the signals rather than a continuously sampled HF recordings can introduce challenges that require a coping strategy. For example, the lack of synchronism in the trigger that asserts the sweep recording makes it challenging to know the exact moment in time it refers to the parallel LF recordings. Knowing this moment in time is crucial since when using the HF sweeps as observational data in a learning approach, one needs to know which sweeps were recorded when a fault was actually happening (fault current conduction). The voltage supply and current conduction started almost immediately after recording began at some times. In other times, the fault current appeared more than a minute after signals started being recorded. Although current conduction usually started when the voltage supply was turned on, it was not true for all tests; some vegetation branches were able to withstand the voltage applied for several seconds before allowing current conduction. These exceptions allowed the existence of tests with valuable *pre-fault* recordings that were further used to validate the proposed methodology. Moreover, the staged faults extensively varied in duration, resulting in a different number of sweeps per test. A high number of sweeps is usually detrimental since only one sweep represents the subsequent second of fault inception that needs to be detected.

The methodology adopted when dealing with the sweep sampling challenges was to get access to the trigger signals and translate them into a time location in the LF recordings. The triggers are given by square waves in the test files with a one-second period that

asserted the recording of each sweep. The moment when the signal turned to a high state, i. e. the rising edge, was assumed as a fair estimative of the starting time of a sweep recording. Coding such a method creates corresponding time locations between sweeps and LF current, allowing for a fair estimation of which sweeps were recorded during fault current conduction (*post-fault* sweeps).

3.1.2 Data cleansing

After being able to import the data and identify the separate sampled channels, initial investigations started to reveal possible issues in the recorded files. The first identified class of issues was the straightforward problem of missing or corrupted files. A number of 1038 tests was officially stated as staged in the test ignition report, but 44 tests were not present or were corrupted in the released data. The second issue was revealed with initial examinations showing that some tests appeared to have no current conduction at all. This fact had to be addressed since the absence of fault current is equivalent to a non-fault occurrence; mistakenly labelling a non-conducting sweep as a fault observation would worsen and derate the process of learning the classifier. The process of filtering these files comprised of coding a script that would scan the LF current signals for values higher than a small threshold (<0.1 A). It resulted in a number of 19 tests that had no current conduction and were then filtered. Most of these filtered recordings referred to the polarising type of fault named 'grass tests'. Grass vegetation samples were able to fully isolate the conduction at some tests, not allowing for any current conduction, or simply behaved as a short circuit in others. The third issue was only revealed by the end of initial investigations when some tests that did not have a consistent number of samples in the LF and HF channels were noticed. Since the LF channel was recorded at 100 kSa/s, and a HF sweep of 40k samples was recorded at every second, the calculated duration in seconds of LF recordings (floored) had to be equal to the number of sweeps given by the HF channel. A code set to calculate the expected number of sweeps and flag inconsistent recordings resulted in 64 tests being filtered. The fourth issue, conceptualized as a preventive measure, relates to the intermittent nature of fault current conduction during some tests. Vegetation samples sometimes behaved as an intermittent impedance where current cycled between flowing for a few cycles and ceasing for a short period. A decision to filter out tests with high intermittency was made due to the lack of confidence that an extracted sweep would come from moments where current conduction was indeed present. A script coded to flag these high-intermittency tests pointed to 19 recordings that were subsequently filtered. Moreover, the final report also labelled 79 of the tests as invalid. These were also excluded from further analysis, although the final report did not give any particular reason this labelling.

In addition to the filtering of problem tests, continuous analysis on the recordings also revealed tests with odd values with particular small ranges. These were only noticed towards the end of the research and were continuously filtered due to possible inference in the performed experiments. After all filtering and cleaning procedures, 769 tests were selected for analysis when an RMS current threshold of 0.1 A is used, and 566 tests remained when considering a threshold of 0.5 A.

The classes of issues found in the recording files can be summarized:

- Files missing or corrupted.
- Tests that did not show current conduction.
- Tests that presented inconsistent sampling.
- Tests with high intermittency in current conduction.
- Tests marked as invalid by the program.
- Tests with odd ranges that did not fit recording standards.

3.1.3 Fault and Non-fault observations

With the decision of using supervised learning, comes the requirement for labelled data. This decision was made based on the knowledge acquired from the literature and findings resulted from initial investigations. The tools used in initial investigations, described in the next sections, corroborated with the overall conclusions in the literature review that the VHIF problem is too complex to be trivially solved. Therefore, it made sense to leverage the statistical predictive capabilities of supervised learning models to conceptualise a VHIF detection method. Doing so, nevertheless, requires the ensemble of a clear data set composed of observations with respective labels of classes that will be learned by the classifier.

Assembling the labelled data set of observations, however, was a much more significant challenge than thought when conceptualising this methodology. The staged tests unfortunately had the drawback of being done in a substation with a dedicated feeder, generating many issues to be dealt with. Although part of a functioning distribution network, a dedicated feeder meant that there were no consumer loads connected to it. This constraint represented the first big issue when compared to the methodology proposed previous works in the related literature. Most methods analyse the effect of the fault in the feeder load current by studying the difference between its pre- and post-fault states. They analyse how the current signal characteristics evolve from a normal steady-state to a faulted one, further using these two classes to validate their method. However, despite having plenty of data of

fault current, there were no observations of the opposite class (non-fault) to perform any validation. This constraint made initial investigations to turn from the current to the voltage signals but only to result in yet another issue. This second and severe constraint came from the tests execution set-up. The branches were put in place between the conductors prior to their energisation, i.e., conduction began immediately after the voltage source was turned on. Such an experiment design resulted in almost no pre-fault voltage signals recorded, except for a few test scenarios.

Non-fault observations

If the choice of using supervised learning was still to be pursued, a different strategy to gain signals from non-fault observations had to be adopted. Despite not having voltage recordings before fault inception took place (in the majority of the tests), there were background voltage recordings made by the project team throughout different times in the test days. These tests were performed by just recording the main system voltage with no vegetation in between conductors in the test rig. The goal was to gather enough data to help characterize the standard background noise of the studied feeder and aid possible discoveries on fault signatures. Indeed, a useful idea that resulted in a large number of sweeps from both channels which could be used as non-fault observations. These recorded sweeps constitute all the non-fault observations in the following described experiments when learning the classifier.

Recordings of background voltage were made throughout most of the test days, from February 23 to March 27, 2015. They were long enough to generate plenty of sweeps. In fact, they were more numerous than the number of tests that generated sweeps from the 'fault' class. However, during most experiments described in the next sections, a decision to adopt an equal number of sweeps between the fault and non-fault classes was made. This balancing was performed by randomly sampling non-fault sweeps to have the same number of fault sweeps, effectively reducing the number of used sweeps when learning the classifier. Another worth-mentioning characteristic of these recordings was that they included moments when the main voltage supply was turned ON and OFF. Having both types of signals labelled as the same class was being harmful to the classifier due to the different characteristics of the signals. Most experiments were performed considering voltage ON and OFF as the same class; only later they were separated in their respective classes, making the number of classes to be three in total: Fault, Non-fault, and Voltage OFF.

Fault observations

The fault observation sweep extraction comprised a more clear but laborious procedure compared to non-fault observations. The observations represented an arbitrary moment of each test recording referred here as fault inception. This moment is defined by the time when the root mean square value of the LF current signal was greater than a threshold. In most experiments discussed in this thesis, this threshold was set based on a finding from the project final report where it was stated that current values greater than 0.5 A rapidly increase the probability of fire [4]. This threshold was adopted particularly when proposing the conceptualized detection method, but 0.1 A was also explored as a threshold for comparison sake. Having the time inception on the LF current recording, nevertheless, only base the fault observation extraction since mainly all results presented here are based on the HF sweeps, rather than LF signals. At this moment, the access to the trigger signals that asserted HF sweep recordings was crucial. The separation of the analyzed fault sweep was done by comparing sweep trigger times with the calculated fault inception time to extract the immediate subsequent sweep recorded after the threshold was met.

It is important to remember that fault experiments had distinct duration times. Some faults lasted for a few seconds and others more than a minute. Since sweeps were recorded at every second, recordings usually had more than one fault sweep per test. However, a choice was made to perform the classification relying only on one single sweep given by the first recording after the fault inception. By doing so, it is guaranteed that the signal considered when testing the classifier came from the first second of the fault occurrence. Having an observation representative of the moment of fault inception is essential because it is usually where the signatures are more subtle and more difficult to detect. As also stated in the final report, a significant decrease in fire ignition can be achieved if the fault is addressed in 5 s or less. As only one sweep is recorded at every second, the extracted fault observation can come anywhere from approximately 0 to 1 s after fault inception.

3.2 Signal processing

Signal processing is in the core of all the experiments presented here as evidence to potential detection of VHIFs. Due to its importance, a considerable amount of time was expended evaluating and refining the choice for signal processing and representation techniques. Although there was an effort to use contemporary edge techniques, the best results usually came from well established/traditional ones.

This section intends to brief the reader in the most relevant signal processing concepts used to derive the results described in Chapter 4. In regards to signal representation, Fourier-based spectrum estimators and wavelet transform were central. One could argue

that it is not surprising that such techniques would perform well given their popularity in this and other signal processing fields. As described in Chapter 2, Fourier- and wavelet features were thoroughly explored in HIF detection. This use is not seeing as an issue in regards to contribution to knowledge since proposing novel signal processing techniques is beyond the scope of this research. Nevertheless, the extraction of VHIF signatures via the application of Shift-Invariant Sparse Coding — a recent unsupervised coding technique — is indeed seen as one signal processing-related original contribution in this thesis. The same could be said for the post-processing subsection; although comprised of simple techniques and calculations, it is an original approach conceptualized as part of a detection method.

3.2.1 Spectrum analysis and Fourier-based features

Representing signals in *frequency domain* is an effective way of processing and analysing their characteristics. Differently from the time domain, which represents how signals evolve through time, a frequency domain representation describes how much a particular signal correlates to certain frequencies of periodic signals. The transformation, usually reversible, is given by a pair of mathematical operators that translate the signal between domains. The Fourier transform is a classical method of decomposing a time-domain signal into frequency domain components. Its continuous-time operators are given in (3.1) and (3.2) where $\mathscr{F}$ and $\mathscr{F}^{-1}$ are the Fourier operator and its inverse, respectively.

$$\mathscr{F}\{f(t)\} = F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{3.1}$$

$$\mathscr{F}^{-1}\{F(\omega)\} = f(t) = \int_{-\infty}^{\infty} f(\omega)e^{i\omega t}d\omega \tag{3.2}$$

Representing the signal in the frequency domain can have many advantages. The most wide-applicable one is the fact that linear differential equations are converted to algebraic equations in the frequency domain. This advantage facilitates complex mathematical analysis of many problems, especially ones in circuit-analysis theory where signals can be represented as phasors.

In symbolic Machine Learning (ML), one of the main advantages of transforming signals to frequency domain results from the sparsity in their representations. Consider, for example, a continuous-time sinusoid with constant frequency and amplitude. To represent this signal in the time domain, one would need an infinite number of coefficients comprising all the continuous-time points. In the frequency domain, however, only two coefficients may be needed: amplitude and phase information. A representation on the frequency domain, therefore, becomes more *sparse* since all the other coefficients despite

the amplitude and phase-related ones would be zero. Given that ML algorithms are highly sensitive to the number of features or parameters in the model, sparsely representing a signal represents a great advantage. If the exact frequencies of interest of a particular signal are known, their components can be individually selected and used as features in a ML model.

When processing real-life digital sequences that are sampled versions of continuous-time signals, the Discrete Fourier Transform (DFT) needs to be employed. For a real discrete sequence of length N, the DFT operator and its inverse are as given in (3.3) and (3.4), respectively. The resulted DFT is a complex finite sequence with the same length as the signal. The frequencies components in the resulted representation are equally-spaced in frequency the same way time-domain samples are uniformly spaced due to the periodical sampling period.

$$DFT\{f[n]\} = F[k] = \sum_{n=0}^{N-1} f_n e^{-i2\pi kn/N} \tag{3.3}$$

$$DFT^{-1}\{F[k]\} = f[n] = \frac{1}{N}\sum_{k=0}^{N-1} F_k e^{i2\pi kn/N} \tag{3.4}$$

Despite effectively used in many applications, applying the DFT for signal representation results in many constraints and issues. In general signal processing, most of the resulting issues are related to the phenomenon of *aliasing*, spectral leakage, and choosing the appropriate sampling rate. In simpler terms, the sampling rate needs to be appropriate to represent the frequency bandwidth of interest and aliasing needs to be addressed to avoid signals of higher frequencies to interfere with coefficient estimations of the investigated band.

In the context of symbolic ML, the problem of spectral leakage is more complex and challenging to address. As an example, consider a 1-second duration signal sampled with at a high rate so frequencies from a wide band can be properly characterised. A DFT of such signal will result in a long sequence of frequency components since it has the same length of the recorded digital signal. If the exact frequencies of interest are known, there needs to be an adjustment of the length of the signal, so the resulted equally-spaced frequency indexes have the exact desired value. If the frequencies of interest are not known (usually the case), and one wants to use the whole coefficients, the feature space will comprise a long sequence with high dimension. As mentioned before, ML models are very sensitive to the dimension of the feature space, meaning that a long sequence of frequency components are highly undesired. This characteristic has a interesting name, often referred as *the curse of dimensionality* [135].

The fact that real-life signals are often noisy and suffer from the effects of interference of other sources only further aggravates these issues. Intermittent and sampling noises can be harmful for frequency estimation, introducing energy in frequency bands that are not originated from the signal of interest. In an effort to mitigate these problems, techniques are introduced to improve the signal spectrum estimation. The main idea when implementing spectrum analysis methods is to leverage the existent data that is larger than the amount required to attain the desired frequency resolution. The data is usually partitioned so that multiple DFTs can be performed and averaged. This averaging reduces the variances introduced by the noise sources, improving the precision of the power estimation of frequency components. The power spectrum, referred to as Power Spectrum Density (PSD) is basically the squared of the absolute values given by the DFT of the signal, as shown in 3.5.

$$PSD = \mathscr{F}\{f[t] * f[-t]\} = F[\omega] \cdot F^*[\omega] = |F[\omega]|^2 \tag{3.5}$$

The technique of averaging multiple DFT applications to get a power spectrum density estimation is called *periodogram*. The Welch's method, firstly presented in [136], is a popular periodogram technique used in most spectrum estimation calculation in this thesis. It starts by breaking the analysed time series into overlapping segments and *windowing* them with an arbitrary function like the Hamming window. The DFT is then applied to each windowed segment, the results have their magnitude squared for power calculation, and an averaging of all the segments is calculated. The result is a modified nonparametric periodogram with reduced variance with calculated as in (3.6). Where, K is the number of segments in the time series, f_n is the analysed power frequencies of segment n and I_k is the PSD function applied to each segment as in (3.7).

$$\hat{P}(f_n) = \frac{1}{K} \sum_{k=1}^{K} I_k(f_n) \tag{3.6}$$

$$I_k(f_n) = c \left| \frac{1}{N} \sum_{t=0}^{N-1} f_k[t] W[j] e^{-i2\pi tn/N} \right|^2 \tag{3.7}$$

In Eq. (3.7), N is the length of the segment, f_k is the k-segment analysed, t is the sample number, W is the window function (Hamming window in this thesis), and c is a constant dependent on the length of the segment and window used.

Since the segments have to be considerably smaller than the original sequence, the number of frequency components can be significantly reduced. Such outcome avoids the scenario of a feature set with high dimensionality and potential problem in terms of overfitting the learning algorithm. The small number of coefficients from Welch's method

are then better estimators of the signal spectrum (reduced variance by averaging) and represent a smaller feature dimension to the ML model. The user can choose to do feature selection to further reduce the feature space or use all the components as class predictors.

The Welch's method was extensively used in the initial analysis. The goal was first to visually check if there were any changes from non-fault and fault observations. The experiment was inspired in the spectrograms illustrated in the tests final report, which showed increases of many high-frequency components in the voltage signals. The spectrogram is a technique similar to periodogram where instead of averaging the PSD of segments of the signal, they are sequentially concatenated to illustrate how the frequency components evolve through time. Spectrograms are usually referred to as a time-frequency analysis technique; Fig. 4.4 in the Results chapter is an example of one. Exception tests that had pre-fault signals were crucial in these experiments. If applied to a sweep recorded at a time with no current conduction (pre-fault), and later to an in-fault sweep, one can clearly see the changes in the high-frequency components. The results of these experiments are described in Section 4.1.

After having clear indications that there was predicting information on the signals, the Welch method was also used to create features (predictors) in the first conceptualisation of the fault classifier. As all hand-engineered features presented in this chapter, the PSD features were a result of many trial and error attempts to properly represent the signals in the frequency domain. The Welch's parameters resulted from these many trials were given by 50% overlapping segments, 450 frequency components with FFT windows of 10k samples. As 450 features represent a relatively high dimension to the size of the used data set, these resulting components also went through a process of feature selection explained in further sections. Doing so meant selecting the frequency bands with high predicting power, consequently resulting in interesting insights about the fault signatures behaviour in the frequency domain. The results of learning the classifier with PSD features are described in Section 4.2.

The learned classifier gave strong evidence to the possibility of detecting VHIFs, but more evidence on the justification of the adopted approach would be beneficial to increase its relevance. With all the constraints that it brings, the adoption of high-frequency signals can still be a source of potential concerns. Therefore, an experiment was set to compare the predicting information content of the low- and high-frequency signals. It was done by extracting features from these two domains and comparing their discriminative potential. The quantitative comparison was made by the resulting accuracy value from learning a classifier with features extracted from the LF and HF channels.

As PSD is such a cornerstone in signal processing techniques, the experiments set to highlight the detection method relevance also used PSD-related calculations. A noticeable aspect of the features in ML models is that they do not necessarily need to be the frequency

components; they can be higher-level representations based on the PSD results. These can be calculations of distribution moments such as standard deviation or variance; peak analysis; entropy; or energy distribution. At a certain point, they were all tested in the features trial and error phases, but one of the most promising was the signal PSD Spectral Flatness (SF). It is also known by tonality coefficient, Wiener entropy or whiteness of a given signal. As described in (3.8), where, $F[n]$ is the power density spectrum with N number of bins, such measure can characterise the noise-like property of a signal in a zero to one range. It describes how the power spectrum coefficients are distributed in a given bandwidth. For example, perfect sinusoids having no distortion would result in 1, while white noise signals would approach zero. As explicitly depicted, the SF is basically the geometric mean of the PSD values over a giving range (N), divided by its algebraic mean. It is a simple, direct, scale-independent measurement chosen after observing that VHIFs tend to create a wide-band noise over the voltages' signals. An interesting note is that it does not necessarily need to be applied only to the whole calculated spectrum, but it can be used at arbitrarily different sub-bands too. In this experiment, the size of these sub-bands was chosen to be a twentieth of the number of bins of the power spectral density, resulting in 20 features. The results from using these features to attest to the classifier relevance are described in Section 4.3.

$$SF = \frac{\sqrt[N]{\prod_{n=1}^{N} F[n]}}{\frac{1}{N}\sum_{n=1}^{N} F[n]} \tag{3.8}$$

Finally, PSD calculations were also used as a benchmark for the classifier. Since their signal representation and analysis capabilities survived the test of time, every trial made with different features was compared to the performance given by PSD features. When a signal representation tool was taken as effective, it meant that it overperformed the PSD features when used to learn a VHIF classifier.

3.2.2 Wavelet analysis and features

The previously discussed sinusoid example is very convenient to discuss the sparsity of the signals in the frequency domain, but the same advantage will not hold for all signals. The sinusoid representation is sparse due to the basis functions used to decompose the signal in the Fourier transform, which are also sinusoidal. However, sinusoidal basis functions may completely lose this advantage depending on the signal characteristics. For example, an impulse signal, which is highly sparse in the time domain, will need many frequency coefficients to be represented in the frequency domain. Conversely to the sinusoid, transforming the impulse function to the frequency domain would have no advantages in terms of information compression. It may be worth remembering that the

most significant advantage of transforming the signals to the frequency domain, in a ML approach, is that their sparse representation results in a reduced feature dimension.

The signals recorded in the tests comprise many transients and signal discontinuities that are troublesome to represent in the frequency domain with sinusoidal bases. As shown in section 4.1, a fault occurrence will increase the energy of many frequency components distributed in particular frequency bands. These bands cannot be easily discriminated as components have their energy increased in different ways at each fault occurrence.

The wavelet transform was notably proposed to overcome the drawbacks introduced when representing signals with discontinuities in the frequency domain. The wavelet functions $\psi(t)$ have compact support in time with finite energy, different from the stationary sinusoids in the Fourier transform. They are adaptable in the sense that one can use any function that fit the defined criteria: being absolutely and square-integrable as in (3.9) and (3.10), respectively; having zero mean as in (3.11), and square norm equals to one as in (3.12).

$$\int_{-\infty}^{\infty} |\psi(t)|\,dt < \infty \tag{3.9}$$

$$\int_{-\infty}^{\infty} |\psi(t)|^2\,dt < \infty \tag{3.10}$$

$$\int_{-\infty}^{\infty} \psi(t)\,dt = 0 \tag{3.11}$$

$$\int_{-\infty}^{\infty} |\psi(t)|^2\,dt = 1 \tag{3.12}$$

Due to its compact support, a wavelet function is shifted through time, generating time-frequency coefficients. As every function represents a band-pass filter with particular characteristics, different bandwidths are considered by the modification of a scaling factor as in (3.13). Where, a represents the scaling factor, and b represents the shifting factor. In this sense, the wavelet transform operator in the time-frequency domain can be defined as (3.14).

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{3.13}$$

$$WT_{\psi}\{x\}(a,b) = \langle x, \psi_{a,b}\rangle = \int_{\mathbb{R}} x(t)\,\psi_{a,b}(t)dt \tag{3.14}$$

As the analysed signals are real sequences of sampled values, one should adopt a discrete version of this transform. The Discrete Wavelet Transform (DWT) employs discrete values for the scaling factor a and shifts the wavelet function with b at every

point of the original sequence. However, since decomposing the signal at every scaling factor would be infeasible, a strategy to choose effective values for a needs to be adopted. The strategy usually employed differs from the Fourier decomposition by not uniformly dividing the frequency spectrum but rather doing it logarithmically. This approach, called Multi-Resolution Analysis (MRA), assumes different exponential values for the scaling factor at each level of the signal decomposition. The definition of a transformed signal $x[t]$ with length n at the m level, is given by (3.15).

$$DWT[x,m,n] = \frac{1}{\sqrt{a^m}} \sum_l x[k]\psi \left[\frac{n - la^m}{a^m} \right] \tag{3.15}$$

The DWT becomes more numerically efficient when the Mallat algorithm [137] is used. In this process, the decomposition is given by the iterative application of a series of low-pass and high-pass filters. The results are time-scaled versions of the original signal which have most of its energy in a defined bandwidth. Each time a couple of filters is applied, the signal is downsampled in a dyadic manner, resulting in fewer samples and its numeric advantage. In the first iteration, the output is given by the convolution of the original time-domain signal with the impulse response function of the low- and high-pass filters. From that, two signals known as approximation (y_a) and detail (y_d) coefficients will result. The iterative process is performed by using the last calculated approximation coefficient as new inputs of the filtering process. The i_{th} level can be generalized as in (3.16) and (3.17) where, $h[t]$ and $g[t]$ are the low-pass and high-pass impulse response function, respectively, and y_d^0 and y_a^0 are defined as x[k].

$$y_d^i[n] = \sum_{k=-\infty}^{\infty} y_d^{i-1}[k] \times h[2n-k] \tag{3.16}$$

$$y_a^i[n] = \sum_{k=-\infty}^{\infty} y_a^{i-1}[k] \times g[2n-k] \tag{3.17}$$

The detailed coefficients, however, are still time-scaled versions of the original signal that cannot be used directly as features. Approaches in the literature usually calculate the energy [138], entropy [139], or standard deviation [88] of these signals to use the values as correlational features.

In this thesis, with the intention of increasing the methods' reliability, several possibilities of wavelet features details were tested: energy percentage (3.18), sum of the absolute values (L_1 norm) (3.19), mean top peaks (3.20), standard deviation (3.22), Shannon's entropy (3.23), and L_2 norms. Where, Et is the sum of the energy of the approximation and detail coefficients, and $\overline{y^i}$ is the mean of the detail vector. The 'peaks' feature, given by S_{peaks}, defined in (3.21), to the best of the author's knowledge, was never presented before in this regard, characterizing an original feature of this thesis. In simpler words, the

feature represents the mean of the top M (sorted) peaks found in the signal. The integer M can be a fixed value or, as used in this thesis, a proportion of the length of the detail vector.

$$e_i = \frac{\sum_{n=1}^{N} \left|y^i[n]\right|^2}{Et} \tag{3.18}$$

$$sum_i = \sum_{n=1}^{N} \left|y^i[n]\right| \tag{3.19}$$

$$pks_i = \frac{1}{M} \sum_{n=1}^{M} (S_{peaks}(n)) \tag{3.20}$$

$$S_{peaks} = sort(\{\left|y^i[n]\right| > \left|y^i[n-1]\right| \wedge \left|y^i[n]\right| \geq \left|y^i[n+1]\right|\}) \tag{3.21}$$

$$std_i = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} \left|y^i[n] - \overline{y^i}\right|} \tag{3.22}$$

$$ent_i = -\sum_{n=1}^{N} y^i[n]^2 log(y^i[n]^2) \tag{3.23}$$

$$L_2 = \sqrt{\sum_{n=1}^{N} \left|y^i[n]\right|^2} \tag{3.24}$$

A note regarding the choice of the mother wavelet should also be made since the low-pass and high-pass impulse response functions are dependent on it. Inspired by claims in previous works that it would heavily influence the performance of a classifier, a prior comparison was executed concerning possible choices of many mother wavelets. The evaluation compared the performance of different wavelet families such as the Haar, Daubechies, Symlets, Coiflets, BiorSplines, ReverseBior and DMeyer, in their different scales.

The decision regarding the number of levels used in the decomposition was made by analyzing the lowest (last) frequency band given by the DWT. By applying the MRA algorithm, the upper bound of the bandwidth of a certain level approaches $\frac{F_s}{2^n}$, and the lowest $\frac{F_s}{2^{n+1}}$, where, F_s is the sampling frequency, and n is the decomposition level. Therefore, for a 7-level decomposition considered in the HF signals, the last detail band results in frequencies ranging from 7.8 to 15.6 kHz. This band was given as sufficient because this channel had a high-pass filter with 10 kHz corner frequency, meaning that investigating lower frequency would not be a useful pursuit.

The wavelet transform and its extracted features were extensively used in most experiments in this thesis. The only exception was the initial investigations where most insights came from analysing the spectrum of the fault signals. Wavelets were used in the first classifier design presented in section 4.2, as a powerful extractor, and when comparing the predicting information of low- and high-frequency signals in section 4.3. When unsupervised learning was applied to capture fault signatures, the wavelet was used as a benchmark, resulting in insights deeper than initially expected. Due to its low time complexity, the wavelet was also the only signal processing tool used in the prototype conceptualized as a fault detection module presented in section 3.4.

Although they were used in the first classifier as an added feature extractor, the last working version heavily relies on the wavelets and does not use any Fourier-based features. The initial intention was to add the wavelet transform to complement the PSD features as a tool to represent the transients in the fault signals. However, it was later realized that the wavelet-based feature explained most of the predicting information in the signals while being more computationally efficient. These features were at last joined by novel measurements, presented in the next subsection, making the proposed fault detection method more effective and fast. The last working version details, and the examples of its performance, can be found in section 4.6.

3.2.3 Shift-Invarint Sparse Coding (SISC)

Regarding HIF detection, the characterization of fault signatures by *high-level* (hand-engineered or not) features can be useful but also ambiguous fault descriptors. They lead to classifiers resulting in high detection accuracy but can fall short at identifying the nature of the causal behaviours responsible for the result. Methods relying on high-level features in machine learning approaches are more prone to suffer from this constraint. All the results presented here using signal processing techniques together with machine learning suffer from this limitation.

Despite being able to classify specific HIFs accurately, such classification approaches are unable (by the nature of the model) to point out the exact time-domain causal disturbances in the recorded signal's time domain. In an effort to address such limitations, this thesis presents the application of the Shift-Invariant Sparse Coding [140] technique as the result of a quest to find an effective technique to describe the fault signatures. This technique can capture patterns in the recorded signals, regardless of their position and convolution with other signals. The resulting patterns are uncoupled and deconvoluted from each other, having their time and frequency domain characteristics individually analyzed. These capabilities are especially useful for wideband signals (hundreds of kHz) such as the ones recorded on the staged experiments.

The VHIF signature extraction methodology is composed of two steps: dictionary learning and basis function validation. The first is responsible for extracting the fault signal patterns with the sparse coding technique, while the second validates some of them as fault signatures. The need for the latter is given by the fact that the network background HF signals are not negligible at the investigated bandwidth. In fact, the recorded signals are the result of the convolution of influences from many sources: electromagnetic interference (EMI) from AM radios, non-linear loads, network transients, and more. The signals from these sources can have energies less or more intense than the fault signal transients. They are represented in the resulting outcomes from sparse coding indiscriminately, despite being fault-resulting behaviours or not. Hence, the task of correlating the found patterns with fault occurrences needed to be addressed by a separate procedure. Such a need certainly reflects some of the challenges of dealing with real sampled signals rather than clean synthetic data from simulations.

The sparse coding methodology was developed as an image processing technique to characterize the primary visual cortex in mammalian receptive cells [141]. The inspiration came from the assumption that natural images have 'sparse structure', i.e., they could be efficiently expressed as a small number of *representations* from a *larger set* of functions. The representations, in the signal processing context, are the basis functions used to describe a particular signal in a linear combination (as sinusoids in the Fourier Transform). The larger set, on the other hand, is the whole *dictionary* of the possible functions used to represent the signal. The relevance of this technique to this work comes mainly from its ability to learn the most efficient dictionary of bases functions to represent signals in a dataset, without any assumption about its prior distributions. In sparse coding, these are often the most efficient basis functions resulted from different underlying convoluted sources, which can be analyzed when uncoupled from one another. Hence, it is not surprising that such a technique can be efficiently used to extract features in supervised learning classification tasks [140].

The problem can be described as levering the access to a data set to learn efficient low-entropy representations. Similar intentions can be found in the popular dimensionality reduction method named Principal Component Analysis (PCA) [142]. Both techniques were inspired by the hypothesis that the most efficient representations to describe a given data set of signals would come from the data itself. Such insight was responsible for the evolution of representation methods that relied on a predefined set of bases, such as the Fourier or Wavelet transforms, to bases derived from *realizations of the data* [143]. However, differently from PCA, sparse coding does not assume that signals come from a known probability distribution [141]. Not assuming a prior distribution for the signals result in increasing *adaptability* when learning the basis functions but also comes with the higher cost of being a much more complex problem to solve. As given by (3.25) and

(3.26), sparse coding can be described as an optimization problem over two objectives: the effectiveness of the bases at approximating the signals in a linear combination and the sparsity of the representation.

$$\min_{a,s} \quad \sum_{i=1}^{m} \|x^{(i)} - \sum_{j=1}^{n} a^{(j)} s^{(i,j)}\|_2^2 + \beta \sum_{i=1}^{m} \sum_{j=1}^{n} |s^{(i,j)}| \tag{3.25}$$

$$s.t. \quad \|a^j\|_2^2 \leq c, \quad 1 \leq j \leq n. \tag{3.26}$$

The input signals $x^{(i)} \in \mathbb{R}^p, i = 1,...,m$ are assumed to be a linear combinations of the dictionary of n basis functions $a^j \in \mathbb{R}^p, j = 1,...,n$ with coefficients $s^{(i,j)} \in \mathbb{R}$. β is a positive constant that determines the trade-off between the *fit* of the bases and the sparsity penalty L_1 norm. The normalization constraint in (3.26) prevents irrelevant solutions that have too small coefficients and very large bases. This problem is not a convex in s and a, meaning that it can't be directly solved in a trivial manner.

An efficient solution for the problem in (3.25) and (3.26) was proposed in [144]. The authors made use of the fact that, although not convex in s and a simultaneously, the optimization problem becomes convex if any of these parameters are considered individually. The two-step proposed algorithm first assumes the bases to be constant vectors while it optimizes the coefficients. Then, oppositely, it freezes the coefficients while optimizing the bases. Convergence is reached by the iterative optimization of these two consecutive steps until the objective function value reaches its minimum.

Although efficient, this formulation suffers from a substantial constraint. Its effect starts to become relevant for longer signals (higher values of p), where patterns can appear in different locations (shifts). Examples of this are large images where a certain object (pattern) may appear in different positions or audio recordings where a particular word (pattern) may appear at any time. These unpredictable appearances have the consequence of either limiting the analysis to piecewise parts of the signals or having different basis functions with shifted versions of the same patterns. None of these represents efficient scenarios.

The solution proposed in [140] for capturing the patterns in longer signals was to conceptualize a shift-invariant version of sparse coding. In such a version, the basis vectors can have a much smaller dimension than the input signal, which allows the capture of smaller shifted repeating patterns. Nevertheless, as the first-mentioned formulation, it also increases the complexity of the algorithm. The product of matrices $a^{(j)} s^{(i,j)}$ becomes a convolution, $a^{(j)}$ assumes a dimension smaller than p, let it be q, and the coefficients become vectors $s^{(i,j)} \in \mathbb{R}^{p-q+1}$, as shown in (3.27) and (3.28). The challenging problem

of optimizing the convolved bases was solved by modifying and extending the algorithms proposed by [144]. It included mathematical manipulations, like the translation of the variables to the frequency domain to solve a Lagrange Dual problem (convolution as multiplication), and further efficient ways to solve for a large number of coefficients resulting from the new problem framing.

$$\begin{aligned}\min_{a,s} \quad & \sum_{i=1}^{m} \|x^{(i)} - \sum_{j=1}^{n} a^{(j)} * s^{(i,j)}\|_2^2 \\ & + \beta \sum_{i=1}^{m} \sum_{j=1}^{n} \|s^{(i,j)}\|_1\end{aligned} \tag{3.27}$$

$$s.t. \quad \|a^j\|_2^2 \leq c, \quad 1 \leq j \leq n. \tag{3.28}$$

When applying this technique, the user needs to set some hyperparameters: number of basis functions in the dictionary, their length (number of samples), number of iterations, batch sizes, and regularization parameter.

The number of bases, to be explored in detail in the results chapter, is probably the most relevant to final results. A large number will lead to an extensive over-complete dictionary, while a small one may not be enough to approximate the signals, resulting in large residuals. Depending on the signal complexity, a large dictionary could lose its *sparseness* by having too many redundant basis functions, while a small one could miss important patterns due to its limited number. The length of the basis is also important in the sense that it would be impossible to capture a pattern longer than a basis function. It basically represents a time windowing effect, as the window size in a periodogram. The batch size is given by the number of signals each iteration of the code optimizes on. A batch size smaller than the total number of signal means that iterations will run on partitions of data. The number of iterations should then be high enough to iterate over the same partition multiple times. For example, a run comprising of a data set with 100 signals, a batch size of 20, and 15 iterations would need 5 iterations to span the total number of signals and would do it 3 times over each batch. Each iteration uses initial bases and coefficients resulted from the last. Therefore, when a partition is re-learned, initials values for $a^{(j)}$ and $s^{(i,j)}$ are incrementally evolved versions from the last time the same partition was learned. The regularization parameter sets the threshold for error tolerance in the approximation of the signals at each iteration. A small number is probably not beneficial since iterations would take an unmanageable amount of time. The main intention is guaranteeing incremental improvements over iterations. The initial tested values were inspired by the original work [140, 144], then changed in many trial and error attempts. The hyperparameter search

space was bounded by values which took anywhere from 12 to 48 hours to run. The resulted learned basis functions are presented in section 4.4 from the results chapter.

The application of sparse coding captures all patterns (fault transients, EMI, and noises) indiscriminately, creating the need for a procedure to measure the correlation between the resulted bases or dictionary with fault occurrences. The proposed method here leverages the existence of labelled data from fault and non-fault signals to calculate the correlation between basis functions and fault signatures.

When testing dictionaries, the correlation measurement is given by the resulted accuracy from a ML classification algorithm application. The features used are extracted using the patterns described in the basis functions of the resulted dictionary. Inspired by the theory of Convolution Neural Networks (CNNs), the extracted features are the result of using the patterns as filters in a cross-correlation operation, measuring its similarity to the input signal at all possible shifts. The convolved signal is then non-linearly summed, as passed through a rectified linear unit (ReLU) activation layer and fed to the classifier as a similarity feature. Each basis function is responsible for creating one feature in every signal. Therefore, the number of features extracted from each signal is equal to the number of basis functions in the testing dictionary.

To label an identified pattern as a fault signature, a correlation score for each basis on a dictionary regarding fault occurrences must be calculated. The proposed method here, in a similar approach to the dictionary evaluation, use the discriminative power from the individual features as a measure of the variance between classes. At this step, each basis is scored by the resulted maximum separability from a single linear split on its related feature.

3.2.4 Post-processing

One of the original contributions presented in this thesis relates to a procedure adopted after the sampled signal is processed before it is sent to the classification agent. The intention of implementing such a procedure is to mitigate the effect of different conditions in the network in the classifier performance, making it more adaptable and resilient.

The post-processing comprises comparing the calculated features from a newly sampled sweep to a value calculated from a buffer of recent previously processed sweeps. The implementation starts by filling a first-in-first-out buffer holding all the features calculated in the last n seconds. A ninety-percentile equivalent is then calculated for each feature of the buffer, resulting in a *feedback* vector with dimension size equal to the number of features. When the newly sampled sweep is processed thereafter, its features are element-wise divided by the feedback vector and fed to the classifier agent. What the classifier

receives, therefore, is a ratio between the newly calculated features to the past state of the network given by the feedback vector.

As getting the previous sweeps to a fault is infeasible in the analysed data set, an equivalent approach had to be adopted. In the methodology used when sampling the staged tests, recordings were only performed in the temporary period of the fault. However, there was metadata informing the day that every test and background recording was produced. This information was used to produce a feedback vector of features for each of the test days, using the sweeps from background recordings. Hence, before feeding the features of fault observations to the classifier, every test had its date information determined so its features could be compared to the correspondent feedback vector.

3.3 Machine learning

As eluded before, the choice of employing machine learning to perform the present classification task was inspired by many reasons. The most decisive is simply given by the task intrinsic complexity. The numerous proposed approaches and solutions discussed in chapter 2 are clear evidence of how difficult detecting HIFs can be. Corroborating to complexity described in the literature, no trivial solutions to discriminate between the fault and non-fault observations in the analysed recordings were found. The data set advantage, however, was that it represents a reliable and sizeable set of labelled data. The most promising tools for levering this labelled data are, undoubtedly, supervised machine learning algorithms. The idea to use ML to solve this task is definitely not original, as proved by its ubiquitous use even in the narrow field of HIF detection. This thesis methodology is certainly inspired by the works in the literature proposing, with different approaches, the application of some supervised learning technique. Due to their effectiveness in solving general problems in many fields of science and engineering, machine learning techniques are also accessible. It is not hard to find application programming interfaces (APIs) or toolboxes for most popular programming languages in science such as Python, MATLAB, R, or even C. MATLAB, for example, has the Statistics and Machine Learning Toolbox™ [131], which comprises most of the presently popular machine learning algorithms. While being the chosen programming environment for the work in this thesis, MATLAB also has useful tools regarding data manipulation, signal processing, and more.

From the several task categories under the field of ML, most of the work performed in this thesis relates to the supervised learning classification tasks. These techniques are classified as supervised learning because it aims to build a statistical model from a data set with observations labelled as distinct classes. As most of the work relates to the tasks of discriminating between two classes, ‘Fault’ and ‘Non-fault’, the approaches could

also be classified as binary classification techniques. The term classification is used to express that the algorithm output can only assume discrete values, which represent the classes of observations. If the algorithm was to output a probability estimation of an observation belonging to a particular class, it would be classified as a regression algorithm. Nevertheless, one can consider the application of sparse coding, as explained in section 3.2.3, as being an unsupervised ML approach. The application does not require labelled data as basically cluster patterns of data as a way to sparsely represent the data set, which are key aspects of an unsupervised learning algorithm. Other types of ML tasks not explored in this thesis include active learning, reinforcement learning, meta learning, and others.

The algorithms performing the classification task in this thesis are related to classical or symbolic ML algorithms. Their complementary types are from a sub-field called deep learning or connectionist ML algorithms. The choice for the former was not only due to the author's limited knowledge but also due to the concept of *causality* and data size. Since symbolic techniques require symbols (referred to as features in this thesis), one could argue that they are closer at explaining the causal relationship between input and output. The features from the input, which are usually hand-engineered, are latent high-level representations of the source of the discriminative information, i.e., the invariant information between classes. In connectionist algorithms such as artificial neural networks, it is troublesome to infer causation between input and classification output once the data are not fed as high-level representations but mainly in raw form. The advantage of a connectionist approach, such as deep learning, is that they usually perform better at most general tasks like face, digit, and speech recognition. However, increases in performance are usually seen for problems with extensively large data sets, resulting in black-box models with reduced causality interpretability.

It is certainly the case that a bottom-up deterministic approach would be preferred over a ML solution. If effective, such an approach would be able to demonstrate and clarify all the aspects, factors, and characteristics involving VHIFs. Nevertheless, it is somewhat safe to say, after many trials at having the insights necessary to employ a bottom-up approach, that it is unlikely it would be found. It took a relatively short time, conversely, to get promising results after the observations were properly organized to be used in a ML approach. In fact, reliable results were achieve in a period short enough to allow the investigation for more supportive evidence for the approach and phenomenon understanding such as the comparison between the predicting information in the low- and high-frequency signals, extraction of time-domain fault features, and the conceptualization of a feasibility prototype.

Machine learning algorithms and concepts were applied in all the experiments presented in the results sections, except for the initial investigations, section 4.1. Their most evident

use was in the conceptualization of the detection method, presented in section 4.2 and 4.6, which has a ML model in its core working as the fault classifier. From the first to the last working versions, the features described in the previous sections of this chapter are used in the ML model to propose a classifier capable of discriminating between three classes: Voltage OFF, Non-Fault, and Fault. To acquire further evidence for the approach taken, ML algorithms were also used to compare the predicting information content of the LF and HF recordings in section 4.3. The experiment involved the extraction of the same features from both domains to evaluate their discriminative potential using the ML concept of information gain from decision trees. Further, in the same experiment, the features from both domains were used to learn two classifiers based on the same algorithm, which are then compared to infer insights about their predicting information content. After being able to effectively classify the signals and generate evidence for this adopted approach, experiments set to further explore the causality were performed in section 4.4. That entailed using sparse coding as an unsupervised learning technique to extract fault signatures in time-domain studied to infer causation. This original methodology, set to associate some of the coded basis functions as fault signatures, also used machine learning concepts of information gain and invariance to attest correlation. After the signals were correlated to fault signatures, they were also used as novel features in the last working version of the classifier.

If statistical learning techniques are going to be unapologetic used, however, one needs to find a method to perform model selection. The sub-field of supervised ML classification methods is still large enough to encompass numerous learning algorithms. Choosing between them is not a trivial task because their methodologies are not necessarily linked to the characteristics of the task to be solved. Therefore, a technique named *cross-validation* was used to evaluate and compare all the tested ML models for a particular set of features. In particular, most of the classification tests were performed using 10-fold cross-validation. Doing so means randomly partitioning the dataset into ten equal-sized samples groups and validating it ten times in an iterative process. At each iteration, nine of the ten parts were used for training the classifier, and one was used for testing. The process ended in the tenth iteration when all the partitions were used for testing. The main advantage of such approach is that all the observations are used for both training and testing. Additionally, as every observation is used for testing at least once, the overall accuracy translates to an evaluation for the whole dataset. The use of every observation for testing does not take place in different validation methods such as holdout.

The models were evaluated by comparing their performance with many classifiers with standard MATLAB default parameters. Machine learning models such as discriminant analysis, support vector machines, k-nearest neighbours, decision trees (and ensembles) were considered. In respect to such comparison, the best results were given by methods

based on ensembling decision trees; these include the methods of *bagging* or *boosting* a set of decision trees. These methods, or even single decision trees, were so powerful at performing this task that they were almost exclusively used in all performed experiments. When they were not directly used as a classifier, variants or related concepts were used to do feature selection/ranking.

3.3.1 Decision trees and ensemble techniques

Decision trees are hierarchical decision support tools where the incoming data go through sequential binary evaluations until the output is provided. As they have a tree-like model, the number of binary evaluations that an observation can go through it is referred to as the depth of the tree. The tree starts from a root node, being the first evaluation point on a single feature. The root points to other subsequent evaluation points, called as nodes of the tree, which will be chosen depending on the value of the evaluated feature. The binary evaluations continue until one of the ends of a tree is reached; they are called leaves. Decision trees were historically done manually and used in operations research and management due to their advantages. With the popularity uptake of machine learning in the last century, they were quickly adapted to be used in data mining. The standard CART (Classification And Regression Trees) algorithm was first presented in [145] and was soon followed by many variations.

The most notable advantage of decision trees is their interpretability. Once a tree is constructed, it is fairly simple to apply it to any process. They are graphically human-friendly, making it easy to understand how the decisions are being made in sequential evaluations, even for non-experts. Decision trees handle big datasets, work with quantitative and qualitative predictors, and easily ignore redundant variables.

Their most notable disadvantage is that they are *greedy*, meaning that they are heuristics quickly converging to a local optimal solution. It might be the case for many problems that a local optimal will approximate the global optimal solution, but it becomes more unlikely for problems with higher complexity. The trees lack robustness in the sense that a relatively small change in the training data can lead to a more significant change in the resulted tree and final predictions. Therefore, decisions trees are usually called *weak learners*, having high variance and may not generalize well.

The problems with decision trees can be exacerbated or mitigated depending on how they are learned. The process of learning or constructing a decision tree includes problems such as defining the depth of the tree, the size of the leaves, how the trees are pruned, what are the evaluation criteria on the nodes, and more. If these factors are not taken into consideration, the resulted tree can suffer from *overfitting* or *underfitting* the data set. Overfitting is a concept that it is extremely relevant for any machine learning model; it

can be thought as a scenario where the model starts to memorize the data set, instead of using it to learn optimal decision boundaries. As it is not actually learning of underlying pattern but only fitting the model to the data set, an overfitted classifier will not generalize well, showing decreasing performance when classifying out-of-sample data. Underfitting also results in decreased performance, but it is more related to a model that is not complex enough to capture potential underlying patterns in the data. There are rules of thumb and general guidelines to help avoid these problems; however, the primary tool to assess the performance of any classifier is cross-validation. As previously stated, cross-validation folds allow all the dataset to be used for training and testing. Therefore, when trying to find the suitable parameters for the proposed model in this thesis — a process referred to as hyperparameter tunning — the performance assessment tool was always the result of a 10-fold cross-validation.

A crucial part of learning decision trees is deciding the binary evaluation test to be performed at each node of the tree. In the classic CART (Classification And Regression Tree) algorithm, a measure called Gini Impurity (GI) is used to decide the binary test performed on a single feature in the data set [145]; it represents a linear decision boundary for one feature. The GI is the result of a calculation set to describe the chance of incorrectly labelling an item in case they were randomly assigned. It can be calculated by following Eq. (3.29), where J is the number of classes, $i \in \{1,2,3,...,J\}$, p_i is the probability of an observation being corrected labelled in i, and p_k is the probability of mistakenly labelling an observation as in i. One can configure the CART algorithm to consider every data point of a particular feature as a potential split. When doing so, the GI is used to evaluate the potential splits to find the one that has the highest information gain. The procedure is simple and can be summarized:

1. Select a data point of a particular feature.
2. Calculate the GI of data pre-split (parent node).
3. Calculate the weighted sum of the GI of both sides of the selected potential split.
4. Calculate the GI difference between the pre-split and post-split scenario.
5. Repeat for all the data points in the data set and select the one with the highest GI difference as the binary test split.

$$GI = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = 1 - \sum_{i=1}^{J} p_i^2 \tag{3.29}$$

The data point that has the highest GI difference is called the split with the highest information gain and thus used as the evaluation criteria. In practice, this will result in a

decision boundary that can best distinguish the classification classes when the data points bound the possibilities of splits. It is an effective procedure but also part of what make the decision trees to be greedy. When applying this algorithm, for example, the first (root) node splits the data set in its highest information-gain point, and all the subsequent binary tests (nodes) will only consider the data points associated with that decision. The decision at the root node may just split a large part of the data set, biasing all the subsequent tests.

To overcome some of the decision trees shortcomings, ML researches proposed ensemble methodologies where the predicting capabilities of many trees are combined in one classifier. The main goal of this approach is to increase the accuracy of individual trees by usually randomizing (random forests) or/and averaging (bootstrap aggregation) the results of many learned trees. Commonly known strategies are random forests [146], tree bagging [147], and tree boosting [148]. These methodologies performed well for all the tasks that they were used, usually over-performing any other technique tested in cross-validation results.

Bagging is based on the statistical concept of bootstrap sampling, where many replicas of the data set are generated by random sampling. The replicas have a smaller number of observations than the original data set and are generated by randomly sampling it with replacement. Each one of the replicas is used to learn a decision tree, which are further averaged in a single classifier. The averaging in classification tasks can be easily performed by following a vote majority scheme. What separates the random forest methodology from simple bagging is that not only the observations are randomly sampled when replicating data sets, but their features are also bootstrapped. If all the features are included in the replicated data sets, the trees are still going to be biased towards the feature that presents the first highest information gain. The effect of bagging, therefore, is basically reducing the variance of the model without increasing its bias. Having a lower variance gives more robustness to the model, making it less vulnerable to noise due to the averaging of many biased decisions.

In addition to result in a classifier with increased performance, the methodology for bagging can also be used to generate many interesting insights. When the random sampling is made, there is a sizeable part of the data set that is omitted in each of the replicas. These observations, referred to as ‘out-of-bag’, can be used to investigate important concepts in ML such as feature importance/selection and validation. The out-of-bag observations can be used in a similar way to a holdout validation method where part of the data is used for training and another for testing. It is more insightful in a sense that, due to the bootstrapping performed on the features, one can have an estimate of the predictive power of particular features. The estimate can be performed by averaging predictions of trees in the ensemble for which the observations are out of the bag. If a random permutation is applied to the features while testing on the out-of-bag data, one can then get a feature

importance estimator. Most of the qualitative statements regarding the effectiveness of a particular group of features made in chapter 4 are based on the insights given by these evaluation capabilities of bagging.

Boosting was another ensemble technique used, especially in the first published version of the classifier [55], which produced some of the highest results in cross valuation. It consists of an averaging and weighting technique in which many learned trees are used to reduced bias and variance. The main difference between boosting and bagging is that the former weights the votes of the many weak learners instead of only averaging them. As the weights are iteratively updated, observations that were misclassified from trees of past iterations receive a higher weight than rightly classified ones. Such a technique has the effect of making the trees to focus more on observations that are, in a sense, more difficult to classify. Between the boosting strategies tested in the present methodology, the one associated with best results is known as AdaBoost [148]. It trains learners sequentially, and for every learner with index t, it computes a weighted classification error as in (3.30). Where, x_n is the feature vector from the observations, y_n is the classification label response vector, h_t is the prediction of the learner with index t, I is the indicator function, and $d_n^{(t)}$ is the weight of observation n at step t. Training such classifier can be thought as the stagewise minimization of the exponential loss E given by (3.31), where w_n are the observation weights normalized to add up to 1, and $f(x_n)$ is the predicted classification score.

$$\varepsilon_t = \sum_{n=-1}^{N} d_n^{(t)} I(y_n \neq h_t(x_n)) \tag{3.30}$$

$$E = \sum_{n=-1}^{N} w_n e^{-y_n f(x_n)} \tag{3.31}$$

A last note on the use of a concept related to decision trees in feature raking should be noted. In the experiments performed to compare the predicting information content of low- and high-frequency signals, features from the two domains were not only used to learn classifiers but were only individually ranked. The concept of decision trees used to rank the features individually was the previously discussed Gini Impurity. As they are used in CART to define the binary evaluations of the tree nodes, they can be used to find the feature with the higher discrimination power between classes in the measurement set. The resulted values for each feature can then be ranked and compared between the two types of signals from the LF and HF channels. The ranking of the best splits was performed by using their post-split GI (weighted sum), referred to as Impurity Index (I.I.). The smaller this index is, the more *pure* is the classification zones given by the decision boundary, representing better discrimination of the data points. Further, in order to add

to this comparison, the three best splits in the whole feature set are then used to train a simple decision tree validated by cross-validation. It is noteworthy that the procedure of learning the classifier only has a validation purpose. In this case, it is a way to demonstrate the potential of these features at classifying the signals since the dataset is going to be split into test and training sets, representing the generalization on out-of-sample data.

3.4 Prototype and hardware experiments

Part of the experiments and methodology conceptualized to gather evidence of the potential of the adopted detection approach was producing a proof-of-concept prototype the results from such undertaken serve as the basis for further claims made regarding the feasibility of a prototype. The following sections describe the methods adopted for the hardware and software pieces, as well as the experiments proposed to evaluate the prototype performance.

3.4.1 Hardware set-up

The experiment set-up was composed of three distinct parts: an ordinary desktop computer, an external USB sound card, and a single-board computer (Beaglebone). The desktop computer's main role was to load the original signals, generate, and stream audio signals which represented the network's HF voltage sweeps from both classes of observations. The USB sound card is an ADC that sampled the audio signals sent by the desktop via a standard TRS cable, representing a data acquisition hardware. With signals digitized and recorded, the board embodied the calculation and decision making module, running the proposed signal processing and machine learning algorithms.

When the set-up is running, the desktop computer basically plays the audio signal through the onboard audio codec, *Realtek's ALC221*, which constitute a 24-bit DAC (Digital to Analog Converter). The analog signal is transmitted to the sound card, *Sound Blaster Play! 3*, which is a simple commercial 24-bit ADC connected to the board's USB port. Regarding their frequency rates, streaming and sampling are set at 48 kHz. At this rate, one second of sampling relates to the 40k values of a voltage sweep, plus a small space of zeros (remaining 8k), in between sweeps. This approach is convenient since the sweeps were also sampled at every second; the small space helps to differentiate each recording and to compensate for any small sample delays that the set-up might introduce. The board is just a small computer that has an ARM 720 MHz processor, 256 MB of RAM, and runs a Linux distribution (Debian) for embedded/IoT devices. The set-up is illustrated in Fig. 3.1, and its respective diagram in Fig. 3.2, with the three distinct hardware parts outlined by different colours. All hardware used in this scheme are commercial, low-cost

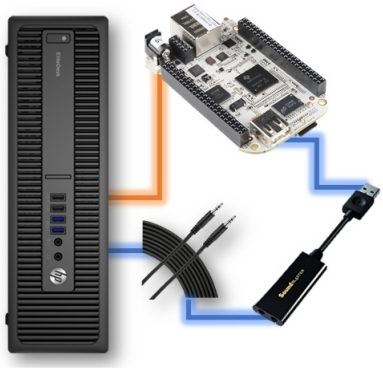


Figure 3.1. Experimental set-up. Blue lines indicate analog and digital signal paths, while the orange line represent a power/communication path.

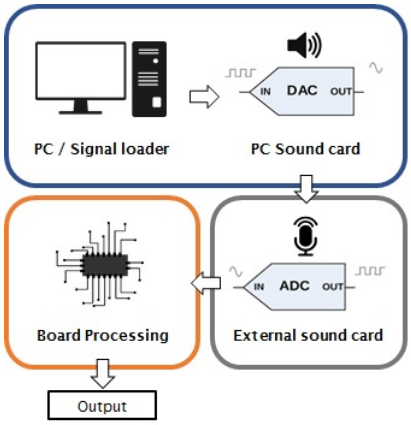


Figure 3.2. Experimental set-up diagram. Desktop computer in blue, external USB sound card in gray, and board in orange.

products that represent a crude and restricted implementation scenario to attest to the method's feasibility.

It is fair to expect, given such a simple hardware arrangement, that sampled signals would result in low Signal to Noise Ratio (SNR) values. This assumption is further confirmed and often not desirable, but the constructed low SNR environment was critical in demonstrating the resilience and feasibility of the developed prototype against noise. In this manner, the high noise environment was valuable to the presented results herein but definitely something to be mitigated when constructing a full prototype subsequent to this proof of concept.

3.4.2 Software set-up

Building software interfaces for different codes and hardware manipulations was certainly the most challenging part. Sampling meant accessing the external Pulse-Code Modulation (PCM) to record streamed values. Processing and classifying implied exporting the codes,

conceptualized in MATLAB environment first, to another language. Moreover, deploying the codes having all the functions in a common language in a cross-compilation from a Windows machine to a Linux ARM architecture executable.

When writing the ensemble of all functions and interfaces, the C programming language was used. In the main code, a buffer of 48k float values is set to store the sampled signals while calculations take place. In this manner, one second of sampling is needed, in the established sampling rate, to fill all the buffer spaces. This buffering results in the first detection delay, also of one second. It is in this buffering delay period when all the calculations of the previous sweep take place. While the next sweeps buffers, the present one is being processed and labelled with the classifier result.

Fortunately, from MATLAB toolboxes to open source APIs, there was a lot of online support for such tasks that avoided the need for coding these pieces from scratch. Accessing PCM samples was possible due to a software framework, *Advanced Linux Sound Architecture (ALSA)*, that provides APIs for sound card devices in Linux. For exporting native codes to C language, a toolbox called the MATLAB Coder [149] was used. It can export functions to C language by generating source files directly from MATLAB codes. The ensemble and compiling of all the source files were made in the Microsoft Visual Studio Community environment, which enabled cross compilation to the ARM architecture. Despite other minor tools, these were essential software that not only facilitated hardware manipulation but also made it possible to deploy full machine learning models to the board embedded system.

3.4.3 Experiments

Four experiments were conceptualized to evaluate the noise introduced by the quantizations, validate the used classifier, and test the board's performance in extracting features, processing, and classifying signals. In the first, noise quantification was performed by streaming, recording and calculating SNR and error measurements of an arbitrary number of signals. The classifier, based on the new proposed feature extraction approach, was validated in the second experiment by a 10-fold cross-validation approach. The third and fourth were set to test the classification performance of the prototype with both *in sample* and *out of sample* data using all the tests from the described dataset.

Noise Quantification

When streaming and sampling signals, two pieces of hardware were major sources of quantification noise: the DAC in the desktop computer and the ADC on the board. Noise evaluation of the whole set-up is, nevertheless, more difficult to estimate, and thus performed by an empirical approach. Rather than relying on estimations by hardware specifications,

the adopted approach consisted of performing four measurements on an arbitrarily high number of samples. The following results were derived from calculations made on 25 random observations of 40k samples each, resulting in one million streamed and sampled values.

Noise quantification was thus performed by comparing four standard signal measurements calculated from original and sampled signals. These were SNR, L_1, L_2, and L_{inf} errors norms. Eqs. (3.32)-(3.35) respectively describe how the measurements were calculated, where, x_i^1 is the original signal sequence, x_i^2 is the discrete signal recorded by the board, and σ^2 is the variance.

$$SNR_{dB} = 10\log_{10}\frac{\sum_{i=1}^{n}|x_i^1|^2}{\sum_{i=1}^{n}|x_i^1 - x_i^2|^2} = 10\log_{10}\frac{\sigma^2_{signal}}{\sigma^2_{noise}} \tag{3.32}$$

$$L_{1err} = \frac{\|x_i^1 - x_i^2\|_1}{\|x_i^1\|_1} \times 100 = \frac{\sum_{i=1}^{n}|x_i^1 - x_i^2|}{\sum_{i=1}^{n}|x_i^1|} \times 100 \tag{3.33}$$

$$L_{2err} = \frac{\|x_i^1 - x_i^2\|_2}{\|x_i^1\|_2} \times 100 = \frac{\sqrt{\sum_{i=1}^{n}|x_i^1 - x_i^2|^2}}{\sqrt{\sum_{i=1}^{n}|x_i^1|^2}} \times 100 \tag{3.34}$$

$$L_{inf} = \left(1 - \frac{\|x_i^2\|_\infty}{\|x_i^1\|_\infty}\right) \times 100 = \left(1 - \frac{\sup_i|x_i^2|}{\sup_i|x_i^1|}\right) \times 100 \tag{3.35}$$

Classifier validation

As feature extraction modifications were made in the data process step previous to the supervised learning part, validation of the classifier had to be again performed. The validation was needed to test the classifier's ability to generalize out of sample data and the existence of overfitting in the learning process. Standard practice was adopted, and the 10-fold cross-validation method was performed. Doing so meant partitioning the data set in ten equal parts. All parts except one were used for training and the remaining one for testing. This is repeated iteratively with all parts until all samples are used for testing exactly once.

Classifying sampled data

Testing the classifier on the same data used in training has no value regarding generalization or classification performance assessment. Nonetheless, it is a meaningful experiment to investigate the noise effects introduced by the experimental set-up. With this goal in mind, the third experiment consisted of training the classification algorithm with the whole data set and deploying it to the board. The desktop then streamed all the original signals,

sampled, and classified by the board. It was expected that the classification result given by the desktop would differ from the board, giving a quantitative measure of the noise effects in the decision boundaries created by the classifier.

Split set

The fourth and last experiment had the same goal as the third, but doing so in the classical dataset split approach. Here, the data were partitioned into two equal parts. One was used to train the algorithm, and the other to test it. The trained algorithm was deployed to the board and tested with the same out of sample data as the desktop. The comparison of classification accuracies was then presented as a quantitative measurement of the feasibility prototype's performance.

3.5 Methods summary

The previous sections discussed the fundamental parts of the methodology adopted in this thesis. It could also be organized and classified in sequential segments of work: initial analysis, pre-processing, processing, post-processing, classification, and validation experiments. Apart from the first and last segments, the other parts reflect the work related to the main goal of the thesis — the production of a VHIF classifier and detection method.

The initial analysis was composed of brief investigations on individual signals, in a search for discriminative information of fault occurrences. Such investigations were performed mainly visually, with the help of spectral analysis tools such as periodograms and spectrograms. Although simple, valuable insights related to harmonic behaviour and high-frequency information content were captured.

The favourable initial results gave confidence to the investment of time and effort to conceptualize the other main segments. The pre-processing comprised the cleansing of problematic tests, the creation of a database of labelled sweeps ('Fault' and 'Non-fault'), and the filtering of signals as a pre-processing measure. The processing part was an ever-changing segment due to the continuous learning and testing of effective signal representation tools. The main goal of it was to extract the features, or discriminative information, in a consistent and representative way before classification. The first feature extractors were periodograms and resulting frequency components, which later was merged with the wavelet transform and features calculated its coefficients. Fourier-based features were further dropped out, with the last version of the classifier using a combination of wavelet- and fault signature-based features. The latter is extracted from filter-like basis functions learned from the application of the unsupervised learning technique called shift-invariant sparse coding. Before sending to classification, however, a method developed

as a post-processing tool is applied. Its main goal is to help the classifier to cope with different environmental conditions and regular changes that the system might experience throughout the day. In a real application, it compares the just-calculated features from a newly acquired signal to a trend of average recent, past calculated features. These new values are then sent to the classifier, which is a machine learning algorithm trained from the database of sweeps. The algorithm is an ensemble of decision trees method known as bagging or random forest, which over-performed all the classical machine learning algorithms previously tested.

Validation of the classification algorithm was always performed by 10-fold cross-validation, but more experiments to support the adopted approach were also produced. The first experiment was conceptualized to support the decision of using high-frequency signals, comparing the discriminative information content between low- and high-frequency signals. Doing so meant extracting Fourier and wavelet-based features from both domains and using them to learn individual classifiers that had their performance compared. The comparison was also performed on a feature-by-feature basis by ranking the best linear discriminator of each respective feature. To support the claims of robustness to noise and real-time capabilities, a low-cost prototype composed of a card-size microcontroller and simple ADCs was also produced. Finally, to prove and illustrate the fault signatures in time-domain, the sparse coding technique was used to learn efficient signal representation basis in the data set. By using cross-correlation and linear discriminators, the correlation of the basis functions to fault signals was calculated, with the highest ones labelled as faults signatures. Discrete, small experiments were also produced throughout the research period, leading to modest insights that will be commented in the next Chapter — Results.

Chapter 4

Results

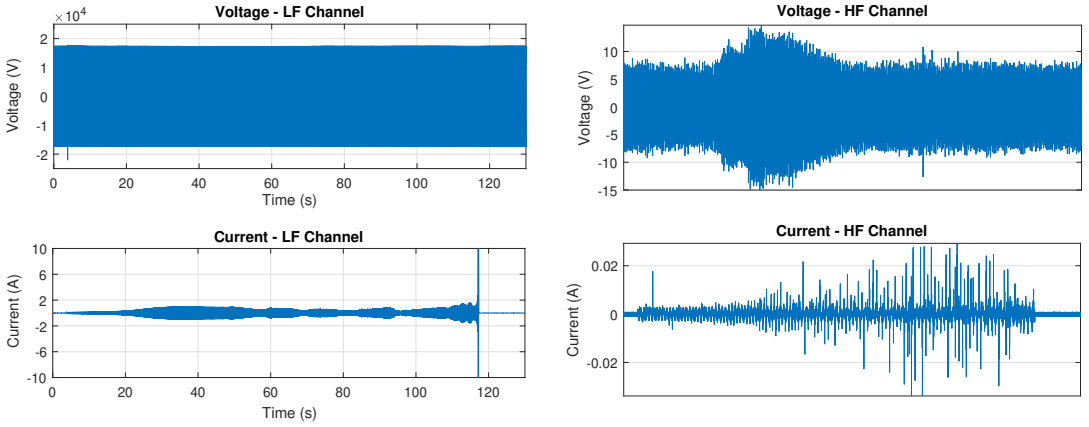
Chapter 4 presents the outcome of the experiments described in Chapter 3 with minimal commentary regards their implications. For a detailed description of their relevance, practical implications, and constraints, please refer to Chapter 5: Discussions.

Brief noteworthy points need to be mentioned to make this presentation clearer. It mainly refers to the required substantial adjustments made to the methodology during the research period. Although the aims and general goals of this thesis suffered no significant changes, the methods applied, signal representation techniques, and data considered did go through many modifications. The results presented in the following sections were obtained between time intervals where iterations of adjustments were performed and validated. The results were also published in between these periods. The following presentation describes the outcome of the experiments closely resembling the form and order they were obtained. Adopting this presentation approach does not only inform the reader about the experiments results but also to reveal insights had throughout the research period. It is hoped that such findings will create supporting evidence for particular methods, possibly saving time of individuals undertaking similar tasks. To bring to the reader up to date, nevertheless, the final section of this chapter is dedicated to describing the final version of the results, working method, and examples of performance.

4.1 Data investigation and initial findings

Comprehensive data from the *Vegetation Conduction Ignition Test* project was made public by the Victorian Government [49]. Accessibility to the fault recordings, however, was particularly demanding due to the released data file format. Staged tests were recorded as individual files with a format defined by the data acquisition hardware company (HBM). Fortunately, the company supplies a software package that facilitates importing the data to the MATLAB environment. After familiarisation with fault recording files and coding

a file-reading interface, efficient data manipulation and visualisation was achieved. As an example, Fig. 4.1 illustrates the LF and HF recordings of the test #36. The current conduction started around 3.82 s into the test with an initial fault current of 0.05 A, reaching its final value at 0.9 A, before ending in flashover. Such a final state can be attested by the quick rise in the fault current at the end when the amplitude gets close to 10 A.



(a) Low-frequency channel recordings.

(b) High-frequency channel recordings.

Figure 4.1. LF and HF recordings from test #36.

Initial investigations were enough to produce relevant insights about the background noise, network, and vegetation fault behaviour. The background noise was mainly composed by narrowband signals around 10 kHz and radio signal carriers at frequencies higher than 300 kHz. The PBSP team pointed out the sources of the multiple narrowband signals around 10 kHz sources to be large industrial loads operating in the system. The higher frequency carriers are most probably related to military and maritime radio navigation from 300 to 450 kHz [150] and AM radio broadcasting from 623 kHz to 1 MHz. Examples are the ABC National radio (623 kHz), 3AW (693 kHz), ABC Gippsland (721 kHz), 771 ABC Melbourne (774 kHz), Sport 927 (927 kHz). The blue trace of Fig. 4.2 illustrates the voltage power spectrum density of a staged fault HF recording, moments before power was supplied to the test rig (background noise). The orange trace represents power spectrum density moments after the energisation of the test rig. The signal around 10 kHz was immensely amplified, pointing out its source to be inherent from the grid. Intense narrowband signals from 100 to 200 kHz also appeared. In the report, they were associated with interference created by large grid loads.

Significant observations regarding the influence of the staged faults in the voltage HF signals can be made by superimposing power spectrum densities plots. The two curves in Fig. 4.3 represents the HF components of signals extracted from test #36, illustrated in Fig. 4.1, moments before and after fault conduction started, i. e., pre- and post-fault. The test was performed around 3 PM with a branch place in between an energized and an

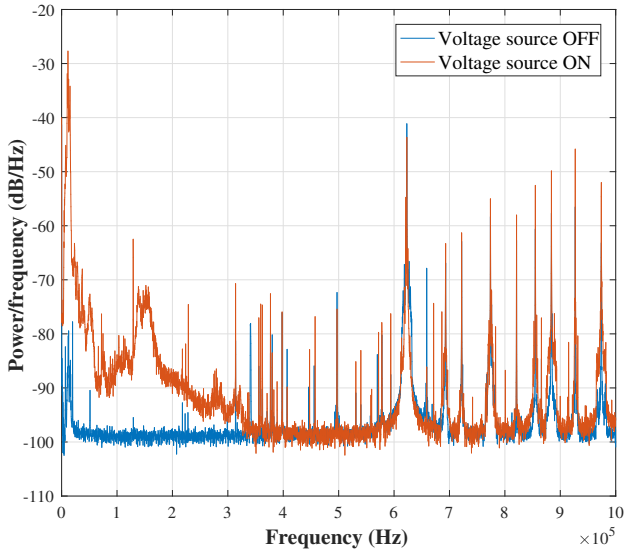


Figure 4.2. Power spectrum density of the background noise and grid voltage.

earthed conductor. Although it did finish in a flashover, the power density representation is from moments after the fault current reached 0.5 A The superimposed plots are effective at illustrating the influence of the fault in the HF signals. The most noticeable difference is the added wideband signals from $\sim$ 50 to 600 kHz. Such a significant comparison, however, could not be performed for every test on the data set. As previously mentioned, only a few tests had the voltage source switched ON before the vegetation contact, test #36 being one of these. In most tests, the current conduction started immediately after energisation since vegetation contact was made before its recording.

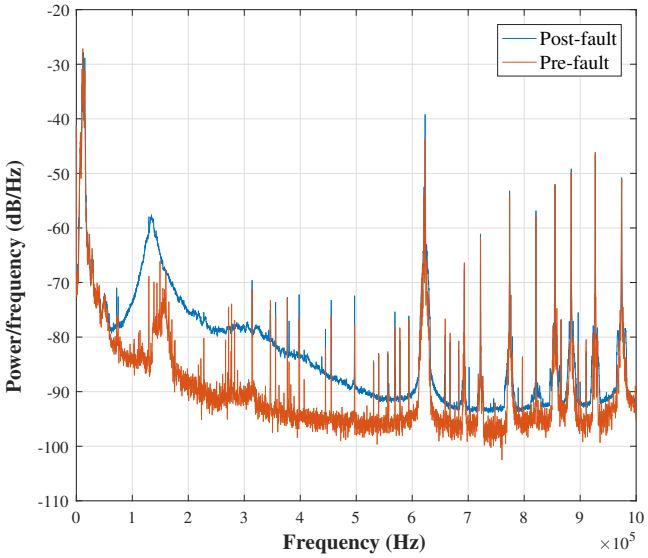


Figure 4.3. Comparison of the pre- and post-fault power spectral density of voltage signals.

As another form of visualisation, Fig. 4.4 shows the spectrogram of the same test. One can observe that the frequency components higher than 500 kHz, and the ones around 10 to 15 kHz, remain approximately constant throughout the test. The same cannot be said for the frequency component in the range of 50 to 400 kHz, especially in the area close to 130 kHz. It is important to note that test #36, as illustrated in Fig. 4.1, start seconds before the recording. All the sudden change happening at the beginning of the timeline in Fig 4.4 is the effect of the minimal fault current of less than 0.1 A. One can also observe the pivot in the frequencies, and the vanishing of the strong 130-kHz centred components at the end of the timeline in the graph. The time ticks in Fig. 4.4 is not the same as Fig. 4.1 because the spectrogram is calculated from HF data, which is a concatenation of many 20-ms sweeps. The beginning of the fault in this image is before the timeline reaches 0.1 s, and the end is around 2.4 s.

One of the most significant initial findings came from observing the harmonic content of some of the tests. The observation was performed by comparing the harmonic content of the fault main current (50 Hz) in two different moments: 5 and 100 seconds after fault inception. Fig. 4.5 shows five cycles of the power frequency from the fault current in time and frequency domain at these two different moments. In the early stage, the current waveform presents close to linear behaviour, with low harmonic content, whereas the latter stage shows a relevant harmonic distortion. Although it might seem trivial, this phenomenon should be widely relevant to the HIF detection field, especially regarding the niche related to vegetation faults. The relevance, to be further discussed in the next chapter, is given by the contextualization of two facts. The first is that many HIF detection methods proposed in the literature rely on the fault contribution to low-order harmonics

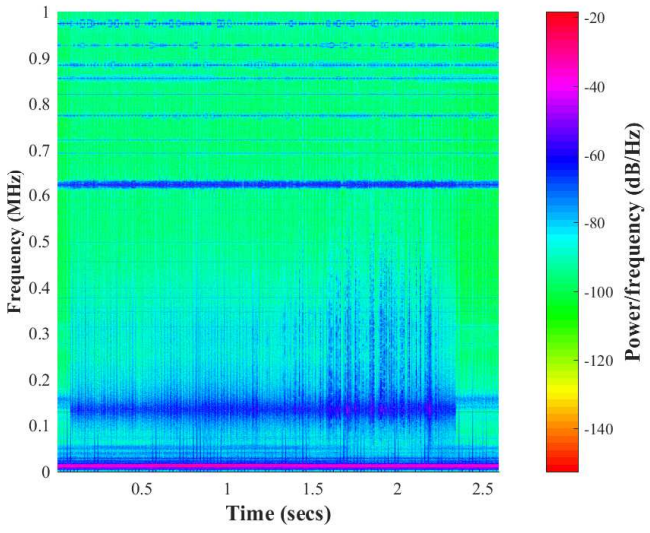


Figure 4.4. Spectrogram of the voltage sampled by the high-frequency channel.

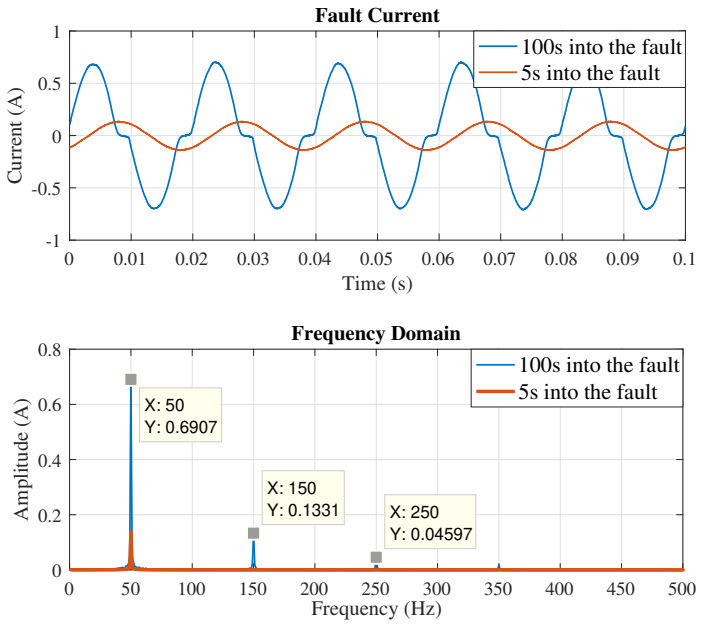


Figure 4.5. Comparison between fault current in time and frequency domain of 5 and 100 seconds into the fault.

[86, 99, 101]. The second is given by one of the PBSP program findings, which states that fire ignition risk will not be severely reduced if these faults are not detected in five seconds or less [4]. If the insight represented in Fig. 4.5 in fact generalizes, it could prevent low harmonic-based methods from reducing fire risk, even if they are effective. Further results comparing the LF and HF information content presents evidence that such an effect is probably generalized in the tested vegetation species.

One of the sources of the fault signals observed in the HF voltage spectrum was partially revealed in the initial investigations. The increases in frequency components shown in Fig. 4.3 was observed to be likely related to fast step discontinuities in the fault current created by the high non-linearity of the vegetation fault impedance. The observation was inspired by insights presented in the project's final report [4] where fault current discontinuities in the HF signals were followed and analysed. As an example, Fig. 4.6 shows a spike in the fault current and the voltage response in time and frequency domain from the HF channel. The frequency domain shows that the responses in the voltage oscillate with a frequency centred at approximately 130 kHz. Such response matches the frequency components increase around 130 kHz, as shown in Fig. 4.3, pointing it to be one of the sources of the added transients.

To illustrate the phenomenon consistency, Fig. 4.7 and 4.8 display the plots of power density spectrum of two more tests that had pre- and post-fault recordings. Those are tests #14 and #916, respectively.

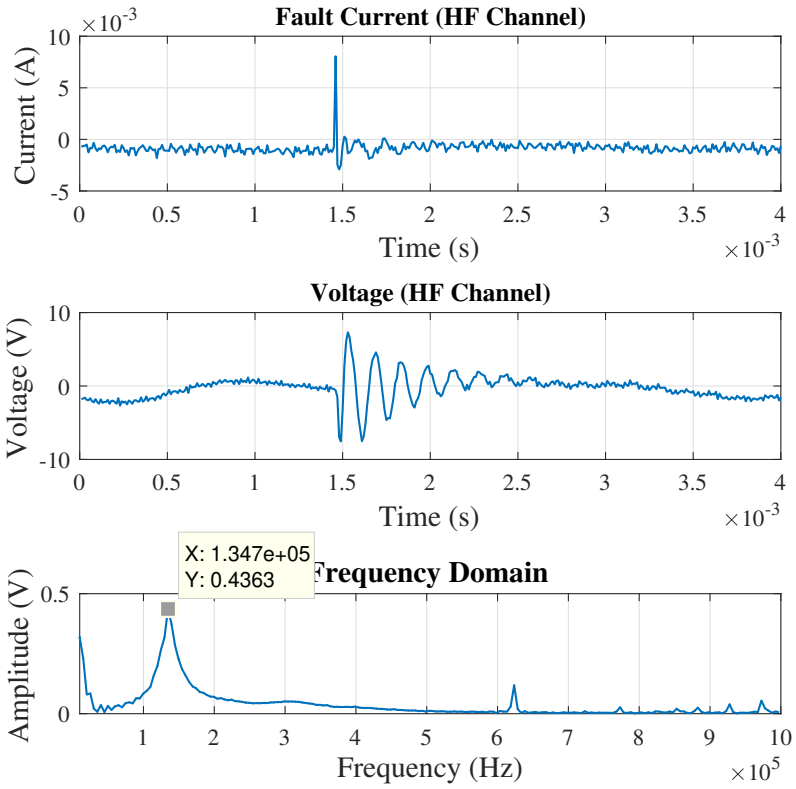


Figure 4.6. Fast discontinuities in the current and voltage waveform and voltage frequency response.

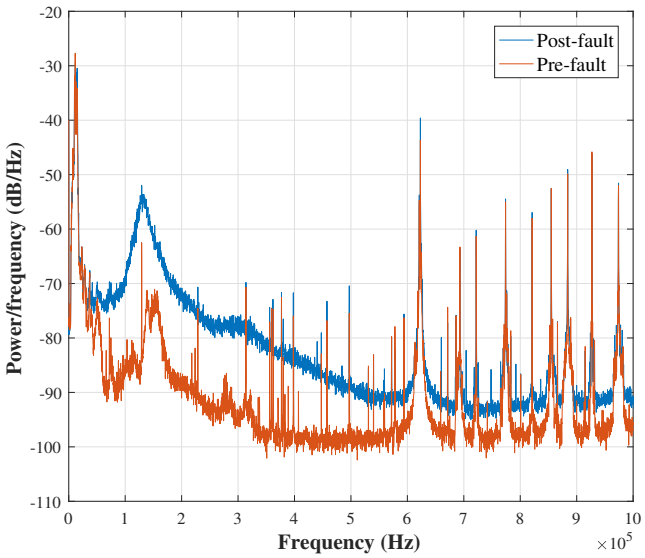


Figure 4.7. Comparison of the pre- and post-fault power spectral density of voltage signals of test #14.

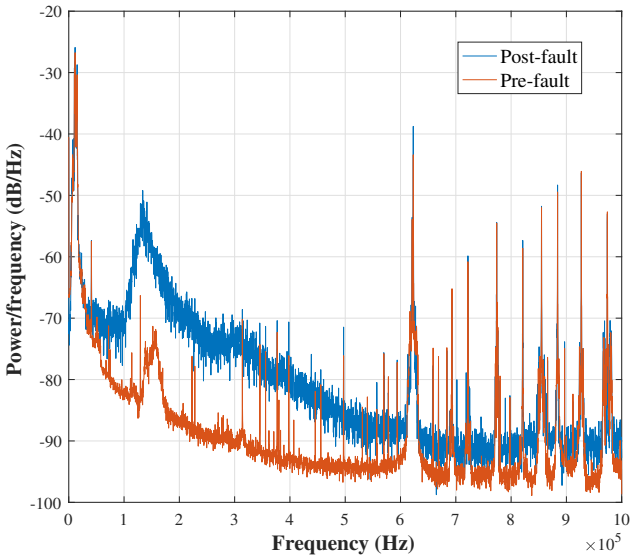


Figure 4.8. Comparison of the pre- and post-fault power spectral density of voltage signals of test #916.

4.2 Classifier design

Initial investigations results were significant enough to endorse the hypothesis that the HF signals might carry sufficient discriminatory information to support a VHIF detection method. Nevertheless, many steps related to data pre-processing were still needed to be taken before conceptualising a classifier. Most were related to generalising data processing, cleansing, and labelling.

4.2.1 Data preprocessing

After being able to access all tests recording data, anomalies that could negatively influence results were identified and filtered. Some tests recordings were missing or had corrupted data, making their use impractical. Current conduction was not present in every test, especially ones using grass as the conducting surface. Grass tests usually have polarising results by either having very low impedance and intense conduction or not conducting at all. In some tests, current conduction was highly intermittent with short conduction times.

Inspired by the staged faults report, a portion of the tests was chosen to be arbitrarily filtered. The report labelled some recordings as invalid, although the reason was not fully disclosed. It also stated in one of its findings that fire ignition risk significantly increased for currents higher than 0.5 amperes. This finding inspired the decision of excluding faults that did not meet such a threshold and to adopt it as the initial time in further data analysis.

From the 1038 tests allegedly performed by the program, 994 had accessible recordings, and 568 remained after the described labelling and filtering process. 351 were ‘phase-to-

earth tests', 193 were 'phase-to-phase tests', 23 were 'bush tests', and 1 was a 'grass test'. In respect to the effective current threshold value of these tests, 5.11% of the experiments were limited to 0.5 A, 50% to 1 A, 39.96% to 2 A, 4.58% to 4 A, and 0.35% were not classified.

An equal number of observations (HIF and non-HIF) was considered to avoid bias in the classification. The analyzed non-HIF observations originated from 11 voltage background recordings made between February 24^{th} to 27^{th} of 2015. Each run was performed in three different periods of each day, resulting in 719 non-fault observations. 548 sweeps were randomly picked from these observations for balancing and added to 20 sweeps of white Gaussian noise. The latter was used to help teach the classifier not to mislabel a situation where there is no connection to the voltage source (supply off) as a fault occurrence.

4.2.2 Feature extraction

As stated in the methodology, many trial and error attempts to find suitable signal features with high predictor potential were made. By the time the classifier methodology was published [55], the best performing method comprised a combination of the wavelet detailed coefficients peaks, energy, entropy, and Power Spectral Density (PSD) measurements. Although wavelet features alone resulted in high accuracy (+90%) in the classification, further analysis showed that the feature group could still be enhanced with frequency-domain measurements (PSD).

When utilising the wavelet multi-resolution analysis (MRA), the output decomposition relates each level to a specific frequency range of the sampled signal. The upper limit of the frequency pass-band in Hertz of each detail coefficient is approximately given by (4.1), and the lower limit by (4.2), where Fs is the sampling frequency and n is the detail level. Each detail coefficients of a signal sampled at 2 MSa/s in the MRA decomposition results in frequency ranges given in Table 4.1.

$$Fb_{up} = \frac{Fs}{2^n} \tag{4.1}$$

$$Fb_{down} = \frac{Fs}{2^n + 1} \tag{4.2}$$

Finding the optimal level of decomposition is not a trivial task. One needs to understand how much information the next level of decomposition might give and the implications for the problem at hand. In this case, however, the signal representation problem is bounded by the fact that signals were passed through a high-pass filter with a 10 kHz corner frequency before recording. The implication is that detailed coefficients greater than the 7^{th} level (7.81 to 15.62 kHz) are not going to provide any meaningful information. Therefore, a

Table 4.1. Frequency range of each detail coefficient

Detail number	Frequency range
1	500 kHz∼1 MHz
2	250 kHz∼500 kHz
3	125 kHz∼250 kHz
4	62.5 kHz∼125 kHz
5	31.25 kHz∼62.5 kHz
6	15.62 kHz∼31.25 kHz
7	7.81 kHz∼15.62 kHz

choice to do an exhaustive search from level 1 to 7 was made. By associating the detailed decomposition levels with the accuracy given by the classifier, it was found that no accuracy was gained for levels of decomposition greater than four levels.

There is a repeated argument that the efficiency of the DWT at representing transients may be heavily influenced by the choice of the mother wavelet [11, 118]. Aiming at investigating such a claim, a prior comparison was executed concerning possible choices of many mother wavelets. The evaluation compared the performance of different wavelet families such as the Haar, Daubechies, Symlets, Coiflets, BiorSplines, ReverseBior, and DMeyer, in their different scales. The performance indeed changed regarding different choices, with the sym4 (Symlets) giving the best overall accuracy. A reasonable explanation for that is given by the similarity between the mother wavelet waveform and the transients created in the HF voltages signals, which have origins on the fast step discontinuities in the HF current. However, the maximum difference in overall accuracy was about 1% between different wavelet families. It did not corroborate with the critiques made by researchers in this particular studied case.

Feature selection from wavelets was performed with support from the previously cited class, 'Tree Bagger', from the Statistics and Machine Learning Toolbox. More precisely, its built-in function for measuring features importance. The process is given by the permutation of the feature order across the observations in the dataset and the resulted effect on the classifier accuracy. Based on it, the features were selected and can be listed: sum of absolute coefficients for 1^{st}, 3^{rd} and 4^{th} level; 1% of the top peaks from 3^{rd} level; and energy percentile of 3^{rd} level.

As the whole HF spectrum components showed relative increases in the presence of a fault, an investigation on reliable and simple PSD features was also considered. The feature selection from the many frequency components was performed with the same method mentioned above. The estimation was performed by the Welch's periodogram technique with 450 frequency bins (0 to 1 MHz), a window size of 10k samples, and 50% of window overlap. Peaks in three different ranges showed a strong correlation with fault occurrences:

approximately 350-370 kHz, 770-775 kHz, and 890-901 kHz. These density power values were used together with the wavelet features, adding to the total of eight features. Please refer to the Methods subsection 3.2.1 for a comprehensive explanation on how the wavelet and PSD features were calculated.

4.2.3 Classification algorithm

The choice of the type of classifier was made by comparing the performance of various classifiers with standard MATLAB default parameters. Machine learning techniques such as discriminant analysis, support vector machines, k-nearest neighbours, decision trees (and ensembles) were considered. In respect to such comparison, the best result was given by the boosting the decision trees technique. Decision trees split the data points strategically in binary decision nodes with indicator functions that evaluate each feature to classify an observation. They can handle big datasets, work with quantitative and qualitative predictors, easily ignore redundant variables, and have relatively high levels of interpretability.

The task of choosing the classifier, however, took place before the feature selection procedure where all the calculated features were given as predictors and overall accuracy was adopted as the technique performance evaluation. The tested classifiers, using 491 predictors, can be listed in ascending performance order: Quadratic discriminant (68.2%), Linear discriminant (77.2%), Weighted KNN (K-Nearest Neighbour) (86.7%), Fine Gaussian SVM (87.1%), Fine KNN (87.7%), Linear SVM (88.7%), Quadratic SVM (91%), Complex decision tree (94.9%), and Boosted Trees (98.06%). Given the relatively large dataset, all performance results were given by an average of the 10-fold cross-validation procedure.

It is probably worth remembering that the cross-validation technique is just a way to test a statistical classifier efficiently. In 10-fold cross-validation, for example, the data set is partitioned in ten equal parts. In each fold, one part is separated for testing while the others as used to learn the classifier. By the end of the ten folds, all parts have been used in testing at least once; hence all the data points are used to learn and test the model. When dependability is discussed, it refers only to the percentage of positive (fault) observations that were classified as such. The security only represents the percentage of negative (non-fault) observations that were classified as negative. The overall accuracy includes all the observations, representing the percentage of classified observations that matched their real label.

After the classifier selection, adjustments of parameters, and feature curation, the resulting performance can be expressed by the confusion matrix shown in Fig. 4.9. This matrix is commonly used in the machine learning field to describe important features

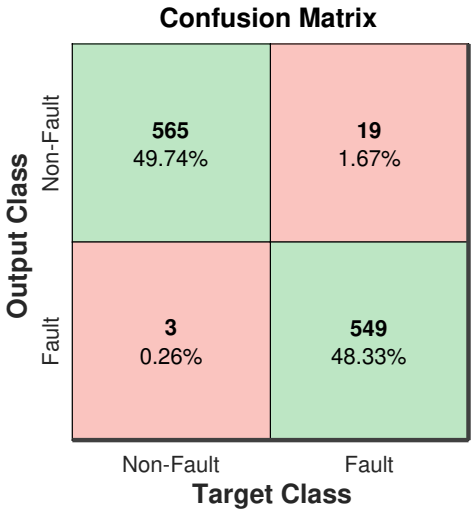


Figure 4.9. Classifier confusion matrix.

from a proposed method. In HIF detection research, these numbers can infer important parameters such as dependability, security, and overall accuracy. The diagonal terms of the matrix, given by the green blocks, represent accurate classification, while the remaining terms show the occurrence of mislabelling. In this case, the first term (1,1) is related to Non-HIF observations being labelled as such, i. e., it translates the security of the classifier (99.47%). The second diagonal term also represents accurate classification but now for HIF observations, i. e., the dependability of the classifier (96.65%). 19 HIF observations were misclassified as Non-HIF and 3 Non-HIF as HIF observations. If aggregated as the total overall accuracy, a result of 98.06% is achieved.

The method's security is undoubtedly the most relevant result from learning this classifier. It is fair to assume that, for a HIF detection method, security should come as a priority over dependability. As previously discussed, and further explored in the Discussion chapter, a false positive may lead to severe undesirable consequences considering load shedding priorities.

4.2.4 Classifier validation

A common practice to validate HIF detection methods is to test the proposed algorithm against data from simulated network switching transients. However, this practice is more relevant for proposed HIF detection algorithms that are current, low-frequency methods (as most in the related literature), which may be greatly affected by such transients. The solution presented here is a voltage and high-frequency based method, slightly diverging from the main goal of such validations. Although less informative, these simulations can still be useful to demonstrate the classifier's security towards simplistic transients

models. To that end, simulations of the given disturbances in Simulink^{TM} environment were performed.

The 4-bus IEEE test node feeder was used to simulate the transients. All its details and characteristics can be found comprehensively described in [151]. The most relevant reasons for this choice were its similarity to a dedicated feeder, like the one where the real tests took place (a short feeder), and the fact that the 4-bus feeder was a system made public to test different transformer connections. The latter is relevant because the original feeder was part of a three-wire distribution system where the transients and faults characteristics can severely change when considering the existence or absence of solid grounding.

The simulation used a step-up transformer (12.47/24.9 kV) set-up, connected in a Delta-Delta configuration. The connected load is linear, with 6 MVA, and 0.8 lagging power factor. The simulated transients were the normal switching events that usually concern HIF algorithm's security: transformer energisation (24.9/415 kV, no load), capacitor energisation (1 and 2/3 of the system's Q), load switching (1.5 MVA, overloading the transformer in 25%), and non-linear load switching (also 1.5 MVA). The time step in the discrete simulation was the same as the sampling frequency in the tests, i.e., 5.10^{-7} s or 2 MHz, with data also fed through a 10 kHz corner frequency filter. Moreover, in regards to noise consideration, white Gaussian noise was added based on the average power of noise in the background noise recordings from the real tests (same noise power). The switching times were simulated considering eight equidistant angles, from 0 to 315°, and the most severe transients are illustrated in Fig. 4.10.

After simulated, filtered, and added noise, the signals were sliced in sweep length sizes and fed into the classifier for testing. None of the 40 different experiments was labelled as faults.

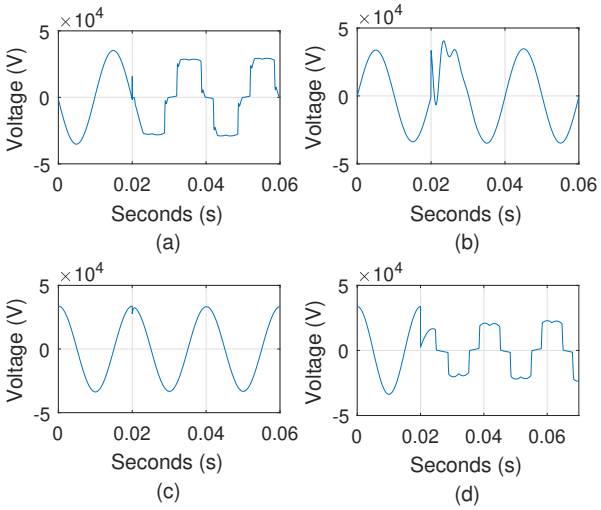


Figure 4.10. Illustration of the simulated transients. (a) Transformer energization (b) Capacitor energization. (c) Load switching. (d) Non-linear load switching.

4.3 High-frequency signals importance

Having an effective classifier learned addressed a significant part of the goals of this thesis but not necessarily justified the adopted HF-based methodology. Working with HF signals can be more demanding and costly since it requires high-speed sampling devices and powerful processing capabilities. Such pressing requirements can be seen as strong constraints or may appear as reasonable criticism from researchers or industry when evaluating the solution. For this reason, aiming at highlighting the relevance adopted approach, the author decided to pursue a comparison of the predictor information content between the LF and HF signals. Such results can reveal the importance of having the onerous but powerful predictor information present in the high-resolution signals.

Since the team undertaking the project had the intention to analyse frequencies up to 1 MHz, the tests also needed to be sampled at a rate of 2 MSa/s. Due to the amount of data that is generated when sampling signals at such a high rate, they decided to use a sweep sampling method in the HF recording channel. When the trigger to turn on the high sampling recording was asserted, both channels (high and low-pass filter) had their signals sampled. This means that despite already being sampled continuously with a 100 kSa/s sampling rate, the LF signals also had sweeps sampled at 2 MSa/s. That gave the HF sweeps a LF counterpart sampled at the same time, with the same amount of samples. Fig. 4.11 shows an example of two signals (faulty and not), from sweeps (40 k samples per power cycle) of both channels.

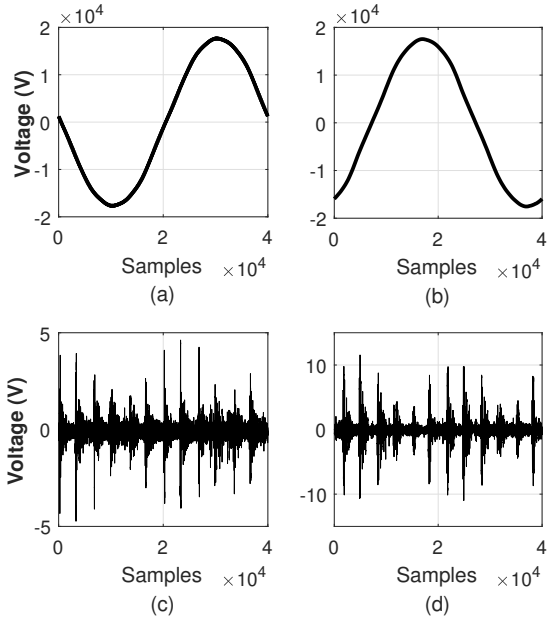


Figure 4.11. Two example of sampled sweeps. LF recordings from a) Non-fault sweep and b) Faulty sweep. HF recordings from c) Non-fault sweep and d) Faulty sweep.

One clear observation from the figure mentioned above is that the fault and non-fault signals are basically indistinguishable. Such illustration is a clear example of the point that such HIFs, with fault currents of single-digit amperes, are much challenging to detect. The main difference between the shown tests is the maximum amplitude of the faulty voltage signal from the HF channel. HF fault signals indeed tend to have higher energy and amplitudes than non-fault ones, but although true, such measure did not show to be consistent enough to be used as a strong predictor. It is mainly because the signal's energy was considerably erratic, not only throughout the test days but also between periods of the day when they were staged.

Although not clearly visible, consistent differences between the two sample stages were identified in the performed experiments. It begins by down-sampling the LF channel sweeps to 100 kSa/s (20 fold), so it could be more computationally efficient. It is worth remembering that the characteristic bandwidth of the channel connected to the low-pass filter was approximately 5 to 50 kHz. This means that, according to the Nyquist-Shannon sampling theorem, the LF channel sweeps at 2 MSa/s was basically oversampling the signals at the task to characterise its related bandwidth. It is also worth remembering that in this experiment, the current threshold for selecting the sweeps was more harsh and limiting. The intention to clearly display the existence of discriminative information, even for every small fault currents, led to a choice of a 0.1 A threshold.

Signal representation techniques were used to create the quantitative measurements used in the comparison. Explained in section 3.2, these were coefficients of popular and renowned signal processing techniques: Fourier-based transform (PSD) and wavelet transform. Table 4.2 briefly illustrates the dimensionality of the working features. The described scheme resulted in a set of six measurements, two from Fourier domain, and four from the DWT. It might seem a small number of measurements, but each one represents a set of multidimensional features. The PSD of the LF signal, for example, results in a feature set of 1001 dimensions. There is an argument for averaging these energy bins to reduce the number of dimensions, but doing so would result in a loss in frequency resolution. In this configuration, they are separated by values of 50 Hz that are all multiples of the fundamental in the LF channel.

The ranking of the features was performed using an index based on the Gini Impurity measurement from the classic Classification and Regression Tree algorithm. Also explored in the Method chapter, the Impurity Index (II) was calculated to represent the best potential decision boundary on a particular feature. A small index means a more *pure* classification zones given by a decision boundary, representing better discrimination of the data points.

The single best split (highest information gain) of each set of features is depicted in Table 4.3 for the LF channel, and in Table 4.4 for the HF channel. The splits are referred not by their number but by their frequency range or centre frequency (PSD) in their respective

Table 4.2. Set of measurements' dimensions

		LF	**HF**
Fourier	PSD	1001	20001
	SF	20	20
Wavelet	EP	8	8
	IQR	8	8
	L_1	8	8
	L_2	8	8

Table 4.3. Split ranking from the LF channel

	Split	**I.I.**	**Sep.**
PSD	$PSD\{11.55\ kHz\} > 3.61 \cdot 10^{-6}$	0.47	0.58
SF	$SF\{15 \sim 17.5\ kHz\} > 0.43$	0.45	0.59
EP	$EP\{25 \sim 50\ kHz\} < 3.24 \cdot 10^{-7}$	0.48	0.59
IQR	$IQR\{12.25 \sim 25\ kHz\} > 2.26$	0.43	0.64
L_1	$L_1\{12.5 \sim 25\ kHz\} > 985.85$	0.44	0.63
L_2	$L_2\{25 \sim 50\ kHz\} > 22.98$	0.45	0.62

braces. In the tables, "Sep." represent the separability potential of the split as a decision boundary. In other words, the separability indicates the percentage of observations that the decision boundary can correctly separate. Such calculation is given by the ratio of correct classifications by the total amount of observations. It is similar to the accuracy in case of using such split to classify the whole dataset between the two classes (faulty or not) of sweeps. It may be worth noting that the tables do not discriminate between faulty or non-faulty observations. The I.I. and Sep. values need to be calculated considering the whole data set to make sense.

The impurity index shows that the HF measurements overperform the LF features at every comparison, although close in some splits. In the same manner, the separability showed that such measurements in the LF channel, when used as predictors to classify such faults, is not much more reliable than a coin toss at labelling the observations.

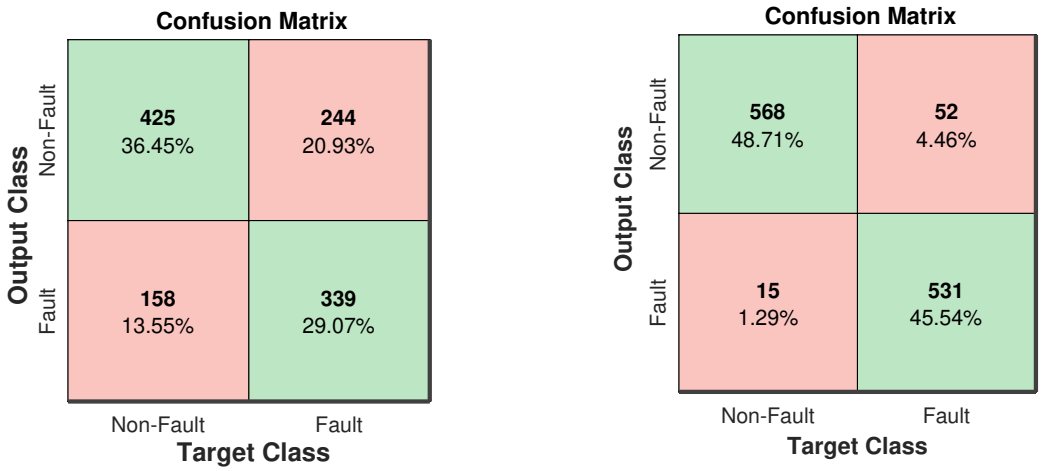
Two features extracted at the HF channel indicated reasonable decision boundaries for fault occurrences separability, namely one IQR and one L_1 measure. The IQR from the HF channel showed the lowest impurity and higher significance. With an impurity index of 0.15 and separability of 0.91, it represented a promising feature from the studied type of faults. Having separability of 0.91 means that if used as a stand-alone predictor, such feature would correctly separate 91% of the dataset samples. The DWT showed to be superior to the Fourier measurements for both channels. Although a consensus in the literature [88, 152, 153], it confirmed the Wavelet transform ability to better represent fast transients in the fault signals.

Table 4.4. Split ranking from the HF channel

	Split	I.I.	Sep.
PSD	$PSD\{34.5\ kHz\} > 5.12 \cdot 10^{-6}$	0.4	0.68
SF	$SF\{1 \sim 50\ kHz\} > 0.02$	0.42	0.66
EP	$EP\{31.25 \sim 62.5\ kHz\} > 1.21$	0.39	0.69
IQR	$\mathbf{IQR\{62.5 \sim 125\ kHz\} > 0.05}$	**0.15**	**0.91**
L_1	$L_1\{62.5 \sim 125\ kHz\} > 250.02$	0.22	0.87
L_2	$L_2\{15.62 \sim 31.25\ kHz\} > 16.67$	0.32	0.77

Nevertheless, a high separability is not evidence of generalisation in the sense of correctly predicting new data (out of sample observations). To attest for generalisation, a further comparative experiment was done. The three best splits over the whole set of features of each channel were selected to fit a simple decision tree, validated in 2-fold cross-validation. This means dividing the dataset in two, using half 1 to learn the tree, and half 2 to test it. Also, doing it the other way around (fitting with 2 and testing with 1) and reporting the average out of sample error. In this experiment, the current threshold of 0.5 A was used.

The confusion matrix, attesting for the features generalisation power and resulted from learning classifiers from both channels, is given in Fig. 4.12. When considered by overall accuracy, the simple tree fitted with three HF features correctly classified 94.2% of the observations, while the tree with LF features only labelled 65.5% of observations correctly.



(a) LF tree's confusion matrix

(b) HF tree's confusion matrix

Figure 4.12. Confusion matrixes from the best three splits in both LF and HF channels.

4.4 Fault signatures

The results presented so far are significant at asserting the discriminative information present in the fault recordings. Nevertheless, acquiring more insightful evidence regarding VHIFs behaviour would certainly help to highlight the relevance of this argument. Differently from presenting observational examples such as the one discussed in Fig. 4.6, producing quantitative evidence about the transients created by VHIFs is much more challenging. This is mainly due to the complex nature of the responses in the voltage signals, and the presence of background noise. Fortunately, signal processing techniques can help reveal patterns from specific types of fault given the availability of the data set of sampled data from real faults.

This section describes the use of the Shift-Invariant Sparse Coding (SISC) technique on the data set of fault recordings to help reveal VHIF signatures. Explained in detail in section 3.2.3, the SISC is an unsupervised learning technique that results in a set of coded basis functions to represent a data set of signals. The basis functions are not necessarily the fault signatures itself, but with the methodology proposed here, some of them are shown to be highly correlated with fault recordings only. These are then labelled as fault signatures.

To exemplify the outcome of the sparse coding algorithm, Fig. 4.13 presents the returned learned dictionary when the number of bases is set to 32, performed over 100 iterations. This length represents a duration of 125 μs, which is a bit more than the period of the lowest considered frequency (10 kHz), limited by the high-pass filter of the sampling channel. In particular, the narrow frequency band close to 10 kHz was intensely present in the HF signals. Bases such as the one shown in the fourth column and fifth row (highlighted) of Fig. 4.13 are an example of this.

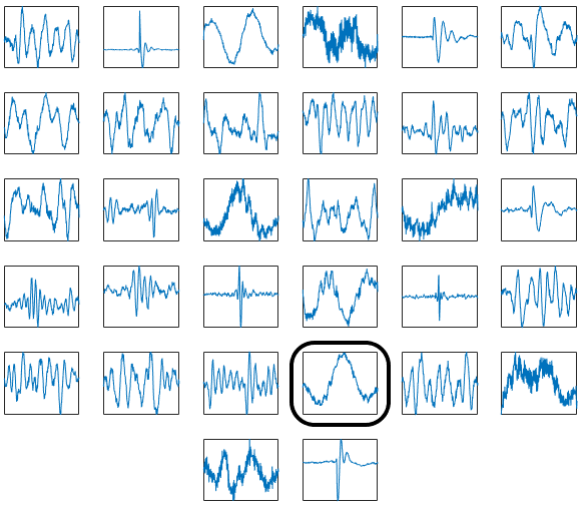


Figure 4.13. Example of a learned 32-basis dictionary.

As seen in Fig. 4.13, when the number of 32 functions is chosen as a hyperparameter, the resulting dictionary includes a considerable number of redundant bases trying to describe a similar pattern. This redundancy suggests that the number of possible underlying sources creating these patterns may not be so numerous, which is somewhat expected from a 50 Hz power system at higher frequencies. Such a redundant number of bases, however, is not advantageous when trying to approximate fault signatures with sparse and effective representations. The underlying hypothesis is that the information to discriminate between fault and non-fault signals will be *diluted* in the redundant high-level representations (features), unnecessarily increasing the complexity of the statistical model. If such a hypothesis is right, the number of basis functions becomes an important hyperparameter to consider.

Changing the number of bases in the dictionary was actually the only hyperparameter that significantly changed the results. It addressed the problem of redundancy, forming a more insightful dictionary. The results in Table 4.5 illustrate the performance change in dictionaries of different size. The accuracy is the dictionary score, resulted from learning and validating a classifier with features extracted using the basis functions.

The features are given by convolution and sum operators, and the accuracy is the immediate result from the 10-fold cross-validation. True positives represent the algorithm dependability, i. e., the percentage of signals from class 'fault' that were correctly classified. True negatives represent the algorithm security, which expresses the percentage of signals from class 'non-fault' that were correctly classified.

The higher separability suggests that there are bases in the dictionary that have a high correlation only with fault signals (fault signatures). It also suggests that, in regards to separability, a fewer number of basis functions works best. The returned 8-basis dictionary, which resulted in higher separability and now used for the remaining discussions in this paper, is illustrated in Fig. 4.14. For comparison's sake, Table 4.5 also presents the accuracy results for dictionaries made of Symlet wavelets, used to extract features in the same manner as the learned bases. It not only attests the effectiveness of wavelets in signal representation but also demonstrates that sparseness is a critical concept from other signal decomposition techniques. As shown in Table 4.5, the 8-basis dictionary can correctly discriminate between fault and non-fault classes with more than 94% accuracy. This is especially relevant since the features used are given by simple cross-correlation calculations between the bases and signals.

Associating a particular basis function as a fault signature, however, requires a more detailed description of the *effectiveness* of each pattern. This association was done by creating a simple linear decision boundary in the feature resulting from each basis. Instead of using all the basis-resulting features to learn an ensemble of decision trees, this approach used only the one-dimensional data related to each feature to create a one-split linear

Table 4.5. Discriminative potential vs. number of basis functions

Dict. size	**Acc. (%)**	Dep. (%)	Sec. (%)	Wavelet acc. (%)
8	94.52	92.4	96.64	94.43
16	94.08	91.52	96.64	93.28
32	93.37	89.93	96.82	93.02
64	90.11	85.51	94.70	92.05
128	88.69	82.69	94.70	92.31

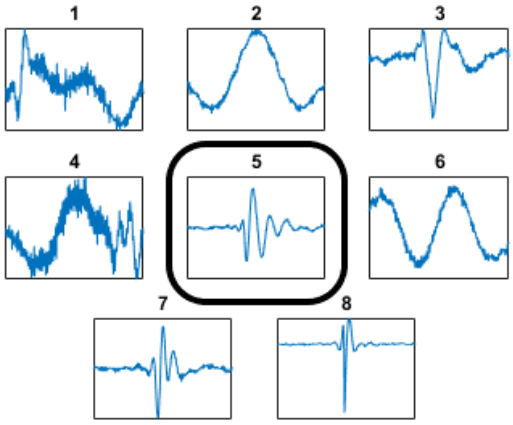


Figure 4.14. Learned 8-basis dictionary.

separator. The result, listed in descending order from the most to less effective basis, is shown in Table 4.6.

The results presented in Table 4.6 can be read as a score correlating individual basis to fault occurrences. It shows that function #5 is invariant to fault signals and can predict them with 90% accuracy. Other basis functions have a reasonable correlation with fault signals and could be argued as fault signatures. Note also that the patterns shown in Fig. 4.14 can be separated in functions that tried to fit the stationary sinusoidal components such as #2, #4, and #6, and others resulted from fitting transients with finite, short existence. This, together with the results shown in Table 4.6, suggests that an added transient component gives most of the effect of a vegetation HIF in the HF signals.

Table 4.6. Individual discriminative potential of each basis listed in descending order.

Functions number	Separability (%)
5	90.46
8	88.78
7	86.48
3	77.74
2	63.52
6	63.16
4	62.90
1	59.72

Another hyperparameter considered was the basis function length in the learned dictionary or, more simply, their number of samples. All the given results here considered 250 samples (or 125 us duration) as the length of the bases. The reason for this choice was that no relevant differences in results were found when varying the length anywhere from 25 to 500 samples. The length of the illustrated fault signatures was then chosen by the ones with more convenient visualisation aspects.

As the fault signatures found in the voltage signal are responses to transients in the current signals created in a fault occurrence, a dictionary was also learned using the HF current signals. Fig. 4.15 illustrates an 8-basis dictionary learned from the HF current signals with the same hyperparameters used to create previous dictionaries. To clarify, the basis functions shown in this figure are the ones learned from patterns in the fault current; they do not have a direct relationship to the ones showed in Fig. 4.14.

Examples of the correlation between the current and voltage basis functions can nevertheless be found by applying a cross-correlation operator in the fault recordings. For example, if one cross-correlate basis no. 5 of Fig. 4.14 with a fault sweep, it is possible to find places where it appears by following the peak of the resulting signal. The zoomed part of Fig. 4.16 was located by such method. In the voltage zoom-in plot, one can see the appearance of basis no. 5, superimposed in the background noise. If the same moment in time is zoomed-in in the current, one can see the impulse-like transient that created the voltage disturbance. It is noticeable that the current discontinuity has such a relatively high amplitude, but it is understandable since the voltage transient that it creates is also so apparent in the signal. Less noticeable, but also interesting, is the fact that the discontinuity in the fault current is also similar to the upper-middle basis shown in Fig. 4.15. Nevertheless, although identified in the signals, such examples are not common or easily located. There are a few reasons for this: (1) the transients are usually convoluted with other signals, (2) the peaks created in the voltage HF signals are usually smaller than

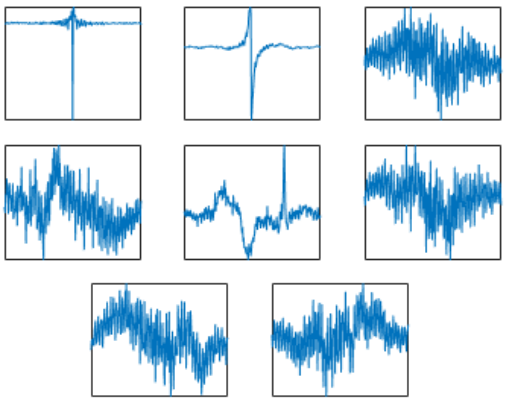


Figure 4.15. 8-basis dictionary learned from the current signals.

background noise (as in the image), and (3) their appearance in time seems stochastic (shifted and non-deterministic). HF current bursts are often given by isolated or shifted convoluted discontinuities buried in noise, such as the ones shown in Fig. 4.16 around the $12k^{th}$ sample on the bottom left plot.

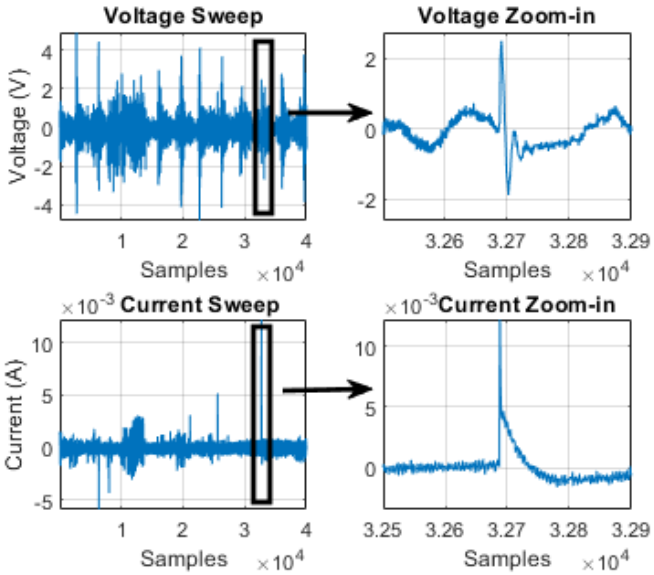


Figure 4.16. Example of a in-fault first voltage and current sweeps zoomed in at strong HF current transient.

4.5 Proof-of-concept prototype

With quantitative evidence supporting the existence of predictive information located mainly in the HF signals, it is desirable to demonstrate the feasibility of adopting the proposed solution. That would mean addressing possible concerns of working with the demanding HF signals and asserting a reasonable implementation cost. The main concerns associated with HF signals could be summarised in two main points: computational complexity due to high-dimensional data calculations (HF) and noise resilience issues. As it is fair to argue that only empirical evidence can adequately address the existent concerns, a proof-of-concept prototype dealing with real signals was produced. It is the result of deploying the proposed machine learning classifier to an embedded system and evaluating its performance. Since computational complexity concerns automatically raise the issue of computational cost, a low-cost embedded set-up was adopted.

The prototype was used to sample, process, and classify HF voltage signals. However, there are some constraining points regarding the adjustments made while using the board that needs to be cleared. As the signals were streamed and sampled by commercial sound

cards, the reproducing/sampling rate was limited to 96 kSa/s. That is much smaller than the originally used sampling rate when recording the HF fault signals (2 MSa/s). An adjustment based on the used sweep sampling mode was then proposed to match these frequencies. It used the effect introduced by the sweep sampling mode that results in plenty of unused periods (98% of each second) in between sweeps. Its implementation consisted of stretching the 20 ms-sweep to last close to a full second, basically reducing the effective sampling frequency to 40 kHz. As explained in the 3.4, this stretching consists of the transmission of the 40k samples of a sweep, in a lower updating rate. A sweep has a duration of 20 ms, leaving 98% of the second in idle time; if stretched, the same signal can be transmitted with much less idle time, in a lower updating rate. Despite this drastic procedure, the board still receives the same amount of information as would the decision-making module of a fault detection apparatus relying on the sweep sampling (one sweep per second). This constraint can be solved by simply adding a high-speed analog-to-digital converter to sample the signals and should not interfere at attesting the method feasibility.

Another noteworthy point relates to the feature and sweep extraction procedures as they are modifications from the ones previously presented. In order to guarantee low computational complexity, a simplification of the feature extraction procedure was adopted while maintaining most of the predictive power. It comprises mainly in using only the wavelet transform as a feature extractor, in conjunction with a linear and a non-linear sum operator. In regards to the sweep considered, this experiment implements a methodology shift by not taking the immediate sweep after the fault current reaches 0.5 A but by considering the following three sweeps after the threshold is met. Such an idea was inspired by current intermittency observations and translates to significant differences in classification results. These procedures and their important implications are further discussed in the following section and chapter 5.

In regards to noise quantification measurements, Table 4.7 is set to illustrate the experiment's results. Despite having outliers, the observed signal measurements compose a skewed (towards minimum value), and narrow, probability distribution. The table's 'max' values were given by an outlier signal that had short and rapid spikes in the HF sweeps. The authors' best hypotheses for that, inspired by empirical testing, is that fast transitions and saturated values were more difficult to accurately reproduce in the streaming part and to sample in the data acquisition part by the hardware.

The 10-fold cross-validation results for the model using the simplified features in this experiment are shown in Fig. 4.17. As clearly visible, the confusion matrix is much different from the ones shown thus far. By this point, the author realized the importance of having more data in the learning phase, even though it made the number of classes unbalanced. Moreover, the number of classes was updated to three. The new class

Table 4.7. Noise quantification results

	SNR_{dB}	$L_{1err}(\%)$	$L_{2err}(\%)$	$L_{inf}(\%)$
Mean	17.56	17.92	14.73	16.65
Min	9.97	5.03	5.83	6.01
Max	24.68	53.43	31.72	41.89

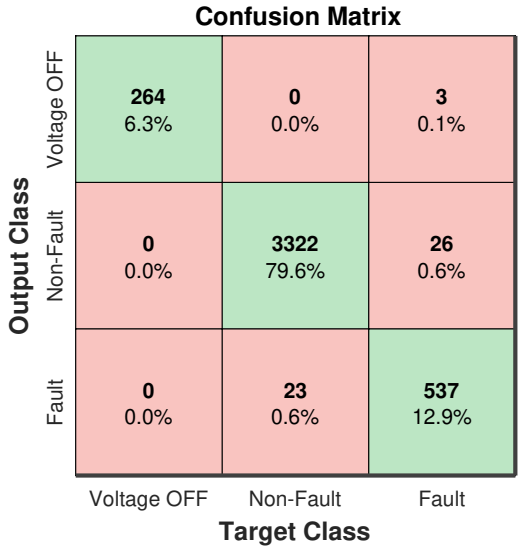


Figure 4.17. Classifier's confusion matrix.

'Voltage OFF' was set to include moments where the voltage source was OFF (test rig not energized), which are observations with distinct characteristics. Such changes had severe implications regarding the model presented in section 4.2, and they will be discussed in the next chapter. Although this model has smaller dependability than the previous, it makes up for having more data for non-fault observations, increasing security (the most important measurement). It is remarkable that this model presents practically the same accuracy as before, now 98.75%, but requiring way less computational effort (only DWT).

The comparison between the prototype versus the off-line MATLAB classification accuracy was made as described in the third and fourth experiments from the methods section 3.4. Their results, showed in Table 4.8, demonstrate that the algorithm presented high resilience to environmental noise with a slight change in classification accuracy. From this point onward, the word *desktop* is used to describe the results obtained in MATLAB environment on the desktop computer, and *board* to depict any results from the experiments made in the Beaglebone.

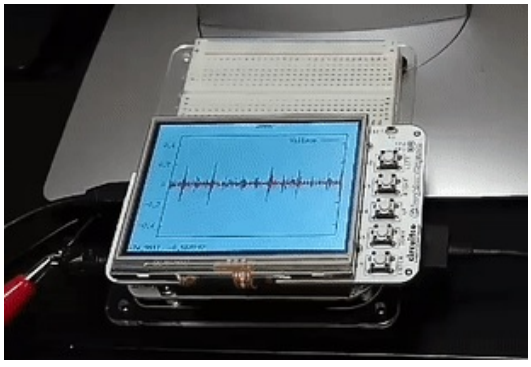
Table 4.8. Board vs. Desktop classification results.

	Desk. acc.	Board acc.
Whole data*	100%	96.47%
Split data	97.53%	95.23%

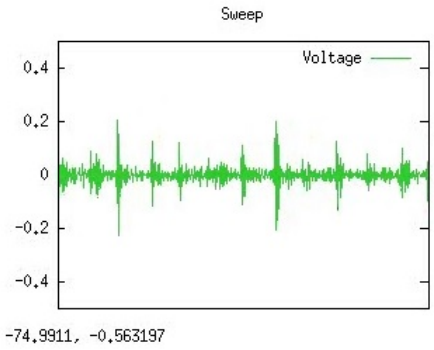
*No generalization value

The board was set to not only process and classify the sweeps online but also to plot the classified signals in an LCD cape while doing it. Fig. 4.18 shows the LCD, sold as a board add-on, plotting the classified sweeps. The plot is updated at every second after the sampling buffer gets flushed, and the values of the present sampled sweep are classified. After classification, the signals were down-sampled to 512 values (reduced only for illustration processing purposes) and displayed in red, if a fault is detected, or in green, if it is labelled as a non-fault sweep. The images, products of the free graph utility *Gnuplot*, were exemplified in Fig. 4.18b and 4.18c as sweeps classified in both classes, Non-fault and Fault, respectively.

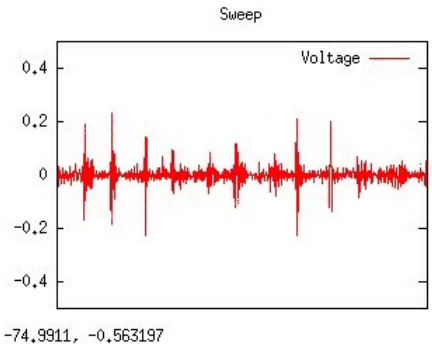
By testing the prototype's accuracy in a noisy and simple sampling procedure, the presented results helped to address environmental noise concerns. Nonetheless, in order



(a) BeagleBone board with a LCD cape plotting sampled signals.



(b) Plot of a signal classified as a Non-Fault sweep.



(c) Plot of a signal classified as a Fault sweep.

Figure 4.18. Implementation of a LCD cape to plot the classified signals.

to further solidify this claim, a noise versus accuracy experiment was set to illustrate the overall algorithm's robustness to noise. The results are illustrated in Fig. 4.19, showing the accuracy (by cross-validation) versus different noise levels (in dB) given by the desktop when artificial white noise was added to the recorded HF signals. A noteworthy feature of this figure is pointed out by the marker at 18 dB. By adding the same noise level as the one measured in the prototype environment (18 is the closest integer to 17.56), the desktop accuracy was close (96.64%) to the one presented by the board classification in the split test scenario (95.23%). It shows that the white noise added in the MATLAB environment has similar effects as the noise obtained in the real hardware set-up. It is noticeable that the slight difference may be due to the use of cross-validation in this test. The same effect was presented when comparing the cross-validation and split test scenario accuracy given by the desktop (98.5% to 97.53%).

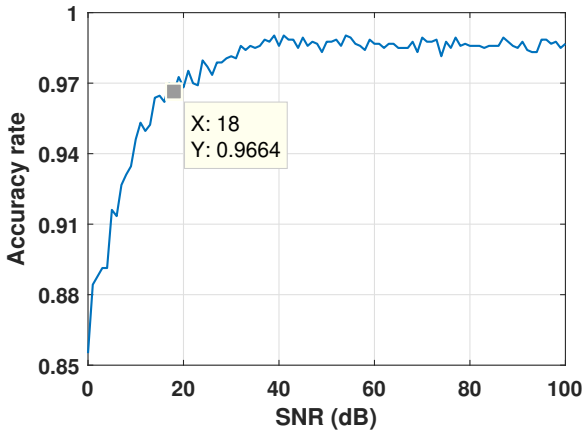


Figure 4.19. Accuracy versus signal-to-noise ratio plot.

4.6 Last working version and performance examples

Choosing the parameters for the last working version of the algorithm is not a trivial task. One needs to decide conditions such as non-fault and fault observations to include, the current threshold for sweep extraction, signal pre-processing techniques, features, and classifier. The decision concerning each of these subjects was made based on classification results from cross-validation:

- Fault observations: the previously mentioned 566 tests, which do not have any problems with data, LF intermittency, compatible sampling and sweeps, and have current conduction.
- Non-fault observations: all available background recording tests, including phase-to-earth and phase-to-phase tests.
- Current threshold: 0.5 A based on fire ignition mitigation recommendations in the staged faults report [4].
- Fault samples: Extraction of the sweeps immediately after the current threshold was met.
- Signal classes: Voltage OFF, Non-fault, and Fault.
- Signal pre-processing: third-order Butterworth filter with a corner frequency of 500 kHz.
- Features: linear and non-linear sum operators on wavelet coefficients (8 features) and cross-correlation with SISC basis functions (8 features).
  - 4-level wavelet multi-resolution decomposition with the Symlet4 as mother wavelet.
  - Sparsely coded dictionary (SISC) with a size of 8 basis functions.
- Post-processing: feedback methodology based on an averaging buffer of features.
- Classifier: 350 Bagged decision trees with a minimum of 20 samples on the tree leaves, validated with 10-fold cross-validation.

Although performing reasonably well, the classifier presented in section 4.2 suffered from being biased. The non-fault observations were extracted from background recordings from three consecutive days, while the tests were performed along multiple days. Not including these tests meant not included phase-to-phase background recordings as they

were recorded in the last test days. These recordings are not only significant for increasing the generalization ability of the classifier but primarily because there were also tests recorded through phase-to-phase measurements. As the noise levels from phase-to-earth measurements are different from phase-to-phase, not having background noise from the same type negatively influenced the classifier. This version, therefore, includes all background recordings available from all test days.

Some updates in the final version include the addition of a signal class, more efficient feature extraction, and enhancement in pre- and post-processing. Having the phase-to-phase signals made the classification of signals more challenging but also promoted improvement ideas. For example, the classifier was under-performing due to the lack of discrimination from non-fault signals had grid-voltage connected or not. A new class of signal named 'Voltage OFF' was added to address this problem. Efficiency was also gained by using only signal processing methods that have $O(n)$ complexity for feature extraction such as cross-correlation with basis functions. A further efficiency boost was also obtained by decimating the signals after a low-pass filter application as signal pre-processing. Results did not change and slightly improved in some experiments by using half of the bandwidth of the signals (10 to 500 kHz). Decimating the signals with a factor of two consequently reduced feature extraction computational complexity by the same factor ($O(n)$). Nevertheless, the probably most substantial contribution from these methodology enhancements was the adoption of a feedback approach in signal post-processing. To be detailed discussed in the next chapter, it mainly consisted of buffering a n number of features from previous sweeps and comparing it to the one to be classified. The feedback practice allows the practical implementation of the method in networks with different levels of background noise and enhanced classifier security.

Results from learning the classifier with the described parameters are illustrated in the confusion matrix in Fig. 4.20. It resulted in an average accuracy of 98.8%, dependability of 97%, and security of 99.09%.

Validating examples from the classifier performance at discriminating the three mentioned signal classes are shown in Figs. 4.21, 4.22, 4.23, and 4.24. These were fortunate exception tests that had voltage supply connected before current conduction started, allowing the testing of the three signal classes in the same recording. It should be noted that the first two plots of voltage and current in the graphs are LF waveforms to better illustrate the fault development. The data fed to the classifier, however, were 20 ms HF sweeps sampled at every second interval. The third plot can be read as a 3-level logical graph where the lowest level means the voltage supply was not ON, the second means voltage connected but signals are healthy, and the third level represents the detection of the fault. As the detection is made based on one HF sweep, sampled once every second, the classifier output also has a one-second step for every classification result.

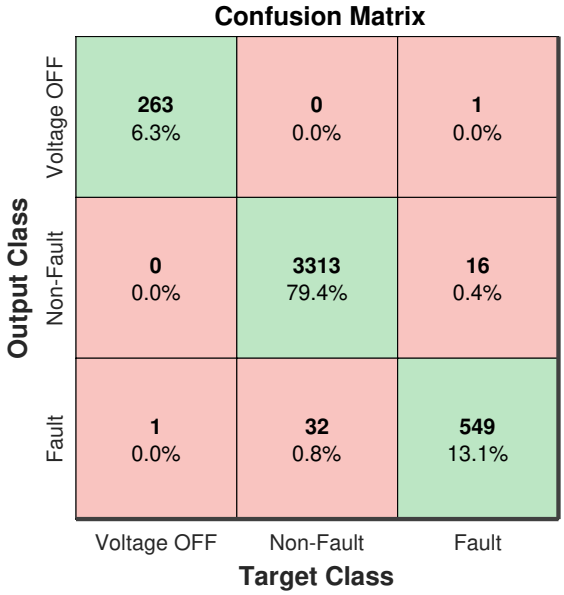


Figure 4.20. Confusion matrix of the last working classifier version.

Current conduction in test #335, shown in Fig. 4.21, started much higher than the current threshold and was immediately picked up by the classifier once the fault sweep was sampled. Test #504, shown in Fig. 4.22, had current conduction started at a minimal level, smaller than 0.1 A. The two parallel black lines in the third plot mark the sweep in which the current threshold was met, namely the 14^{th} in test #504. The noteworthy point of this test is that the classifier correctly detects the fault much earlier, in the 9^{th} sweep, attesting for its high sensibility and accuracy. A similar case happened with test #517, with only the last conducting sweep having RMS current higher than the threshold. Fault detection was asserted since the 9^{th} sweep, following the fault until the end. Test #552 is particularly interesting at illustrating the classifier capabilities. The voltage was turned ON for more than 30 seconds without current conduction. It was turned OFF for a bit longer than 10 seconds and then turned ON again, starting the current conduction. The current started with a small value and did not introduce much detectible fault signatures until more than 15 seconds after its inception. One can see two attempts of fault detection from the classifier between the 60^{th} and 70^{th} second before the detection was asserted. Nevertheless, although taking almost 20 seconds to assert the detection, it was still made it much before the hazardous threshold of 0.5 A was met and for all sweeps after.

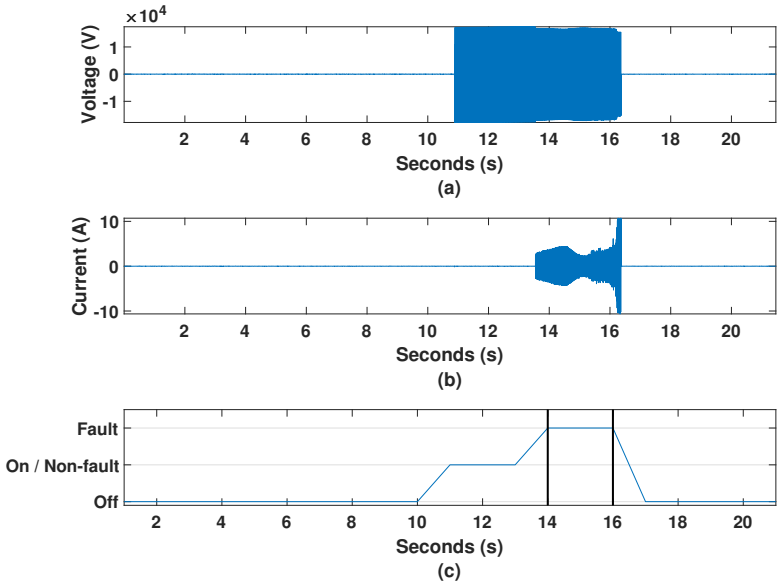


Figure 4.21. Performance example on test #335. a) LF voltage recording. b) LF current recording. c) Classifier output.

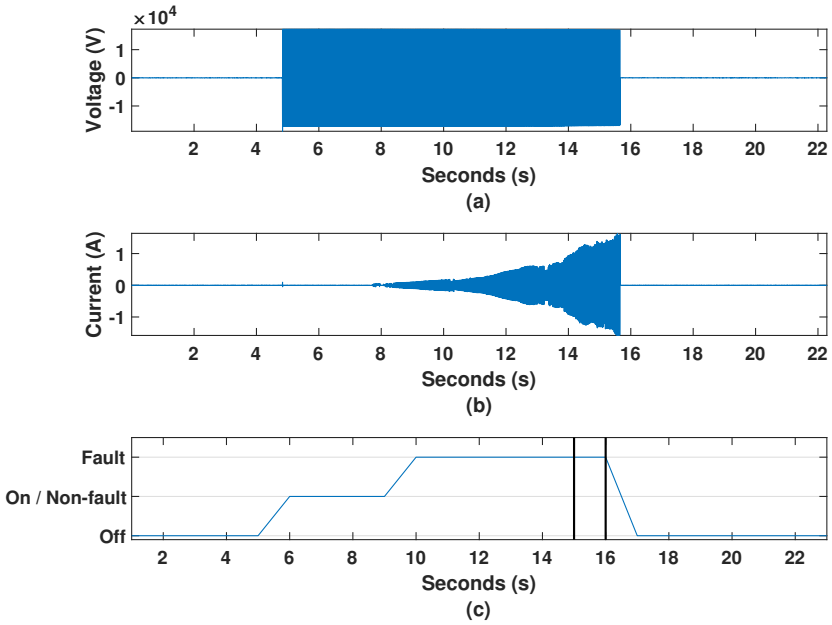


Figure 4.22. Performance example on test #504. a) LF voltage recording. b) LF current recording. c) Classifier output.

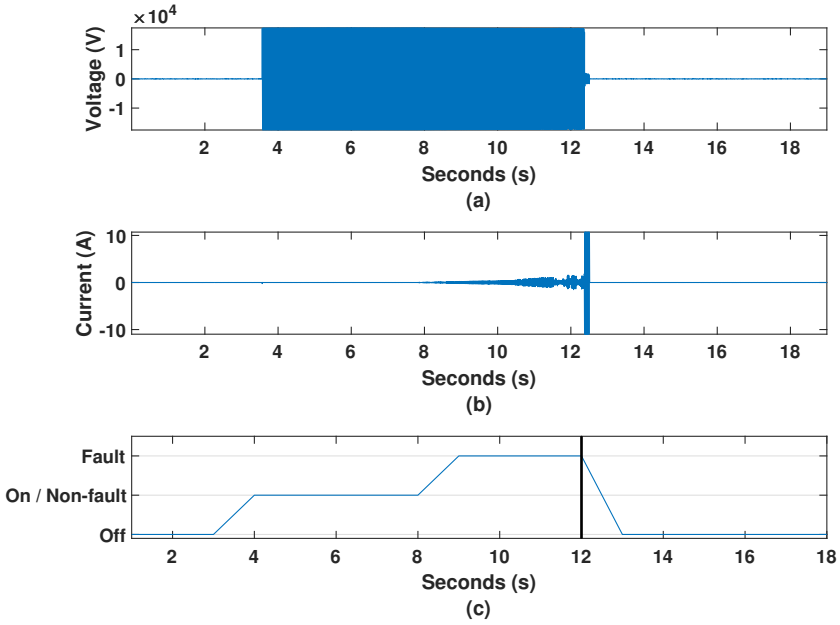


Figure 4.23. Performance example on test #517. a) LF voltage recording. b) LF current recording. c) Classifier output.

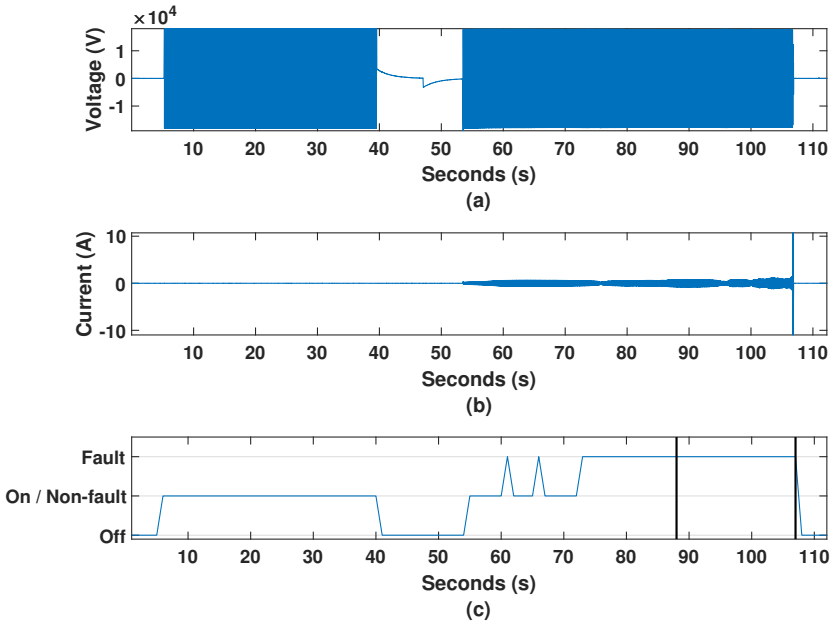


Figure 4.24. Performance example on test #552. a) LF voltage recording. b) LF current recording. c) Classifier output.

4.7 Results summary

The results presented in the previous sections are the outcome of applying the concepts discussed in Chapter 3. Each section builds on the previous, taking steps closed to a better classifier and understanding of the VHIF phenomenon.

Initial investigations started by analysing the background noise and ended with evidence of discriminative information in some recordings. Frequency components with the highest energy were noticed in a narrow band centred at 10 kHz and in a more wider band with discrete carriers in between 250 to 700 kHz. The former was believed to be due to large grid loads [4] and the latter was linked to radio navigation and AM radio broadcasts. With the help of some exception tests that had pre-fault recordings, the initial investigation confirmed that a fault occurrence could introduce substantial frequency components, especially in the range from 100 to 450 kHz. It also confirmed that low-frequency harmonics might not be noticeable in the first seconds of a fault, which are crucial for detection targeting fire mitigation. An example of fault current discontinuity creating an oscillatory transient in the voltage signal at the same band affected by the fault was also illustrated.

These results gave the confidence to continue the classifier conceptualisation but were not evidence that such insights could be generalised. As such evidence would require a large sample size, the first step was cleansing the data from potential problematic recordings; tests that were missing, corrupted or had no conduction were excluded. A data set of labelled observation was assembled by using the first sweep sampled after the fault current reached 0.5 A (fault), and from background noise recordings (non-fault). A combination of Fourier-based (PSD) and wavelet-based features shown to be effective at classifying the faults. However, the wavelet features such as sum, peaks, and energy percentile of the coefficients were much superior in explaining the invariance between classes. Many machine learning algorithms such as linear or quadratic discriminants, SVM, and KNN were tested, but the ensemble of decision trees constantly over-performed them.

The classifier learning showed evidence that the existence of discriminative information is generic, but it still was not evidencing that using the HF signals was a necessary approach. An experiment was then conceptualised to compare the discriminative information content between the LF and HF signals. When analysed under the same methodology, it showed that the LF signals do not present enough information to be a consistent predictor of faults as HF signals. Such results were also evidence that the non-appearance of low-harmonic content in the initial moments of fault is indeed also generic.

Although initial investigations showed an example of oscillatory voltage transients created by the fault current, it still did not present evidence of their recurrence and generality. An experiment composed of applying the sparse coding technique to find explanatory bases gave evidence of their causal relationship and illustrated the transients in time-domain. They also helped to enhance the classifier by serving as pattern-matching filters that allow better discrimination of fault occurrences.

Since all the mentioned experiments were offline, this research could benefit from evidence that such classification could be done in real-time, in a deterministic manner. The production of a prototype helped to corroborate this possibility while presenting itself as a

low-cost solution. It also showed to be a reasonably accurate classifier in the presence of considerable environmental noise, showing possible robustness for real applications.

The last version of the classifier implemented the lessons learned by all the experiments while assuming three different classes, as opposed to the first binary versions. It also comprises the basis-based features and the comparative post-processing method, which is expected to enhance the adaptability of the method further. The constraints, advantages, and further discussions of the results and approach adopted in this thesis are presented in the next Chapter — Discussions.

Chapter 5

Discussions

5.1 Results relevance

As any discrete scientific research, the relevance of the results described in Chapter 4 is bounded by layers of uncertainty. For an explicit discussion on the relevance of the results, it is useful to start from the most unequivocal findings presented by the results and then expand into potential applications.

5.1.1 Certainty, variance, and bias

The clear difference between the classes of signals that were analysed is probably the fact with the highest certainty. It is essential to make this distinction clear. The classes of signals represent recordings sharing particular characteristics. The 'Non-fault' class represents voltage recordings from the substation, sampled in the same days and throughout the staged vegetation tests. The 'Voltage-off' class are simply recordings from electromagnetic interference captured by the test rig apparatus. These two classes and the one from the voltage recordings made when staging the faults — 'Fault' — individually present *invariance*. As far as certainty goes, the countless cross-validations performed in this data set show that signals from these classes are different in some way. This affirmation may sound obvious, but as seen in the comparative visual examples from 'Fault' and 'Non-fault' signals in Fig. 4.11, their difference is not much evident. Their similarity is the reason why discriminating signals from VHIF occurrences is not trivial. The classifier may have reasonably high accuracy, but it does not completely separate the observations; the exact cause for this lack of total separability (100% accuracy) is not entirely known. It is possible to hypothesize that the few miss-labelled observations were sweeps recorded in intermittent moments of the fault. An experiment made by selecting the sweep with the highest energy, between the first three sweeps after fault inception, points to this hypothesis

to be relevant due to a resulting increased accuracy. However, validating such a hypothesis would be impractical for it would require a drastic amount of data. Regarding the layers of uncertainty, the next level beyond class invariances includes aspects such as bias, variance, and generalization.

Discussing the classifier's ability to generalize is hard, it is easy to overstate it, and there are incentives to do it. A discussion on the significance of its resulted accuracy can clarify it. The classification results from learning and testing a classification algorithm on a data set it is probably not exactly how it is going to behave when tested in real life. It would only be the case if the data had no bias and a low enough variance that its sample size would be able to represent effectively. This chapter presents some arguments and methodology aspects designed to deal with possible issues resulted from bias and variance, but some aspects of which are unavoidable. It is easy to argue, for example, that the variance of the VHIF problem is enormous. There are many types of network configuration, equipment, and environmental conditions that may affect the fault behaviour, and consequently, the performance of the classifier. Therefore, it would be necessary to get data from all these conditions to represent all possibilities that the classifier may face. To most conditions outside the ones represented in the data set, all statements regarding real accuracy, dependability, or security, are only well-guided speculations. Nevertheless, despite not often discussed, any other top-down HIF detection method proposed in the literature relying on *a posteriori* knowledge suffers from these issues. It appears that this lack of transparency and standard data sets (discussed in further sections) cuts deep through the difficulty of establishing a consensus regarding HIFs; the fact that it is a challenging and high-variance problem only exacerbates it.

The bias of the data set is mainly related to its specificity, which has positive and negative aspects. There are several characteristics that make the tests specific: fault surface (vegetation), limited fault currents, network grounding type, distance from the fault point, electromagnetic interference, network load, and intrinsic noise sources. Some of these are arguably more important or beneficial than others. For example, having a limitation on the fault current results in a more precise estimation of the method's fire mitigation potential while resulting in almost no adverse effects for its generalization. Such a general positive effect is mainly given by the fact that faults with higher currents will primarily result in more intense effects on the network and will consequently be more easily detected. This effect is somewhat expected, but it was also attested when learning classifiers with data filtered to include signals with different current thresholds. The results showed lower accuracy for a 0.1 A threshold in comparison to faults with a 0.5 A detection threshold. Conversely, differently from the current limitation, some biases will generally result in adverse effects to overall generalization; the load noise, weather conditions, and electromagnetic interference are good examples. It is also understandable that these issues

might generate concerns from technicians, engineers, and research peers. Nevertheless, given that there is a limitation on resources, data, and investigation time, having to deal with a level of uncertainty is unavoidable.

There are some practices that help to deal with the effects of variance and bias in the potential generalization of the method. The commonly discussed ones are normalization and standardization, which can be applied before the learning stage. The need for those practices is evident in any application of this method since testing signals of different scale would completely alter the performance of the classifier. Adding to standard practices, moreover, original methods were also developed to help mitigate variance and bias effects. The main one was called the 'feedback method' as it functions by introducing information about the current network state to the data before it is fed to the classifier. It consists of storing feature vectors from the previously sampled sweeps in a first-in-first-out buffer and calculating the 90^{th} percentile of each set of feature. Then, they are directly compared to the newly sampled sweep features before used for classification. The comparison is made by calculating the ratio (element-wise) between the newly calculated features to the percentile value of the stored features in the feedback vector. Such an approach means that the classifier does not receive the information of the actual feature value, but the ratio of that feature to a previous state estimator of the network. It is hoped that such a method will make the classifier more flexible towards different conditions. For example, if a network is particularly noisy at a specific bandwidth that is relevant for detection, it could derate the security of the classifier by making it produce false positives. With the feedback method, nevertheless, the classifier is learned to discriminate changes in states, rather than specific values.

5.1.2 Classification results

It is important to be honest to what the accuracy value really means: a crude estimator of the method real performance. Arguably, accuracy is not as important to this work as it appears to be in many machine learning-based publications. To many of these works, the novelty or contribution comes from conceptualizing a method that over-performs previous approaches. In the same manner, comparisons are also expected from peers when proposing a HIF detection methodology. Comparing accuracies with previous methods makes sense when both are working with the same data or model. However, standards data sets for tasks such as facial or handwriting recognition are easily found online while the same can not be said for HIFs. Yet, it is possible to find papers comparing results with works that use entirely different data or models. In this thesis, in particular, the novelty comes from building a method on top of a novel data set never formally investigated for this purpose. Even though the accuracy presented here is indeed higher than other

methods, making such a comparison would be futile and misleading. A recent literature review publication cites the performance of existing methods [11], in case the reader is interested in such estimates. Regarding useful performance comparisons, it is desirable and welcoming that other authors also create methods using this data set to possibly find better alternatives to the contributions made in this thesis.

The key novel aspects of this work go beyond just attesting high accuracy. It is related to its specificity regarding fault type (vegetation HIFs), its sensitivity (0.5 A), and the fact that it uses the feeder's high-frequency voltage signals. The fact that results were consistent for either phase-earth and phase-phase faults is also relevant since HIF is mainly discussed by a conductor breakage scenario (phase to ground). The accuracy, in this sense, is useful to attest the positive results from this investigation and its potential application to detect VHIFs and possibly fire mitigation. To make a more certain claim: the presented results mainly support the merit of continuing to investigate the real applicability of the adopted approach.

5.1.3 High-frequency signals and patterns

For such positive results achieved, having access to HF signals was paramount. It is unlikely that the staged faults could be detected without information of higher resolution since the HF components are low in magnitude. The indicating evidence for that was presented in section 4.3 where the predicting information content of the LF and HF channels was compared. The experiment opens the question of how high the resolution must be, or rather, what is the lowest sampling rate that will result in adequately representation of meaningful, discriminative information. These kind of questions are part of the worth-discussing, complex problems that are touched in this thesis but that also will remain open until more focused research is done. Some progress can be achieved by starting with the signals sampling bandwidth responsible for such positive results: 10 kHz to 1 MHz. Experiments performed with the data can inform some of these frequency bands importance for accurate detection. The first comprised the adoption of a third-order Butterworth filter with a corner frequency of 500 kHz as a pre-processing stage of the signals. It effectively cuts the bandwidth in half, filtering most of the high-frequency components. Its effect on the classification results was from negligible to a slight improvement on the accuracy, pointing the discriminative information to be concentrated in frequency bands lower than 500 kHz. Moreover, experiments with wavelets and sparse coding showed a large part of the discriminative information to be located between 125 and 250 kHz. When their respective feature is tested individually, it showed that they were responsible for representing a substantial part of the invariance between classes. Nevertheless, having other frequency

bands represented in complementary features is necessary to reach the classifier highest accuracy values.

The experiments with the sparse coding technique pointed that the damped oscillating transients in the voltage are a substantial part of the VHIF behaviour. The simplest explanation for the transient events is related to the second-order circuit response to impulses or steps in the HF current signals. If these transients are indeed second-order responses, they would considerably change given different line RLC parameters, which may represent a strong utility constraint. Notwithstanding, as seen in the 32-basis dictionary in Fig. 4.13, when the same test rig, network, and fault distance are considered, the transient response can still drastically vary due to the different characteristic resistance and reactance of the contact surface (vegetation sample). Therefore, even when experiencing high variance in transient responses, the features extracted from the resulted basis functions, as well from the wavelet transform, were still able to effectively predict and represent the invariance between classes (fault and non-fault). The wavelet transform, in specific, is a powerful tool for the task of fault detection since the filters' frequency response is smooth and can capture tuned bandwidths of resonant transients created in the HF signals. Hence, even if the RLC parameters change considerably, the coefficients would probably be significantly activated and useful for detection support.

It is important to note that the patterns presented by the application of the sparse coding were still somewhat present in the other classes of signals. Otherwise, the observations would be completely separable. The critical information is that some were highly correlated with fault occurrences while others not; highly-correlated patterns had feature values tens of times higher in the 'Fault' class than others. Such a difference means that although present in both, the patterns were much more recurrent in fault occurrences. It also means that unsupervised learning techniques like sparse coding could have considerable potential to improve HIF and other disturbances detection, profiting from the existence of representative data. Learning from available data could aid the further understanding of disturbances behaviour in the network and increase the method's accuracy. The resulting patterns could be added as additional discriminative information. HIFs are often discussed as random events with characteristics hard to quantify such as build-up, non-linearity, and high intermittency [2, 11, 16]. These appearances may however have fewer random characteristics than initially thought, especially when considering specific conducting surfaces.

5.1.4 Applicability considerations

Although achieved in a narrow scenario, the results point to the possibility that the proposed method could have positive effects on fire mitigation. As per the final report from the staged

vegetation tests [4], a detection sensitivity of 0.5 A in less than 2 seconds could reduce fire ignition risk in tenfold. The results from learning the classifier and proof-of-concept prototype certainly attest these constraints. One aspect that could generate concerns about detection speed relates to the adoption of the sweep sampling method, which introduces some delay in detection and other possible disadvantages (further discussed). Nonetheless, even sampling one diagnostic sweep per second still results in time enough to meet the time constraints. In a worst-case scenario, a VHIF may begin right after the recording of the last sample of a given sweep, leading to a longer detection delay. This event means that sampling could possibly take one second, though not likely, and that detection should be asserted in the next second. Despite comprising high-resolution signals with longer arrays, such detection time was achieved even with low-cost microcontrollers. It is therefore arguable that higher computational capabilities would further alleviate such overhead and much more could be achieved. Computation complexity issues are also further discussed in the following sections.

If sensible, discussions on the fire risk mitigation by any method will include more nuanced arguments than plain accuracy. The layers of complexity include aspects from dependability and security to how the detection signal is going to be eventually used. Regarding the more quantifiable issues of dependability and security, discussions started to appear in the early 90s [73], when authors started moving to real implementation. The relationship between these two parameters needed to be considered because they represent distinct sides of a trade-off. For example, if the goal of the detection method is never missing an event occurrence, one should care to improve the dependability of the classifier. Making it more sensible to all occurrences, however, could make the classifier more prone to detect false positives: classifying nominal operating events as fault occurrences, as in this thesis case. Caring for dependability over security is a legitimate option for problems such as cancer diagnosis where a false-positive result in minimal damage while a false negative could be fatal. For fault detection, nevertheless, it is arguable that the case is the opposite, and false negatives are somewhat preferred. The reality is that this dichotomy represents a complex optimization problem involving many factors such as the economics of service discontinuity, load importance/priority, weather conditions, and area under the feeder. A realization of these factors would result in values to be assigned as the weight of the observations in the learning process. The methodology presented here is unable to cope with all these factors, but ideas were conceptualized to help enhance the method's security and simplify its trade-off with dependability.

It is certainly not desirable to interrupt service to customers or to disconnect priority loads due to a false positive fault detection. Fortunately, for most of the experiments performed in this research, including the last working version of the classifier, results were always better for security than dependability. As shown in the results section 4.6, security

results reached more than 99% in the last version of the classifier. Although being a crude estimator, as previously discussed, implementation methods can still further improve the classifier's real security. The idea proposed here is, instead of acting on the first positive detection result, to only assert detection after two or three consecutive positive results. In the best-case scenario, where the observations are considered as independent events, the error will exponentially reduce by every consecutive sweep needed for the assertion. The downside of adopting such an approach is that every consecutive sweep added to the assertion requirement will add one second in the fault detection delay. Nevertheless, two points should be made regarding such fault delay. The first concerns the findings in the vegetation ignition test final report [4], which stated that fault detection under five seconds might still achieve substation fire risk reduction. In a scenario where an one-sweep-per-second approach is implemented, three consecutive sweeps would still be under the five-second threshold. Moreover, sampling one sweep per second is not fundamental to the application and could be increased given more computational power. Sampling two sweeps per second, for example, would result in a fault delay potentially under the two-second threshold. The second point regards the sensitivity on the method. The remarks and suggestion for fault detection delay in the tests report were based on a 0.5 A sensitivity. However, as shown in examples in section 4.6, the method will start detecting a fault before the current reaches 0.5 A and it did present a reasonably high accuracy when a 0.1 A threshold was adopted, as shown in section 4.3. If the method can indeed present such sensitivity, its detection delay constraints could potentially be more relaxed. Another idea for the usage of this method regarding service continuity includes using it as an extra function in automatic circuit recloser devices. The potential benefits would be a short period of service interruption, allowing for self-extinguishing faults to be removed, and higher overall security, as it only locks out after repeated fault detections. All these ideas could be further improved with adjustments to different weather conditions where dry and wind days have a more restrictive setting than other conditions, for example.

5.2 Implementation

An implementation of the proposed approach could be discussed in three individual parts: Signal acquisition, Processing, and Communication. The scope of the discussions in this chapter only involves the first two aspects due to their large intrinsic complexity. This section will start with the part that is more closely related to the primary thesis goal, the processing stage. Further discussions on signal acquisition are relevant since different sampling techniques result in more relaxed or stricter constraints.

5.2.1 Processing and classification

An essential aspect of the signal acquisition is the mode it uses to sample signals. The one adopted in the ignition tests, sweep sampling, is just one of the many ways it could be done; they could assume larger or smaller durations, or even different periods. Sweep sampling definitely offers some advantages in comparison to continuous sampling. The reduced hardware overhead is probably the most convenient one. In the case of the adopted approach, for example, the 20-ms sweep represents 2% of the sampling period (one second), allowing the remaining 98% to be available for signal processing, diagnostics, and communication.

One might think that adopting a sweep sampling approach to be detrimental for having less information, but all the results presented in the previous chapter attest for their sufficient discriminative information content. The confirmation of the hardware overhead benefits was observed first offline in MATLAB environment, and then in the proof-of-concept prototype (time complexity is also briefly discussed). An example of the former appeared when testing a fault recording of 21 s duration (21 sweeps), which ran a script that loaded the signal, calculated the features, and labelled the sweeps; it took less than 9 s to perform these actions, attesting for its real-time implementation potential. This *slack* in overhead is especially notable due to some MATLAB characteristics: it is scripted (not compiled), its code is not vectorized, and it represents a higher level language compared to others like C and C++. More direct observations of the potential for real-time implementation came when developing the fault detection module prototype. Despite not sampling the signals at the same rate as in the tests, it still had the same amount of time to make every decision (one second). In its final form, the microcontroller had to request continuous sampling from the audio card, while calculating the wavelet-based features, making the classification, and plotting a representation of the just-sampled signal in an LCD extension cape. One of the motivating reasons to build the proof-of-concept prototype was indeed to see if it could handle sampling and classifying the signals in real-time, backing up claims of potential feasibility. Even with the demands of processing, a slack on the microcontroller processor was observed, which is a sign of being deterministic. Therefore, the sweep sampling mode is one of the main drivers of reducing the possible cost of a full-size prototype since, by only taking snapshots of the signal, even a card-size computer like the *Beaglebone* can handle its computational requirements.

Time complexity theoretical constraints are probably worth discussing since they are one of the main aspects to be considered before implementation. Although the processes of feature extraction and classification might generate some concerns, there are arguments to label them as somewhat reasonable when compared to other signal processing techniques. The features adopted here are extracted by a convolution between the signal and basis

functions (in the sparse coding application) and wavelet filter, followed by the application of a sum operator. There could be a problem if the signals being convoluted were of the same dimensions since the convolution of signals with size n and m would yield approximately nm multiplications, resulting in an approximate $O(N^2)$ time complexity. However, this estimation would not apply to the feature extraction by basis functions due to their limited and fixed dimension size, which are much smaller than the signals (approx. 160 times). The proposed procedure is more similar to the basic filtering process of the wavelet transform, which has approximately the same number of multiplications as the input signal. This procedure would then result in a time complexity approximating $O(N)$ for the convolution and wavelet transform, compared to the $O(NlogN)$ for the Fast Fourier Transform (FFT). Moreover, even if the FFT could effectively represent the features of VHIFs — and evidence points that it cannot (see section 4.3) — it is unlikely that it would be directly used to extract features in a 40k-sample signal. It is more likely that an estimator of the power spectral density such as the Welch method would be used to reduce variance and dimension of the feature space. An average estimator like Welch would result in many runs of the FFT algorithm at each observation, further increasing the computational complexity.

A comparison between the Welch's estimator, wavelet, and proposed signatures for feature extraction was performed to exemplify the time complexity points. The test considers the run time from the input signal to the complete matrix of features. As fewer features would result in more calculation in the Welch-based feature extraction method, the number of 100 was chosen as the size of feature space (FFT size of 200); no overlap between windows was considered. The wavelet dictionary was composed of 8-levels of Symlet4 wavelets, with features also extracted by convolution and sum operators. The run time was given by the MATLAB profiler tool, which can output the time needed to run each function. The average time for ten runs for each feature extraction is shown in Table 5.1. These results, moreover, present itself as a counter-argument to other works that may quickly dismiss the use of the wavelet transform for this task when proposing a novel and more complex technique.

Table 5.1. Run time comparison of different feature extraction methods.

Welch (s)	Wavelet (s)	Signatures (s)
0.1356	0.0477	0.0103

Although the method proposed here needs high-speed sampling, there is an argument to be made regarding the relative simplicity of the classification and its advantages. There are trends and incentives in the related literature to use increasingly complex methods to detect HIFs, as discussed in Chapter 2. That weighed heavily in the period of conceptualization

of this method, as the author searched for proper ways to address the classification task. However, it is arguable that the classifier should be as simple and interpretable as possible, without extra complexities that do not add to its performance. This argument is based on the concept of VC (Vapnik-Chervonenkis) dimension [154], which is one of the bases of statistical learning theory. To put it simply, the VC dimension is a measure of the complexity of the space of functions in a classification algorithm. As a comparison example, a linear discriminator or perceptron has a much smaller VC dimension than a neural network composed of many layers. The perceptron can only create linear decision boundaries, being much more restricted in ways to fit a dataset; a neural network composed of many layers, however, could fit complex patterns that could go beyond of the capacity of polynomials of high order. Therefore, when tackling complex multi-dimensional classification problems, one needs algorithms that have the capacity equivalent to the problem complexity and desired VC dimension. However, the higher the VC dimension of a classifier, the more prone it is to *overfit* the data set. Algorithms with high capacity can easily *memorize* data sets and compromise generalization ability if the data set is not large enough. This effect is known as the *variance-bias trade off* [155]. In general, a model with high bias will pay less attention to the training data and possibly result in oversimplified decision boundaries A high variance model, conversely, will fit many aspects of the data and possibly not generalize well to out-of-sample observations. Therefore, given the problems of variance in the data discussed at the beginning of this chapter, one needs to be careful not to overfit the particular conditions of the test and deliver a classifier with a poor chance of generalizing. Notwithstanding, if the model is too simplistic, it might just have poor performance regardless. Regarding the VHIF detection problem, amount of data, and the approach taken here, it is arguable that the methodology presents a relatively well-balanced complexity: a relatively small number of features, eight wavelet and eight basis-functions, and a classifier given by an ensemble of shallow trees.

The perfect scenario would be to have a low-bias low-variance deterministic classifier with parameters that completely addressed the physical phenomena of HIFs. As previously stated, however, complexity is an intrinsic characteristic of this problem, and one has only so much data and time to conceptualize a reasonable way to detect these faults. It is possible that a decision of going towards more complex models such as deep neural networks could (1) not result in optimal performance due to the size of data or (2) completely lose any interpretability by delivering a black-box classifier. With the reduced number of features, the work done with the linear boundaries in section 4.3, and the sparse coding results in section 4.4, it is arguable that interpretability was still present in some level, which resulted in some VHIFs behaviour insights. It is also arguable that a classifier with clearly understood inputs and additional work developed to attest its potential would be received with more confidence by decision-makers and field engineers.

5.3 Signal acquisition

The signal acquisition part needs to be addressed because it is a crucial part and one of the most constraining aspects of a real implementation. The challenge of acquiring the signals it is not primarily related to data sampling; analog-to-digital converters are reasonably accessible at the required sampling rates. The complication is instead linked to the requirement for high-voltage transducers, which will transform the primary feeder voltage (12.7 kV) to a data acquisition-friendly voltage level. There are different ways to achieve such a task; each has particular associated costs, reliability, and popularity. In regards to the latter, novel signal acquisition technologies might just acutely increase the economic viability of the hardware implementation. Moreover, as the methodology relies on high-frequency sampling, considerations on the signal attenuation, cost, and potential use of technology become relevant. If the decision of installing such hardware is indeed taken, the potential uses of it, in particular, are much vast than only HIF detection.

Although already present in the field, traditional signal transducers may not be suitable for the application of the method presented here. One of the reasons for this incompatibility is their inductive nature. Voltage transformers used for relaying in power distribution substations are similar to power transformers in the sense that they use inductive coils of different ratios to scale down the primary voltage. The resultant inductance has an attenuating effect on the high-frequency signals, which are essential information for the detection. An alternative such transformers would be capacitive coupled voltage transformers (CCVTs), which are used in transmission systems in power line communication systems. However, having expensive high-voltage CCVTs installed in power distribution substations for detection vegetation HIFs is not ideal either.

The ingenious configuration for signal acquisition used in the vegetation tests could be a fair compromising solution between cost and dependability. The equipment used a capacitive voltage divider in a similar configuration as a CCVT but much less costly. They used a high-voltage coupling capacitor for dropping the voltage, in series with a smaller capacitor that had lower, sampling-friendly voltage levels. The clever part of it was using the high-voltage coupling capacitor as a voltage dropper when it was originally commercialized as assessing equipment for partial discharge tests. It was an Omicron MCC 124 coupling capacitor, with a nominal capacitance of 1.1 nF and a maximum voltage of 24 kV. When combined with a 100 nF bottom-end capacitor and a 220 Ω shunt resistor, the transformation ratio was about 100:1. As the high-frequency channel also contemplated high-pass filters, selecting out the high-energy, low-frequency components, the voltage signals resulted in RMS values at about 1.4 V. More information about the data acquisition methodology can be found in tests report [4] or in a brief description of the tests rig set-up and configuration in Appendix A.

The capacitive voltage divider is certainly a more accessible solution than high-voltage CCVTs, but non-traditional sampling technologies might reduce the cost even further. These innovative technologies are mainly sensor/antennae sampling methods that could be used to access the HF voltage signals with the added advantage of not being invasive. Recent patents publications [111, 112] show the increasing interest in the application of such solutions and its possibly more affordable commercialization. The document [111] disclosures a solid-state electric-field sensor that can sample signals for a target located at a distance (powerlines) by the change in the electric field in a voltage-controlled capacitor. In the same manner, the other patent [112] discloses an early fault detection system that samples signals with antenna sensors, also without the need for physical connection (at a distance). Early fault detection systems based on the patent [112] have actually being installed in Australian systems and are presenting initial positive results [156]. This increasing interest in sensor-based sampling technologies was a supporting argument when considering using the voltage signals in this thesis, which is not a common approach in the literature.

This reduction in cost is probably essential to the implementation of this and other diagnostic technologies mainly because they might have to be installed in more than one location in the monitoring feeder. Another potential relevant concern from the presented approach is the stray attenuation effect of HF signals in overhead powerlines. In the same manner that inductive coils can attenuate these signals with their nature reactance, the stray capacitance of powerlines acts as a smooth low-pass filter. This effect might just signify that the sensing hardware will need to be installed in multiple locations throughout the feeder. However, although this put in question the practicability of the methodology, there is evidence that the number of hardware may not be prohibitively high for economic feasibility. The formerly mentioned fact that there are commercially viable technologies that use similar principles as the one presented here [156] is the first evidence. As reported in the tests [4], the company hardware has sampling rates around tens of MHz, making use of much higher frequencies. The other evidence is a Masters thesis [60], produced in the Australian college (RMIT), studying the propagation of partial discharges. It pointed through simulations that frequencies up to 25 MHz may still be sensible up to tens of kilometres in overhead-line systems. It may be the case that these points are not decisive evidence and that there are questions to be answered regarding the requirement of sensing hardware to guarantee the complete monitoring of a feeder (discussed in suggestions for further research). Nevertheless, there is also the undeniable fact that there are players in the market, making similar technologies commercially viable in the present day.

It should be lastly mentioned the potential benefits that would come for having such hardware in the field. Although the primary goal discussed here is to detect VHIFs, having access to such signals could bring substantial benefits to the field of grid diagnostics.

These benefits include not only the chance to implement the signals as part of a protection system but also to acquire data to reveal a more comprehensive state of the network and disturbances patterns. Current tools and trends, such as the use of unsupervised learning to reveal patterns and their consequence classification with supervised learning, could reveal massive insights and produce precise protection performances. Still, further reasonable points could also be cited to highlight the interest for the similar technologies. One is the fact that it is in the direction of a future, highly probable, smart grid scenario with more distributed and sophisticated sampling methods. Likewise, there is the possibility to aid other problems and disturbances diagnostics such as the lasting problem of accurate fault location and power quality estimation, which increases in complexity with the growing penetration of distributed generation.

5.4 Related discussions

This brief section is dedicated to discussing hypotheses formalized on the aspects of the meta-discussion in the HIF field. A suitable start is the somewhat surprising existence of the lack of consensus on how to address HIFs. It is remarkable, especially due to the relative mature awareness of the problem, which dates for decades. Part of this effect is clearly due to the problem complexity, which is arguably more complex than researchers initially thought. However, one can also argue that because it is not an immediate and asset-damaging problem, there are also not enough incentives to animate more elaborate solutions. There is a challenging aspect in creating such incentives since producing and staging the necessary experiments is an onerous and expensive activity. It is also the reasons why the Victorian Government should be praised for funding the massive vegetation conduction experiment that resulted in the data set used here. Nevertheless, the incentives that motivated such funding were a long history of dealing with bushfires, plus the devastating effects of Black Saturday. Not surprisingly, the Australian company developing early fault detection technology it is also starting to find markets in California [157], a place also marked by recurring devastating wildfires [158].

When other organizations end up doing real experiments, the fact that they are so burdensome results in negative incentives to sharing the experimental data. If the HIF phenomenon is indeed as high variant as concluded in Chapter 2, one should expect that conclusive solutions for all types of conduction surfaces to only come from large data sets with a massive number of experiments. The lack of standard data sets results in convoluted literature where many solutions are presented for different sub-problems inside the HIF detection field. This confusion leads to the incapability of comparing different methods performances and leave much of the proposed knowledge without any specific

use or application. Therefore, the author would like to propose a call to action for future researchers to include the vegetation ignition data set in their method validation. Another goes to the companies to make their data publicly available so researchers can have proper assessments of their methods and so others can build an open-source data sets of HIFs. The author believes that this will be the fastest way to solve the HIF detection problem and probably save the community from the massive amount of damages resulting from powerline-ignited fires.

Chapter 6

Further research and Conclusions

Unfortunately, a part of the ideas for experiments, analysis, and work on the data set was not performed due to time constraints. As the author believes that they could generate potential insights for the HIF phenomenon understanding and an even more elaborate classifier, the main ideas for continuing the research presented here will be put forth. Ideas based on the constraints of the work are also going to be discussed as they can serve as further validation of the proposed approach. The main implications and conclusions can be find in the end of the chapter in section 6.2.

6.1 Ideas for further research

6.1.1 Modelling

One of the most promising potential research developments from the adopted data set is using it to create realistic VHIF models. As previously explained, modelling works are important because they can be used in simulations to generate synthetic data and to consequently base and validate detection methods. Similar work has been presented in a recent paper [19] but it was not conceptualized with the same data set or intended methodology.

One potentially promising methodology suggested here comprise the use of recently proposed machine learning-based generative models for HIF modelling. Although classic machine learning algorithms were extensively explored in HIF detection as supervised learning tasks, deep learning and generative models are still relatively recent in the field. Generative Adversarial Network (GAN), for example, is a class of machine learning able to generate data with similar characteristics as the ones present in a data set. The model is composed of two neural networks, which have competing objective functions while playing distinct roles: one generates data candidates, the other evaluates them as real or

synthetic. The objective function of the network that generates new data is to increase the error rate of the discriminative network. An optimization process is applied to both networks with the goal of making the generative network to produce better images to fool the discriminator, while it gets better at discriminating real from synthetic data. GANs have proved to be drastically flexible, being applied to many different fields: from art [159] to simulations in dark energy research [160]. They have been heavily used in the field of image processing but could be used for other signals as well. In general, the idea is not limited to GANs but any other generative model that could represent the latent space in the fault signals to create new synthetic data.

After model selection, a data set composed of all the fault currents recorded in the tests would be assembled and used for training. Although being a black-box model of fault currents, i. e. causality is not explicit, it could then be used to generate similar fault currents to be incorporated in simulations. They could be simulated at different locations, in different configurations, and different types of grounding. Studies on the feasibility and method limitations could then be trivially done with Monte Carlo-like simulations. It would be portable; anyone that wanted to test their method against VHIFs would only need the black-box parameters. Moreover, the model could then be studied to understand the latent space and distributions of features of a VHIF current, generating insights about the phenomenon behaviour.

A quick mention regarding the modelling of the network at higher frequencies is also appropriate. There are few works that target more comprehensive modelling of the network frequency response at higher frequencies. More realistic and detailed models could be used to rate the suitability of networks with different characteristics for methods relying on specifics bandwidths. It could also play a significant role in predicting the resonating frequencies that partial discharges — symptoms of early fault detection and useful for predictive maintenance — would create so they could be better monitored.

6.1.2 Feasibility Prototype

Building and testing a full-stack prototype, from signal acquisition to communication, would be a validation experiment that could bring much confidence to the proposed approach. As previously explained, the prototype presented here represented the processing part of the implementation; it receives the signals from external devices and classifies signals in real-time. However, one could build a whole application prototype with the signal acquisition devices (capacitive voltage divider or sensor), high-frequency sampling (filters and ADCs), and a communication framework to signal a fault detection. Initial experiments made with high-speed ADCs showed that even the Beaglebone microcontroller would be able to handle sampling rates up to 1 MHz. Doing so would require some expertise

because the only way to run an ADC at 1 MHz sampling rate with the Beaglebone is via its peripheral processors called programmable real-time units (PRUs). The author was able to access and code these processors to run the ADC via a parallel port; they only have Assembly compilers, so the code needs to be written at a low-level language. This much laborious process could nevertheless be replaced by adding a low-cost FPGA between the microcontroller and ADC. Still, that would also be at the cost of the expertise in hardware-level coding to program the FPGA.

6.1.3 Signal attenuation experiments

Another idea for further research based on potential constraints of the current methodology is assessing signal attenuation/propagation characteristics on power distribution networks. Such systems are particularly complex due to the number of equipment, the fact that it can be severely branched with many lateral feeders, presence of non-linear loads, and distributed generation. Works proposing methods based on travelling-wave theory often have to make coarse considerations about the system characteristics resulting is less realistic models. Factors such as ground resistivity, knowledge of the topology of the line, ending and branching of feeders, and even the consideration of loads can be arbitrarily assumed, greatly influencing results. One of the problems with signal attenuation/propagation studies, therefore, is creating a framework where the insights from testing one system could be generalized to another.

There is the further complication that making such tests in a real functioning feeder is expensive and onerous. However, the author would like to leave a suggestion for future researchers conducting HIF tests to make the measurements in different parts of the system, rather than just at the fault point or substation. In this manner, one can hypothesize theories about the signal propagation and reach, leading to more confidence in the systems based on the resulted experimental data, as well as a better understanding of their limitations. Likewise, to make an attempt at trying to use higher sampling rates so the phenomenon could also be characterized at different bandwidths. One can argue that it is very probable that different disturbances will have particular signatures at higher frequencies that could be used to conceptualize better protection systems. The modern tools of clustering information and identifying patterns can allow the discovery of compelling insights that were formerly not possible with just human expertise.

The issue of signal propagation is also linked to the one formerly discussed of network modelling. One can confidently state that how the signal is going to be propagated has a substantial influence on the parameters of the system. Therefore, these problems are somewhat unable to be separately addressed and will require considerable efforts until better standards and guidelines are achieved.

6.2 Conclusions

Although the primary goal of this thesis is to propose a VHIF detection method, the process of achieving it also resulted in notable, related insights. Some came from analysing trends and distributions in the literature, while others came from hypothesis testing and evidence gathering for the suitability of the proposed approach. Part of the insights is not counted as original contribution as they came from contextualizing the information in published documents. However, findings from original experiments described here were published in peer-review journals [55–57] as a contribution to knowledge. Both are important for a holistic discussion of the consequent implications of the work produced in this thesis.

Contextualizing the documents in the related literature resulted in the confirmation of a few HIF characteristics that are widely accepted, but also revealed the lack of consensus and proved practical solutions. The most significant evidence for the latter was the $750 million dollars invested in the creation of the Powerline Bushfire Safety Program. The investment was recommended by the 2009 Victorian Bushfires Royal Commission when they came to the conclusion that effective commercial solutions to mitigate the powerline-ignited fires were lacking. The survey of formal peer-review journals corroborates with this conclusion in most senses. Up to this date, researchers are still proposing novel and elaborated HIF detection approaches while finding knowledge gaps is previous ones. Despite key commercial players such as GE, ABB, and Sweitzer had also proposed their solutions, it is hard to prove that a definite solution is nowhere near. The vegetation ignition tests presented one further evidence of the unsuitability of the commercial solutions. A relay with HIF detection function was present during all the tests did not detect any of the staged faults.

Concerning the relatively elevated number of papers published in this field in the last decade, they might only exist due to the progress made in HIF modelling. Researchers that knowledgeably tried to discover the patterns in their data sets and embed them in models paved the way for many subsequent simulation-based works. Despite their numbers, most of the modelling-based methodologies still leave many questions open regarding their capability the generalize to real scenarios. Moreover, another notable conclusion from investigating the literature is that hardly any works specifically target vegetation as HIF surface. Although the need for doing so is not apparent, insights on the considerable variance of HIFs, the fact that commercial solutions can not guarantee their detection, and their capability to ignite fires make such specification much more relevant. Therefore, focusing on local vegetation species tested in real, staged faults considerably differentiates this work from most in the literature.

The amount of data resulted from the vegetation ignition tests allowed the production of a machine learning-based classifier. This approach was advantageous because machine

learning avoids the need for a bottom-up method, which might be infeasible given the complexity of the problem. The sizeable data set also allowed for proper validation as some of the observations could be partitioned as out-of-sample testing data. The problem of classifying the observations as faulted or not constituted a supervised learning task, which means that the algorithm was trained on labelled data with respective classes. In the last version of the classifier, such classes were 'Voltage-OFF', 'Non-fault', and 'Fault'. The first class was composed of HF sweeps extracted from moments where the HV source was not connected; hence it only includes the background noise from the test rig. Observations from the 'Non-fault' class were sweeps from recordings made throughout the test days, with the connection of the HV source, as a way to characterise the network signals. The observations from the latter class, 'Fault', came from a more arbitrary extraction method; they were the sweeps immediately sampled after the current RMS reached 0.5 A. After their assembling in a labelled data set, the observations were used to learn many machine learning algorithms such as KNN, SVM, and decision trees. Ensembles of decision trees, boosting and bagging, over-performed all tested algorithms in most of the tests. The results from learning the classifier were favourable: 98.8% accuracy, 97% dependability, and 99.09% security.

The positives results from learning the classifier are significant to VHIF detection and fire mitigation when their relevancy is considered. Discriminating between the 'Non-fault' and 'Fault' classes is very challenging, as seen in illustrations and results from Chapter 4. In fact, when only the LF observations are considered, the classifier is not much better than a coin toss at predicting faults. Although accuracy is a relevant parameter, it is not the most significant part of the main results; one can find works that claim higher accuracy values than this. It is instead the fact that such promising results were achieved with real data (not simulations) from a specific type of HIF having vegetation as conducting surface. This main contributions here is particularly relevant to the Australian context, which has people and property seasonally damaged by powerline-ignited bushfires. From the classifier results, it is also notable that the security value (non-false positives) was the highest one. The problem of detecting HIF is tightly linked with service continuity commitment, especially to critical and industrial loads. Tripping the power off the line in case of false-positive results could be disastrous if exacerbated. Therefore, strategies to further increase the already high-certainty security were also proposed. The most trivial one is the implementation of a counter before the trip signal where consequent detections are needed before disconnecting the power. The less trivial and original idea was proposed as a feedback method where the previous state of the network is saved in the form of a feature vector. In this approach, which is adopted in the last version of the method, the newly sampled signal features are compared to the feedback feature vector and then fed to the classifier. It is expected that the positive classification results and implementation strategies

result in generalization on the field, but it is important to be honest and not to exaggerate the results. One does not need to claim that such a method will effectively detect all types of faults and conducting surfaces to be relevant. However, it is possible to speculate that there is a potential use of this approach to detect other similar disturbances given such positives results and the nature of the phenomena. In any case, considering the favourable results and the cost of alternatives to help mitigate powerline-ignited fire risk, the solution proposed here arguably presents a tremendous potential for further investments.

As the supervised learning task is solved through a symbolic approach, the choice of features and signal processing techniques played part of the role in the method success. Initial investigations started with power spectral estimators and spectrograms, which are all Fourier-based techniques that can also be used as signal-representative features. However, signal representation methods will perform differently depending on signal characteristics such as if they are stationary or not, or if discontinuities are relevant information. As expected, Fourier-based predictors underperformed when compared to wavelet ones. This fact is somewhat in accordance with the knowledge in the literature as wavelets became the prevalent technique once they were introduced to the field. As they are well localized in time and logarithmically split the frequency domain, wavelets are more efficient at representing signals discontinuities and anomalies. Towards the conceptualization of the final version, wavelet-based features were combined with features from the basis functions learned from the data set. Such originally developed features serve as matching filters to recurrent patterns found in the faulted signals.

Results from the secondary work and evidence gathering experiments imply some valuable VHIFs insights and how they relate to common knowledge in the field. The probably most relevant one is the use of low-frequency harmonic content for VHIF detection. As mentioned in Chapter 2, such measurements were used since the beginning of the field. The third-order harmonic, in specific, was often presented as a reliable HIF predictor due to the non-linearity of the fault current around zero-crossings. However, as shown in initial investigations and rarely mentioned in the literature, the HIF fault surface behaves almost linearly during initial moments of vegetation conduction. This phenomenon should receive increased importance since fire ignition risk can only be avoided in the initial moments of a fault. Further evidence came from experiments set to compare the information content from LF and HF signals. It revealed that only the latter could be used to predict fault observations reliably, and that although valid for the single-digit vegetation fault current studied here, it means that HF sampling may be imperative for their detection. In regards to the nature of phenomena, the application of unsupervised learning helped to illustrate how the fault signatures are formed. As another novel application in this thesis, the patterns were deconvolved from other signals, isolated, and illustrated in the time domain to further result in insights about faulted signals. They corroborated with the descriptions of HIFs

having an impulsive nature at higher frequencies that creates signal discontinuities on the feeder current. This impulses in the current, in turn, generate damped oscillatory responses in the voltage signals due to the second-order nature of the system.

Part of the evidence-gathering work also helped to attest some of the practical feasibility of the method. The proof-of-concept prototype can be presented as evidence that the method can be implemented in low-cost computers to work deterministically in real-time. The concern for computational complexity when working with HF signals is completely justified. The only reason why the card-sized computer was able to perform the classification in real-time is that it was processing sweeps of 20 ms, instead of continuously sampled data. In this sense, classifying signals via HF sweeps is an entirely novel approach that could alleviate a substantial part of the computational requirements. Given the favourable results presented here, one could argue that the sweep approach is an unexplored area that could be useful for this and other similar problems.

In summary, the results obtained in this thesis attest that investigating the vegetation ignition data set to create a VHIF detection method was a credible idea. The results suggest that fault signals can be accurately discriminated and that the approach is practically feasible. The classification had the favourable effect of working with both phase-to-earth and phase-to-phase faults. They also show that there are fault signatures in the HF signals, and for detecting VHIFs, such high-resolution sampling is possibly a requirement. The methodology and consequent results also have the additional validation aspect of being accepted in high-ranking peer-reviewed journals. Considering the gravity of the problem and the specificity of the results regarding the Australian scenario, one can argue that this solution has a high-potential value for the community. This argument is exceptionally easy to make when considering that Australia has tens of thousands of kilometres of SWER lines that are unsuitable for even the most costly and promising three-phase solution. If anything, the body of work in this thesis attest the value in continuing the research direction and production of a complete prototype for further testing.

References

[1] N. D. Tleis, "1 - Introduction to power system faults," in *Power Systems Modelling and Fault Analysis*, ser. Newnes Power Engineering Series, N. D. Tleis, Ed. Oxford: Newnes, 2008, pp. 1 – 27.

[2] M. Sedighizadeh, A. Rezazadeh, and N. I. Elkalashy, "Approaches in high impedance fault detection a chronological review," *Advances in Electrical and Computer Engineering*, vol. 10, no. 3, pp. 114–128, 2010.

[3] A. C. Depew, M. G. Adamiak, B. D. Russell, C. L. Benner, R. W. Dempsey, and J. M. Parsick, "Field experience with high-impedance fault detection relays," in *2005/2006 IEEE/PES Transmission and Distribution Conference and Exhibition*, 2006, Conference Proceedings, pp. 868–873.

[4] T. Marxsen, "Vegetation conduction ignition test report - final," Marxsen Consulting Pty Ltd., Department of Economic Development Jobs Transport and Resources, 2015.

[5] 2009 Victorian Bushfires Royal Commission. (2010, Jul.) Final report. State Government of Victoria. [Online]. Available: http://www.royalcommission.vic.gov.au/Commission-Reports/Final-Report/Summary/Interactive-Version.html

[6] C. Miller, M. Plucinski, A. Sullivan, A. Stephenson, C. Huston, K. Charman, M. Prakash, and S. Dunstall, "Electrically caused wildfires in victoria, australia are over-represented when fire danger is elevated," *Landscape and urban planning*, vol. 167, pp. 267–274, 2017.

[7] B. M. Aucoin and B. D. Russell, "Distribution high impedance fault detection utilizing high frequency current components," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 6, pp. 1596–1606, June 1982.

[8] M. Aucoin and B. D. Russell, "Detection of distribution high impedance faults using burst noise signals near 60 hz," *IEEE transactions on power delivery*, vol. 2, no. 2, pp. 342–348, 1987.

[9] J. Carr, "Detection of high impedance faults on multi-grounded primary distribution systems," *IEEE Transactions on Power Apparatus and Systems*, no. 4, pp. 2008–2016, 1981.

[10] M. Aucoin, "Status of high impedance fault-detection," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, no. 3, pp. 638–644, 1985.

[11] A. Ghaderi, H. L. Ginn Iii, and H. A. Mohammadpour, "High impedance fault detection: A review," *Electric Power Systems Research*, vol. 143, pp. 376–388, 2017.

[12] M. Mishra and R. R. Panigrahi, "Taxonomy of high impedance fault detection algorithm," *Measurement*, vol. 148, p. 106955, 2019.

[13] D. C. T. Wai and X. Yibin, "A novel technique for high impedance fault identification," *IEEE Transactions on Power Delivery*, vol. 13, no. 3, pp. 738–744, 1998.

[14] D. Hou, "High-impedance fault detection—field tests and dependability analysis," in *Proceedings of the 36th Annual Western Protective Relay Conference, Spokane, WA*, 2009.

[15] M. Aucoin, B. D. Russell, and C. L. Benner, "High impedance fault detection for industrial power systems," in *Conference Record of the IEEE Industry Applications Society Annual Meeting,*, Oct 1989, pp. 1788–1792 vol.2.

[16] S. Nam, J. Park, Y. Kang, and T. Kim, "A modeling method of a high impedance fault in a distribution system using two series time-varying resistances in EMTP," in *Power Engineering Society Summer Meeting, 2001*, vol. 2. IEEE, 2001, Conference Proceedings, pp. 1175–1180.

[17] P. E. Farias, A. P. de Morais, J. P. Rossini, and G. Cardoso, "Non-linear high impedance fault distance estimation in power distribution systems: A continually online-trained neural network approach," *Electric Power Systems Research*, vol. 157, pp. 20–28, 2018.

[18] W. C. Santos, F. V. Lopes, N. S. D. Brito, and B. A. Souza, "High-impedance fault identification on distribution networks," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 23–32, 2017.

[19] N. Bahador, F. Namdari, and H. R. Matinfar, "Modelling and detection of live tree-related high impedance fault in distribution systems," *IET Generation, Transmission & Distribution*, vol. 12, no. 3, pp. 756–766, 2017.

[20] J. C. Chen, B. T. Phung, D. M. Zhang, T. Blackburn, and E. Ambikairajah, "Study on high impedance fault arcing current characteristics," in *2013 Australasian Universities Power Engineering Conference (AUPEC)*, 2013, Conference Proceedings, pp. 1–6.

[21] D. Hou and N. Fischer, "Deterministic high-impedance fault detection and phase selection on ungrounded distribution systems," in *2006 Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources*, 2006, Conference Proceedings, pp. 112–122.

[22] M. G. Adamiak, B. D. Russell, C. L. Benner, R. W. Dempsey, J. M. Parsick, A. C. Depew, M. G. Adamiak, B. D. Russell, C. L. Benner, R. W. Dempsey, J. M. Parsick, and A. C. Depew, "Field experience with high-impedance fault detection relays," in *2005/2006 IEEE/PES Transmission and Distribution Conference and Exhibition*, May 2006, pp. 868–873.

[23] A. R. Sedighi, M. R. Haghifam, O. P. Malik, and M. H. Ghassemian, "High impedance fault detection based on wavelet transform and statistical pattern recognition," *IEEE Transactions on Power Delivery*, vol. 20, no. 4, pp. 2414–2421, 2005.

[24] C. J. Kim, B. D. Russell, and K. Watson, "A parameter-based process for selecting high impedance fault detection techniques using decision making under incomplete knowledge," *IEEE Transactions on Power Delivery*, vol. 5, no. 3, pp. 1314–1320, July 1990.

[25] B. D. Russell and R. P. Chinchali, "A digital signal processing algorithm for detecting arcing faults on power distribution feeders," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 132–140, Jan 1989.

[26] W. C. dos Santos, B. A. de Souza, N. S. D. Brito, F. B. Costa, and M. R. C. Paes, "High impedance faults: From field tests to modeling," *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 6, pp. 885–896, 2013.

[27] H. S. Jain, S. Devabhaktuni, and T. Sairama, "In-depth analysis of charge leakage through vegetation in transmission and distribution lines," in *Applications of Artificial Intelligence Techniques in Engineering*. Springer Singapore, 2019, Conference Proceedings, pp. 21–29.

[28] N. Bahador, F. Namdari, and H. R. Matinfar, "Modelling and detection of live tree-related high impedance fault in distribution systems," *IET Generation, Transmission & Distribution*, vol. 12, no. 3, pp. 756–766, 2018.

[29] N. Bahador, F. Namdari, and H. R. Matinfar, "Tree-related high impedance fault location using phase shift measurement of high frequency magnetic field," *International Journal of Electrical Power & Energy Systems*, vol. 100, pp. 531–539, 2018.

[30] J. A. Wischkaemper, C. L. Benner, and B. D. Russell, "Electrical characterization of vegetation contacts with distribution conductors - investigation of progressive fault behavior," in *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, 2008, Conference Proceedings, pp. 1–8.

[31] D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "High-frequency spectral analysis of high impedance vegetation faults on a three-wire system," in *2017 Australasian Universities Power Engineering Conference (AUPEC)*, 2017, Conference Proceedings, pp. 1–5.

[32] T. Marxsen, "New technology to cut victoria's powerline fire risk," in *Arboriculture Australia National Conference*, 2016.

[33] J. W. Mitchell, "Power line failures and catastrophic wildfires under extreme weather conditions," *Engineering Failure Analysis*, vol. 35, pp. 726 – 735, 2013, special issue on ICEFA V- Part 1.

[34] A. G. McArthur, N. P. Cheney, and J. Barber, *The fires of 12 February 1977 in the Western District of Victoria*. CSIRO, 1982.

[35] V. C. F. Authority, *The major fires originating 16th February, 1983*. CFA, 1983.

[36] 2009 Victorian Bushfires Royal Commission. (2010, Jul.) Commission reports. State Government of Victoria. [Online]. Available: http://www.royalcommission.vic.gov.au/Commission-Reports/Final-Report.html

[37] T. Marxsen, "REFCL Technologies Test Program – Final Report," Marxsen Consulting Pty Ltd., Report, 2015.

[38] A. Emanuel, D. Cyganski, J. Orr, S. Shiller, and E. Gulachenski, "High impedance fault arcing on sandy soil in 15 kv distribution feeders: contributions to the evaluation of the low frequency spectrum," *IEEE Transactions on Power Delivery*, vol. 5, no. 2, pp. 676–686, 1990.

[39] BBC News, "Australia fires: A visual guide to the bushfire crisis," 2020. [Online]. Available: https://www.bbc.com/news/world-australia-50951043

[40] M. A. Moritz, M.-A. Parisien, E. Batllori, M. A. Krawchuk, J. Van Dorn, D. J. Ganz, and K. Hayhoe, "Climate change and disruptions to global fire activity," *Ecosphere*, vol. 3, no. 6, pp. 1–22, 2012.

[41] T. Marxsen, "REFCL Technologies Test Program – Final Report," Marxsen Consulting Pty Ltd., Report, 2015.

[42] T. Marxsen, "New technology to cut victoria's powerline fire risk," in *Arboriculture Australia National Conference.*, 2016, Conference Proceedings.

[43] K. M. Winter, "The RCC Ground Fault Neutralizer—A novel scheme for fast earth-fault protection," in *CIRED 18th International Conference and Exhibition on Electricity Distribution*. IET, 2005, Conference Proceedings, pp. 1–4.

[44] Victoria State Government. (2018) Research and development. Victoria State Government. [Online]. Available: https://www.energy.vic.gov.au/__data/assets/pdf_file/0017/41624/Jan-18-R-and-D-fact-sheet-pdf.pdf

[45] The Nous Group. (2010) National workshop on rural electricity network options to reduce bushfire risk. Government of Victoria. [Online]. Available: https://www.energy.vic.gov.au/__data/assets/pdf_file/0020/41663/SWER-Workshop-21-April-2010-Final-Report-June-2010.pdf

[46] Victoria State Government. (2016) Electricity safety (bushfire mitigation) amendment regulations 2016. Department of Environment, Land, Water and Planning. [Online]. Available: https://www.energy.vic.gov.au/safety-and-emergencies/powerline-bushfire-safety-program/electrical-safety-bushfire-mitigation-further-amendment-regulations-2016

[47] Victoria State Government, "Electricity safety amendment (bushfire mitigation civil penalties scheme)," No. 18 of 2017, The Parliament of Victoria, 2017.

[48] Powercor, "Contingent project application REFCL program: tranche two," Powercor, Report, 2018.

[49] Department of Environment, Land, Water & Planning, "Powerline bushfire safety program - vegetation conduction ignition test report and data," 2016. [Online]. Available: https://discover.data.vic.gov.au/dataset/powerline-bushfire-safety-program-vegetation-conduction-ignition-test-report

[50] General Electric Grid Solutions. F60 guideform specifications. [Online document; accessed 2-July-2018]. [Online]. Available: https://www.gegridsolutions.com/products/specs/f60/f60_7_60%20specs.doc

[51] ABB. Feeder protection and control REF620 ANSI. [Online document; accessed 18-July-2018]. [Online]. Available: https://library.e.abb.com/public/1493c1549ae145e0b3d9c32c5b3d9552/REF620_appl_757651_ENd.pdf

[52] B. M. Aucoin and R. H. Jones, "High impedance fault detection implementation issues," *IEEE Transactions on Power Delivery*, vol. 11, no. 1, pp. 139–148, 1996.

[53] A. Ghaderi, H. A. Mohammadpour, and H. Ginn, "High impedance fault detection method efficiency: simulation vs. real-world data acquisition," in *Power and Energy Conference at Illinois (PECI), 2015 IEEE*. IEEE, 2015, Conference Proceedings, pp. 1–5.

[54] Victoria State Government. (2017) Vegetation detection challenge. [Online]. Available: https://www.energy.vic.gov.au/safety-and-emergencies/powerline-bushfire-safety-program/research-and-development/vegetation-detection-challenge

[55] D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "High-sensitivity vegetation high-impedance fault detection based on signal's high-frequency contents," *IEEE Transactions on Power Delivery*, vol. 33, no. 3, pp. 1398–1407, 2018.

[56] D. P. S. Gomes, C. Ozansoy, A. Ulhaq, and J. C. de Melo Vieira Júnior, "The effectiveness of different sampling rates in vegetation high-impedance fault classification," *Electric Power Systems Research*, vol. 174, p. 105872, 2019.

[57] D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "Vegetation high-impedance faults' high-frequency signatures via sparse coding," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2019.

[58] B. M. Aucoin and R. P. Heller, "Overcurrent and high impedance fault relaying using a microcomputer," in *Proceedings of the 7th Texas Conference on Computing Systems*, 1978, pp. 2–5.

[59] G. Hashmi, M. Isa, and M. Lehtonen, "Modeling on-line three-phase PD monitoring system for MV overhead covered-conductors," in *International Conference on Power Systems Transients (IPST2009)*, 2009.

[60] K. J. Khor and K. L. Wong, "Partial discharge sensing in overhead distribution line," in *2008 Australasian Universities Power Engineering Conference*, Dec 2008, pp. 1–5.

[61] K. Pandakov, H. K. Høidalen, and J. I. Marvik, "Misoperation analysis of steady-state and transient methods on earth fault locating in compensated distribution networks," *Sustainable Energy, Grids and Networks*, vol. 15, pp. 34–42, 2018.

[62] R. Lee and M. Bishop, "A comparison of measured high impedance fault data to digital computer modeling results," *IEEE Transactions on Power Apparatus and systems*, no. 10, pp. 2754–2758, 1985.

[63] R. E. Lee and R. H. Osborn, "A microcomputer based data acquisition system for high impedance fault analysis," *IEEE transactions on power apparatus and systems*, no. 10, pp. 2748–2753, 1985.

[64] J. Carr and G. Hood, "High impedance fault detection on primary distribution systems," *CEA Final Report, Project*, no. 78-75, 1979.

[65] C.-L. Huang, H.-Y. Chu, and M.-T. Chen, "Algorithm comparison for high impedance fault detection based on staged fault test," *IEEE transactions on power delivery*, vol. 3, no. 4, pp. 1427–1435, 1988.

[66] S. H. Mortazavi, Z. Moravej, and S. M. Shahrtash, "A hybrid method for arcing faults detection in large distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 94, pp. 141–150, 2018.

[67] J. R. Dunki-Jacobs, "The effects of arcing ground faults on low-voltage system design," *IEEE Transactions on Industry Applications*, no. 3, pp. 223–230, 1972.

[68] B. D. Russell and C. L. Benner, "Arcing fault detection for distribution feeders: security assessment in long term field trials," *IEEE Transactions on power delivery*, vol. 10, no. 2, pp. 676–683, 1995.

[69] C. T. Leondes, *Expert systems: the technology of knowledge management and decision making for the 21st century*. Elsevier, 2001.

[70] W. H. Kwon, G. W. Lee, Y. M. Park, M. C. Yoon, and M. H. Yoo, "High impedance fault detection utilizing incremental variance of normalized even order harmonic power," *IEEE Transactions on Power Delivery*, vol. 6, no. 2, pp. 557–564, 1991.

[71] A. F. Sultan, G. W. Swift, and D. J. Fedirchuk, "Detection of high impedance arcing faults using a multi-layer perceptron," *IEEE Transactions on Power Delivery*, vol. 7, no. 4, pp. 1871–1877, 1992.

[72] F. Jota and P. Jota, "High-impedance fault identification using a fuzzy reasoning system," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 145, no. 6, pp. 656–662, 1998.

[73] R. Patterson, W. Tyska, B. D. Russell, and B. M. Aucoin, "A microprocessor-based digital feeder monitor with high-impedance fault detection," in *47th Annual Conference for Protective Relay Engineers Texas A&M University, College Station, Texas, USA*, 1994.

[74] A. Mamishev, B. D. Russell, and C. L. Benner, "Analysis of high impedance faults using fractal techniques," in *Proceedings of Power Industry Computer Applications Conference*. IEEE, 1995, pp. 401–406.

[75] C. L. Benner and B. D. Russell, "Practical high-impedance fault detection on distribution feeders," *IEEE Transactions on Industry Applications*, vol. 33, no. 3, pp. 635–640, 1997.

[76] General Electric. High impedance fault detection on the multilin f60 feeder protection system. Accessed: June, 2019. [Online]. Available: https://www.gegridsolutions.com/multilin/products/hiz/index.htm

[77] S. Ebron, D. L. Lubkeman, and M. White, "A neural network approach to the detection of incipient faults on power distribution feeders," *IEEE Transactions on Power Delivery*, vol. 5, no. 2, pp. 905–914, 1990.

[78] A. A. Petrosian and F. G. Meyer, *Wavelets in signal and image analysis: from theory to practice*. Springer Science & Business Media, 2013, vol. 19.

[79] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[80] Wolfram Mathematica. Wavelet analysis. Accessed: June, 2019. [Online]. Available: https://www.wolfram.com/mathematica/new-in-8/wavelet-analysis/

[81] S.-J. Huang and C.-T. Hsieh, "High-impedance fault detection utilizing a morlet wavelet transform approach," *IEEE Transactions on Power Delivery*, vol. 14, no. 4, pp. 1401–1410, 1999.

[82] A. A. Girgis, W. Chang, and E. B. Makram, "Analysis of high-impedance fault generated signals using a kalman filtering approach," *IEEE Transactions on Power delivery*, vol. 5, no. 4, pp. 1714–1724, 1990.

[83] D. Jeerings and J. Linders, "Unique aspects of distribution system harmonics due to high impedance ground faults," *IEEE Transactions on Power delivery*, vol. 5, no. 2, pp. 1086–1094, 1990.

[84] K.-Y. Lien, S.-L. Chen, C.-J. Liao, T.-Y. Guo, T.-M. Lin, and J.-S. Shen, "Energy variance criterion and threshold tuning scheme for high impedance fault detection," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 810–817, 1999.

[85] M. Sarlak and S. M. Shahrtash, "High impedance fault detection using combination of multi-layer perceptron neural networks based on multi-resolution morphological gradient features of current waveform," *IET Generation, Transmission & Distribution*, vol. 5, no. 5, pp. 588–595, 2011.

[86] T. Baldwin and F. Renovich, "Analysis of fault locating signals for high-impedance grounded systems," *Ieee Transactions on Industry Applications*, vol. 38, no. 3, pp. 810–817, 2002.

[87] H. Daqing, "Detection of high-impedance faults in power distribution systems," in *2007 Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources*, 2007, Conference Proceedings, pp. 85–95.

[88] J. Chen, E. Ambikairajah, D. Zhang, T. Phung, and T. Blackburn, "Detection of high impedance faults using current transformers for sensing and identification based on features extracted using wavelet transform," *IET Generation, Transmission & Distribution*, vol. 10, no. 12, pp. 2990–2998, 2016.

[89] C. G. Wester, "High impedance fault detection on distribution systems," in *1998 Rural Electric Power Conference Presented at 42nd Annual Conference*. IEEE, 1998, pp. c5–1.

[90] A. Lazkano, J. Ruiz, E. Aramendi, and L. A. Leturiondo, "A new approach to high impedance fault detection using wavelet packet analysis," in *Ninth International Conference on Harmonics and Quality of Power. Proceedings (Cat. No.00EX441)*, vol. 3, 2000, Conference Proceedings, pp. 1005–1010 vol.3.

[91] T. M. Lai, L. A. Snider, E. Lo, and D. Sutanto, "High-impedance fault detection using discrete wavelet transform and frequency range and RMS conversion," *IEEE Transactions on Power Delivery*, vol. 20, no. 1, pp. 397–407, 2005.

[92] M. Michalik, W. Rebizant, M. Lukowicz, L. Seung-Jae, and K. Sang-Hee, "Wavelet transform approach to high impedance fault detection in mv networks," in *2005 IEEE Russia Power Tech*, 2005, Conference Proceedings, pp. 1–7.

[93] M. Michalik, W. Rebizant, M. Lukowicz, L. Seung-Jae, and K. Sang-Hee, "High-impedance fault detection in distribution networks with use of wavelet-based algorithm," *IEEE Transactions on Power Delivery*, vol. 21, no. 4, pp. 1793–1802, 2006.

[94] A. R. Sedighi, M. R. Haghifam, and O. P. Malik, "Soft computing applications in high impedance fault detection in distribution systems," *Electric Power Systems Research*, vol. 76, no. 1, pp. 136–144, 2005.

[95] A. R. Sedighi, M. R. Haghifam, O. P. Malik, and M. H. Ghassemian, "High impedance fault detection based on wavelet transform and statistical pattern recognition," *IEEE Transactions on Power Delivery*, vol. 20, no. 4, pp. 2414–2421, 2005.

[96] A. H. Etemadi and M. Sanaye-Pasand, "High-impedance fault detection using multi-resolution signal decomposition and adaptive neural fuzzy inference system," *Iet Generation Transmission & Distribution*, vol. 2, no. 1, pp. 110–118, 2008.

[97] M. Sarlak and S. M. Shahrtash, "High impedance fault detection in distribution networks using support vector machines based on wavelet transform," in *2008 IEEE Canada Electric Power Conference*, 2008, Conference Proceedings, pp. 1–6.

[98] A. M. Sharaf and W. Guosheng, "High impedance fault detection using feature-pattern based relaying," in *2003 IEEE PES Transmission and Distribution Conference and Exposition (IEEE Cat. No.03CH37495)*, vol. 1, 2003, Conference Proceedings, pp. 222–226 Vol.1.

[99] Y. Sheng and S. M. Rovnyak, "Decision tree-based methodology for high impedance fault detection," *IEEE Transactions on Power Delivery*, vol. 19, no. 2, pp. 533–536, 2004.

[100] S. R. Samantaray, P. K. Dash, and S. K. Upadhyay, "Adaptive kalman filter and neural network based high impedance fault detection in power distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 4, pp. 167–172, 2009.

[101] S. R. Samantaray, L. N. Tripathy, and P. K. Dash, "Combined EKF and SVM based High Impedance Fault detection in power distribution feeders," in *2009 International Conference on Power Systems*, 2009, Conference Proceedings, pp. 1–6.

[102] C. Tao, D. Xinzhou, B. Zhiqian, A. Klimek, and A. Edwards, "Modeling study for high impedance fault detection in mv distribution system," in *2008 43rd International Universities Power Engineering Conference*, 2008, Conference Proceedings, pp. 1–5.

[103] N. I. Elkalashy, M. Lehtonen, H. A. Darwish, M. A. Izzularab, and A. I. Taalab, "Modeling and experimental verification of high impedance arcing fault in medium voltage networks," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 14, no. 2, pp. 375–383, 2007.

[104] X. Wang, J. Gao, X. Wei, Z. Zeng, Y. Wei, and M. Kheshti, "Single line to ground fault detection in an non-effectively grounded distribution network," *IEEE Transactions on Power Delivery*, pp. 1–1, 2018.

[105] J. Roberts, H. J. Altuve, and D. Hou, "Review of ground fault protection methods for grounded ungrounded and compensated distribution systems," *USA, SEL*, pp. 1–40, 2001.

[106] N. I. Elkalashy, M. Lehtonen, H. A. Darwish, A. M. I. Taalab, and M. A. Izzularab, "DWT-Based detection and transient power direction-based location of high-impedance faults due to leaning trees in unearthed MV networks," *Ieee Transactions on Power Delivery*, vol. 23, no. 1, pp. 94–101, 2008.

[107] N. I. Elkalashy, M. Lehtonen, H. A. Darwish, A. M. I. Taalab, and M. A. Izzularab, "Verification of DWT-based detection of high impedance faults in MV networks," *IET Conference Proceedings*, pp. 344–348(4), January 2008.

[108] E. C. Senger, W. Kaiser, J. C. Santos, P. M. S. Burt, and C. V. S. Malagodi, "Broken conductors protection system using carrier communication," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 525–530, 2000.

[109] G. M. Hashmi and M. Lehtonen, "On-line PD measuring system modeling and experimental verification for covered-conductor overhead distribution lines," in *2007 Mediterranean Conference on Control & Automation*, 2007, Conference Proceedings, pp. 1–6.

[110] G. M. Hashmi and M. Lehtonen, "On-line PD detection for condition monitoring of covered-conductor overhead distribution networks-A literature survey," in *2008 Second International Conference on Electrical Engineering*, 2008, Conference Proceedings, pp. 1–6.

[111] M. A. Noras, "Solid-state electric-field sensor," May 2017, uS Patent 9,664,721.

[112] K. L. Wong and A. Bojovschi, "Fault detection system," Mar 2017, uS Patent 9,606,164.

[113] J. Das, M. Stoupis, R. Maharsi, S. Nuqui, and R. Kunsman, "Ground alert: Reliable detection of high-impedance faults caused by downed conductors," *ABB review*, vol. 1, 2004.

[114] Schweitzer Engineering Laboratories, "Arc Sense Technology (AST) - High-Impedance Fault Detection," SEL Inc, Report, 2016. [Online]. Available: https://cdn.selinc.com/assets/Literature/Product%20Literature/Flyers/Arc-Sense_PF00160.pdf

[115] A. S. Bretas, M. Moreto, R. H. Salim, and L. O. Pires, "A novel high impedance fault location for distribution systems considering distributed generation," in *2006 IEEE/PES Transmission & Distribution Conference and Exposition: Latin America*, 2006, Conference Proceedings, pp. 1–6.

[116] S. R. Samantaray, B. K. Panigrahi, and P. K. Dash, "High impedance fault detection in power distribution networks using time–frequency transform and probabilistic neural network," *IET Generation, Transmission & Distribution*, vol. 2, no. 2, p. 261, 2008.

[117] A. Sedighi and M. Haghifam, "Simulation of high impedance ground fault in electrical power distribution systems," in *2010 International Conference on Power System Technology*. IEEE, 2010, pp. 1–7.

[118] N. R. Bahador, F. Namdari, and H. R. Matinfar, "Feature extraction of tree-related high impedance faults as a source of electromagnetic interference around medium voltage power lines' corridors," *Progress In Electromagnetics Research B*, vol. 75, pp. 13–26, 2017.

[119] M. Sarlak and S. M. Shahrtash, "High-impedance faulted branch identification using magnetic-field signature analysis," *IEEE Transactions on Power Delivery*, vol. 28, no. 1, pp. 67–74, 2013.

[120] V. C. de Paula and J. R. Macedo, "High-impedance fault detection in power distribution systems through the analysis of the magnetic fields in the surroundings of the conductors," in *Simposio Brasileiro de Sistemas Eletricos (SBSE)*, 2018, Conference Proceedings, pp. 1–6.

[121] J. R. Macedo, D. Carvalho, J. W. Resende, F. C. Castro, and C. A. Bissochi, "Proposition of an interharmonic-based methodology for high-impedance fault detection in distribution systems," *IET Generation, Transmission & Distribution*, vol. 9, no. 16, pp. 2593–2601, 2015.

[122] S. Gautam and Brahma, "Detection of high impedance fault in power distribution systems using mathematical morphology," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1226–1234, 2013.

[123] A. Ghaderi, H. A. Mohammadpour, H. L. Ginn, and Y. J. Shin, "High-impedance fault detection in the distribution network using the time-frequency-based algorithm," *IEEE Transactions on Power Delivery*, vol. 30, no. 3, pp. 1260–1268, 2015.

[124] O. E. Batista, R. A. Flauzino, M. A. de Araujo, L. A. de Moraes, and I. N. da Silva, "Methodology for information extraction from oscillograms and its application for high-impedance faults analysis," *International Journal of Electrical Power & Energy Systems*, vol. 76, pp. 23–34, 2016.

[125] A. N. Milioudis, G. T. Andreou, and D. P. Labridis, "Enhanced Protection Scheme for Smart Grids Using Power Line Communications Techniques; Part I: Detection of High Impedance Fault Occurrence," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1621–1630, 2012.

[126] A. N. Milioudis, G. T. Andreou, and D. P. Labridis, "Enhanced Protection Scheme for Smart Grids Using Power Line Communications Techniques—Part II: Location of High Impedance Fault Position," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1631–1640, 2012.

[127] A. N. Milioudis, G. T. Andreou, and D. P. Labridis, "Detection and location of high impedance faults in multiconductor overhead distribution lines using power line communication devices," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 894–902, 2015.

[128] L. U. Iurinic, A. R. Herrera-Orozco, R. G. Ferraz, and A. S. Bretas, "Distribution systems high-impedance fault location: A parameter estimation approach," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1806–1814, 2016.

[129] A. H. A. Bakar, M. S. Ali, C. Tan, H. Mokhlis, H. Arof, and H. A. Illias, "High impedance fault location in 11kv underground distribution systems using wavelet transforms," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 723–730, 2014.

[130] N. Bahador, H. Matinfar, and F. Namdari, "A framework for wide-area monitoring of tree-related high impedance faults in medium-voltage networks," *Journal of Electrical Engineering & Technology*, vol. 13, no. 1, pp. 1–10, 2018.

[131] *Signal Processing Toolbox. Release 2018a*, The MathWorks, Inc., Natick, Massachusetts, United States. [Online]. Available: https://www.mathworks.com/products/signal.html

[132] *Statistics and Machine Learning Toolbox. Release 2018a*, The MathWorks, Inc., Natick, Massachusetts, United States. [Online]. Available: https://www.mathworks.com/products/statistics.html

[133] *MATLAB and Wavelet Toolbox Release 2012b*, The MathWorks, Inc., Natick, Massachusetts, United States. [Online]. Available: https://www.mathworks.com/products/wavelet.html

[134] R. Grosse, "Roger grosse - pulications." [Online]. Available: http://www.cs.toronto.edu/~rgrosse/publications.html

[135] E. Keogh and A. Mueen, *Curse of Dimensionality*. Boston, MA: Springer US, 2017, pp. 314–315.

[136] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[137] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[138] F. B. Costa, B. Souza, N. Brito, J. Silva, and W. Santos, "Real-time detection of transients induced by high-impedance faults based on the boundary wavelet transform," *IEEE Transactions on Industry Applications*, vol. 51, no. 6, pp. 5312–5323, 2015.

[139] M. H. Dhend, "Fault diagnosis of smart grid distribution system by using smart sensors and symlet wavelet function," *Journal of Electronic Testing*, vol. 33, no. 3, pp. 329–338, 2017.

[140] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariant sparse coding for audio classification," in *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, ser. UAI'07. Arlington, Virginia, United States: AUAI Press, 2007, pp. 149–158.

[141] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, p. 607, 1996.

[142] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37 – 52, 1987, proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[143] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[144] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801–808.

[145] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. wadsworth & brooks," *Monterey, CA*, 1984.

[146] L. Breiman, "Random forests," *UC Berkeley TR567*, 1999.

[147] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[148] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *icml*, vol. 96, 1996, Conference Proceedings, pp. 148–156.

[149] *MATLAB Coder. Release 2018a*, The MathWorks, Inc., Natick, Massachusetts, United States. [Online]. Available: https://au.mathworks.com/products/matlab-coder.html

[150] Australian Communications and Media Authority, "Australian radiofrequency spectrum plan 2017," ACMA, Report, January 2017.

[151] IEEE Power and Energy Society, "Distribution test feeders." [Online]. Available: http://ewh.ieee.org/soc/pes/dsacom/testfeeders.html

[152] S. H. Mortazavi, Z. Moravej, and S. M. Shahrtash, "A hybrid method for arcing faults detection in large distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 94, pp. 141–150, 2018.

[153] K. Chul-Hwan, K. Hyun, K. Young-Hun, B. Sung-Hyun, R. K. Aggarwal, and A. T. Johns, "A novel fault-detection technique of high-impedance arcing faults in transmission lines using the wavelet transform," *IEEE Transactions on Power Delivery*, vol. 17, no. 4, pp. 921–929, 2002.

[154] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," in *Measures of complexity*. Springer, 2015, pp. 11–30.

[155] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[156] T. Marxsen, “EFD SWER Trial - final report,” IND Technology Pty Ltd, Report, June 2019.

[157] IND Technology, “Intelligent network diagnostic technology,” 2020. [Online]. Available: https://ind-technology.com.au/

[158] J. D. Miller, C. Skinner, H. Safford, E. E. Knapp, and C. Ramirez, “Trends and causes of severity, size, and number of fires in northwestern california, usa,” *Ecological Applications*, vol. 22, no. 1, pp. 184–203, 2012.

[159] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, “Can: Creative adversarial networks, generating “art" by learning about styles and deviating from style norms,” *arXiv preprint arXiv:1706.07068*, 2017.

[160] M. Mustafa, D. Bard, W. Bhimji, Z. Lukić, R. Al-Rfou, and J. M. Kratochvil, “Cosmogan: creating high-fidelity weak lensing convergence maps using generative adversarial networks,” *Computational Astrophysics and Cosmology*, vol. 6, no. 1, pp. 1–13, 2019.

Appendix A

Experiment and measurement set-up

The real, staged HIF tests were performed in a purpose-built facility in the city of Melbourne. The test rig was assembled in a local substation comprised of two ship containers. A dedicated feeder connected to the substation transfer bus was designed as the test rig energy supply. Fig. A.1 illustrates the simplified single-line diagram with the CB (Circuit Breaker), ACRs (Automatic Circuit Recloser), RCGSs (Remote Control Gas Switch), and HV resistors. The CB and ACRs have overcurrent, earth fault, and earth fault sensitivity protection. The illustrated capacitor is a 1.1 nF, 24 kV, coupling capacitor and HV resistors were included to avoid internal flashovers.

The substation is connected to a sub-transmission system of 66 kV and step down voltages via two Y-Y transformers. Their grounding connection is floating on the primary side and impedance grounded on the secondary by neutral earth resistors. Brief information regarding the feeders was disclosed but the fact that exists at least ten consumer load feeders, including industrial loads.

In regards to the measurement hardware, the 24-kV coupling capacitors were combined with bottom-end capacitors to form the dual-channel capacitive voltage divider (LF and HF channels). Two 125-V bidirectional voltage limiting diode and 350-V spark gaps were included in the channels to serve as over-voltage protection. In the HF channel, a 110 nF bottom-end capacitor with a 220 ohms shunt resistor branch was introduced to provide a ratio of 100:1 at high frequencies and high-pass characteristic with 10 kHz corner frequency. The output signal was processed by a Frequency Device, active 4-pole Butterworth filter to eliminate the 50 Hz signals and low-order harmonics. Anti-Aliasing filters, represented in the diagram as the "Low-pass ($<$1 MHz) filter", were also included.

The Gen3i HBM was adopted as the data acquisition mainframe, fitted with an HBM GN401 four-channel optical input card fed by four HBM GN110-2 optoisolated digitizers. The digitizers operate at a constant 100 MS/s, using anti-aliasing filters and data decimation to achieve the 2 MSa/s (HF) and 100 kSa/s (LF) sampling rates. The information presented

in this Appendix were extracted from the *Vegetation Conduction Ignition Test Report* [4], where further details are given.

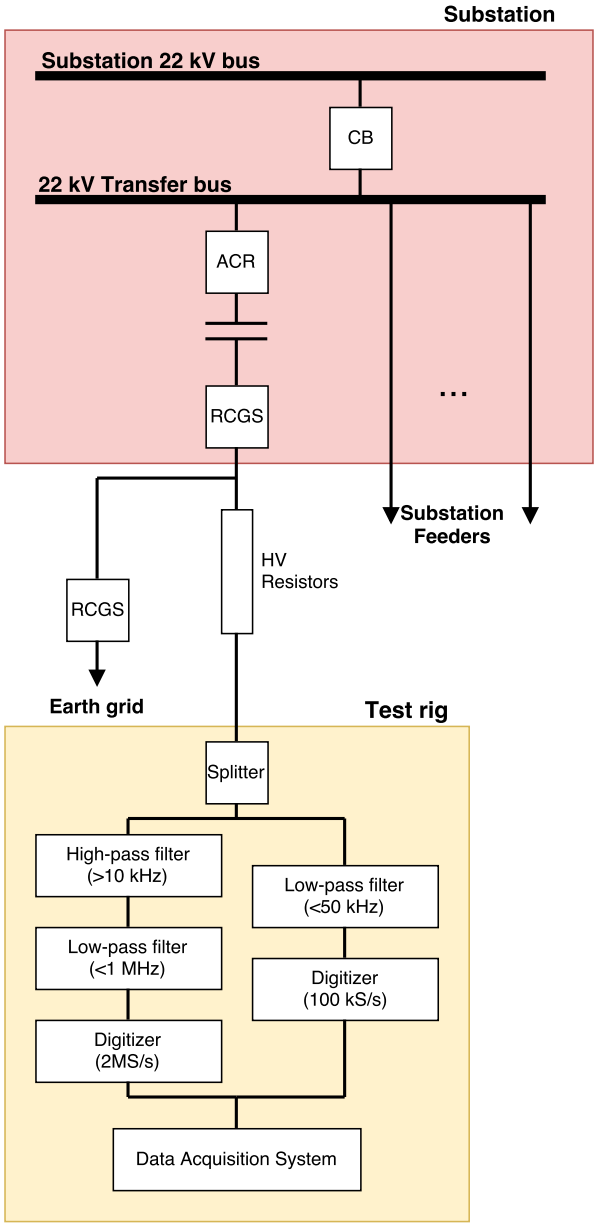


Figure A.1. Unifilar diagram of the feeder and test rig.

Appendix B

Codes

The following are MATLAB codes that exemplify the application of the concepts discussed in Chapter 3, Methodology. The purpose of all the scripts and functions are summarized:

- 'Main script example' - A script that describes the whole process of learning and testing the classifier; from loading the data set of signals to the validation results from 10-fold cross-validation.
- 'sweep_select' function - Receives the data set of signals and sweep extraction parameters as input, and returns the selected in-fault sweeps for learning.
- 'cal_select' function - Receives the data set of background recordings and extraction parameters as input, and returns the selected non-fault sweeps for learning.
- 'psd_calc' function - Receives the selected sweeps and frequency bands parameters, and returns the PSD-based features.
- 'wav_calc' function - Receives the selected sweeps and wavelet decomposition parameters, and returns the wavelet-based features.
- 'app_feed' function - Post-processing stage. It receives the calculated features and compares with the features of signals sampled in the same test day.
- 'classification' function - Receives the features and labels of observations, and returns the results of cross-validation.
- 'load_databases' script - Loads the voltages and current signals of the fault tests and background recordings into the workspace, as well as information about the sweep triggers, fault current thresholds, and other metadata.
- 'def_prob_day' function - Receives the metadata of tests enumeration, and returns the problematic tests to skip in the analysis.

- ‘tests_ensemble’ script - Script gathering all the test recordings, further loaded as ‘all_voltages_tests.mat’ and ‘all_current_tests.mat’ in the ‘load_databases’ script.
- ‘calibration_ensemble’ script - Script gathering all the calibration recordings, further loaded as ‘currents_cal.mat’ and ‘voltages_cal.mat’ in the ‘load_databases’ script.
- ‘calibration_ensemble’ script - Script gathering all the triggers recordings, further loaded as ‘triggers.mat’ in the ‘load_databases’ script.
- ‘Prototype main routine’ script - Example of the C++ routine ran by the Beaglebone while sampling and classifying signals using MATLAB functions via MATLAB Coder.

**The shift-invariant sparse coding technique is not included but can be found in the author’s website [134].

Main script example

```
%% Initializing
close all; clc; clearvars -except Voltages Voltages2 Voltages_cal Currents Currents_cal tests_info triggers

trainning_method=1; %1=yes 0=no
tests_under_analysis=1:1036;
type_to_test=0; %0=all 1=ph_e 2=ph_ph
current_thres=0.5; %Current threshold
max_eval_win_no=1; %How many best windows after fault to evaluate
low_freq_hi_samp_need=0; %Flag to select low_freq signals in win_pos (Voltage2)
current_hf_need=0; %Flag to select current sweeps

%Loading data
load_databases; %Voltages Voltages2 Voltages_cal Currents Currents_cal tests_info triggers

%% Information of tests by day

% February tests = 75:504
% March tests = 505:1038
%
% Cal ph_e = 36:95
% Cal ph_ph = 100:112
%
% Tests with feedback with ph_e cals (std n_cal) = 336 - 821 / Feb 23 - Mar 20
% Tests with feedback with ph_ph cals = 821 - end / Mar 20 - end

%% Capturing data

%flag tests to tests to skip and valid tests

[skipping_tests, all_tests]=def_prob_day(tests_info);

%Fault singals observations from pos-fault sweeps

%Calling sweep_select function
[~, ~, ~, ~, win_pos, ~, fault_ident, fault_type, day_ident, ~, ~, ~, ~]=...
    sweep_select(Voltages, Voltages2, Currents, all_tests, triggers, fault_currents,...
        tests_under_analysis,...
    trainning_method, current_thres, max_eval_win_no, type_to_test, skipping_tests,...
        low_freq_hi_samp_need, current_hf_need,...
    test_days);

%Non_fault signals observtions from calibration recordings

n_cal=[36 37 38 39 40 41 42 43 44 47 48 49 50 51 52 53 55 57 58 59 60 61 62 63 64 65 66 77 78 79 80 81 82 83 87 88 89 93 94 95 100 101 102 103 104 105 106 107 108 109 110 111 112];

volt_source_consi=0; %Discriminate between voltage source on or not 1-yes 0-no (can only to current and low_freq_hi_samp with 'no'
balancing=0; %Balance the amount of observations between pos and cal 1-yes 0-no
add_noise=0; % subtitute some cal obs for white noise
cal_current_hf_need=0; %Flag to select current sweeps
cal_low_freq_hi_samp_need=0; %Flag to select low_freq signals

[win_cal, cal_ident, voltage_off]=cal_select(Voltages_cal, Currents_cal, n_cal, volt_source_consi, balancing, add_noise, cal_days, cal_current_hf_need, cal_low_freq_hi_samp_need,length(win_pos));
```

```
%TO DO: FIX BALACING, VOLTAGE_SOURCE CONS, AND ADD NOISE WITH 'voltage_off' data

%to balance
%win_cal(voltage_off)=[];
%win_cal=win_cal(datasample([1:2466],566,'Replace',false));

%% Pre-processing
trainning_method=1;

enclosed_norm=0;
std_norm=0;
downsample_sig=1;

%Build signals structure
signals=struct();
for i=1:length(win_cal)+length(win_pos)
    if i<=length(win_cal)
        signals(i).voltage=win_cal(i).voltage;
    else
        signals(i).voltage=win_pos(i-length(win_cal)).voltage';
    end
end

%Standardization
if std_norm==1
    for i=1:length(signals)
        meann=mean(signals(i).voltage);
        signals(i).voltage=signals(i).voltage-meann;
        stdd=std(signals(i).voltage);
        signals(i).voltage=signals(i).voltage./stdd;
    end
end

%Normalization
if enclosed_norm==1
    for i=1:length(signals)
        signals(i).voltage=signals(i).voltage-mean(signals(i).voltage);
        signals(i).voltage=(signals(i).voltage)./22;
    end
end

%Downsample
if downsample_sig==1
    for i=1:length(signals)
        signals(i).voltage=filter(Butter_filter_3rd_order_500k,(signals(i).voltage));
    end

    for i=1:length(signals)
        signals(i).voltage=downsample(signals(i).voltage,2);
    end
end

%% Feature calculation (PSD/Wavelet)

inc_psd=0;
    regions=[358 367; 774 778; 898 909];
inc_wavelet=0;
```

```
    wav_lvl=4;
    idxvar=[1:4 13:16];
inc_basis=1;

feat_test=[];

%PSD features
if inc_psd==1
    psd_feat=psd_calc(signals,regions);
    feat_test=[feat_test psd_feat];
end

%Wavelet features
if inc_wavelet==1
    %Sum, peaks, std, entropy, kurtosis, ene
    wav_feat=wav_calc(signals,wav_lvl);
    feat_test=[feat_test wav_feat(:,idxvar)];
end

%Basis features
if inc_basis==1
    A=load('basis_pre_250s_566_tests_8_bases_100_int_beta_10.mat');
    A=A.A;

    if downsample_sig==1
        for i=1:size(A,4)
            A(:,1,1,i)=filter(Butter_filter_3rd_order_500k,A(:,1,1,i));
            A1(:,1,1,i)=downsample(A(:,1,1,i),2);
        end
        A=A1;
        clear A1
    end

    clear test_conv test_conv2 test_conv3 %fix it
    for o=1:size(signals,2)
        for i=1:size(A,4)
            test_conv(:,i)=conv(flip(A(:,1,1,i)),signals(o).voltage);
        end
        test_conv2(o,:)=sum(abs(test_conv));

        for i=1:size(test_conv,2)
            test_conv3(o,i)=wentropy(test_conv(:,i)./1e3,'log energy');
        end
    end

    feat_test=[feat_test test_conv2];
end

%% Post-processing

trainning_method=1;

apply_feedback=0; %on-off %CAL_IDENT AND DAY_IDENT NEEDS TO BE ALLIGNED w/ SIGNALS

if apply_feedback==1
    [feat_test, feedback_vec]=app_feed(feat_test,feat_test,trainning_method,win_cal,win_pos, day_ident,
         cal_ident, [], [], [],[]);
```

```
164 end
165
166 resp=ones(1,length(win_cal)+length(win_pos));
167 resp(logical(voltage_off))=0;
168 resp(length(win_cal)+1:end)=2;
169
170 %% Classification
171 [yFit,cv]=classification(feat_test,resp,[],[],[],trainning_method,[]);
```

sweep_select function

```
function [voltage_sweep_initial, current_sweep_initial, current_sweep_final, signal_duration, win_pos,
     win_cal, fault_ident, fault_type, day_ident, voltage_hf, voltage_lf, current_lf,
     voltage_lf_in_fault]=...
    sweep_select(Voltages, Voltages2, Currents, all_tests, triggers, fault_currents,
         tests_under_analysis,...
    trainning_method, current_thres, max_eval_win_no, type_to_test, skipping_tests,
         low_freq_hi_samp_need, current_hf_need, test_days)

voltage_sweep_initial=[];
current_sweep_initial=[];
current_sweep_final=[];
signal_duration=[];
win_pos=[];
win_cal=[];
fault_ident=[];
fault_type=[];
day_ident=[];
voltage_lf_in_fault=[];



%% Getting time inception, fault observations and data

if trainning_method==1
    clear win_pos current_lf_in_fault n_win_fault fault_ident fault_type voltage_lf_in_fault
end

for a=1:length(tests_under_analysis)

    test=find(all_tests==tests_under_analysis(a));

    if ~any(test)
        test=0;
    end

    %exclude by fault type if selected
    if type_to_test==1&&trainning_method==1
        if test>=536
            test=0;
        end
    elseif type_to_test==2&&trainning_method==1
        if test>=916||test<=535
            test=0;
        end
    end

    if (~any(test==skipping_tests)&&test~=0)||(trainning_method==0&&test~=0) %Test skipping, invalid,
         missing tests

        clear voltage_lf current_lf voltage_hf current_hf
        voltage_lf=Voltages(test).Voltage_LF;
        current_lf=Currents(test).Current_LF;
        voltage_hf=Voltages(test).Voltage_HF;
        current_hf=Currents(test).Current_HF;

        clear incept_time
```

```
clear Current_LF_RMS
wdn_size=2000; %Rms calculation window size
win_over=1000; %Window overlap in rms calculation
rms1=1;
incp1=1;
flag_in_fault=0;
flag_rms=0;
flag_seq=0;
incept_time=-1;
for o=1:win_over:length(current_lf)-wdn_size %Calculating current rms and extracting sample of
     fault injection and extinguishing
    Current_LF_RMS(rms1,1)=rms(current_lf(o:o+wdn_size));
    if Current_LF_RMS(rms1,1)>current_thres&&flag_in_fault==0
        if flag_rms==0
            flag_rms=1;
            flag_inception=o;
            flag_seq=o;
        elseif any(flag_rms==1:6)&&o-flag_seq==win_over
            flag_rms=flag_rms+1;
            flag_seq=o;
        elseif any(flag_rms==1:6)&&o-flag_seq~=win_over
            flag_rms=1;
            flag_inception=o;
            flag_seq=o;
        elseif flag_rms==7
            incept_time(incp1)=flag_inception+2000;
            incp1=incp1+1;
            flag_in_fault=1;
            flag_rms=0;
        end
    elseif Current_LF_RMS(rms1,1)<0.01&&flag_in_fault==1
        if flag_rms==0
            flag_rms=1;
            flag_inception=o;
            flag_seq=o;
        elseif any(flag_rms==1:10)&&o-flag_seq==win_over
            flag_rms=flag_rms+1;
            flag_seq=o;
        elseif any(flag_rms==1:10)&&o-flag_seq~=win_over
            flag_rms=1;
            flag_inception=o;
            flag_seq=o;
        elseif flag_rms==11
            incept_time(incp1)=flag_inception;
            incp1=incp1+1;
            flag_in_fault=0;
            flag_rms=0;
        end
    end
    rms1=rms1+1;
end

clear source_on_off
clear Voltage_LF_RMS
wdn_size=2000; %Rms calculation window size
win_over=1000; %window overlap in rms calculation
rms1=1;
```

```
108         incp1=1;
109         flag_in_fault=0;
110         flag_rms=0;
111         flag_seq=0;
112         source_on_off=-1;
113         for o=1:win_over:length(current_lf)-wdn_size %Calculating voltage rms and extracting sample of
                voltage turn on and off
114             Voltage_LF_RMS(rms1,1)=rms(voltage_lf(o:o+wdn_size));
115             if Voltage_LF_RMS(rms1,1)>100&&flag_in_fault==0
116                 if flag_rms==0
117                     flag_rms=1;
118                     flag_inception=o;
119                     flag_seq=o;
120                 elseif any(flag_rms==1:6)&&o-flag_seq==win_over
121                     flag_rms=flag_rms+1;
122                     flag_seq=o;
123                 elseif any(flag_rms==1:6)&&o-flag_seq~=win_over
124                     flag_rms=1;
125                     flag_inception=o;
126                     flag_seq=o;
127                 elseif flag_rms==7
128                     source_on_off(incp1)=flag_inception+2000;
129                     incp1=incp1+1;
130                     flag_in_fault=1;
131                     flag_rms=0;
132                 end
133             elseif Voltage_LF_RMS(rms1,1)<100&&flag_in_fault==1
134                 if flag_rms==0
135                     flag_rms=1;
136                     flag_inception=o;
137                     flag_seq=o;
138                 elseif any(flag_rms==1:20)&&o-flag_seq==win_over
139                     flag_rms=flag_rms+1;
140                     flag_seq=o;
141                 elseif any(flag_rms==1:20)&&o-flag_seq~=win_over
142                     flag_rms=1;
143                     flag_inception=o;
144                     flag_seq=o;
145                 elseif flag_rms==21
146                     source_on_off(incp1)=flag_inception;
147                     incp1=incp1+1;
148                     flag_in_fault=0;
149                     flag_rms=0;
150                 end
151             end
152             rms1=rms1+1;
153         end
154
155         if  incept_time(1)~=-1
156             %Fixing fault extingshing and source turn off time until the end of the test
157             if length(incept_time)==1
158                 incept_time(2)=length(current_lf);
159             end
160
161             if length(source_on_off)==1
162                 source_on_off(2)=length(voltage_lf);
163             end
```

```
164
165         %Sweeping taking and selection
166         least_time=1000; %find the initial sweep of fault inception
167         for o=1:length(triggers(test).trigger_time)
168             time_dif=triggers(test).trigger_time(o)-incept_time(1)/1e5;
169             if time_dif<least_time&&time_dif>0
170                 least_time=time_dif;
171                 current_sweep_initial=o;
172             end
173         end
174
175         least_time=-1000; %find the sweep of fault extinguishing
176         for o=1:length(triggers(test).trigger_time)
177             time_dif=triggers(test).trigger_time(o)-incept_time(2)/1e5;
178             if time_dif>least_time&&time_dif<0
179                 least_time=time_dif;
180                 current_sweep_final=o;
181             end
182         end
183
184         least_time=1000; %find the initial sweep of voltage source turned on
185         for o=1:length(triggers(test).trigger_time)
186             time_dif=triggers(test).trigger_time(o)-source_on_off(1)/1e5;
187             if time_dif<least_time&&time_dif>0
188                 least_time=time_dif;
189                 voltage_sweep_initial=o;
190             end
191         end
192
193         %signalling tests with possible pre_fault_signals (old)
194         if voltage_sweep_initial<current_sweep_initial
195             pre_fault_flag=1;
196         else
197             pre_fault_flag=0;
198         end
199
200         %printing first fault sweeps from the recording
201         if trainning_method==0
202             voltage_sweep_initial
203             current_sweep_initial
204             current_sweep_final
205             signal_duration=length(current_lf)/1e5
206             continue
207         end
208
209         %Collecting low frequency fault signals
210         clear voltage_lf_off_fault %Create vector with low-frequency Voltage sweeps in fault
211         if incept_time(1)-source_on_off(1)>2e4
212             voltage_lf_off_fault(:,1)=voltage_lf(source_on_off(1):incept_time(1));
213         end
214
215         clear voltage_lf_in_fault %Create vector with low-frequency Voltage sweeps in fault
216         voltage_lf_in_fault(:,1)=voltage_lf(incept_time(1):incept_time(2));
217
218         clear current_lf_in_fault %Create vector with low-frequency current sweeps in fault
219         current_lf_in_fault(:,1)=current_lf(incept_time(1):incept_time(2));
220
```

```
%Creating vector post and pre-fault
clear voltage_hf_in_fault %Create vector with High-frequency Voltage sweeps in fault
voltage_hf_in_fault(:,1)=voltage_hf(40000*(current_sweep_initial-1)+1:40000*(
    current_sweep_final));

clear current_hf_in_fault %Create vector with High-frequency current sweeps in fault
if length(current_hf)>=40000*(current_sweep_final)
    current_hf_in_fault(:,1)=current_hf(40000*(current_sweep_initial-1)+1:40000*(
        current_sweep_final));
else
    current_hf_in_fault=[];
end

clear voltage_hf_off_fault %Create vector with High-frequency Voltage sweeps pre fault
if pre_fault_flag==1
    voltage_hf_off_fault=voltage_hf(40000*(voltage_sweep_initial-1)+1:40000*(
        current_sweep_initial-1));
    faults_pre_fault(test,1)=test;
end


% Separating all post-fault sweeps
n_win_fault(a)=length(voltage_hf_in_fault)/4e4; %Indicator of how many HF sweep the test
    have
win_size1=1*4e4; %Window size

if length(voltage_hf_in_fault)>3e4 % If and fors for different length or position of sweep
    taken
    samp_count=1;
    while samp_count<=max_eval_win_no&&samp_count<=n_win_fault(a)
        win_pos(a).voltage((samp_count-1)*win_size1+1:win_size1*samp_count)=
            voltage_hf_in_fault((samp_count-1)*win_size1+1:win_size1*samp_count);
        samp_count=samp_count+1;
    end
    fault_ident(a)=tests_under_analysis(a);
end

if (current_hf_need||max_eval_win_no>1)
    if length(current_hf_in_fault)>3e4 % If and fors for different length or position of
        sweep taken
        samp_count=1;
        while samp_count<=max_eval_win_no&&samp_count<=n_win_fault(a)
            win_pos(a).current((samp_count-1)*win_size1+1:win_size1*samp_count)=
                current_hf_in_fault((samp_count-1)*win_size1+1:win_size1*samp_count);
            samp_count=samp_count+1;
        end
    end
end

% Separating a single sweep of the post-fault ones / best win
% select or not
if max_eval_win_no>1&&length(current_hf_in_fault)>3e4
    clear win_eval
    if  n_win_fault(a)<=max_eval_win_no
        for o=1:n_win_fault(a)
            %[c,l] = wavedec(win_pos(a).voltage((o-1)*4e4+1:o*4e4),3,'sym4');
            %cd3 = detcoef(c,l,3);
```

```
270                     %win_eval(o)=sum(abs(cd3));
271                     %win_eval(o)=wentropy(cd3,'log energy');
272                     win_eval(o)=(rms(win_pos(a).current((o-1)*4e4+1:o*4e4)));
273                 end
274             else
275                 for o=1:max_eval_win_no
276                     %[c,l] = wavedec(win_pos(a).voltage((o-1)*4e4+1:o*4e4),3,'sym4');
277                     %cd3 = detcoef(c,l,3);
278                     %win_eval(o)=sum(abs(cd3));
279                     %win_eval(o)=wentropy(cd3,'log energy');
280                     win_eval(o)=(rms(win_pos(a).current((o-1)*4e4+1:o*4e4)));
281                 end
282             end
283
284
285             if length(current_hf_in_fault)>3e4
286                 best_win=find(win_eval==max(win_eval));
287             else
288                 best_win=1;
289             end
290
291             swp_no(a)=best_win-1+current_sweep_initial;
292             if swp_no(a)==0
293                 flagx=a;
294             end
295             %test_ident(a)=test;
296
297             win_pos(a).voltage=win_pos(a).voltage((best_win-1)*4e4+1:best_win*4e4);
298             win_pos(a).current=win_pos(a).current((best_win-1)*4e4+1:best_win*4e4);
299
300             if (low_freq_hi_samp_need==1)
301                 if (test==213||test==420)
302                     win_pos(a).voltage2=Voltages2(test).Voltage_LF_HiSamp(end-4e4+1:end);
303                 else
304                     win_pos(a).voltage2=Voltages2(test).Voltage_LF_HiSamp((swp_no(a)-1)*4e4+1:swp_no
                            (a)*4e4);
305                 end
306             end
307         else
308             if (test==213||test==420)&&(low_freq_hi_samp_need==1)
309                 win_pos(a).voltage2=Voltages2(test).Voltage_LF_HiSamp(end-4e4+1:end);
310             elseif any(current_hf_in_fault)
311                 win_pos(a).voltage=win_pos(a).voltage(1:4e4);
312                 if low_freq_hi_samp_need==1
313                     win_pos(a).voltage2=Voltages2(test).Voltage_LF_HiSamp((current_sweep_initial-1)
                            *4e4+1:current_sweep_initial*4e4);
314                 end
315             end
316         end
317
318         % Signalling of type of test analyzed
319         if length(voltage_hf_in_fault)>3e4
320             if test<536
321                 fault_type(a,1)=1; %First windown
322             elseif test<916&&test>535
323                 fault_type(a,1)=2; %First windown
324             elseif test>915&&test<971
```

```
                    fault_type(a,1)=3; %First windown
                elseif test>970&&test<995
                    fault_type(a,1)=4; %First windown
                end
            end

        end
    end
end

if trainning_method==1


    fault_type=fault_type(find(fault_type));
    fault_ident=fault_ident(find(fault_ident));

    %swp_no=swp_no(find(swp_no));

    %test_ident=test_ident(find(test_ident));

    %taking 0 coluns of win_pos
    zero_coluns=0;
    for i=1:length(win_pos)
        if ~any(win_pos(i).voltage)
            zero_coluns(i)=i;
        end
    end
    win_pos(zero_coluns(find(zero_coluns)))=[];
    voltage_lf_in_fault(zero_coluns(find(zero_coluns)))=[];

    %Day of test identification
    clear day_ident
    for i=1:length(fault_ident)
        if any(fault_ident(i)==test_days.feb_12)
            day_ident(i)=121;
        elseif any(fault_ident(i)==test_days.feb_13)
            day_ident(i)=131;
        elseif any(fault_ident(i)==test_days.feb_16)
            day_ident(i)=161;
        elseif any(fault_ident(i)==test_days.feb_17)
            day_ident(i)=171;
        elseif any(fault_ident(i)==test_days.feb_18)
            day_ident(i)=181;
        elseif any(fault_ident(i)==test_days.feb_19)
            day_ident(i)=191;
        elseif any(fault_ident(i)==test_days.feb_20)
            day_ident(i)=201;
        elseif any(fault_ident(i)==test_days.feb_23)
            day_ident(i)=231;
        elseif any(fault_ident(i)==test_days.feb_24)
            day_ident(i)=241;
        elseif any(fault_ident(i)==test_days.feb_25)
            day_ident(i)=251;
        elseif any(fault_ident(i)==test_days.feb_26)
            day_ident(i)=261;
        elseif any(fault_ident(i)==test_days.feb_27)
            day_ident(i)=271;
```

```
382         elseif any(fault_ident(i)==test_days.feb_27)
383             day_ident(i)=271;
384         elseif any(fault_ident(i)==test_days.mar_3)
385             day_ident(i)=032;
386         elseif any(fault_ident(i)==test_days.mar_5)
387             day_ident(i)=052;
388         elseif any(fault_ident(i)==test_days.mar_6)
389             day_ident(i)=062;
390         elseif any(fault_ident(i)==test_days.mar_13)
391             day_ident(i)=132;
392         elseif any(fault_ident(i)==test_days.mar_16)
393             day_ident(i)=162;
394         elseif any(fault_ident(i)==test_days.mar_17)
395             day_ident(i)=172;
396         elseif any(fault_ident(i)==test_days.mar_18)
397             day_ident(i)=182;
398         elseif any(fault_ident(i)==test_days.mar_19)
399             day_ident(i)=192;
400         elseif any(fault_ident(i)==test_days.mar_20)
401             day_ident(i)=202;
402         elseif any(fault_ident(i)==test_days.mar_23)
403             day_ident(i)=232;
404         elseif any(fault_ident(i)==test_days.mar_24)
405             day_ident(i)=242;
406         elseif any(fault_ident(i)==test_days.mar_25)
407             day_ident(i)=252;
408         elseif any(fault_ident(i)==test_days.mar_26)
409             day_ident(i)=262;
410         elseif any(fault_ident(i)==test_days.mar_27)
411             day_ident(i)=272;
412         end
413     end
414 end
415 
416 end
```

cal_select function

```
%% Gathering calibration data

function [win_cal, cal_ident, voltage_off]=cal_select(Voltages_cal, Currents_cal, n_cal,
     volt_source_consi, balancing, add_noise, cal_days, cal_current_hf_need, cal_low_freq_hi_samp_need,
     n_balancing)

clear win_cal cal_ident cal_ident2
stop_len=0;
n_win_cal=1;
cal_ident_flag=0;
cal_ident_idx=1;
voltage_off=[];

% %Discriminative between voltage source on/off
if volt_source_consi==1
    for a=1:length(n_cal)
        cal=n_cal(a);

        % Load the calibration test and figure out start and stop time of the
        % test
        clear voltage_cal
        voltage_cal=Voltages_cal(cal).Voltage_HF;

        clear incept_time
        clear voltage_cal_RMS
        wdn_size=2000; %Rms calculation window size
        win_over=1000; %Window overlap in rms calculation
        rms1=1;
        incp1=1;
        flag_in_fault=0;
        flag_rms=0;
        flag_seq=0;
        incept_time=-1;
        for o=1:win_over:length(voltage_cal)-wdn_size %Calculating calibration starting
            voltage_cal_RMS(rms1,1)=rms(voltage_cal(o:o+wdn_size));
            if voltage_cal_RMS(rms1,1)>0.3&&flag_in_fault==0
                if flag_rms==0
                    flag_rms=1;
                    flag_inception=o;
                    flag_seq=o;
                elseif any(flag_rms==[1:10])&&o-flag_seq==win_over
                    flag_rms=flag_rms+1;
                    flag_seq=o;
                elseif any(flag_rms==[1:10])&&o-flag_seq~=win_over
                    flag_rms=1;
                    flag_inception=o;
                    flag_seq=o;
                elseif flag_rms==11
                    incept_time(incp1)=flag_inception+2000;
                    incp1=incp1+1;
                    flag_in_fault=1;
                    flag_rms=0;
                end
            elseif voltage_cal_RMS(rms1,1)<0.3&&flag_in_fault==1
                if flag_rms==0
                    flag_rms=1;
```

```
                flag_inception=o;
                flag_seq=o;
            elseif any(flag_rms==[1:10])&&o-flag_seq==win_over
                flag_rms=flag_rms+1;
                flag_seq=o;
            elseif any(flag_rms==[1:10])&&o-flag_seq~=win_over
                flag_rms=1;
                flag_inception=o;
                flag_seq=o;
            elseif flag_rms==11
                incept_time(incp1)=flag_inception;
                incp1=incp1+1;
                flag_in_fault=0;
                flag_rms=0;
            end
        end
        rms1=rms1+1;
    end

    if length(incept_time)==1
        incept_time(1)=incept_time(1);
        incept_time(2)=length(voltage_cal);
    end

    % Gathering of calibration sweeps in window slices
    win_size2=1*4e4; %window size

    measu_time=incept_time(2)-incept_time(1)+1;
    sweeps_cal=fix(measu_time/win_size2); %amount of sweeps of the loaded test

    n_sweeps_call(a)=sweeps_cal; %signalizer

    for i=1:sweeps_cal %
        win_cal(stop_len+i).voltage=voltage_cal(incept_time+((i-1)*win_size2):incept_time+((i-1)*
             win_size2)+(win_size2-1));
    end
    stop_len=length(win_cal);


    if any(n_cal(a)==cal_days.cal_feb_23)&&cal_ident_flag==0
        cal_ident(cal_ident_idx:sweeps_cal)=231;
        cal_ident_flag=1;
        cal_ident_idx=sweeps_cal;
    elseif any(n_cal(a)==cal_days.cal_feb_23)
        cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=231;
        cal_ident_idx=cal_ident_idx+sweeps_cal;
    elseif any(n_cal(a)==cal_days.cal_feb_24)
        cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=241;
        cal_ident_idx=cal_ident_idx+sweeps_cal;
    elseif any(n_cal(a)==cal_days.cal_feb_25)
        cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=251;
        cal_ident_idx=cal_ident_idx+sweeps_cal;
    elseif any(n_cal(a)==cal_days.cal_feb_26)
        cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=261;
        cal_ident_idx=cal_ident_idx+sweeps_cal;
    elseif any(n_cal(a)==cal_days.cal_feb_27)
        cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=271;
```

```
111             cal_ident_idx=cal_ident_idx+sweeps_cal;
112 
113         elseif any(n_cal(a)==cal_days.cal_mar_3)
114             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=032;
115             cal_ident_idx=cal_ident_idx+sweeps_cal;
116         elseif any(n_cal(a)==cal_days.cal_mar_5)
117             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=052;
118             cal_ident_idx=cal_ident_idx+sweeps_cal;
119         elseif any(n_cal(a)==cal_days.cal_mar_6)
120             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=062;
121             cal_ident_idx=cal_ident_idx+sweeps_cal;
122         elseif any(n_cal(a)==cal_days.cal_mar_13)
123             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=132;
124             cal_ident_idx=cal_ident_idx+sweeps_cal;
125         elseif any(n_cal(a)==cal_days.cal_mar_16)
126             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=162;
127             cal_ident_idx=cal_ident_idx+sweeps_cal;
128         elseif any(n_cal(a)==cal_days.cal_mar_17)
129             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=172;
130             cal_ident_idx=cal_ident_idx+sweeps_cal;
131         elseif any(n_cal(a)==cal_days.cal_mar_18)
132             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=182;
133             cal_ident_idx=cal_ident_idx+sweeps_cal;
134         elseif any(n_cal(a)==cal_days.cal_mar_19)
135             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=192;
136             cal_ident_idx=cal_ident_idx+sweeps_cal;
137         elseif any(n_cal(a)==cal_days.cal_mar_20)
138             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=202;
139             cal_ident_idx=cal_ident_idx+sweeps_cal;
140         elseif any(n_cal(a)==cal_days.cal_mar_23)
141             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=232;
142             cal_ident_idx=cal_ident_idx+sweeps_cal;
143 
144         elseif any(n_cal(a)==cal_days.cal_mar_24)
145             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=242;
146             cal_ident_idx=cal_ident_idx+sweeps_cal;
147         elseif any(n_cal(a)==cal_days.cal_mar_25)
148             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=252;
149             cal_ident_idx=cal_ident_idx+sweeps_cal;
150         elseif any(n_cal(a)==cal_days.cal_mar_26)
151             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=262;
152             cal_ident_idx=cal_ident_idx+sweeps_cal;
153         elseif any(n_cal(a)==cal_days.cal_mar_27)
154             cal_ident(cal_ident_idx:cal_ident_idx+sweeps_cal)=272;
155             cal_ident_idx=cal_ident_idx+sweeps_cal;
156         end
157 
158     end
159 end
160 
161 
162 %Balancing no. of windows
163 if balancing==1
164     [win_cal,idx]=datasample(win_cal,n_balancing,'Replace',false);
165     cal_ident=cal_ident(idx);
166 
167 end
```

```

if add_noise==1
    for i=length(win_cal)-19:length(win_cal)
        win_cal(i).voltage=wgn(4e4,1,10*log10(0.60));
    end
    cal_ident(length(cal_ident)-19:length(cal_ident))=datasample(cal_ident,20);
end
```

psd_calc function

```
%% Calculation of FT features
function psd_feat=psd_calc(signals,regions)

clear pxx

n_regions=size(regions,1);

fft_bin=450; %No. of calculated fft bins
win_length=1e3;
bin_length=1e6/fft_bin;

for i=1:length(signals)
    [pxx(i,:),f1] = pwelch(signals(i).voltage,win_length,[],fft_bin*2,2e6);
end

%Normalization
% for i=1:size(pxx,1)
%     total_power=sum(pxx(i,:));
%     pxx(i,:)=pxx(i,:)./total_power;
% end

psd_feat=zeros(size(pxx,1),n_regions);
for i=1:n_regions
    init_bin=floor(regions(i,1)*1e3/bin_length);
    final_bin=floor(regions(i,2)*1e3/bin_length);
    psd_feat(:,i)=max(pxx(:,init_bin:final_bin),[],2);
end

end
```

wav_calc function

```
function wav_feat=wav_calc(signals,level)

% Calculation of wavelet features
% 'signals' is the structure organized as signals(i).voltage with the signals
% to be calculated
% 'level' standard decimated wavelet level
% The six different features need to be activated below

n_level=level; %wavelet level used
calc_sum=1; %sum features 1-on 0-off
calc_peaks=1; %peaks features 1-on 0-off
calc_std=1; %std features 1-on 0-off
calc_ene=1; %energy percentual features 1-on 0-off
calc_ent=1; %entropy features 1-on 0-off
calc_kurt=1; %kurtosis features 1-on 0-off

clear sum_feat peak_feat std_feat ent_feat kurt_feat ene_feat wav_feat
%declaring feature vectors
sum_feat=zeros(length(signals),n_level);
peak_feat=zeros(length(signals),n_level);
std_feat=zeros(length(signals),n_level);
ent_feat=zeros(length(signals),n_level);
kurt_feat=zeros(length(signals),n_level);
ene_feat=zeros(length(signals),n_level+1);

for i=1:length(signals)

    [c,l] = wavedec(signals(i).voltage,n_level,'sym4');

    for o=1:n_level
        coef=detcoef(c,l,o);
        if calc_sum==1 %Sum of coefficients features
            sum_feat(i,o)=sum(abs(coef));
        end
        if calc_peaks==1 %Peak features
            [~,~,~,p] = findpeaks(double(abs(coef)),'SortStr','descend');
            peak_feat(i,o)=mean(p(1:fix(length(coef)*0.01)));
        end
        if calc_std==1 %STD features
            std_feat(i,o)=std(coef);
        end
        if calc_ent==1 %entropy features
            ent_feat(i,o)=wentropy(coef,'log energy');
        end
        if calc_kurt==1 %Kustosis features
            kurt_feat(i,o)=kurtosis(coef);
        end
    end
    if calc_ene==1 %Energy features
        [Ea,Ed] = wenergy(c,l);
        ene_feat(i,1)=Ea;
        ene_feat(i,2:end)=Ed';
    end
end

wav_feat=[sum_feat peak_feat std_feat ent_feat kurt_feat ene_feat];
```

```
57
58  return
```

app_feed function

```
function [feat_test, feedback_vec]=app_feed(input,feat_test_input,training_flag,win_cal,win_pos,
    day_ident,cal_ident,tests_under_analysis,test_days,testing_win,feedback_vec)

% Feedback approach
%if apply_feedback==1

tests_days=[121 131 161 171 181 191 201 231 241 251 261 271 032 052 062 132 162 172 182 192 202 232 242
    252 262 272];
%feedback_for_tests=[231 231 231 231 231 231 231 231 241 251 261 271 032 052 062 132 132 172 182 192 202
     232 242 252 262 272];
feedback_for_tests=[231 231 231 231 231 231 231 231 241 251 261 271 032 052 062 132 132 172 182 192 202
    202 202 202 202 202];
cal_tests_days=[231 241 251 261 271 032 052 062 132 162 172 182 192 202 232 242 252 262 272];



if training_flag==1

    feedback_vec=zeros(size(feat_test_input,2),length(cal_tests_days));

    for i=1:length(cal_tests_days)
        %feedback_vec(:,i)=mean(feat_test_input(cal_ident==cal_tests_days(i),:));
        feedback_vec(:,i)=prctile(feat_test_input(cal_ident==cal_tests_days(i),:),90);
    end

    for i=1:length(win_cal)
        for o=1:length(cal_tests_days)
            if cal_ident(i)==cal_tests_days(o)
                feat_test(i,:)=input(i,:)./feedback_vec(:,o)';
            end
        end
    end

    for i=1:length(win_pos)
        for o=1:length(tests_days)
            if day_ident(i)==tests_days(o)
                feat_test(length(win_cal)+i,:)=input(length(win_cal)+i,:)./feedback_vec(:,cal_tests_days
                    ==feedback_for_tests(o))';
            end
        end
    end
else

    day_of_test=[];

    if any(tests_under_analysis==test_days.feb_12)
        day_of_test=121;
    elseif any(tests_under_analysis==test_days.feb_13)
        day_of_test=131;
    elseif any(tests_under_analysis==test_days.feb_16)
        day_of_test=161;
    elseif any(tests_under_analysis==test_days.feb_17)
        day_of_test=171;
    elseif any(tests_under_analysis==test_days.feb_18)
        day_of_test=181;
    elseif any(tests_under_analysis==test_days.feb_19)
```

```
52          day_of_test=191;
53      elseif any(tests_under_analysis==test_days.feb_20)
54          day_of_test=201;
55      elseif any(tests_under_analysis==test_days.feb_23)
56          day_of_test=231;
57      elseif any(tests_under_analysis==test_days.feb_24)
58          day_of_test=241;
59      elseif any(tests_under_analysis==test_days.feb_25)
60          day_of_test=251;
61      elseif any(tests_under_analysis==test_days.feb_26)
62          day_of_test=261;
63      elseif any(tests_under_analysis==test_days.feb_27)
64          day_of_test=271;
65      elseif any(tests_under_analysis==test_days.feb_27)
66          day_of_test=271;
67      elseif any(tests_under_analysis==test_days.mar_3)
68          day_of_test=032;
69      elseif any(tests_under_analysis==test_days.mar_5)
70          day_of_test=052;
71      elseif any(tests_under_analysis==test_days.mar_6)
72          day_of_test=062;
73      elseif any(tests_under_analysis==test_days.mar_13)
74          day_of_test=132;
75      elseif any(tests_under_analysis==test_days.mar_16)
76          day_of_test=162;
77      elseif any(tests_under_analysis==test_days.mar_17)
78          day_of_test=172;
79      elseif any(tests_under_analysis==test_days.mar_18)
80          day_of_test=182;
81      elseif any(tests_under_analysis==test_days.mar_19)
82          day_of_test=192;
83      elseif any(tests_under_analysis==test_days.mar_20)
84          day_of_test=202;
85      elseif any(tests_under_analysis==test_days.mar_23)
86          day_of_test=232;
87      elseif any(tests_under_analysis==test_days.mar_24)
88          day_of_test=242;
89      elseif any(tests_under_analysis==test_days.mar_25)
90          day_of_test=252;
91      elseif any(tests_under_analysis==test_days.mar_26)
92          day_of_test=262;
93      elseif any(tests_under_analysis==test_days.mar_27)
94          day_of_test=272;
95      end
96
97      feed_to_use=feedback_for_tests(day_of_test==tests_days);
98      for i=1:length(testing_win)
99          feat_test(i,:)= input(i,:)./feedback_vec(:,cal_tests_days==feed_to_use)';
100     end
101 end
```

classification function

```
% Trainning classifier/classificating new data

function [yFit,cv]=classification(feat_test,resp,feat_test_input,resp_input,cv,trainning_method,
     classifier_trained)

if trainning_method==1

    t = templateTree('NumVariablesToSample','all','MinLeafSize',20);

    %to test in kfold
    cv = fitcensemble(feat_test,resp,'Method','Bag','NumLearningCycles',350,'Prior','uniform',...
        'kfold',10);

    %For 3 classes
    [yFit,sFit] = kfoldPredict(cv);
    conf_mat=confusionmat(resp,yFit)
    accuracy=sum(diag(conf_mat))/size(feat_test,1)
    dependability=1-(sum(conf_mat(3,1:2))/sum(conf_mat(3,:)))

%For 2 classes
%         [yFit,sFit] = kfoldPredict(cv);
%         conf_mat=confusionmat(resp,yFit)
%         accuracy=(conf_mat(1 ,1)+conf_mat(2,2))/length(resp)
%         dependability=conf_mat(2,2)/(conf_mat(2,1)+conf_mat(2,2))
%         security=conf_mat(1,1)/(conf_mat(1,1)+conf_mat(1,2))

else
    if classifier_trained==0

        t = templateTree('NumVariablesToSample','all','MinLeafSize',20);
        %t = templateTree('MaxNumSplits',10,'NumVariablesToSample','all','MinLeafSize',5,'Prior','
            uniform');

        % prior uniform
        cv = fitcensemble(feat_test_input,resp_input,'Method','Bag','NumLearningCycles',350,'Learners',t
            );

    end

    yFit = predict(cv,feat_test);

end
```

load_databases script

```
if (exist('Voltages') ~= 1)
    load('all_voltages_tests.mat'); %Voltage tests database
end

if (exist('Currents') ~= 1)
    load('all_current_tests.mat'); %Current tests database
end

if (exist('Currents_cal') ~= 1)
    load('currents_cal.mat'); %Calibration currents
end

if (exist('Voltages_cal') ~= 1)
    load('voltages_cal.mat'); %Calibration voltages
end

if (exist('Voltages2') ~= 1)
    if low_freq_hi_samp_need==1
        load('voltage_tests2.mat'); %High frequency (sweeps) sampling of the LF channel
    else
        Voltages2=[];
    end

end

load('triggers.mat'); %HF time triggers
load('tests_info.mat'); %Tests numbering and types
load('fault_currents.mat'); %fault current threshold adopted in the tests
load('voltage_cal_off_index.mat'); %sweeps of calibration with no voltage source (most used n_cal config
    )
load('test_days.mat'); %Day which the tests were executed with numbers
load('cal_days.mat'); %Day which the cal were executed with numbers

load('Butter_filter_3rd_order_500k.mat'); %Load butterworth filter for downsample

load('butter_3rd_5kk_samp_1kk_CF.mat');
```

def_prob_day function

```
function [skipping_tests, all_tests]=def_prob_day(tests_info)

%% Defining types of faults (ph_e_tests, ph_ph_tests, bush_tests, grass_tests)

%Corrupted/missing
broken_tests=[1,19,330,085,105,160,708:745];

ph_ph_tests=zeros(380,1);
bush_tests=zeros(55,1);
ph_e_tests=zeros(535,1);
grass_tests=zeros(24,1);

p1=1;
p2=1;
p3=1;
p4=1;
for i=1:length(tests_info)
    if ~any(tests_info(i,1)==broken_tests)
        if tests_info(i,2)==1
            ph_ph_tests(p1,1)=tests_info(i,1);
            p1=p1+1;
        elseif tests_info(i,2)==2
            bush_tests(p2,1)=tests_info(i,1);
            p2=p2+1;
        elseif tests_info(i,2)==3
            ph_e_tests(p3,1)=tests_info(i,1);
            p3=p3+1;
        elseif tests_info(i,2)==4
            grass_tests(p4,1)=tests_info(i,1);
            p4=p4+1;
        end
    end
end

all_tests=[ph_e_tests;ph_ph_tests;bush_tests;grass_tests];

% Tests recording
% From 1 to 535 = Phase to earth faults
% From 536 to 915 = Phase to Phase faults
% From 916 to 970 = Bush faults
% From 971 to 994 = Grass faults

%% Defining problem tests

% ph_e_tests
%Tests with more than one fault inception
tests_more_inception1=[15;179;413];
%Tests with no fault current
tests_no_current1=[45;46;47;119;236];
%Tests with no fault sweeps
tests_no_sweeps1=[83;163;168;227;282;291;294;328;523;525];
%Faults that do not have sweeps with 40k samples
non_std_samp1=165;
%tests with pre_fault signal
tests_pre_fault1=[1;2;14;395];

```

```
%ph_ph_faults
%Tests with more than one fault inception
tests_more_inception2=[570;571;574];
%Tests with no fault current
tests_no_current2=567;
%Tests with no fault sweeps
tests_no_sweeps2
    =[594;597;603;606;609;620;621;623;630;640;646;656;661;662;665;666;671;675;748;779;785;790;794;795;
821;831;862;872;679;680;681;682;683;684];
%Faults that do not have sweeps with 40k samples
non_std_samp2=[536;537;538;539;540;541;542;543;544;545;546;547];
%tests with pre_fault signal
tests_pre_fault2=[548;549;550;551;552;553;554;555;556;557;558;559;560;561;562;563;564;569;573;913];

%bush_faults
%Tests with more than one fault inception
tests_more_inception3=[919;927;928;937;941;943;945;951;952;953;963;965];
%Tests with no fault current
tests_no_current3=[];
%Tests with no fault sweeps
tests_no_sweeps3=[931;954;955];
%Faults that do not have sweeps with 40k samples
non_std_samp3=[];
%tests with pre_fault signal
tests_pre_fault3=[916;917;918;919;927;933;935;936;940;944;956;957;967;968;969;970];

%grass_faults
%Tests with more than one fault inception
tests_more_inception4=974;
%Tests with no fault current
tests_no_current4=[975;976;978;980;986;987;988;989;990;991;992;993;994];
%Tests with no fault sweeps
tests_no_sweeps4=[971;972;983;985];
%Faults that do not have sweeps with 40k samples
non_std_samp4=[];
%tests with pre_fault signal
tests_pre_fault4=981;

%Defining tests to skip
skipping_tests1=[tests_more_inception1; tests_no_current1; tests_no_sweeps1; non_std_samp1];
skipping_tests2=[tests_more_inception2; tests_no_current2; tests_no_sweeps2; non_std_samp2];
skipping_tests3=[tests_more_inception3; tests_no_current3; tests_no_sweeps3; non_std_samp3];
skipping_tests4=[tests_more_inception4; tests_no_current4; tests_no_sweeps4; non_std_samp4];
skipping_tests=[skipping_tests1; skipping_tests2; skipping_tests3; skipping_tests4];

%Tests with low_range/questionable sampling
tests_low_range
    =[958,959,960,961,962,964,966,686,687,688,689,690,691,693,694,695,696,697,698,699,700,701,702,703,704,705,

706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724];

%Tests marked invalid by the report
invalid_tests
    =[536;537;538;539;540;541;542;543;544;545;546;547;548;549;550;551;552;553;554;555;556;557;558;559;
560;561;562;563;564;916;917;918;919;1;2;3;565;566;567;568;569;570;571;572;573;574;575;576;577;578;
579;580;581;582;583;4;5;6;7;8;9;10;11;12;13;584;14;15;585;16;586;587;38;971;972;973;974;400;401;
668;669;670;672;674;676;678;483;484;778;795;796;797;825;834;835;863];
```

```
110
111
112 %Tests with perciaveble pre_faults
113 tests_pre_fault=[tests_pre_fault1; tests_pre_fault2; tests_pre_fault3; tests_pre_fault4];
114
115 %Tests with perciaveble pre_faults that are not invalid
116 real_tests_pre_fault=[395;913;927;933;935;936;940;944;956;957;967;968;969;970;981];
117
118 %Final vector skipping_tests
119 skipping_tests=[skipping_tests; tests_low_range'; invalid_tests];
```

tests_ensemble script

```
close all; clc; clear all;
load('tests_info.mat'); %Load types of fault

%% Defining broken tests
asd=[708:745];
asd=asd';
broken_tests=[1;19;330;085;105;160;asd];

%% Defining types of faults
p1=1;
p2=1;
p3=1;
p4=1;
for o=1:1038 % create ph_ph_tests, bush_tests, ph_e_tests, grass_tests
    if ~any(broken_tests==o)
        if tests_info(find(tests_info(:,1)==o),2)==1
            ph_ph_tests(p1,1)=o;
            p1=p1+1;
        end
        if tests_info(find(tests_info(:,1)==o),2)==2
            bush_tests(p2,1)=o;
            p2=p2+1;
        end
        if tests_info(find(tests_info(:,1)==o),2)==3
            ph_e_tests(p3,1)=o;
            p3=p3+1;
        end
        if tests_info(find(tests_info(:,1)==o),2)==4
            grass_tests(p4,1)=o;
            p4=p4+1;
        end
    end
end

all_tests=[ph_e_tests;ph_ph_tests;bush_tests;grass_tests];

% Tests recording
% From 1 to 535 = Phase to earth faults
% From 536 to 915 = Phase to Phase faults
% From 916 to 970 = Bush faults
% From 971 to 994 = Grass faults

%% Data gathering

for test_gat=1:length(all_tests)
    Test=num2str(all_tests(test_gat,1));
    if all_tests(test_gat,1)<100
        Test=strcat('0',Test);
    end
    if all_tests(test_gat,1)<10
        Test=strcat('0',Test);
    end

    Filename=strcat('VT',num2str(Test),'.pnrf');
    FromDisk=actxserver('Perception.Loaders.PNRF');
    Data=FromDisk.LoadRecording(Filename);
```

```
57
58          for PLoop=1:4
59              if PLoop==1||3
60                  ItfData = Data.Recorders.Item(1).Channels.Item(PLoop).DataSource(1);
61                  ItfData.get;
62                  SegmentsOfData = ItfData.Data(-200, 200);
63                  WaveformData = SegmentsOfData.Item(1).Waveform(4, 1, 1e8, 1);
64                      if PLoop==1
65                          Voltages(test_gat).Voltage_LF(:,1)=WaveformData(1,:);
66                      end
67                      if PLoop==3
68                          Currents(test_gat).Current_LF(:,1)=WaveformData(1,:);
69                      end
70              end
71
72              if PLoop==2||4
73                  ItfData = Data.Recorders.Item(1).Channels.Item(PLoop).DataSource(2); %Locate Data
74                  ItfData.get; %Data info
75                  SegmentsOfData = ItfData.Data(-200, 200); %Select time interval of data
76                  if PLoop==2
77                      for o=1:SegmentsOfData.get.Count
78                          WaveformData = SegmentsOfData.Item(o).Waveform(4, 1, 1e8, 1); %Collect Data
79                          position=length(WaveformData)*(o-1);
80                              for p=1:length(WaveformData)
81                                  Voltages(test_gat).Voltage_HF(p+position,1)=WaveformData(1,p);
82                              end
83                      end
84                  end
85                  if PLoop==4
86                      for o=1:SegmentsOfData.get.Count
87                          WaveformData = SegmentsOfData.Item(o).Waveform(4, 1, 1e8, 1); %Collect Data
88                          position=length(WaveformData)*(o-1);
89                              for p=1:length(WaveformData)
90                                  Currents(test_gat).Current_HF(p+position,1)=WaveformData(1,p);
91                              end
92                      end
93                  end
94              end
95          end
96  end
97  clc;
```

calibration_ensemble script

```
%% Data gathering
broken_tests=[6; 7; 8; 9; 10; 46; 56;67];
for test_gat=1:67
    if ~any(test_gat==broken_tests)
        Test=num2str(test_gat);
        if test_gat<100
            Test=strcat('0',Test);
        end
        if test_gat<10
            Test=strcat('0',Test);
        end

        Filename=strcat('VT_Test_site_calibration',num2str(Test),'.pnrf');
        FromDisk=actxserver('Perception.Loaders.PNRF');
        Data=FromDisk.LoadRecording(Filename);

            for PLoop=1:4
                if PLoop==1||3
                    ItfData = Data.Recorders.Item(2).Channels.Item(PLoop).DataSource(1);
                    ItfData.get;
                    SegmentsOfData = ItfData.Data(-200, 200);
                    WaveformData = SegmentsOfData.Item(1).Waveform(4, 1, 1e8, 1);
                        if PLoop==1
                            Voltages_cal(test_gat).Voltage_LF(:,1)=WaveformData(1,:);
                        end
                        if PLoop==3
                            Currents_cal(test_gat).Current_LF(:,1)=WaveformData(1,:);
                        end
                end

                if PLoop==2||4
                    ItfData = Data.Recorders.Item(2).Channels.Item(PLoop).DataSource(2); %Locate Data
                    ItfData.get; %Data info
                    SegmentsOfData = ItfData.Data(-200, 200); %Select time interval of data
                    if PLoop==2
                        for o=1:SegmentsOfData.get.Count
                            WaveformData = SegmentsOfData.Item(o).Waveform(4, 1, 1e8, 1); %Collect Data
                            position=length(WaveformData)*(o-1);
                                for p=1:length(WaveformData)
                                    Voltages_cal(test_gat).Voltage_HF(p+position,1)=WaveformData(1,p);
                                end
                        end
                    end
                    if PLoop==4
                        for o=1:SegmentsOfData.get.Count
                            WaveformData = SegmentsOfData.Item(o).Waveform(4, 1, 1e8, 1); %Collect Data
                            position=length(WaveformData)*(o-1);
                                for p=1:length(WaveformData)
                                    Currents_cal(test_gat).Current_HF(p+position,1)=WaveformData(1,p);
                                end
                        end
                    end
                end
            end
    else
         Voltages_cal(test_gat).Voltage_LF=[];
```

```
57          Currents_cal(test_gat).Current_LF=[];
58         Voltages_cal(test_gat).Voltage_HF=[];
59         Currents_cal(test_gat).Current_HF=[];
60     end
61 end
62 clc;
```

Prototype main routine

```
/* ============ Includes ============== = */

#include <stdio.h>                /*Std libraries*/
#include <stdlib.h>
#include <errno.h>
#include <alsa/asoundlib.h>
#include <iostream>
#include <fstream>
#include <unistd.h>

//#include "rt_nonfinite.h"          /*Matlab header*/
#include "classify1.h"
#include "feat_calc.h"
#include "sep_sweep.h"

//#include "gperftools/profiler.h"   /*Performance analiser*/

/* ============ ALSA Related Globals ============== = */
static const char *device = "plughw:1,0";
static snd_pcm_t *capture_handle;

/* hwparams and default settings */
#define HWPARAMS_FORMAT SND_PCM_FORMAT_S24_LE
#define HWPARAMS_CHANNELS 1
#define HWPARAMS_RATE 48000
#define HWPARAMS_PERIOD_FRAMES 48000

static struct {
        snd_pcm_format_t format;
        unsigned int channels;
        unsigned int rate;
        snd_pcm_uframes_t period_frames;
        snd_pcm_uframes_t buffer_frames;
} hwparams = {
        .format = HWPARAMS_FORMAT,.channels = HWPARAMS_CHANNELS,.rate =
        HWPARAMS_RATE,.period_frames = HWPARAMS_PERIOD_FRAMES };

/* ============ Globals ============== = */

#define BUFSIZE (48000)  /*Amount of samples to read*/
int const size_print = 512;  /*Samples to print for plotting*/
int buf[BUFSIZE*2];  /*Buffer to sample size*/
static double buf2[BUFSIZE * 2]; /*Buffer to separate sweep*/
static double sweep[40000]; /*sweep buffer*/
static double features[8]; /*features buffer*/

static int first_time = 1; /*first time execution flag*/
static int output;
static float sweep_print[size_print];

                /*Flags*/
static int secs1; /*iteration flag*/
static int secs2; /*flag of first sweep*/
//static int avails[60];

static int open_stream(snd_pcm_t **handle, const char *name, char* dir)
```

```
{
        snd_pcm_hw_params_t *hw_params;
        snd_pcm_sw_params_t *sw_params;
        //snd_pcm_uframes_t *period_size = &hwparams.period_frames;
        //snd_pcm_uframes_t buffer_size;

        const char *dirname = (strcmp(dir, "SND_PCM_STREAM_PLAYBACK") == 0) ? "PLAYBACK" : "CAPTURE";
        int err;

        /*Opening pcm*/
        if (strcmp(dir, "SND_PCM_STREAM_PLAYBACK") == 0) {
                if ((err = snd_pcm_open(handle, name, SND_PCM_STREAM_PLAYBACK, 0)) < 0) {
                        fprintf(stderr, "%s (%s): cannot open audio device (%s)\n",
                                name, dirname, snd_strerror(err));
                        return err;
                }
        }
        if (strcmp(dir, "SND_PCM_STREAM_CAPTURE") == 0) {
                if ((err = snd_pcm_open(handle, name, SND_PCM_STREAM_CAPTURE, 0)) < 0) {
                        fprintf(stderr, "%s (%s): cannot open audio device (%s)\n",
                                name, dirname, snd_strerror(err));
                        return err;
                }
        }

        /*Allocating pcm structure*/
        if ((err = snd_pcm_hw_params_malloc(&hw_params)) < 0) {
                fprintf(stderr, "%s (%s): cannot allocate hardware parameter structure(%s)\n",
                        name, dirname, snd_strerror(err));
                return err;
        }

        if ((err = snd_pcm_hw_params_any(*handle, hw_params)) < 0) {
                fprintf(stderr, "%s (%s): cannot initialize hardware parameter structure(%s)\n",
                        name, dirname, snd_strerror(err));
                return err;
        }

        if ((err = snd_pcm_hw_params_set_access(*handle, hw_params, SND_PCM_ACCESS_RW_INTERLEAVED)) < 0)
              {
                fprintf(stderr, "%s (%s): cannot set access type(%s)\n",
                        name, dirname, snd_strerror(err));
                return err;
        }

        if ((err = snd_pcm_hw_params_set_format(*handle, hw_params, hwparams.format)) < 0) {
                fprintf(stderr, "%s (%s): cannot set sample format(%s)\n",
                        name, dirname, snd_strerror(err));
                return err;
        }

        //if ((err = snd_pcm_hw_params_set_rate(*handle, hw_params, hwparams.rate, NULL)) < 0) {
        if ((err = snd_pcm_hw_params_set_rate(*handle, hw_params, hwparams.rate, 0)) < 0) {
            fprintf(stderr, "%s (%s): cannot set sample rate(%s)\n",
                        name, dirname, snd_strerror(err));
                return err;
        }
```

```
113
114         if ((err = snd_pcm_hw_params_set_channels(*handle, hw_params, hwparams.channels/*2*/)) < 0) {
115                 fprintf(stderr, "%s (%s): cannot set channel count(%s)\n",
116                         name, dirname, snd_strerror(err));
117                 return err;
118         }
119
120         if ((err = snd_pcm_hw_params_set_period_size(*handle, hw_params,
121                 hwparams.period_frames, 0)) < 0) {
122                 fprintf(stderr, "%s (%s): error in period assign(%s)\n",
123                         name, dirname, snd_strerror(err));
124                 return err;
125         }
126
127         if ((err = snd_pcm_hw_params(*handle, hw_params)) < 0) {
128                 fprintf(stderr, "%s (%s): cannot set parameters(%s)\n",
129                         name, dirname, snd_strerror(err));
130                 return err;
131         }
132
133         snd_pcm_hw_params_free(hw_params);
134
135         if ((err = snd_pcm_sw_params_malloc(&sw_params)) < 0) {
136                 fprintf(stderr, "%s (%s): cannot allocate software parameters structure(%s)\n",
137                         name, dirname, snd_strerror(err));
138                 return err;
139         }
140         if ((err = snd_pcm_sw_params_current(*handle, sw_params)) < 0) {
141                 fprintf(stderr, "%s (%s): cannot initialize software parameters structure(%s)\n",
142                         name, dirname, snd_strerror(err));
143                 return err;
144         }
145         if ((err = snd_pcm_sw_params_set_avail_min(*handle, sw_params, BUFSIZE)) < 0) {
146                 fprintf(stderr, "%s (%s): cannot set minimum available count(%s)\n",
147                         name, dirname, snd_strerror(err));
148                 return err;
149         }
150         if ((err = snd_pcm_sw_params_set_start_threshold(*handle, sw_params, 0U)) < 0) {
151                 fprintf(stderr, "%s (%s): cannot set start mode(%s)\n",
152                         name, dirname, snd_strerror(err));
153                 return err;
154         }
155         if ((err = snd_pcm_sw_params(*handle, sw_params)) < 0) {
156                 fprintf(stderr, "%s (%s): cannot set software parameters(%s)\n",
157                         name, dirname, snd_strerror(err));
158                 return err;
159         }
160
161         return 0;
162 }
163
164 static int sampling() {
165
166         int err;
167
168         if (first_time == 1) {
169                 if ((err = open_stream(&capture_handle, device, "SND_PCM_STREAM_CAPTURE")) < 0)
```

```
170                 {
171                         fprintf(stderr, "cannot open strem\n",
172                                 snd_strerror(err));
173                         return err;
174                 }
175
176                 if ((err = snd_pcm_start(capture_handle)) < 0) {
177                         fprintf(stderr, "cannot prepare audio interface for use(%s)\n",
178                                 snd_strerror(err));
179
180                         return err;
181                 }
182
183                         printf("%s", "PCM ready\n");
184
185         }
186
187         memset(buf, 0, sizeof(buf));
188
189         int count = 0;
190         while (count < 48000) {
191                 int avail;
192
193                 int err;
194
195                 if ((err = snd_pcm_wait(capture_handle, 500)) < 0) {
196                         fprintf(stderr, "poll failed(%s)\n", strerror(err));
197                         return err;
198                 }
199
200                 //avails[secs1]=snd_pcm_avail_update(capture_handle);
201
202                 while (avail < BUFSIZE) {
203                         avail = snd_pcm_avail_update(capture_handle);
204                 }
205
206                 snd_pcm_readi(capture_handle, buf, avail);
207
208                 for (int i = 0; i < 48000; i++) {
209                         buf2[i] = buf2[48000 + i];
210                 }
211
212                 for (int i = 0; i < 48000; i++) {
213                         buf2[48000+i] = buf[i];
214                         count += 1;
215                 }
216
217                 if (first_time == 1) {
218                         count = 0;
219                         first_time = 0;
220                 }
221
222                 if (avail > 48000) {
223                         printf("Warning: More samples in pcm buffer than allocated %u \n", (avail-48000)
                                ;
224                         //snd_pcm_drop(capture_handle);
225                         //return err=1;
```

```
226                         //return err=-1;
227                 }
228         }
229
230
231         //std::ofstream file;
232         //file.open("result2", std::ios::app);
233         ////for (int p = 0; p < 40000; p++) {
234         //file << avail << " ";
235         ////}
236         //file.close();
237         ////fflush(file);
238
239
240         //std::ofstream file;
241         //file.open("buf2", std::ios::app);
242         //for (int p = 0; p < 96000; p++) {
243         //file << buf2[p] << " ";
244         //}
245         //file.close();
246
247         //secs1++;
248
249         return err;
250
251 }
252
253 int main(int argc, char *argv[]){
254
255         printf("%s", "Starting... \n");
256
257         FILE * f_flags = fopen("flags", "w");
258         f_flags = fopen("flags", "w");
259         fprintf(f_flags, "data_ready=1\nfault_flag=0\n");
260         fclose(f_flags);
261
262         FILE * temp = fopen("result", "wb");
263         fwrite(sweep_print, sizeof(float), size_print, temp);
264         fclose(temp);
265
266         sleep(5);
267
268         int err;
269         if ((err = system("amixer -c 1 set Mic 0%")) == 1) {
270                 fprintf(stderr, "Fail at setting capture gain. Is sound card connected?");
271                 system("lsusb -v 2>&1 1>/dev/null");
272         }
273
274         //ProfilerStart("classifier.prof");
275
276 //      for (int secs = 0; secs < 1; secs++) {
277 //      while(true){
278
279                 //for (int secs = 0; secs < 2; secs++) {
280                 while (true) {
281
282                         err = sampling();
```

```
283
284                     sep_sweep(buf2, sweep);
285
286                     /*Check sweep output*/
287                     //if (sweep[1] != 0) {
288                     //      if (secs2 == 0) {
289                     //              FILE * temp2 = fopen("result2", "w");
290                     //              for (int i = 0; i < 40000; i++) {
291                     //                      fprintf(temp2, "%f\n", sweep[i]);
292                     //              }
293                     //              fclose(temp2);
294                     //      }
295                     //      else {
296                     //              FILE * temp2 = fopen("result2", "a");
297                     //              for (int i = 0; i < 40000; i++) {
298                     //                      fprintf(temp2, "%f\n", sweep[i]);
299                     //                      fflush(temp2);
300                     //              }
301                     //      }
302                     //}
303
304                     if (sweep[1] != 0) {
305
306                             /*Check buf*/
307                             //if (secs1 == 0) {
308                             //      FILE * temp2 = fopen("result2", "w");
309                             //      for (int i = 0; i < 96000; i++) {
310                             //              fprintf(temp2, "%f\n", buf2[i]);
311                             //      }
312                             //      fclose(temp2);
313                             //}
314                             //else {
315                             //      FILE * temp2 = fopen("result2", "a");
316                             //      for (int i = 0; i < 96000; i++) {
317                             //              fprintf(temp2, "%f\n", buf2[i]);
318                             //              fflush(temp2);
319                             //      }
320                             //}
321
322                             feat_calc(sweep, features);
323
324                             output = classify1(features);
325
326                             /*Check output*/
327 /*                          if (sweep[1] != 0) {
328                                     if (secs2 == 0) {
329                                             FILE * temp2 = fopen("result2", "w");
330                                             fprintf(temp2, "%d\n", output);
331                                             fclose(temp2);
332                                     }
333                                     else {
334                                             FILE * temp2 = fopen("result2", "a");
335                                             fprintf(temp2, "%d\n", output);
336                                             fclose(temp2);
337
338                                     }
339                             }
```

```
340                                 secs2++;        */
341
342                                 /*Sweep_print to plot*/
343                                 FILE * f_flags = fopen("flags", "w");
344                                 fprintf(f_flags, "data_ready=0\nfault_flag=%d\n", output);
345                                 fflush(f_flags); /*fclose(f_flags);*/
346
347                                 for (int p = 0; p < size_print; p++) {
348                                         sweep_print[p] = sweep[p * (40000 / size_print)];
349                                 }
350
351                                 FILE * temp = fopen("result", "wb");
352                                 fwrite(sweep_print, sizeof(float), size_print, temp);
353                                 fclose(temp); /*fflush(f_flags);*/
354
355                                 f_flags = fopen("flags", "w");
356                                 fprintf(f_flags, "data_ready=1\nfault_flag=%d\n", output);
357                                 fclose(f_flags); /*fflush(f_flags);*/
358
359                         }
360
361                         secs1++;
362
363                         if (err < 0) {
364                                 printf("%s","Error in main\n");
365                                 //break;
366                         }
367
368             }
369
370                 //ProfilerStop();
371                 snd_pcm_close(capture_handle);
372                 //break;
373                 printf("%s", "Restarting program... \n");
374
375
376 //      }
377 }
```