AUTOMATED WAVELET-BASED FAULT DETECTION AND DIAGNOSIS FOR SMART DISTRIBUTION SYSTEMS AND MICROGRIDS

by

Tamer Sayed Abdelhamid Abdelgayed Sewilam

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in

The Faculty of Engineering and Applied Science

Electrical and Computer Engineering

University of Ontario Institute of Technology

August 2017

Abstract

The legacy electric power system is defined as a one-way power flow from a centralized power generation plant to customers (consumers). In the smart distribution systems, the customers are allowed to produce electricity through the distributed energy resources (e.g. solar photovoltaics), as well as to consume energy, hence, the smart distribution systems can be defined as a two-way power flow. The Micro-Grid system is defined as a part of the smart distribution system that may include distributed energy resources, energy storage systems and loads. In addition, the Micro-Grid system can operate in two modes, grid-connected or non-grid-connected (i.e., islanded mode). The protection of the Micro-Grid system represents one of the major operational challenges, in particular when considering the integration of distributed energy resources, which may result in different fault current levels, especially in islanding mode. However, the capability of protection system equipment to be more accurate and dependable for faults diagnostic in the Micro-Grid is considered a challenge until now.

In this thesis, an automated wavelet-based fault detection and diagnosis technique based on a combination of Wavelet Transform, Harmony Search Algorithm, and Machine Learning approaches is developed for fault diagnosing in the Micro-Grid systems. The harmony search algorithm as an optimization technique is used to identify the optimum wavelet function(s) and the optimum wavelet decomposition level(s) to extract the most prominent features that are hidden in the current/voltage waveforms when applying the discrete wavelet transform. This is unlike previous works in which only one arbitrary wavelet function is used based on a trial and error process.

In order to automate the fault classification process in Micro-Grid system, and to examine the effectiveness of the automated wavelet-based fault detection and diagnosis method against

other approaches, two machine learning techniques (i.e. Decision Tree as an eager learner, and K-Nearest Neighbor as a lazy learner) are used. The performance of the two classifiers is estimated using the Monte Carlo stratified cross validation method. The Consortium for Electric Reliability Technology Solutions Micro-Grid is used as a test-bed system after modelling in Power Systems Computer Aided Design/Electromagnetic Transient Direct Current software package. The study also takes into consideration different operating modes, different fault types, different fault resistances, and also different fault locations.

The results of implementing the proposed automated wavelet-based fault detection and diagnosis technique shows a significant improvement in the classification accuracy compared to other previous approaches reaching an overall accuracy of 95.63% in the Micro-Grid test-bed system. In addition, the proposed technique has been verified experimentally, and the results of the experimental set-up confirmed the validity/effectiveness of the proposed approach in real-time implementation.

Keywords: Micro-grid, distributed energy resources, fault diagnosis, fault protection, feature extraction, wavelet transforms, harmony search algorithm, and machine learning.

Acknowledgements

I would like to express my sincere gratitude to my research supervisor Dr. Walid Morsi Ibrahim, for the continuous support of my Doctoral study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would also like to express my sincere gratitude to my research co-supervisor Dr. Tarlochan Sidhu for his fruitful discussions and helpful guidance.

My appreciation and deep thanks to all my friends and colleagues in Dr. Walid's Smart Grid research group at UOIT.

Special thanks and love go to my mother, Fatma. Words cannot express how grateful I am to my mother, for all of the sacrifices that you've made on my life. Your prayer for me was what sustained me thus far. I would also like to thank my beloved wife Walaa and Thank you for supporting me for everything, and especially I can't thank you enough for encouraging me throughout this experience. To my beloved sons, Ziad, Seif, and Hamza, I would like to express my thanks for being such good boys always cheering me up

Table of Contents

List of Tables

Chapter 2

Chapter 3

Chapter 4

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Nomenclature

ANSI	American National Standards Institute
Cd_j	Approximation coefficient at level j
A	Approximation Decomposition Level
ANNs	Artificial Neural Networks
bw	Bandwidth
Bt.S	Battery Storage Source
bior	Biorthogonal Wavelet Family
Acc_K	Classification Accuracy
Coif	Coiflets Wavelet Family
CSV	Comma Separated Values
CI	Confidence Interval
CERTS	Consortium for Electric Reliability Technology Solutions
CWT	Continuous Wavelet Transform
Ir	Current Iteration
Db	Daubechies Wavelet Family
DT	Decision Tree
Ca_j	Detail coefficient at level j
D	Detail Decomposition Level
DFT	Discrete Fourier Transform
dmey	Discrete Meyer Wavelet Family
DWT	Discrete Wavelet Transform
DERs	Distributed Energy Resources

LL	Double-Line Fault
LL-G	Double-Line-to-Ground faults
Tr.	Electrical Power Transformer
EMTDC	Electromagnetic Transients including DC
g_{aj}	Energy of approximation coefficient at level *j*
g_{dj}	Energy of detail coefficient at level *j*
ED_t	Euclidean Distance at Fault *t*
D_t	Euclidean Distance Vector
FFT	Fast Fourier Transform
FCLs	Fault Current Limiters
FT	Fourier Transform
$Gini$	Gini Index
GC	Grid-Connected
HMC_r	Harmony Memory Considering Rate
H_s	Harmony Memory Size
HSA	Harmony Search Algorithm
f_{l_1}	High frequency filter
HT	Hilbert–Huang Transform
IM	Induction Motor
Acc_t	k-Fold Mean Classification Accuracy
K-NN	K-Nearest Neighbor
f_{l_0}	Low frequency filter
ML	Machine Learning

bw_{max}	Maximum Bandwidth
MIr	Maximum Iteration
PA_{rmax}	Maximum Pitch Adjusting Rate
y_{max}	Maximum value of decision variables
μ	Mean Measure
MU	Measurement Uncertainty
MGDS	Micro-Grid Distribution System
MGs	Micro-Grids
bw_{min}	Minimum Bandwidth
PA_{rmin}	Minimum Pitch Adjusting Rate
y_{min}	Minimum value of decision variables
δ	Monte Carlo Mean Classification Accuracy
MRA	Multi Resolution Analysis
y^{new}	New harmony vector
NGC	Non-Grid-Connected
NK_{miss}	Number of Incorrectly Predicted Cases
N_K	Number of testing Cases
PSO	Particle Swarm Optimization
Ph	Phase (A,B, and C)
A-G	Phase A to Ground Fault
AB	Phase A to Phase B Fault
AB-G	Phase A to Phase B to Ground Fault
AC	Phase A to Phase C Fault

AC-G	Phase A to Phase C to Ground Fault
B-G	Phase B to Ground Fault
BC	Phase B to Phase C Fault
BC-G	Phase B to Phase C to Ground Fault
C-G	Phase C to Ground Fault
PMUs	Phase Measurement Units
PV	Photovoltaic Sources
PA_r	Pitch Adjusting Rate
PCC	Point of Common Coupling
PQ	Power Quality
PSCAD	Power System Computer Aided Design
PNN	Probabilistic Neural Network
$p(\varepsilon\|\gamma)$	Proportional Frequency of class ε at a particular node γ
rand	Random Number
X	Signal Type (current or voltages) and represents the
LG	Single-Line-to-Ground Faults
SWT	Stationary Wavelet Transform
SS	Steady State
SVM	Support Vector Machine
Sym	Symlets Wavelet Family
ABC	Three-Lines Fault
ABC-G	Three-Lines-to-Ground Fault
N_y	Total number of decision variables

TLs	Transmission Lines
σ^2	Variance Measure
VSC	Voltage Source Converter
l_e	Wavelet coefficient length
WSE	Wavelet Singular Entropy
WT	Wavelet Transform
WVD	Wigner-Ville Distribution
y_w	Worst harmony vector

Chapter 1: Introduction

1.1 Background

1.1.1 Legacy Electric Power System

The legacy electric power system is defined as a one-way power flow from a centralized power generation plant to customers (consumers). The legacy electric power system can be separated into three subsystems; 1) the generation system, 2) the transmission system and 3) the distribution system. The transmission system consists of the transmission lines and the transmission substation, where the generation voltages are step-up and transmitted through the transmission lines to supply the distribution substations after the voltages step-down to the level of the distribution substation. The distribution system is used for the supplying the electric power to the end-user customers. Fig. 1.1 below illustrates the electric power system infrastructure [1].

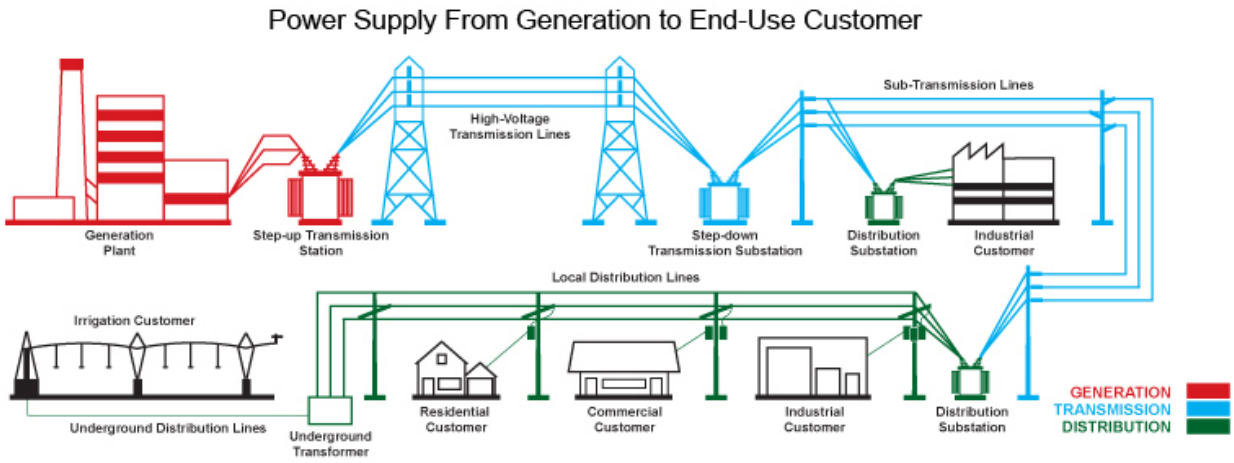


Fig. 1. 1: Typical electric power system [1]

The development of power system growth in terms of size and complexity is essentially required to match the dramatic increase demand and electricity usage. This increased complexity, however, comes at the expense of fault propagation within the power system. Different types of

faults such as the short circuit fault in the network will cause severe economic losses (with the reduction of equipment damage [2]) and compromise the reliability of the system.

The overhead lines in the distribution systems are highly dependent on their surrounding environment, such as large temperature variations, storms, extraordinary winds and other phenomena. High voltage and propagating waves are a result of lighting striking a transmission tower. This causes the deterioration of the isolators, which in return causes the protective relays to operate and interrupt the power flow [2]. In residential areas, the aerial transmission lines are employed to transfer energy to customers, these overhead transmission lines are prone to exposure to birds, animals, wind and trees. These overhead lines are a critical component, which helps ensure reliability of the distribution network.

1.1.2 Smart Distribution Systems

Typically, a power system is designed in a manner to generate electricity through a centralized power generation plant powered by various different fuel types, such as fossil fuel, nuclear and coal. The electricity generated is transferred to the loads in a unidirectional manner via the distribution system. Due to the nature of the fuel type, it is essential to ensure that the use and control of fuel are done in an efficient and optimal manner. Due to the growing concerns with regards to the environmental and economic impact of typical methods of power generation and growing demand, a more efficient and robust system needs to be developed. The system should be smarter in a sense that it allows the utility be more reliable, flexible and secure. The system should integrate communication ability between the various components of the grid, while concurrently allowing the extensive incorporation of renewable energy sources and energy storage devices. In order to address all the issues previously stated, a smart grid needs to be developed, which should enable a bi-directional flow of power in a dependable, efficient and reliable manner. The smart

grid should also maintain the capability of communicating information to the utility in order to improve the control, protection and as a result improve asset lifespan.

Recently, in the smart energy grid, the ability of constructing a modern power distribution system that uses the most advanced tools is considered as one of the most complex challenges. In general, in the smart distribution systems is considered as bidirectional (two-way) power flow as shown in Fig. 1.2, hence the customers are allowed to produce electricity through the distributed energy resources (e.g. solar photovoltaics), as well as consume it.

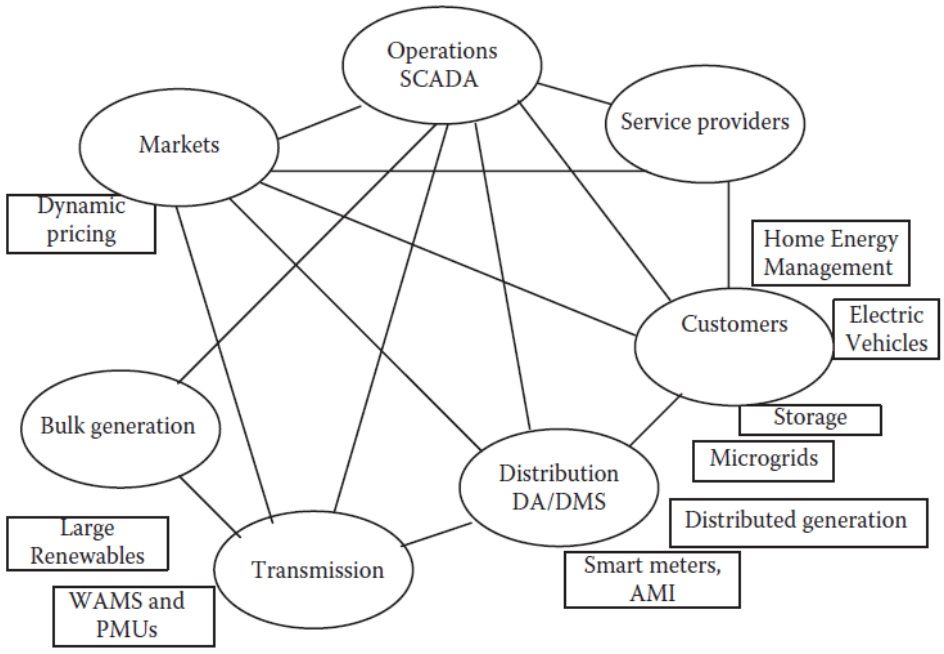


Fig. 1. 2: Structure diagram of smart distribution system [3]

1.1.3 Distribution Management Systems

The Distribution Management Systems (DMS) [3] as shown in Fig. 1.3 include the real-time applications of the Supervisory Control and Data Acquisition (SCADA) system, combined with the relevant functions like topology process and network reconfiguration. In addition, the data gathered from the Intelligent Electronic Devices (IEDs), as well as the network topology processor combined with the Geographical Information Systems (GISs) provide an integrated framework.

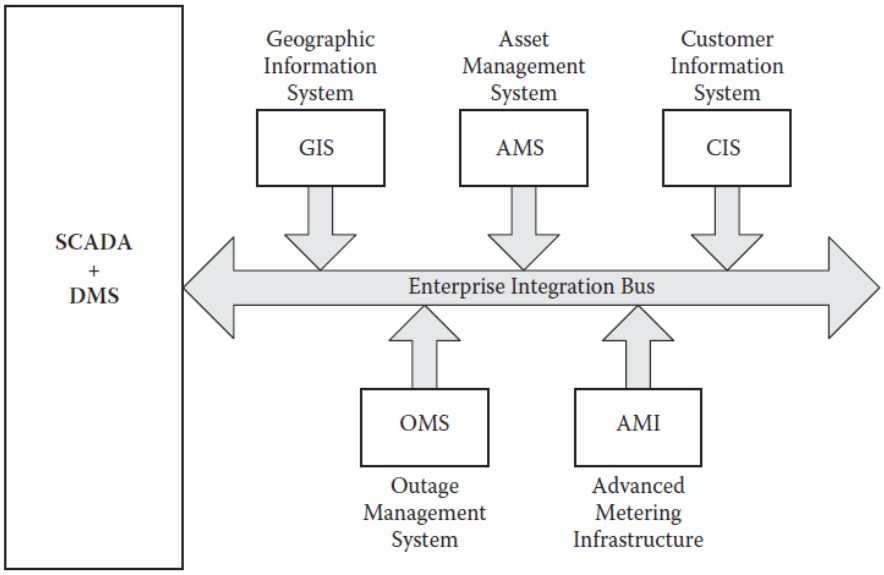


Fig. 1. 3: Structure diagram of distribution management system [3]

1.1.4 Micro-Grid Systems

Micro-Grid system [4] is defined as a part of the smart power distribution system that may include Distributed Energy Resources (DERs), as shown in Fig. 1.4, energy storage systems and loads, which can operate in two modes: 1) grid-connected or 2) non-grid-connected (i.e., islanded mode).

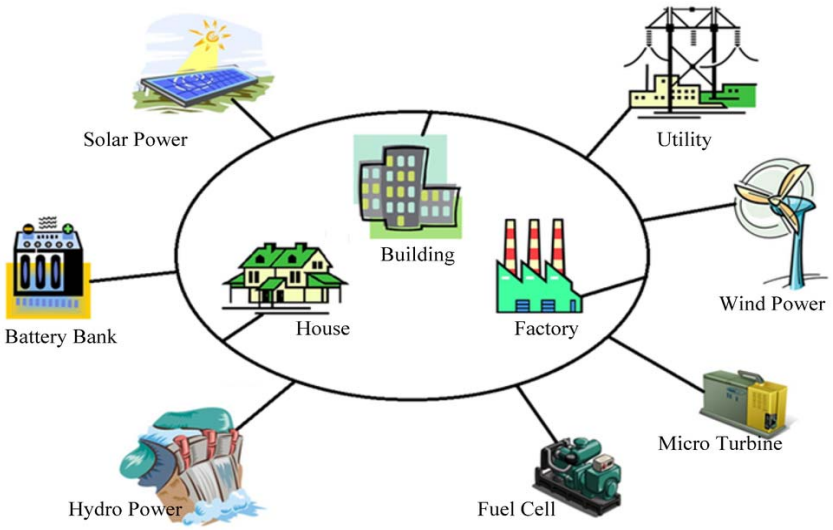


Fig. 1. 4: Structure diagram of Micro-Grid [4]

The aim of the Micro-Grid is to improve the system efficiency, power quality, and reliability [5] and [6]. According to [7]-[9], the Micro-Grid protection represents the main challenge facing its operation, in particular when considering the integration of distributed renewable energy resources (e.g. solar photovoltaic), which may result in bidirectional power flow and different fault current levels especially in islanding mode. This can disturb the operation of the protection relays and hence may result in numerous faults being misclassified, or may go undetected. In addition, in the case of islanded mode, the fault current contribution from DERs do not reach the pick-up level for the protection device to detect the fault. These differences may be considered as a challenging problem for any fault detection approach.

1.1.5 Power System Faults

Faults are mainly categorized according to the different fault types as shown in Fig.1.5 into unsymmetrical faults [10] and symmetrical faults. Examples of unsymmetrical faults are single-line-to-ground faults LG (i.e. A-G, B-G, C-G), double-line faults (i.e. AB, BC, AC), and double-line-to-ground faults (i.e. AB-G, BC-G, AC-G). On the other hand, the symmetrical faults are those faults that involve the three-phases such as three-line faults (i.e. ABC), and three-line-to-ground faults (i.e. ABC-G).

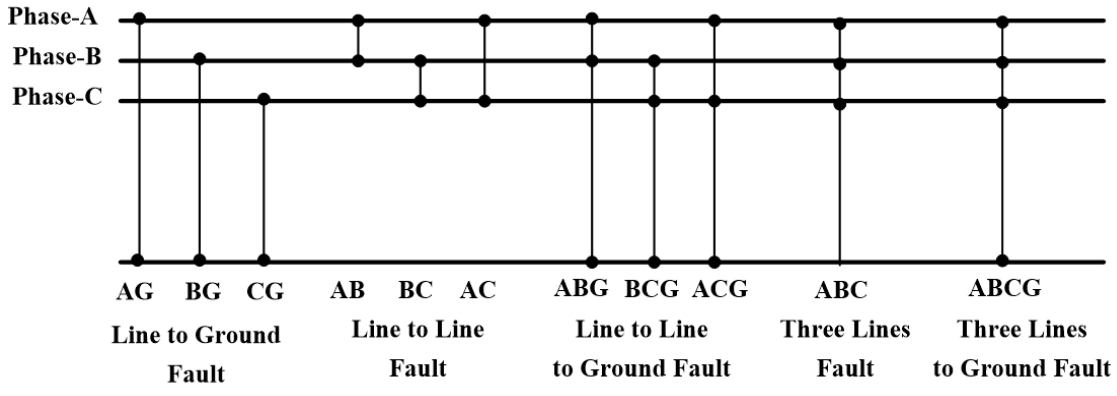


Fig. 1. 5: Power System Faults [3]

A protection system, which is capable of detecting, and classifying any type of fault in the power system is of great importance, which will prevent the equipment of the electric power system from the damage. Faults may cause a deviation of voltage or currents from the nominal values and states such as over/under voltage. This deviation may lead to a severe damage (e.g. winding insulation failure) for the most of electrical loads.

Unlike the transmission line protection system, the protection system of the smart distribution systems and Micro-Grids faces challenges especially with the addition of the distributed energy resources in the system. These challenges can be summarized as follows [4]:

- Bi-directional power flow.
- Fault current levels in the Micro-Grid system are extremely different in the grid-connected and islanded modes,
- In addition, most of the distributed energy resources are mainly connected to the grid through the Voltage Source Converters (VSCs), which mainly affect the fault current level.

1.2 Problem Statement and Motivation

Protection of the smart distribution system and Micro-Grids needs dependable and reliable detection, and classification of faults. Additionally, the separation of the faulty section only as fast as possible. However, most of the data recorded by the protection relays are not being used to its full extent, whereas these data are important to the operator for solving fault diagnosing issues (e.g. Fault detection and type classification). Providing the operators with relatively accurate and reliable knowledge of fault classification will serve restoration efforts and improve power outage restoration, power quality, reliability, and safety [3].

One of the main challenging issues in the smart distribution systems and the Micro-Grids is the protection system. Especially, with the integration of distributed renewable energy resources (e.g. solar photovoltaic and battery storage), which leads to a bidirectional power flow, as well as, different fault current levels with the different operating modes (i.e. Grid connected and Non-grid connected). Such issues may interrupt the operation of the protection relays and may result in several faults being misclassified or undetected. In addition, the fault current contribution from the distributed energy resources in the non-grid connected mode may not reach the pick-up level for the protection device to detect the fault. Consequently, all the aforementioned concerns are considered as challenging problems for any fault detection approach until now.

Recently, most of the research that relied on wavelet analysis used the wavelet coefficients and their energies as attributes to learn and build classifier model. This classifier model was then used for detection and classification of faults. However, the performance of the classifier model, which was based on wavelet analysis, was significantly affected by the choice of the mother wavelet.

The main motivation of this research work is to introduce an effective fault detection and classification approach, which could have a considerable impact in the smart distribution system and Micro-Grids system. However, a new combination of wavelet based on a systematic procedure is introduced, unlike the previous research, which relied on arbitrarily chosen wavelet functions. The new wavelet combination is used to solve the fault diagnosing issues in the smart distribution system and Micro-Grids system. The machine learning techniques are utilized to build the appropriate classifier model for the automation of fault classification procedure, where the prime aim of fault classifiers is to distinguish all fault types with the highest reliability and accuracy.

1.3 Contribution

The main contribution of this thesis can be listed as follows:

- The thesis proposes an Automated Wavelet-Based Fault Detection and Diagnosis (AWBFDD) technique, which is adaptable and flexible with different system topology and/or reconfiguration. In addition, the proposed AWBFDD technique is adaptable with consideration of challenging fault cases in the smart distribution system and Micro-Grids such as a bidirectional power flow, as well as, different fault current levels with the different operating modes (i.e. Grid connected and Non-grid connected).

- The thesis proposes a systematic methodology to identify the best suitable wavelet transform parameters settings. The problem of the wavelet transform parameters selection was framed as an optimization problem to find the best solution from all feasible solutions, where the optimizer (i.e. Harmony Search Algorithm) looks for the best wavelet functions combination and the most suitable wavelet decomposition level(s) among all feasible solutions (i.e. various wavelets and various wavelet decomposition levels). Unlike previous work in which the wavelet parameters were arbitrarily chosen based on a trial and error process, the work presented in this thesis uses a systematic approach that identifies the optimum wavelet function(s) and the optimum wavelet decomposition level(s).

- The thesis proposes the concept of optimal wavelet functions combination in which a combination of mother wavelets is used instead of using only one mother wavelet. The use of wavelet functions combination provides a methodical procedure based on

Harmony Search Algorithm (HSA) technique that determines the best combination of wavelet functions that are capable of extracting the hidden features in the transient signals needed for accurate fault detection and classification in the smart distribution system and Micro-Grids.

- The thesis further proposes the utilization of a combination of wavelet transform, harmony search algorithm, and machine learning approaches in the fault detection and diagnosis application. The role of the wavelet transform is to extract the signal attributes (i.e., features); the harmony search algorithm is used to recognize the best wavelet function(s) and the best wavelet decomposition level(s), while the machine learning classification techniques are applied to automate fault classification process.

- Finally, the thesis proposes the concept of the co-training of two Machine Learning classifiers (i.e. Decision Tree and K-Nearest Neighbor approach), which are used in this work for automating the fault classification process. Different test systems are used for assessing the performance of the proposed fault detection, classification technique considering various fault types, various fault resistances, various fault locations, and a different system topology/configuration.
- The results of this thesis have been published in the top-tier peer-reviewed journals [11- 13]. The first journal paper was entitled “A New Harmony Search Approach for Optimal Wavelets Applied to Fault Classification”. This paper presents a novel approach based on the Harmony Search Algorithm (HSA) to optimally determine suitable wavelet functions and wavelet decomposition levels for accurate fault classification in Transmission Lines (TLs) unlike previous works in which only one arbitrary wavelet function is used.

The second journal paper was entitled “A New Approach for Fault Classification in Microgrids Using Optimal Wavelet Functions Matching Pursuit”. This paper introduces a new approach that uses a combination of wavelet functions and Machine Learning (ML) for fault classification in MicroGrids (MGs). Particle Swarm Optimization is applied to identify the optimal wavelet functions combination that serves as a matching pursuit to extract the most prominent features.

The third journal paper was entitled “Fault Detection and Classification based on Co-Training of Semi-Supervised Machine Learning”. This paper presents a semi-supervised machine learning approach based on co-training of two classifiers for fault classification in both the transmission and the distribution systems with consideration of microgrids. The performance of the proposed method was examined on both transmission and distribution test systems in a simulation, and experimental hardware environment.

1.4 Thesis Organization

The work of this dissertation is organized as follows:

Chapter 2 surveys the different solution methods for the fault diagnosis problem in the previous work including the two analysis methods used, i.e., steady-state analysis based and transient analysis based techniques. The benefits and drawbacks of all pervious methods are outlined. The research gaps of the pervious methods are mentioned, and finally the objectives of proposed research are presented.

Chapter 3 presents the basics of wavelet transform analysis including the discrete wavelet analysis and the wavelet functions. The proposed automated wavelet-based fault detection and diagnosis method for the selection of the wavelet transform parameters (i.e. analysis signals,

wavelet functions, wavelet decomposition levels) are introduced, and the representation of signal's feature vector is presented. In addition, this chapter introduces the different machine learning techniques (i.e. Decision Tree as an eager learner and K-Nearest Neighbor as a lazy learner), which are used to automate the fault classification process. Furthermore, the classification accuracy evaluation method is presented.

Chapter 4 is dedicated to results evaluation, the test system (i.e. transmission and distribution system) are described in this chapter. The generation of faulty cases (with various fault conditions) which are used to test the proposed method is presented. The results of applying the proposed method on both systems are finally tested, evaluated, and compared using the two different classifiers. In addition, the real-time experimental set-up framework is presented in this chapter.

Chapter 5 introduces the main conclusion and recommendations regarding the proposed technique in this thesis, which has proven to be able to perform better than any other existing technique, and finally the future work is discussed.

Chapter 2: Literature Review

2.1 Introduction

The capability of protection system devices to be more accurate, reliable, flexible and dependable for fault diagnosis in the smart distribution system and Micro-Grids system was considered a challenge until now. This chapter presents a comprehensive review of the analysis approaches used in the fault diagnosis in the electrical power system. The chapter also summarizes the previous studies in fault diagnosis, which are grouped into two main categories: steady-state analysis based approaches and transient analysis based approaches. In addition, this chapter ends by identifying the research gaps in the previous studies and the objective of the proposed technique.

2.2 Previous Work on fault diagnosis

This section presents an overview of the work presented in the literature related to various approaches that were proposed for fault analysis in both transmission and distribution systems. The previous work addressing the problem of fault classification in electric power system may be categorized into the steady state analysis based approaches [14] – [22] and the transient analysis based approaches [23]–[41].

2.2.1 Steady-State analysis based approaches

The detection of faults in Micro-Grids relied on the sequence components of the current and voltage waveforms was studied by Casagrande et al. [14]. The study used the random forest technique as a classifier. The approach presented in [15] relied on the magnitude/angle of zero, positive sequence voltage, and an arbitrary threshold for fault classification in Micro-Grids. The analysis presented in [15] was based on the sequence component, which is normally implemented in the steady-state. In the case of a Micro-Grid, which includes Distributed Energy Resources (DERs), the generation unit (DERs) is considered as a part of the distribution system. Most of the

DERs are interfaced to the system through an inverter where in the case of a fault, the response of the generator may completely change, and hence transient analysis is necessary. It is significant to note that in the case of steady-state analysis, most of the signal features in the transient are hidden. These features are very helpful to distinguish between the different fault types in the Micro-Grid system. Furthermore, double phase and three phase faults are not considered in [15]. The coordination of the overcurrent protection relays in different approaches for fault detection in the Micro-Grid system was presented in [16]-[19].

Fault classification in the transmission lines based on the sequence components of currents and voltage signals were studied in [20]–[22]. Rahmati et al., [20] used a criterion index to classify the double-phase-to-ground and single-phase-to-ground fault. The study relied on the values of zero and the negative sequences of the current and voltage waveforms. The classification of faults (i.e. single line to ground (LG) and double line (LL)) in the Transmission Lines (TLs), based on the sample shifting method for sequence component calculation of collected signals from Phase Measurement Units (PMUs), was studied by Samanta et al., [21]. In [22], K-Nearest Neighbor (K-NN) algorithm was used for fault classification in TLs using the fundamental and zero sequence current signals to determine the fault occurrence.

Fault classification approaches, which relied on the sequence components analysis, are commonly completed in steady-state domain. As a result, a portion of features was lost which it at the same time obtained only during the transient analysis. Moreover, the features might help in the detection of the wide diversity of fault types.

2.2.2 Transient Analysis based approaches

2.2.2.1 Discrete Fourier Transform

Fault classification approaches, which relied on the use of Fourier Transform (FT) to catch signals' attributes, were introduced in [23] and [24]. The approach presented in [23] relied on a combination of Fast Fourier Transform (FFT) and support vector machine to classify faults in TLs, whereas FFT was utilized to catch the signal's features from the Phasor Measurement Unit (PMU) data, which were then used as an input to SVM.

In [24], a combination between the Discrete Fourier Transform (DFT) and the Artificial Neural Networks (ANNs) were applied to classify the faults in TLs. Where the DFT was utilized as shown in Equation (2.1) to catch the fundamental component of the current and voltage waveforms. Zhang et al., [25] relied on mathematical morphology, singular value decomposition, and entropy theory to classify the faults in TLs. The study was based on four an arbitrary thresholds in the analysis.

$$\mathrm{DFT[u]} = \frac{1}{\mathrm{N}} \sum_{\mathrm{n=0}}^{\mathrm{N-1}} \mathrm{x[n]} e^{-\mathrm{i}2\pi \mathrm{un}/\mathrm{N}}, \quad \mathrm{u} = 0, \ldots, \mathrm{N}-1 \tag{2. 1}$$

Where $x[n]$ is the time domain representation, N is the waveform length, u and n are the frequency and time variable, respectively.

The utilization of Fourier Transform (FT) offers the features of the signals in the frequency domain only and consequently the features in the time domain is missing. For the sake of extracting the features of the signals in the frequency and time domain together, the wavelet transform as time-frequency transform approach was used in the fault classifications application.

2.2.2.2 Discrete Wavelet Transform

In [26], wavelet was used for fault classification in TLs, and, specifically, third order Daubechies (db3) was used as the mother wavelet. The study relied on the information contained in the first decomposition level to estimate the fault type using an arbitrary threshold. In [27], discrete wavelet transform was utilized as shown in Equation (2.2) for fault detection and classification in TLs using three levels of decomposition while second order Symlets (sym2) was used as the mother wavelet for the analysis.

$$DWT[a,b] = \frac{1}{|a|^{1/2}} \sum_{n=-\infty}^{\infty} x[n]\, \varphi[\frac{n-b}{a}] \qquad (2.2)$$

Where φ is the shifted and scaled version of the used mother wavelet, a and b are the scale and the shift factor, respectively.

He et al., [28] used the wavelet singular entropy (WSE) technique to detect and classify the faults in TLs using (db4) as a mother wavelet and using an arbitrary threshold. Guillen et al., [29] used the Euclidean norm based on wavelet singular entropy to classify the fault in transmission line using (db4) and four wavelet decomposition levels. The detection and classification of fault in transmission line based on the Discrete Wavelet Transform (DWT) and Silva et al., in [30], the artificial neural network was studied by applying (db4) on the voltage and current signals. In [31], the values of wavelet coefficients (i.e. third level of decomposition and using db8), obtained using wavelet Multi Resolution Analysis (MRA) of the current signal in each phase, were used as indices for fault classification in TLs.

In [32], the features in the current signals are extracted using the MRA method and are used for fault classification. The study used the (db4) as the mother wavelet and using six wavelet decomposition levels. In [33], the probabilistic neural network (PNN) with the details coefficients

of wavelet transform was used to classify power system faults. The study used the coefficients of the details at the Fourth wavelet decomposition level using (Meyer) wavelet as an input to the PNN. In [34] the fuzzy logic based multi-criteria approach was used with wavelet transform to classify the faults in transmission line. The mother wavelet (db8) was applied on the current signals and the information in the third wavelet decomposition level was used to classify faults. The fuzzy classifier was used to detect the fault type (LG, LL, and LLG). In [35], wavelet transform, particle swarm optimization, and artificial neural network was used to classify the power system faults. The DWT was applied to the current signals using the (db1) wavelet with seven wavelet decomposition levels while the Particle Swarm Optimization (PSO) was used during the training stage of the ANN. Livani el al., [36] used a combination of wavelet transform and support vector machine classifier to classify power system faults. The DWT was applied to the three phase current and voltage signals using mother wavelet (db4). Four binary SVMs were used to classify the faults based on the energy of the wavelet transform coefficients.

In [37], the spectral energy of the wavelet transform coefficients and the Artificial Neural Network (ANN) were used for fault classification. The study used mother wavelet (db4) for DWT and then used ANN to classify the faults. Costa et al., [38] studied the effect of mother wavelet selection on the transmission lines fault detection and classification. The study concluded that (db1) mother wavelet was the most suitable candidate because it was able to provide high-energy values of wavelet coefficients in case of an AG fault. Geethanjali et al., [39] used a combination of DWT and ANN to detect and classify TL fault. The details and approximation coefficients of DWT of the current signals with mother wavelet (db2) along with five levels were used as inputs to ANN to classify the fault. Jana et al., [40] used the wavelet transform with mother wavelet (db4)

to classify the faults. The detail coefficients of eight decomposition levels were selected based on their energy for single-line-to-ground fault.

In [41], Mishra et al., used a combination of discrete wavelet transform, decision tree, and random forest classifiers for fault classification in Micro-Grids. An arbitrarily chosen wavelet was used in [41], namely first order Daubechies (db1), and the third decomposition level was chosen by trial and error to perform wavelet analysis. The study did not provide proper justification or methodical steps to demonstrate the choice of the wavelet function and/or the decomposition levels. The classification accuracy presented in [41] was obtained using a random test group and therefore these accuracies may be over/under estimated. In [41], the single phase to ground faults (i.e. AG, BG, and CG) were considered as one type of fault. This was the same in the case of double-phase and double-phase-to-ground faults. Hence, it was very difficult to individually distinguish between all faults. The work in [42]-[49] showed a systematic procedure to design the wavelets and it was applied to islanding detection of distributed generation in [42] and [43], and in Non-Intrusive Load Monitoring (NILM) in [44] - [49].

It is worth noting that the previous work in [26]-[41] relied on various wavelet functions in the faults classification application in the electric power system. Nevertheless, there is no clear procedures to support the choice of the wavelet functions used. Table 2.1 outlines the used current and/or voltage signals, the used decomposition levels, and the used wavelet functions in the previous works of the fault classification in the electric power system. Fig. 2.1 shows the percentage occurrence of wavelet functions utilized in the fault classification in [26]-[41]. The visual inspection of Fig. 2.1 revealed that the mother wavelet function that was commonly used in the literature was Daubechies-4 (db4) but without a clear systematic process. Finally, it can concluded that, the main limitation of the previous studies is the absence of a well-defined process

for the choice of the most appropriate wavelet function(s) in the classification of fault in the electric power system. In addition, the choice of the most suitable of decomposition levels' number is not mentioned in the previous works, which might considerably influence in the analysis.

Table 2. 1: Wavelet transform analysis based approaches

Ref.	Current signal	Voltage signal	Wavelet Transform	
			Decomposition Level	Function
[26]	✓	×	A1	db3
[27]	✓	×	D1,A1,A2,A3	sym2
[28]	✓	✓	D1,D2,D3,D4	db4
[29]	✓	×	D1,D2,D3,D4	db4
[30]	✓	×	D1	db4
[31]	✓	×	D3	db8
[32]	✓	×	D6	db4
[33]	×	✓	D4	Meyer
[34]	✓	×	A3	db8
[35]	✓	×	D7	db1
[36]	✓	✓	D1	db4
[37]	✓	×	D1,D2,D3,D4,D5 D6,D7,D8,D9,D10	db4
[38]	✓	✓	D1	db1
[39]	✓	×	D1,D2,D3,D4,D5	db2
[40]	✓	×	D8	db4
[41]	✓	✓	D3	Db1

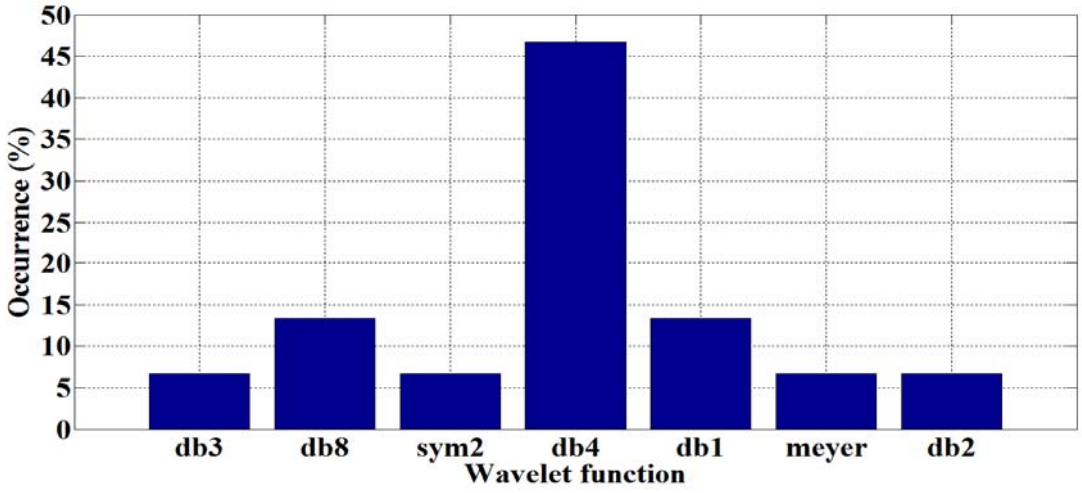


Fig. 2. 1: Occurrence percentage of the used wavelet functions in fault classification application in [26]-[41]

2.3 Research Gaps

The research gaps in the literature can be summarized as follows:

- The lack of the adaptability and flexibility of the existing techniques with the consideration of challenging fault cases in the smart distribution system and Micro-Grids.
- The lack of a well-defined process regarding the choice of the most appropriate wavelet function(s) in the classification of fault in the smart distribution system and Micro-Grids. In addition, the appropriate number of wavelet decomposition levels has not been carefully addressed in the previous works.

2.4 Research Objectives

- This work aims to introduce a systematic methodology in order to determine the most suitable wavelet function(s) and the appropriate number of wavelet decomposition level(s) that hold the prominent features needed for fault detection and classification in smart distribution system and Micro-Grids with considering the challenging fault cases, such as the bidirectional power flow and different fault current levels.

- In addition, this work seeks to introduce an automated wavelet-based fault detection and diagnosis technique to be adaptable and flexible with the different system topology or reconfiguration in the smart distribution system and Micro-Grids.

2.5 Summary

This chapter presented a comprehensive literature study of the previously published work on fault diagnosis on the smart distribution system and Micro-Grids system. In the case of the steady-state analysis based approaches, it was concluded that, a portion of features were lost, which were obtained only during the transient analysis. Furthermore, these features might help in the detection of the wide diversity of fault types. On the other hand, in the case of the transient analysis based approaches, the Fourier transform was investigated which offer the features of the signals in the frequency domain only, and consequently the features in the time domain is missing. The wavelet transform as time-frequency transform approach was used in the fault classifications application in order to extracting the features of the signals in the frequency and time domain together.

Finally, in the case of wavelet transform analysis based approaches, it can be concluded that the main limitation is the absence of a well-defined process for the choice of the most appropriate wavelet function(s) in the classification of fault in the electric power system. In addition, the choice of the most suitable of decomposition levels' number is not clear in previous works, which might considerably influence in the analysis. The chapter concludes by identifying the research gaps (e.g. the absence of systematic way in the choice of the wavelet functions and level) and proposes a set of objectives to fill these gaps. The next chapter explains the details of the methodology required to fill the research gaps, as well as achieve the research objectives.

Chapter 3: Methodology

3.1 Introduction

This chapter presents a detailed description of the proposed Automated Wavelet-Based Fault Detection and Diagnosis (AWBFDD) approach. The chapter starts by giving an overview of the discrete wavelet transform, the harmony search algorithm, and the machine learning techniques that represent the backbone of the proposed AWBFDD approach for fault classification in this work. The wavelet transform represents the signal-processing tool that was applied to capture the features contained in the signal. The harmony search algorithm was implemented to identify the most suitable wavelet function(s) and the optimal number of wavelet decomposition level(s) needed to extract the hidden features in the analysis signal when applying the wavelet transform analysis. The machine learning classifier is used to automate the fault classification process by considering a lazy and an eager machine learner. Furthermore, this chapter presents the mathematical background of the measures used for assessing the performance of the classifiers, which are used in this work to evaluate the performance of the proposed AWBFDD approach.

3.2 Wavelet Transform

The Fourier transform has been widely used as the main signal analysis technique that can provide an amplitude-frequency domain representation. However, the main limitation with the Fourier transform is the lack of time-domain related information. To overcome this limitation, the Short Time Fourier Transform (STFT) and the Wavelet Transform (WT) have emerged as time-frequency transforms which can provide both time and frequency information of the analysis signal. The STFT is defined as:

$$STFT[e,g] = X[k] = \sum_{n=-\infty}^{\infty} x[n]g[n-e]e^{-j\omega n} \quad (3.1)$$

Where, x is a time domain signal, X is the transformed signal in the frequency domain, g is the window function, e is the index to define the size of the fixed window function, and ω is the angular frequency. The expression in (3.2) is for the WT, where x[t] is the target signal, w[t] is the chosen wavelet and υ and φ are the scale and shift parameters, respectively.

$$X[\varphi,\upsilon] = \frac{1}{\sqrt{\upsilon}} \sum_{t=-\infty}^{\infty} x[t]w\left[\frac{t-\varphi}{\upsilon}\right] \quad (3.2)$$

Fig. 3.1 exemplifies the comparison between the domain representation of the Fourier transform, short time Fourier transform, and wavelet transform. It can observed from Fig. 3.1 that, the short time Fourier transform suffer the fixed size window subsequently any effort to increase the frequency resolution comes at the expense of reducing the time resolution and vise-versa. On the other hand, the wavelet transform uses a variable size window, which allows for a good time and frequency resolution in the analysis with the comparison to the short time Fourier transform.

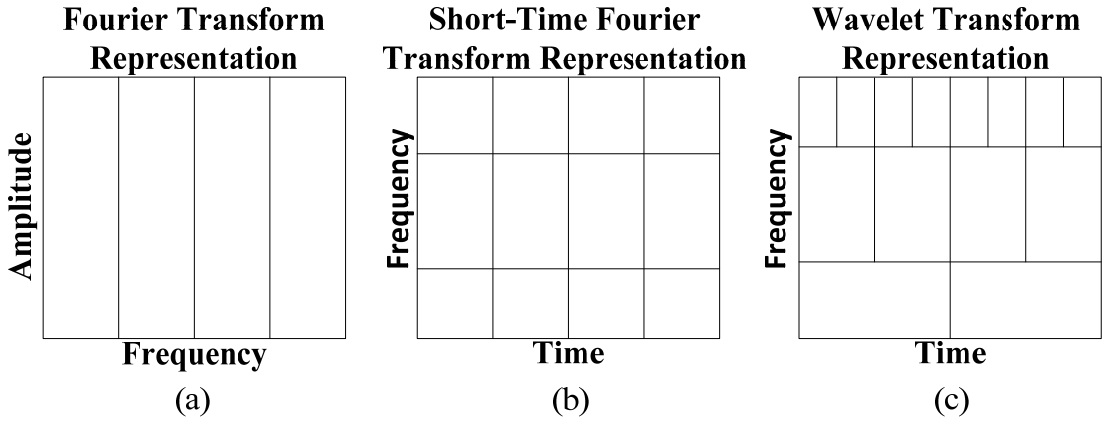


Fig. 3. 1: Time-Frequency domain representation

3.2.1 Hilbert–Huang Transform

The Hilbert–Huang transform [50] (HHT) relies on the instantaneous frequency computation, which results from the Hilbert transform of the analyzed signal. In general, the Hilbert transform for the $x(n)$ signal is defined as follows [50]:

$$HHT[x(n)] = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(n)}{t-n}dn \tag{3.3}$$

The Hilbert–Huang transform has significant limitations in terms of generating undesired components in the low-frequency band, and reducing the ability to distinct some low energy components of the analyzed signal hence they might not seem in the spectrum.

3.2.2 Wigner-Ville distribution

The Wigner-Ville distribution [51] (WVD) is a joint time-frequency analysis for non-stationary signals. The main difficulty with the WVD is its bilinear characteristic, which leads to cross terms in the time-frequency domain. The WVD for the $x(t)$ signal is defined as follows [51]:

$$WVD_X(t,w) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t+\frac{\tau}{2})\, x^*\,(t-\frac{\tau}{2})e^{-j\omega\tau}d\tau \tag{3.4}$$

Where x^* is the conjugate of x. Hence, the Wigner integral is the Fourier transform, with respect to the delay variable τ, of $x(t-\frac{\tau}{2})\, x^*\,(t-\frac{\tau}{2})$, which defines the instantaneous correlation.

3.2.3 Stationary Wavelet Transform

The Stationary Wavelet Transform [52] (SWT) of the translated signal is the same as the translated version of the transformed signal. Consequently, the SWT has the time-invariance property, which can be restored by up sampling the filters coefficients. The resulting algorithm is of the redundant type. The approximation and detail SWT coefficients can be expressed as follows [52]:

$$SCa_j(e) = \sum_m f_{l_0}(m) SCa_{j-1}(e + 2^{j-1}m) \tag{3.5}$$

$$SCd_j(e) = \sum_m f_{l_1}(m) SCa_{j-1}(e + 2^{j-1}m) \tag{3.6}$$

Where SCa_j and SCd_j are the SWT approximation and detail coefficients respectively, f_{l_0} and f_{l_1} are the low and high pass filter , respectively and *m* is any arbitrary integer.

3.2.4 Continuous Wavelet Transform

In wavelet transform analysis, the wavelet coefficients are used as a measure of how close the analysis signal and the mother wavelet are by using a scaled and translated versions of the mother wavelet. According to [53], the Continuous Wavelet Transform (CWT) is defined as:

$$CWT(a,b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(n) \varphi(\frac{n-b}{a})\ dn \tag{3.7}$$

Where, φ is the shifted and scaled version of the used mother wavelet, *a* and *b* are the scale and the time shift indices, respectively.

3.2.5 Discrete Wavelet Transform

In the Discrete Wavelet Transform (DWT), the signals' features are extracted in both time and frequency domain using the Multi-Resolution Analysis (MRA) [54]. At each wavelet decomposition level *j*, the discrete-time signal is decomposed into approximation wavelet coefficients Ca_j (low frequency), and detail wavelet coefficients Cd_j (high frequency). The approximation and the detail wavelet coefficients can be computed using the following:

$$Ca_j(e) = \sum_{l_e} f_{l_0}(l_e - 2e) Ca_{j-1}(l_e) \tag{3.8}$$

$$Cd_j(e) = \sum_{l_e} f_{l_1}(l_e - 2e) Ca_{j-1}(l_e) \tag{3.9}$$

Where, f_{l_0} is the low pass filter and f_{l_1} is the high pass filter. The energy of the wavelet coefficients at the details g_{dj} and the approximations g_{aj} of the j^{th} decomposition level can be calculated using the following:

$$g_{aj} = \sum_{e} \left| Ca_j(e) \right|^2 \tag{3.10}$$

$$g_{dj} = \sum_{e} \left| Cd_j(e) \right|^2 \tag{3.11}$$

3.2.6 Comparison of Transform Analysis Techniques

The Discrete Wavelet Transform (DWT) offers dyadic (the positions and the scale were chosen based on powers of two) representation of the analyzed signal, which provides frequency sub-bands at different resolution. This is a great advantage over the continuous wavelet transform since with DWT multi-resolution analysis can be performed. Also, instead of computing the wavelet coefficients at every time index and at each scale as in the continuous wavelet transform, the discrete wavelet transform provides significant reduction in the computational complexity by just computing the wavelet coefficients at the frequency sub-bands. Therefore, the computational complexity for the DWT is only $O(n)$, where n is the data size, which is significantly less compared to the CWT and the stationary wavelet transform that are considered redundant transforms.

The DWT contains an extensive library of the wavelet basic functions, which makes this transform suitable for transient analysis and, hence, provides time-frequency spectrum at different resolutions. On the other hand, the Hilbert–Huang transform has significant limitations in terms of generating undesired components in the low-frequency band, and reducing the ability to distinct some low energy components of the analyzed signal which may not be seen in the spectrum. In

addition, other transforms such as the Wigner-Ville distribution and the short time Fourier transform suffer from the trade-off among the time and frequency resolution.

In this thesis the following four transforms; 1) Stationary Wavelet Transform (SWT), 2) Hilbert–Huang Transform (HHT), 3) Continuous Wavelet Transform (CWT), and 4) Wigner-Ville Distribution (WVD) have been implemented using MATALB on an Intel Core i7, 2.9 GHz with 16 GB RAM PC. Table 3.1 shows the comparison results between the five transforms (i.e. SWT, HHT, CWT, WVD and DWT) in terms of the computational time. The table shows the DWT outperforming the other techniques SWT, HHT, CWT, and WVD by providing the smallest computational time value. Therefore, in this study, the DWT is selected as the signal processing technique to solve the problem in hand and it is utilized to extract the prominent features in the analyzed signal.

Table 3. 1: Computational Complexity of transform analysis

Technique	SWT	DWT	HHT	CWT	WVD
Computational time (sec)	0.1954	0.0049	0.2410	0.2415	0.0807

3.2.7 Wavelet Functions

In the wavelet transform analysis, several mother wavelet functions with various characteristics, as outlined in Table 3.2, may be used. It is worth noting that, the selection of the wavelet function for the analysis strongly affects the values of the energy of the wavelet coefficients, which will have direct impact on the classification accuracy since it represents the feature vector.

Fig. 3.2 shows one sample wavelet function from each wavelet family. The visual inspection of the figure revealed the differences between each wavelet in terms of their respective wave-shapes and the discrete wavelet filter coefficients.

Table 3.2: Wavelet Family Properties

Wavelet Family	General form	Members	Orthogonal	Biorthogonal	Compact support	Filters length	Symmetry	Vanishing moments number
Daubechies	dbN	db1-db45	Yes	Yes	Yes	2N	Far from	N
Symlets	symN	sym2-sym31	Yes	Yes	Yes	2N	Near from	N
Coiflets	coifN	coif1-coif5	Yes	Yes	Yes	6N	Near from	2N
Biorthogonal	biorNr. Nd	bior1.1-bior6.8	No	Yes	Yes	max(2Nr, 2Nd)+2	Yes	Nr
Discrete Meyer	Dmey	1	Yes	Yes	Yes	102	Yes	-

N is the wavelet order, r for reconstruction, and d for decomposition

For the sake of comparison between different wavelet functions from different families, the discrete wavelet transform was applied using six mother wavelet functions: fifth order symlet (i.e. sym5); biorthogonal 1.5 (i.e. bior1.5); third order coiflet (i.e. coif3); fourth order Daubechies (i.e. db4); twentieth order Daubechies (i.e. db20); and discrete Meyer (i.e. dmey). The energies of the wavelet coefficients of four decomposition levels were calculated and were depicted in Fig. 3.3.

It can observed from Fig. 3.3 that there are differences in the magnitude of the energy of the wavelet coefficients in every decomposition level and for each wavelet function. The presence of such differences may be considered as an indication of the discrepancies between the analyzed signal and the mother wavelet functions wave-shapes. Subsequently, the energy of the wavelet coefficients may be used as an attribute for feature extraction in the classification process.

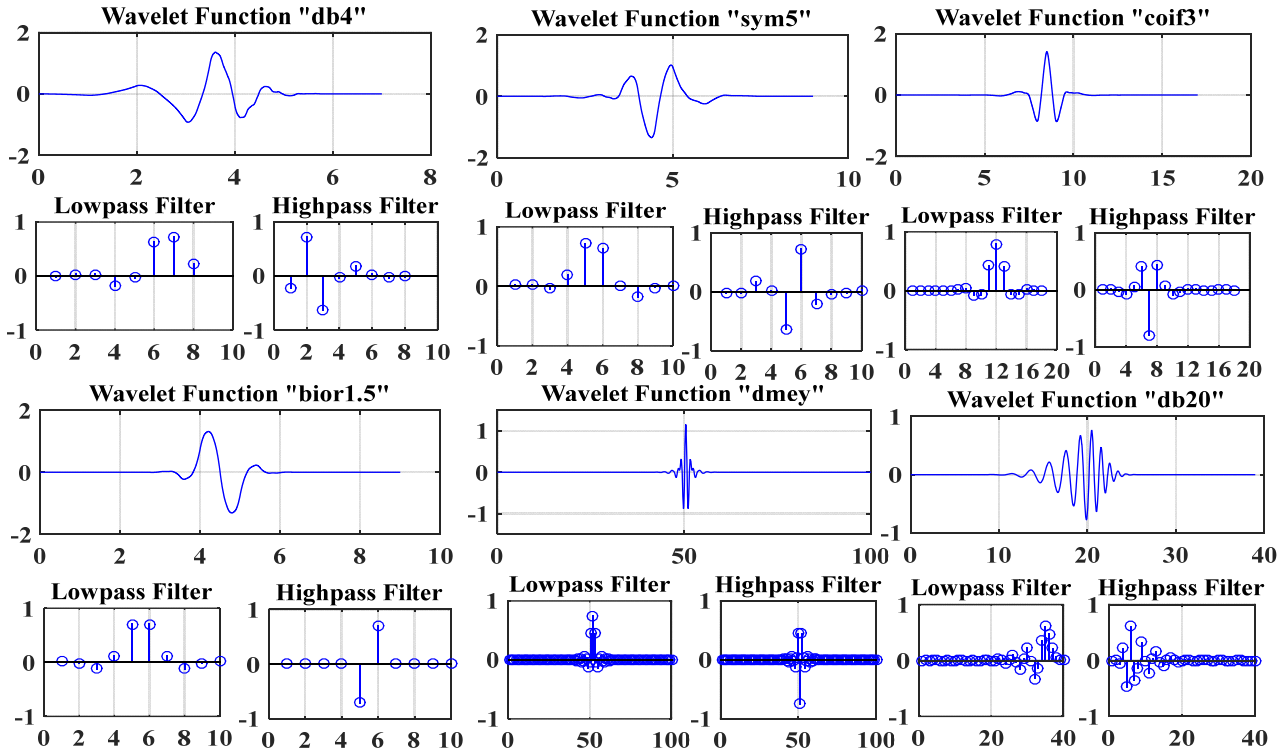


Fig. 3.2: Different wavelet functions and filters

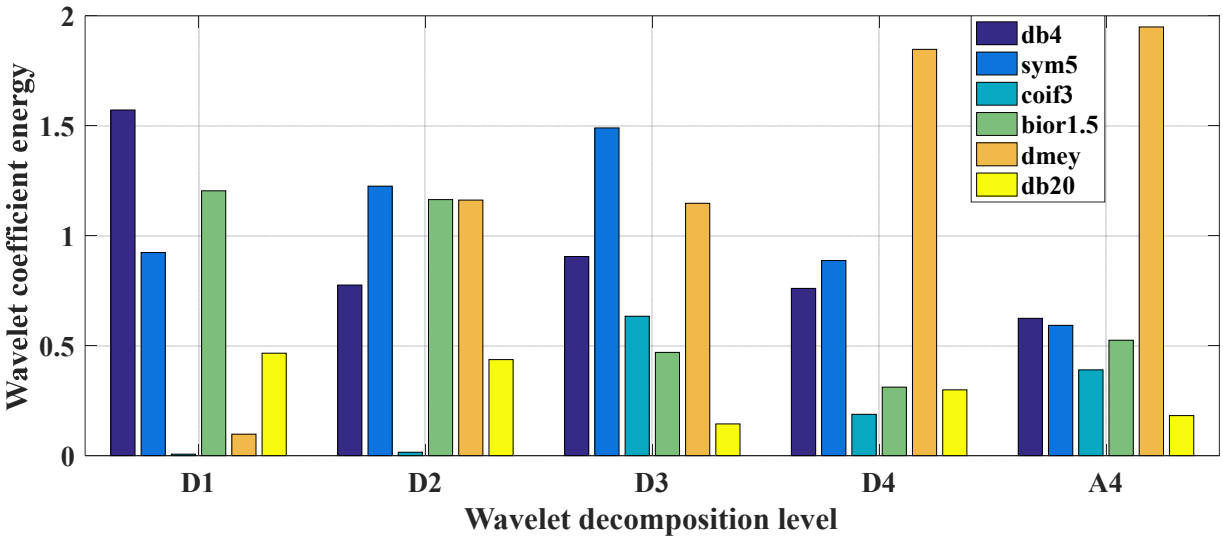


Fig. 3.3: Wavelet Coefficients energies at different functions and levels

The main goal of this work is to develop a systematic approach that determines the best combination of wavelet function(s) and the number of wavelet decomposition level(s) that are capable of extracting the hidden features in the transient signals needed for accurate fault diagnosis

in the smart distribution systems and Micro-Grids. In fact, the proposed approach in this work addresses the limitations of the previously published work, which relied on a trial and error process searching for the wavelet parameters that help extracting the features in the measured signals.

The literature has reported the use of different order of Daubechies wavelets (e.g. db1 [41], db2 in [39], and db4 [37]) but again without providing any justifications or presenting any systematic procedure. The analysis presented in [41] was based on the (db1) wavelet (i.e. haar) as a mother wavelet based on a visual observation made on just one fault type (i.e. single line to ground) and using only single resistive value (i.e. 2Ω) . There was no clear systematic approach to choose either the best wavelet(s) or the best wavelet decomposition levels. The authors in [41] stated the following:

"A comparison of the wavelet detailed coefficients of the fault and no-fault conditions are shown in Fig. 5. It is observed that the detailed coefficient at level-3 (d3) provides the most distinct deviation during fault condition."

By following the footsteps of the work in [41] through visually inspecting the detail decomposition level and computing the wavelet coefficients. The analysis was performed within this thesis using the same parameters chosen in [41] and the detail coefficients were calculated as shown in Figs. 3.4 – 3.11 considering the (db1) wavelet and four other wavelets (i.e. fourth order Symlets (sym4), Biorthogonal (bior1.3), and fourth order Daubechies (db4)). These wavelets were utilized to detect two different faults types: Three-line-to-ground fault (3L-G) and Double-line fault (DL).

From Figs. 3.4 – 3.11, it can be observed that there is a clear difficulty in choosing the wavelet function(s) and the number of decomposition level(s) through only visually inspecting the figures. The problem becomes more challenging when considering the changes due to the fault

type (unlike the work in [41], in which only AG fault was used to satisfy the choice of the wavelet and decomposition level). For example, according to Fig. 3.6, in which the wavelet coefficients were generated with the wavelet function (bior1.3) in the case of DL Fault, the distinct decomposition levels are levels D_1, D_2, and D_3 and not decomposition level-3 (D3) with wavelet (db1) as recommended in [41]. On the other hand, according to Fig. 3.10, in which the wavelet coefficients were generated with the same wavelet function (bior1.3) but with 3LG Fault, the distinct decomposition levels are levels only (D_1, and D_2). Following a similar approach to [41] and through visual inspection the figures of the wavelet coefficients. Table 3.3 summarizes the outcome of such examination showing that the detail level 1 (D1), (and not (D3) as reported in [41]) is the level holding some information in the case of DL and 3L-G faults.

Accordingly, the above discussion triggers the requirement for a methodical method that recognize the optimal wavelet function(s) and the levels of decomposition combination. Obviously, the approaches presented in the literature lack the adaptability and the flexibility in dealing with varying system conditions while the proposed approach provides flexibility and adaptability to these changes by applying the optimal wavelet function and levels.

Table 3.3: The distinct wavelet function and level

Fault Type	Decomposition level	Mother Wavelet
Double Line Fault	D1	db1 (haar)
	D1, D2, and D3	db4
	D2	bior1.3
	D1, D2, D3, and D4	sym4
Three Line to Ground Fault	D1 and D3	haar
	D1, D2, D3, and D5	db4
	D1 and D2	bior1.3
	D1, D2, D3, D4, and D6	sym4

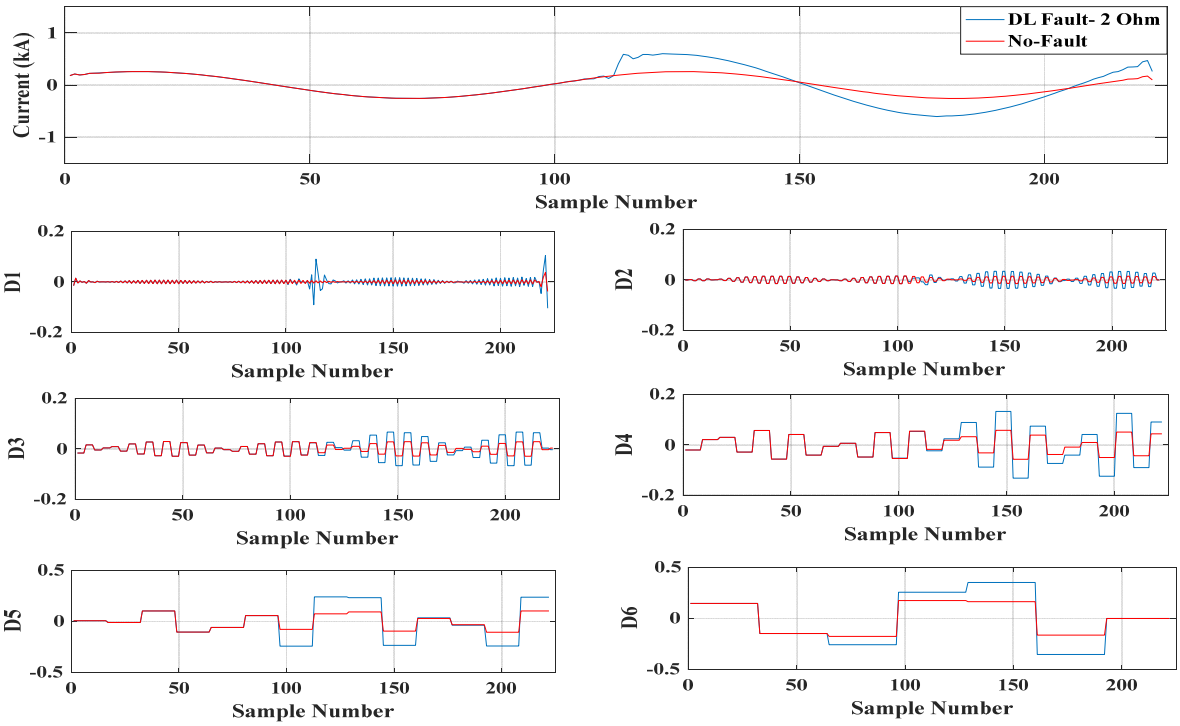


Fig. 3.4: Wavelet transform coefficients generated with (haar) Wavelet at DL Fault.

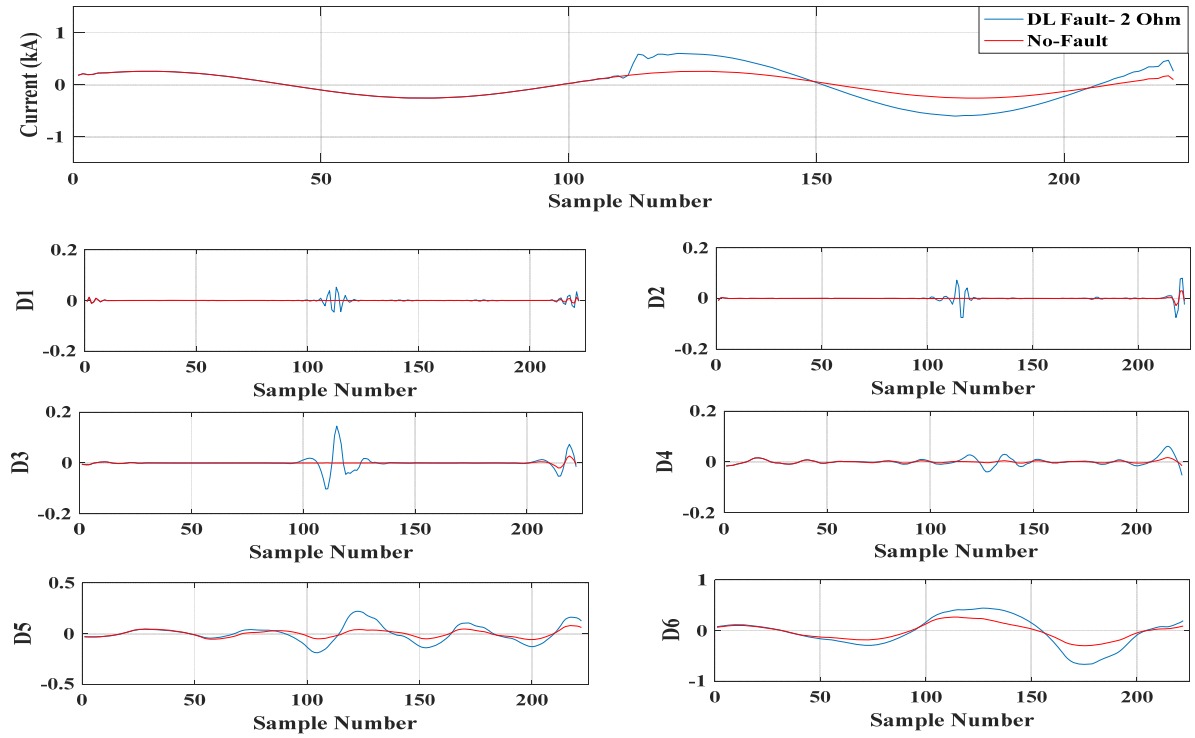


Fig. 3. 5: Wavelet transform coefficients generated with (db4) Wavelet at DL Fault.

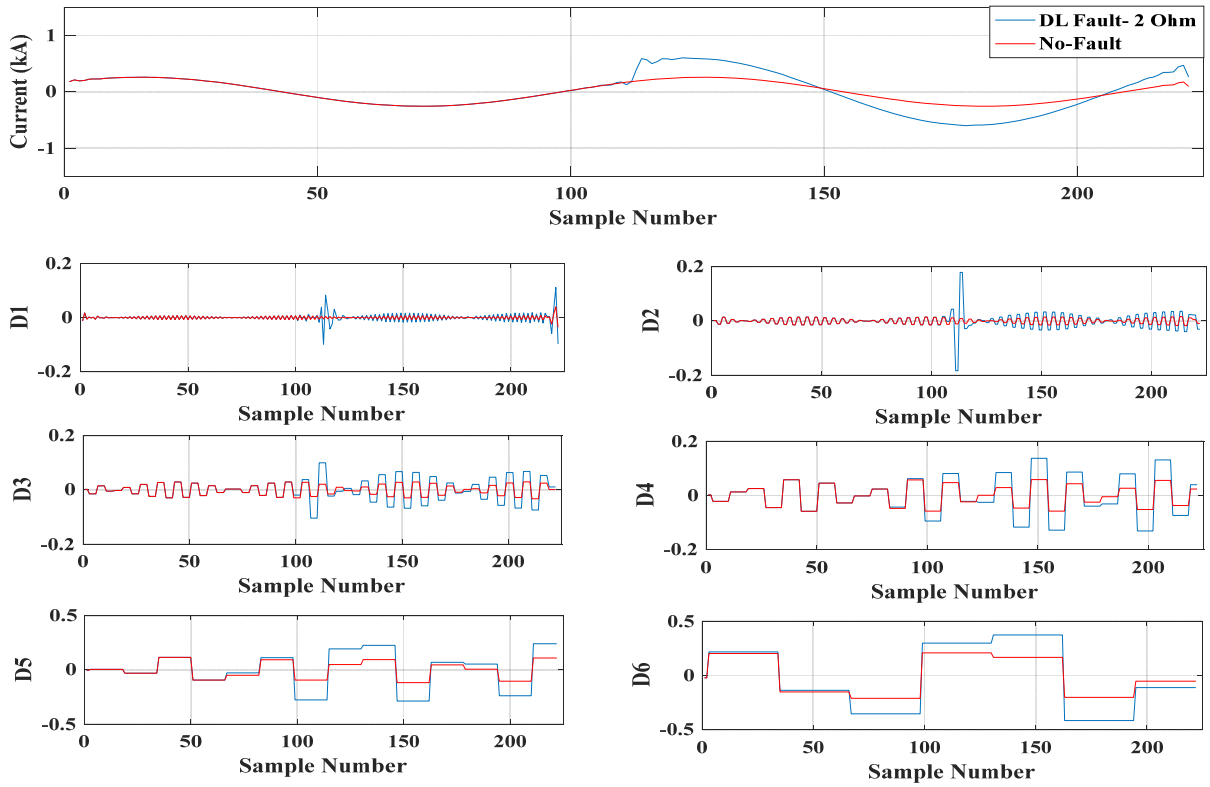


Fig. 3. 6: Wavelet transform coefficients generated with (bior1.3) Wavelet at DL Fault.

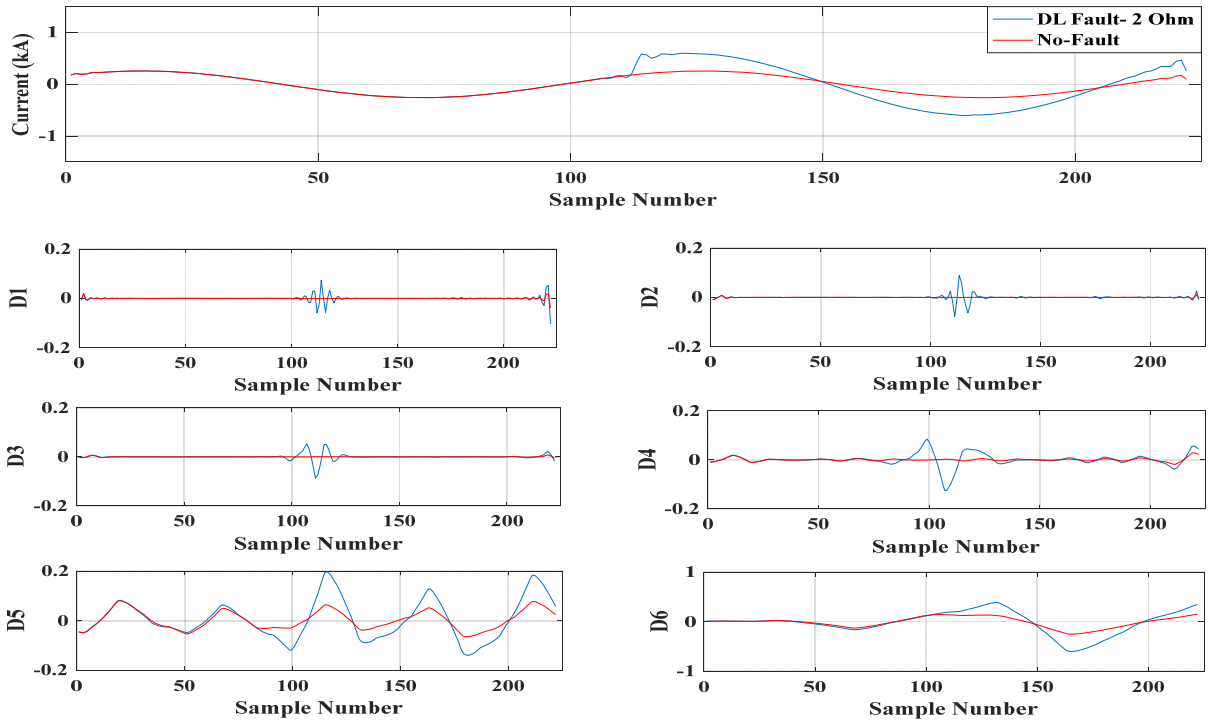


Fig. 3. 7: Wavelet transform coefficients generated with (sym4) Wavelet at DL Fault.

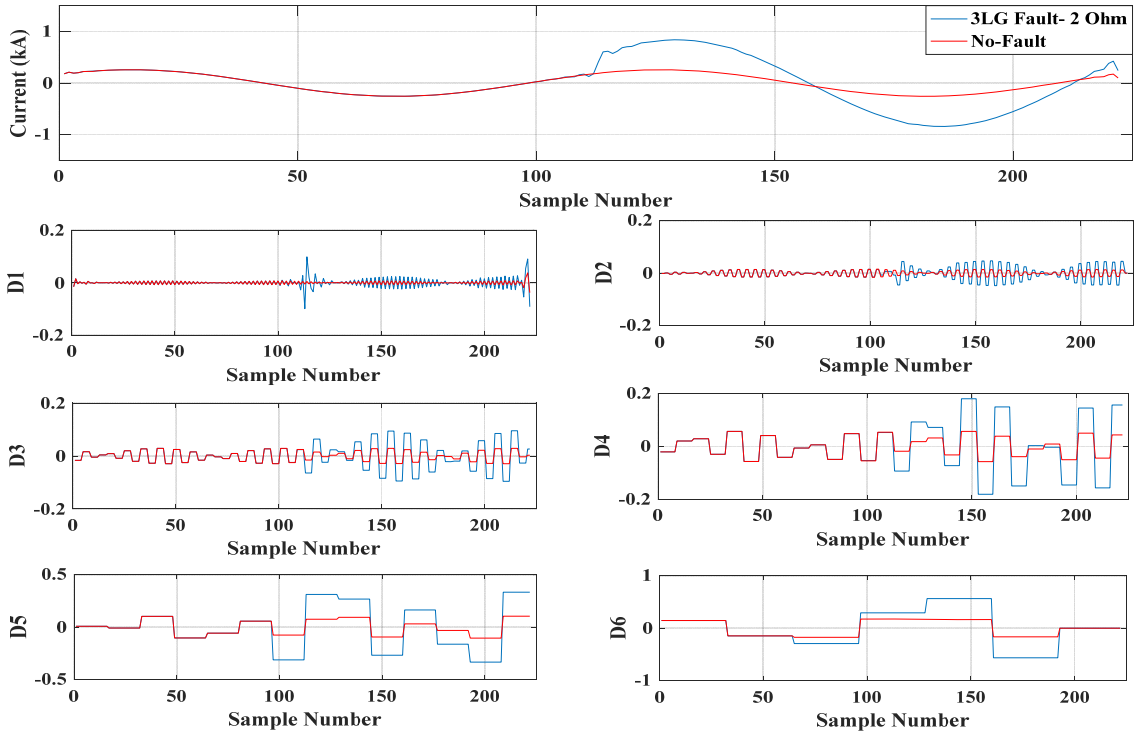


Fig. 3. 8: Wavelet transform coefficients generated with (haar) Wavelet at 3LG Fault.

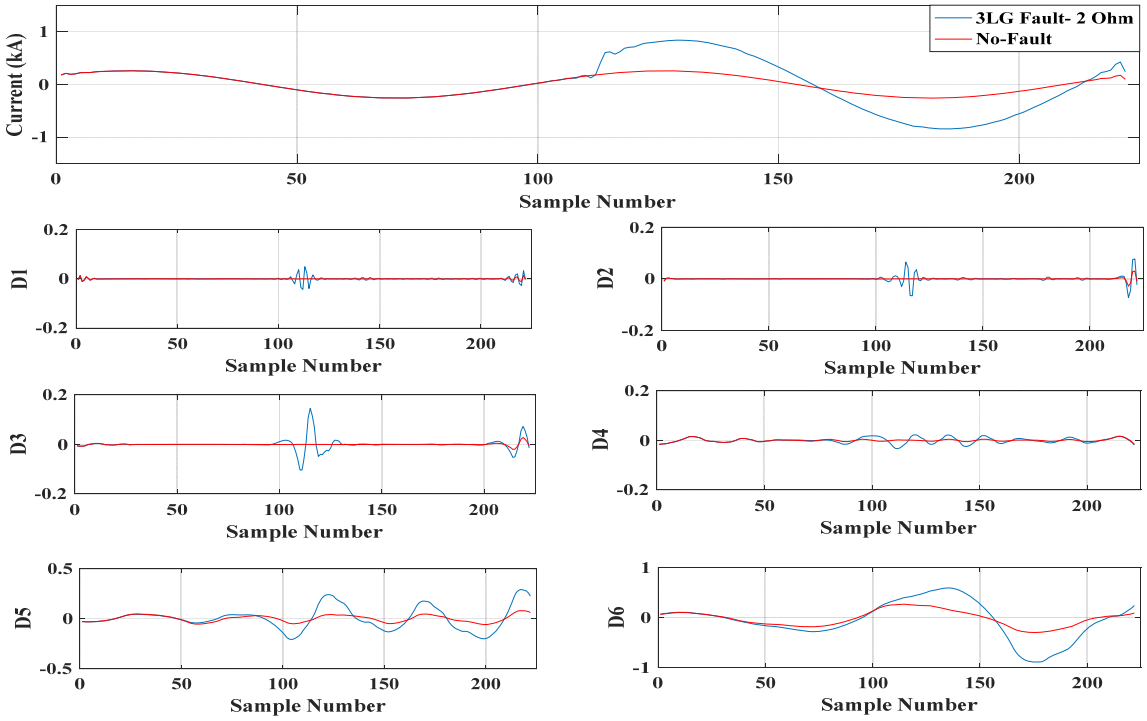


Fig. 3. 9: Wavelet transform coefficients generated with (db4) Wavelet at 3LG Fault.

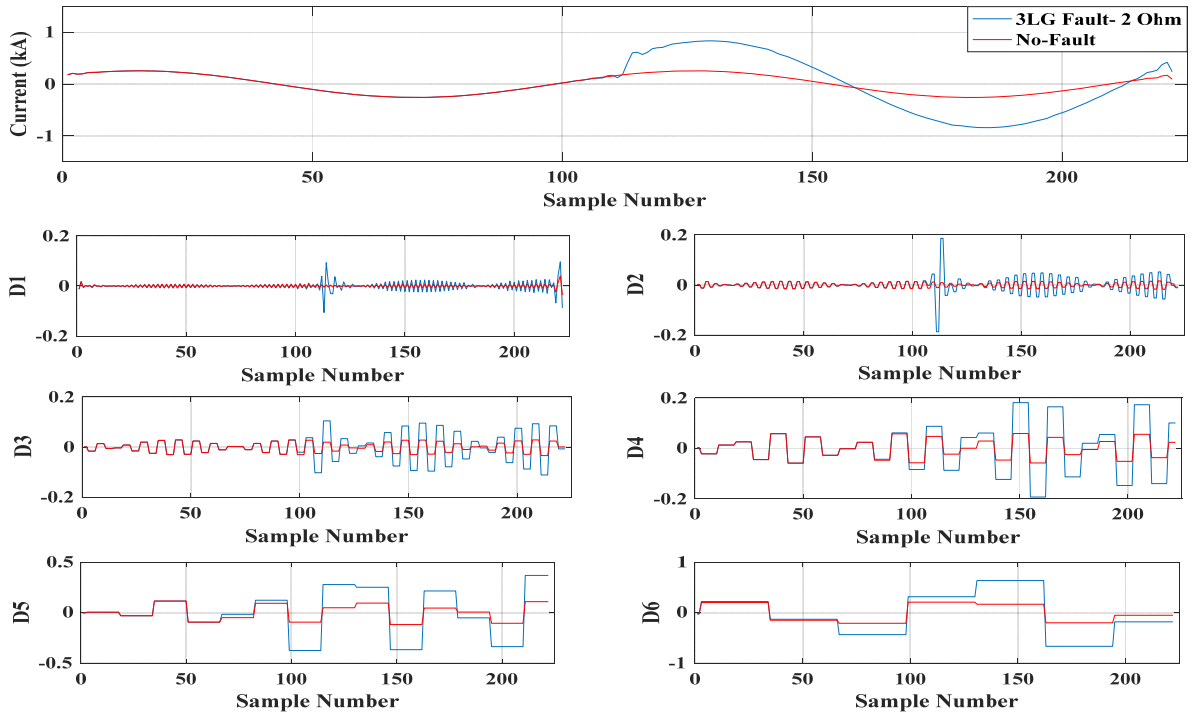


Fig. 3. 10: Wavelet transform coefficients generated with (bior1.3) Wavelet at 3LG Fault.

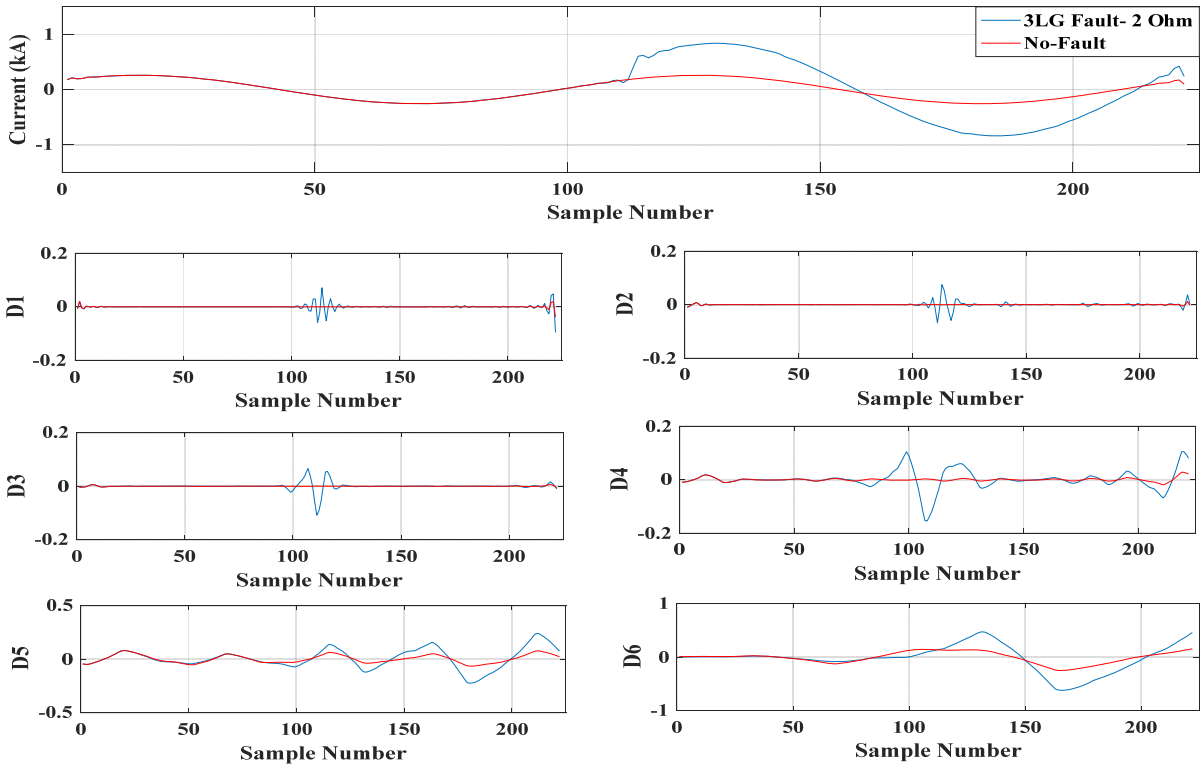


Fig. 3. 11: Wavelet transform coefficients generated with (sym4) Wavelet at 3LG Fault.

3.3 Proposed Automated Wavelet-Based Fault Detection and Diagnosis Technique

The proposed Automated Wavelet-Based Fault Detection and Diagnosis (AWBFDD) technique aims to find the optimum combination of mother wavelets and the number of wavelet decomposition levels that help extracting the most important attributes from the signal, which are needed for fault diagnosis in the smart distribution systems and Micro-Grids. Fig. 3.12 shows the hierarchical structure of the selection of wavelet transform parameters in which three main parameters need to be determined. The first parameter is the wavelet functions (i.e. one function or a combination of functions), the second parameter is the number of decomposition levels, including the approximations and the details, and the third parameter is the signal type, including both voltage and/or current signals.

3.3.1 Feature Vector Representation

3.3.1.1 Signal Pre-processing

The three-phase currents and voltages signals are sampled at a sampling rate of 64 samples per cycle. The corresponding sampling frequency is 3.84 kHz (64 samples/cycle × 60 cycles = 3.84 kHz). This sampling rate represents the typical sampling rate used in most digital protective relays as reported by [55]. In order to remove the steady state information from the signal and keep only the transient information following the fault inception, the difference between the samples in each two successive cycles as shown in Fig. 3.13 is calculated using the following:

$$x_{ph}^{D}(j) = x_{ph}(j+n) - x_{ph}(j) \quad j = 1,2,\dots,n \tag{3.12}$$

Where x is the type of signal (current or voltages) and ph represents the phase (phase A, phase B, and phase C).

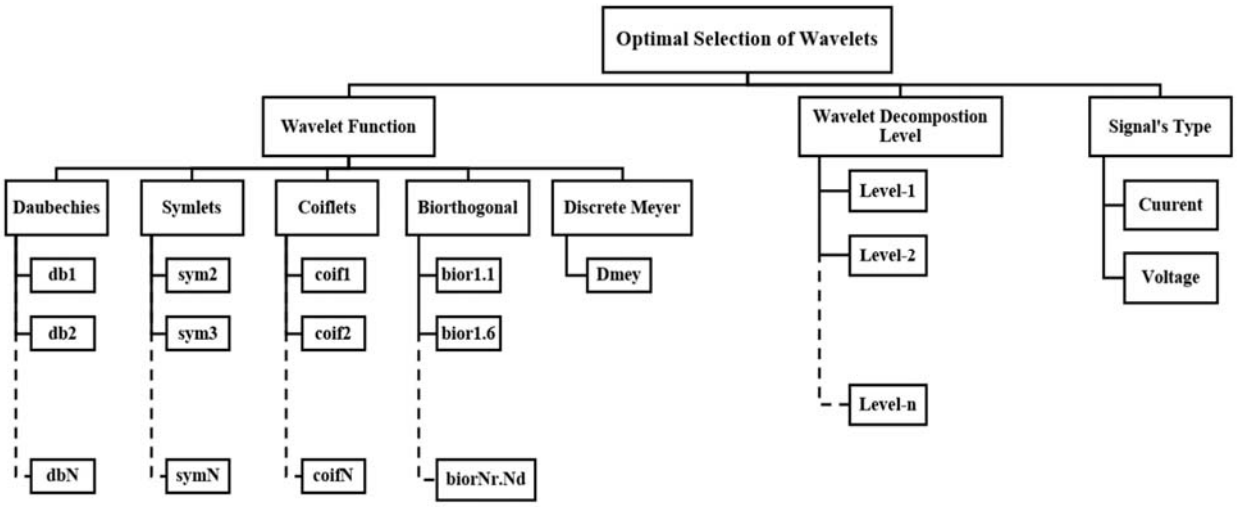


Fig. 3. 12: Wavelet parameters selection

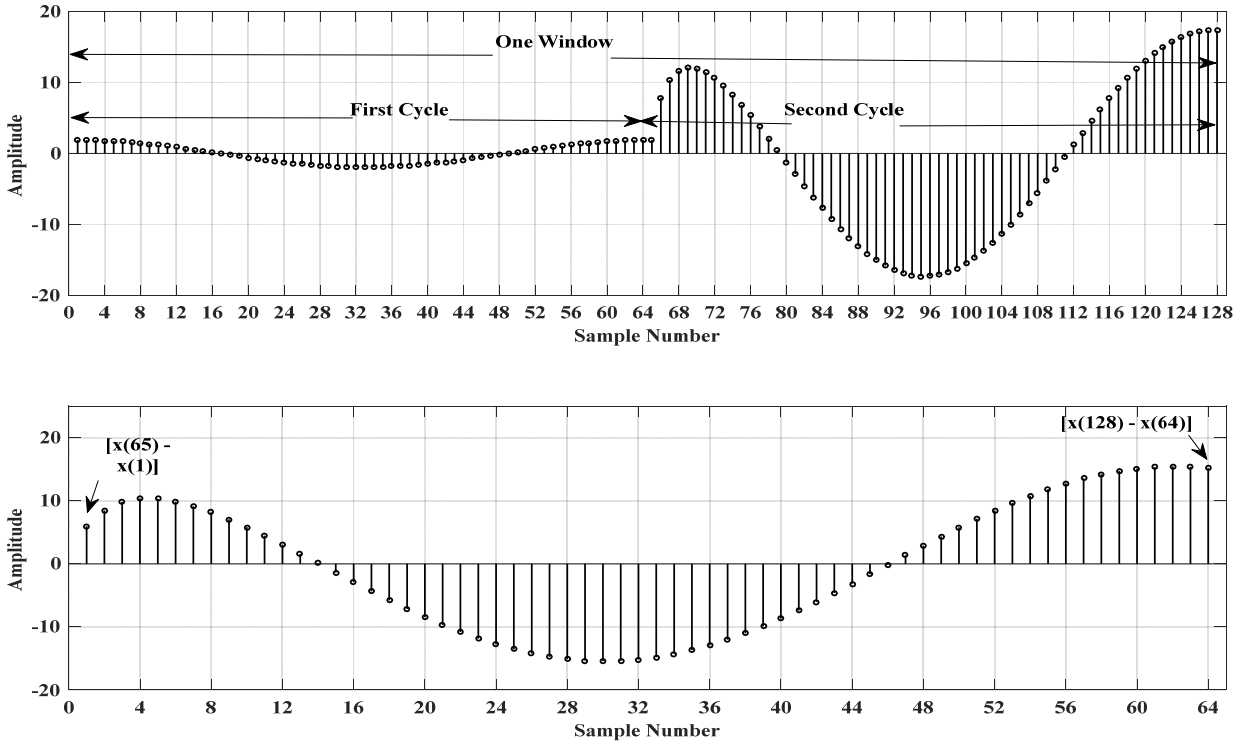


Fig. 3. 13: The analysis window of the sampled signal. (a) One window of two cycle. (b) The difference between two cycles

The discrete wavelet transform is applied to the sequences (x_{ph}^{D}) of the currents and voltages. Equations (3.8) - (3.11) are utilized to calculate the approximation and detail wavelet coefficients and their energies of the four wavelet decomposition levels (i.e. Cd_1, Cd_2, Cd_3, Cd_4, and Ca_4).

The number of decomposition levels is selected to be four to guarantee that the power system frequency (i.e., 60 Hz) is centered at the approximation level (i.e. Ca_4), which spans between 0 and 120 Hz as shown in Fig. 3.14.

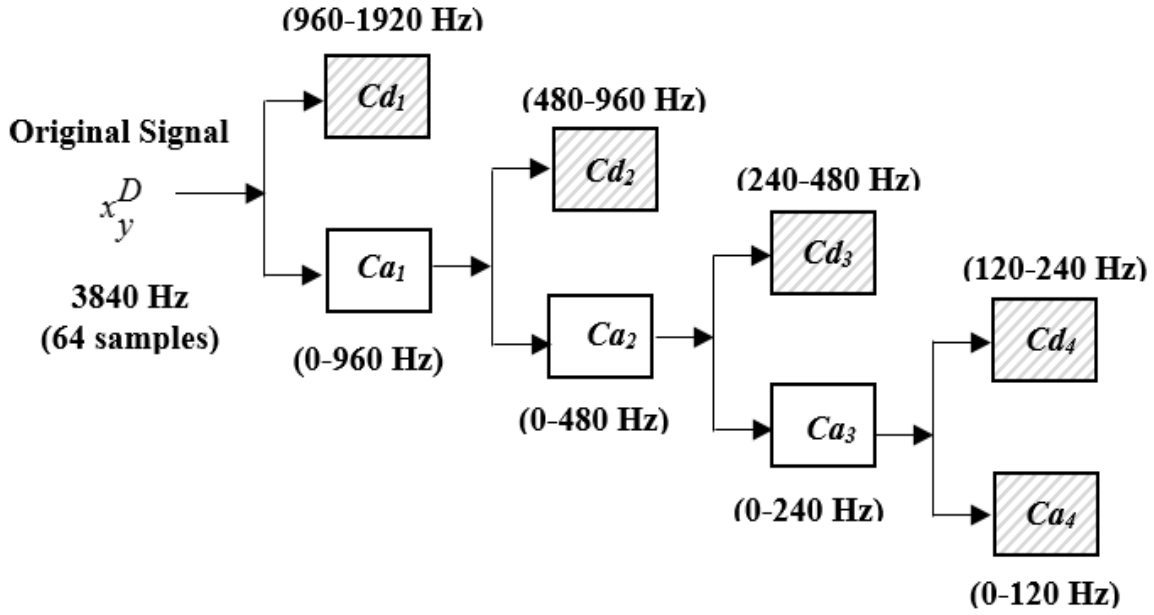


Fig. 3. 14: The levels of the wavelet decomposition of the main signal.

The outcome of the wavelet transform analysis is a vector consisting of the wavelet coefficients, which is then used to compute the energy of the wavelet coefficients of the details (g_{d1}- g_{d4}) and the approximation (g_{d4}) for each current or voltage sampled signals. The steps of implementing the DWT analysis, which are described earlier, are then repeated for various types of faults, as listed in Table 3.4, to calculate the energies of the wavelet coefficients for each fault type (i.e. E_{ag}, E_{bg}, etc.) and then sore them into one array W_c, which consists of the all fault types coefficients energies.

Table 3. 4: Fault type representation

Fault Type	Assigned Symbol
single-line-to-ground fault (LG)	AG BG CG
double-line fault (LL)	AB BC AC
double-line-to-ground fault LLG	ABG BCG ACG
three-lines fault (LLL)	ABC
three-lines-to-ground fault (LLLG)	ABCG

$$
\begin{aligned}
g_{Ia} &= [g_{d1_{Ia}} \quad g_{d2_{Ia}} \quad g_{d3_{Ia}} \quad g_{d4_{Ia}} \quad g_{a4_{Ia}}] \\
g_{Ib} &= [g_{d1_{Ib}} \quad g_{d2_{Ib}} \quad g_{d3_{Ib}} \quad g_{d4_{Ib}} \quad g_{a4_{Ib}}] \\
g_{Ic} &= [g_{d1_{Ic}} \quad g_{d2_{Ic}} \quad g_{d3_{Ic}} \quad g_{d4_{Ic}} \quad g_{a4_{Ic}}] \\
g_{Va} &= [g_{d1_{Va}} \quad g_{d2_{Va}} \quad g_{d3_{Va}} \quad g_{d4_{Va}} \quad g_{a4_{Va}}] \\
g_{Vb} &= [g_{d1_{Vb}} \quad g_{d2_{Vb}} \quad g_{d3_{Vb}} \quad g_{d4_{Vb}} \quad g_{a4_{Vb}}] \\
g_{Vc} &= [g_{d1_{Vc}} \quad g_{d2_{Vc}} \quad g_{d3_{Vc}} \quad g_{d4_{Vc}} \quad g_{a4_{Vc}}] \\
E_{ag} &= [g_{Ia} \quad g_{Ib} \quad g_{Ic} \quad g_{Va} \quad g_{Vb} \quad g_{Vc}] \\
W_c &= [E_{ag} \quad E_{bg} \quad E_{cg} \quad E_{ab} \quad E_{bc} \quad E_{ac} \\
&\qquad E_{abg} \quad E_{bcg} \quad E_{acg} \quad E_{abc} \quad E_{abcg}]^T
\end{aligned}
\tag{3.13}
$$

3.3.1.2 Normalization

The energy of the wavelet coefficients, which are described by the vector shown in Fig. 3.15, usually needs normalization. There is a need for normalization to ensure that, there is no unbalance in the values of the energy of the wavelet coefficients that may arise due to the large values of the approximation coefficients compared to those at the detail levels. In order to address this issue, the energies vector (E_{ag}) of the wavelet coefficients is normalized to get the normalized value (Z_{Eag}) of the energy of the wavelet coefficient vector, and the remaining elements in the energies vectors for all other fault types are computed in a similar way.

$$Z_{Eag}(k) = \frac{E_{ag}(k) - \mu(E_{ag})}{\sigma(E_{ag})} \tag{3.14}$$

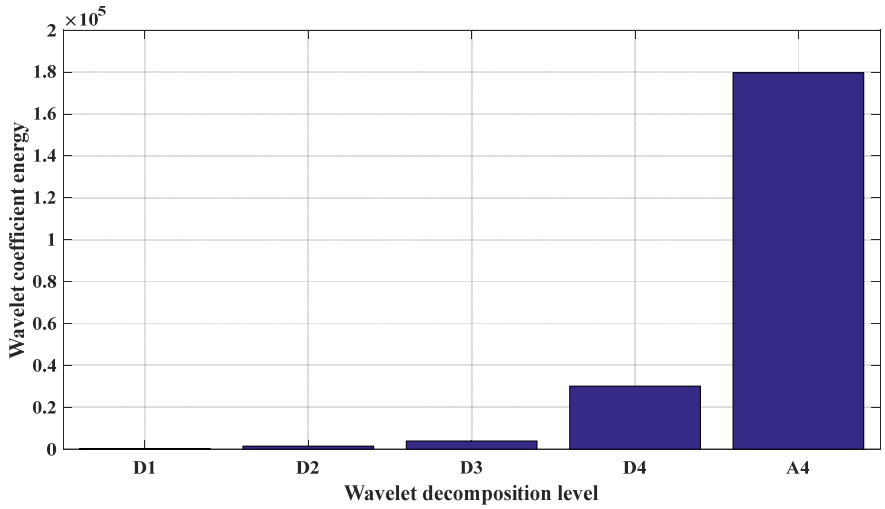


Fig. 3. 15: wavelet coefficients energies vector

3.3.1.3 Reference Vector

Each type of fault is represented by a vector, which contains the energy of the wavelet coefficients. In order to represent each fault by only one value, another vector was generated, which uses as a reference vector. Consequently, the distances between the vectors containing the

energy of the wavelet coefficients for all other fault types and this vector (reference vector) can be easily calculated.

In order to generate a reference vector, the DWT is applied to the three phase sampled currents and voltages signals at a non-fault (i.e. healthy) case and then the wavelet coefficients energies of that non-faulty case are sorted in a vector E_h and then normalized to get Z_{Eh} using Equation (3.15).

$$Z_{Eh}(k) = \frac{E_h(k) - \mu(E_h)}{\sigma(E_h)} \tag{3.15}$$

3.3.1.4 Euclidean Distance

The distance (i.e. Euclidean Distance *ED*) among the vector Z_E of every faulty case in the array W_c and the non-faulty case vector Z_{Eh} is computed as in Equation (3.16).

$$ED_t = \sqrt{\sum_r [Z_{gt}(r) - Z_{gh}(r)]^2} \tag{3.16}$$

All the values of the Euclidean Distances among the fault types are sorted in a single vector D_t. Fig. 3.16 illustrates the Euclidean distance values for the all fault types. Algorithm 1 describes in detail the procedures of generating the feature vector representation.

$$\begin{aligned} D_t = [&ED_{ag} \quad ED_{bg} \quad ED_{cg} \quad ED_{ab} \quad ED_{bc} \quad ED_{ac} \\ &ED_{abg} \quad ED_{bcg} \quad ED_{acg} \quad ED_{abc} \quad ED_{abcg}] \end{aligned} \tag{3.17}$$

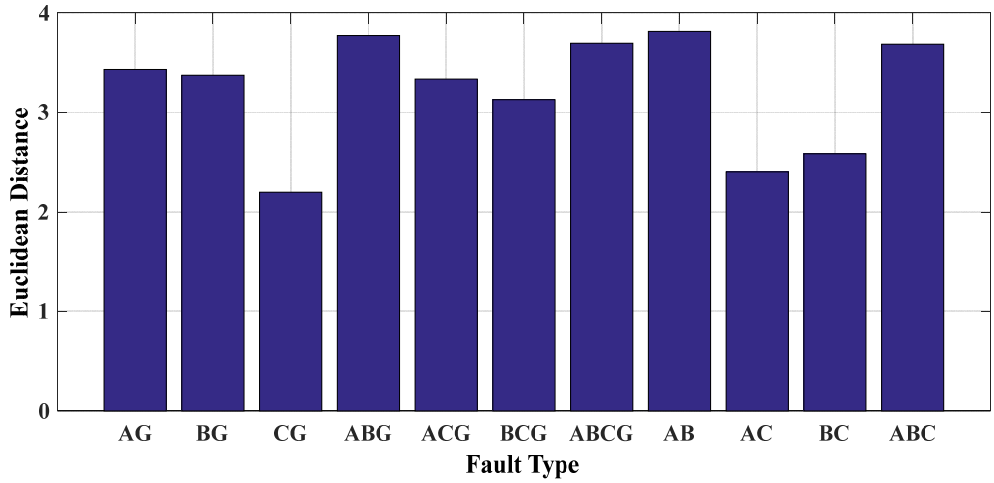


Fig. 3. 16: The Euclidean distances at different fault type.

Algorithm 1. Feature vector representation [12]

1: **Start:** Input, Three-phase voltage and current signals

2: **For** F=1:1: N_F **Do**

F: Fault type (AG, BG,……., N_F)

3: **Calculate:** Difference value between the samples

$$x_{ph}^{D}(j) = x_{ph}(j+n) - x_{ph}(j) \quad j = 1,2,\dots,n$$

x: Signal type (voltage or current)

ph: Phase (A, B, and C),

n: Number of samples per cycle

D: Difference value between the two samples

4: **Calculate**: Wavelet coefficient and their energies using

$$Ca_j(e) = \sum_{l_e} f_{l_0}(l_e - 2e)Ca_{j-1}(l_e), \text{and } Cd_j(e) = \sum_{l_e} f_{l_1}(l_e - 2e)Ca_{j-1}(l_e)$$

$$g_{aj} = \sum_{e} \left|Ca_j(e)\right|^2, \text{and } g_{dj} = \sum_{e} \left|Cd_j(e)\right|^2$$

5: **Arrange:** Wavelet coefficient energies in one vector E_r

$$W_c = [E_{ag} \quad E_{bg} \quad E_{cg} \quad E_{ab} \quad E_{bc} \quad E_{ac}$$

$$E_{abg} \quad E_{bcg} \quad E_{acg} \quad E_{abc} \quad E_{abcg}]^T$$

6: **Standardize:** Wavelet coefficient energies vector using

$$Z_{Eag}(k) = \frac{E_{ag}(k) - \mu(E_{ag})}{\sigma(E_{ag})}$$

7: **Calculate, Arrange, and Standardize:** Wavelet coefficient energies at healthy case Z_{gh}

8: **Calculate**: Euclidean Distance between Z_{gt} and Z_{gh}

$$ED_t = \sqrt{\sum_r [Z_{gt}(r) - Z_{gh}(r)]^2}$$

9: **End For**

10: **Formulate**: Euclidean distance vector

$$D_t = [ED_{ag} \quad ED_{bg} \quad ED_{cg} \quad ED_{ab} \quad ED_{bc} \quad ED_{ac}$$

$$ED_{abg} \quad ED_{bcg} \quad ED_{acg} \quad ED_{abc} \quad ED_{abcg}]$$

11: **End**

3.3.1.5 Variance

The variance σ_{Dt}^2 between the distances in the distance vector D_t is calculated as:

$$\sigma_{D_t}^2 = \frac{1}{m-1}\sum_P [D_t(P) - \mu(D_t)]^2 \tag{3.18}$$

The variance in the distance vector is a measure of how distinct each fault is from the remaining fault types. Therefore, the value of the variance $\sigma_{D_t}^2$ needs to be maximized for maximum differences between all fault types to be able to identify each fault type. Moreover, the variance is affected by the wavelet functions combination choice used in the analysis.

Figs. 3.17 - 3.19 shows an instance for the Euclidean distance values for different types of faults (i.e. 11 faults) by applying five different arbitrary combinations of mother wavelet functions as listed in Table 3.5. In every combination, a different mother wavelet was used for the decomposition level. For example, in the case of combination 2 (comb.2), the mother wavelet db4 as recommended in [28] and [29] is used in all decomposition levels. Visual inspection of Figs.

3.17 - 3.19 reveals that in the case of varying the wavelet function combination used, the maximum and minimum values of variance for the distance vector are 2.43, and 0.25 respectively.

Table 3. 5: Wavelet Family Combination

Decomposition level	Comb.1	Comb.2	Comb.3	Comb.4	Comb.5
Cd_1	sym10	db4	bior1.5	bior3.5	sym5
Cd_2	coif5	db4	bior1.3	bior1.5	sym5
Cd_3	bior3.5	db4	db25	db4	sym5
Cd_4	dmey	db4	sym4	db8	sym5
Ca_4	db20	db4	bior1.5	db41	sym5

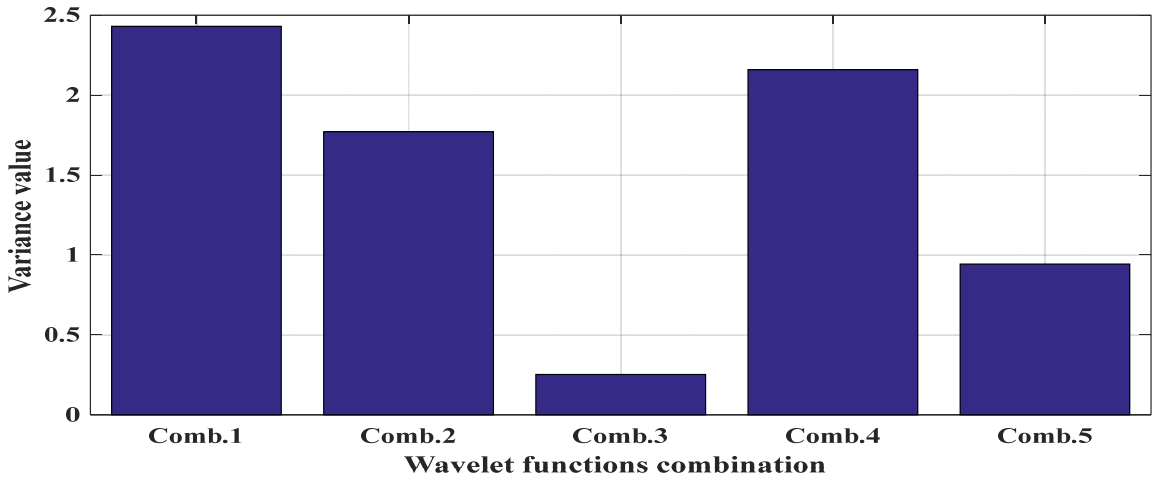


Fig. 3. 17: Variance of Different fault type with different wavelet functions combination.

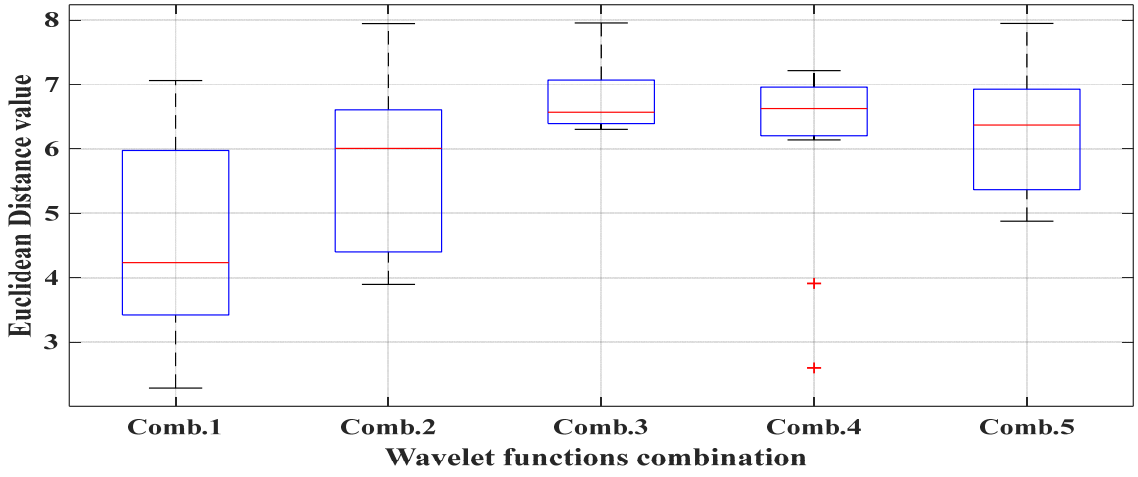


Fig. 3. 18: Boxplot of Different fault type with different wavelet functions combination.

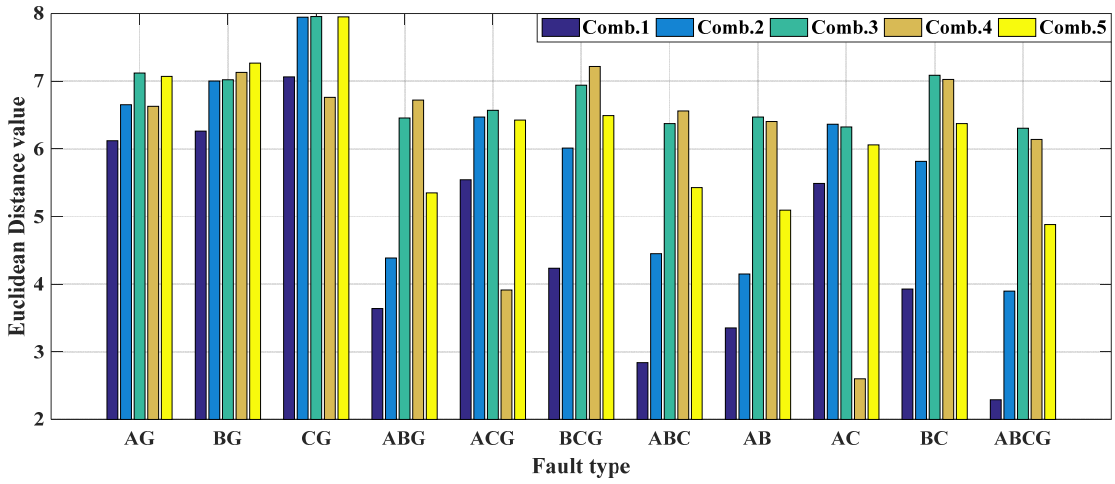


Fig. 3. 19: Euclidean Distance values of different fault type with different wavelet functions combination.

In order to solve the problem of obtaining different values of the variance, which is mainly dependent on the choice of wavelet functions and levels, the problem is formulated as an optimization problem where the variance σ_{Dt}^2 considers the fitness function needed to be maximized. In this work, the Harmony Search Algorithm (HSA) as an optimization technique is utilized to systematically identify the most suitable wavelet function(s) and the appropriate decomposition levels that would maximize the variance.

In this study, the HSA as an evolutionary optimization technique was used to explore the search space of the wavelet families. In addition, the algorithm searches for the suitable wavelet function for each level in each signal.

For example:

a. In the case of two signals (e.g. voltage and current), the available options are (2^2 = 4).

b. In the case of five decomposition levels for each signal (e.g. D_1, D_2, D_3, D_4, and A_4), the available options are (2^5 x 2^5 = 1024).

c. In the case of 85 wavelet families (e.g. db1, db2... etc.) for each levels (5 levels) in each signal (2 signals), the available options ($85^5 \times 85^5 = 19 \times 10^{18}$).

d. The total available options will be ($2^2 \times 2^5 \times 2^5 \times 85^5 \times 85^5 = 8 \times 10^{22}$).

3.3.2 Optimization Tool: Harmony Search Algorithm

The Harmony Search Algorithm (HSA) mimics the improvisation of the musical process looking for an accurate state of harmony as exemplified in Fig. 3.20. The HSA is a meta-heuristic optimization technique, which was developed by Geem et al., [56] and has been recently utilized in several power system benchmark with success [57].

The HSA is a simple concept and is easy to implement since it does not require extensive mathematical analysis and less parameters are needed. Furthermore, it does not need a setting for the initial value of decision variables.

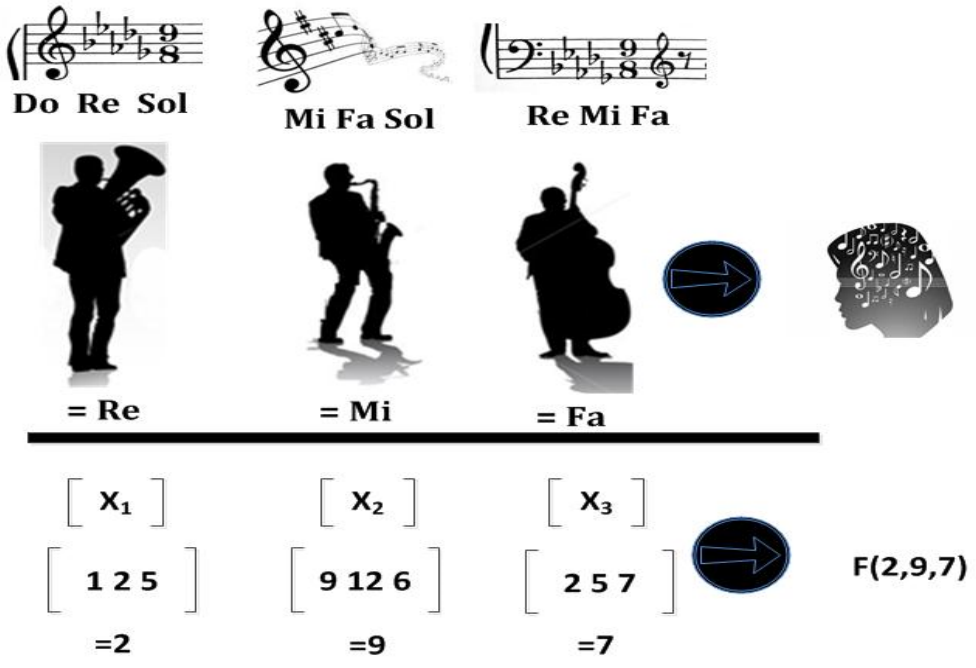


Fig. 3. 20: Musical improvisation and engineering problem optimization [56].

The limitations of the existing optimization techniques, such as the need for initial values and considerable gradient details, were highlighted in [58]. Furthermore, the study in [58] concluded that the HSA was capable of providing a superior accuracy in comparison with present meta-heuristic optimization algorithms (e.g. particle swarm, genetic algorithm, and tabu search). In addition, the HSA requires less mathematical requirements, which makes it preferable to use in the complex optimization problems.

In order to identify the most suitable optimizer for the problem at hand, the performance of the existing optimization algorithms was examined in this thesis. Specifically, the HSA, GA, and PSO algorithms were applied using the same number of population (i.e. 50) for each algorithm (individuals in GA, particles in PSO, and harmony memory size in HSA) and their performances were compared. The fitness function tolerance value (i.e. 1×10^{-6}) was used as a termination criterion for all algorithms. Table 3.6 summarizes the parameters used for each algorithm. The table also shows the superiority of HSA (highlighted in grey) compared with the other algorithms (i.e. PSO and GA) with the highest fitness function value.

Therefore, in this study, the Harmony Search Algorithm (HSA) is selected as the optimization technique to solve the problem at hand and it is utilized to systematically identify the optimal wavelet(s) and level(s) of decomposition for accurate fault classification. The optimization steps 1–4 of the HSA can be outlined as follows [59]:

1) Optimization problem preparation and decision variables description.

2) Harmony memory initialization.

3) Generation of a new harmony solution.

4) Harmony memory Updating.

Table 3. 6: Optimization techniques comparison

Technique	Parameters	Fitness Function Tolerance	Fitness Function Value
GA	Individuals: 50 Crossover rate: 0.8 Selection function: Roulette Mutation rate: 0.01	1 × 10-6	3.49
PSO	Number of Particles: 50 Cognitive acceleration: 2 Social acceleration:: 2 Minimum inertia weight: 0.4 maximum inertia weight: 0.9	1 × 10-6	3.48
HSA	Harmony memory size: 50 Harmony memory consideration rate: 0.95 Pitch adjusting rate: 0.99 Bandwidth: 0.35	1 × 10-6	4.47

The complete description of these steps summarized in the following:

Step 1: *Optimization problem preparation and decision variables description*

In this work, the variance σ_{Dt}^2 represents the objective function needed to be maximum.

$$Maximize \quad \{\sigma_{D_t}^2\} \tag{3.19}$$

The decision variable set is *y*, the total number of decision variables is N_y, each decision variable y_x are restricted by the maximum value y_{max} and minimum value y_{min}.

The HSA randomly allocates the values for each decision variable. In every iteration, the decision variable vector *y* consists of various elements, where each element is allocated a numeric

value. In the proposed method, the elements of decision variable vector represent the following as shown in Fig. 3.21:

1) The type of the signal (i.e. current or/and voltage).
2) The wavelet functions.
3) The wavelet decomposition levels.

In every iteration of the optimization algorithm (i.e. HSA), the current and/or voltage signals are used during the optimization procedures according to the decision variables' values of V_{r1} (current) and V_{r2} (voltage). For instance, if the value of $V_{r1} = 0$, that means the current signal is not selected in this iteration. The HSA algorithm thereafter choose the decomposition levels of voltage and current signals corresponding to the decision variables' values of V_{r3} to V_{r12}. Moreover, the wavelet functions required in the analysis is selected according to the decision variables' values of the V_{r13} to V_{r222}. The decision variables (V_{r13} to V_{r22}), which represent the wavelet functions, are assigned with numbers as described in Table 3.7. Fig. 3.21 illustrates an example for the vector of the decision variables (V_{r1} to V_{r22}) of a specific trial.

Where the vector of the decision variable in this example consists of:

- The current signal is chosen since the value of V_{r1} =1.
- The voltage signal is chosen since the value of V_{r2} =1.
- The three wavelet decomposition detail levels, Cd_1, Cd_2, and Cd_4, are chosen in the case of the current signal since the values of V_{r3} =1, V_{r4} =1, and V_{r6} =1. While the wavelet decomposition detail level Cd_3 and the wavelet decomposition approximation level Ca_4 are not chosen in this example since the values of V_{r5} =0, and V_{r7} =0.
- The three wavelet decomposition detail levels, Cd_1, Cd_2, and Cd_3, are chosen in the case of the voltage signal since the values of V_{r8} =1, V_{r9} =1, and V_{r10} =1. While the wavelet

decomposition detail level Cd_4 and the wavelet decomposition approximation level Ca_4 are not chosen in this example since the values of V_{r11} =0, and V_{r12} =0.

- The mother wavelet functions, *db45*, *db1*, and *bior1.1*, are chosen in the case of the current signal since the values of V_{r13} =45, V_{r14} =1, and V_{r16} =70. Because of the wavelet decomposition detail level Cd_3 and the wavelet decomposition approximation level Ca_4 are not chosen in this example (V_{r5} =0, and V_{r7} =0), the wavelet function (V_{r15} and V_{r17}) for this two levels (Cd_3, and Ca_4) are not chosen as well.
- The mother wavelet functions, *coif1*, *sym2*, and *dmey*, are chosen in the case of the voltage signal since of the values of V_{r18} =46, V_{r19} =51, and V_{r20} =85. Because of the wavelet decomposition detail level Cd_4 and the wavelet decomposition approximation level Ca_4 are not chosen in this example (V_{r11} =0, and V_{r12} =0), the wavelet functions (V_{r21} and V_{r22}) for these two levels (Cd_4, and Ca_4) are not chosen as well.

Table 3.7: Wavelet families' library

Wavelet families	Daubechies	Coiflets	Symlets	Biorthogonal
Wavelets	db1 ... db45.	coif1... coif5.	sym2 ... sym20.	bior1.1 … bior6.8.
Assigned number	1 ... 45.	46 ... 50.	51 ... 69.	70 ... 84.

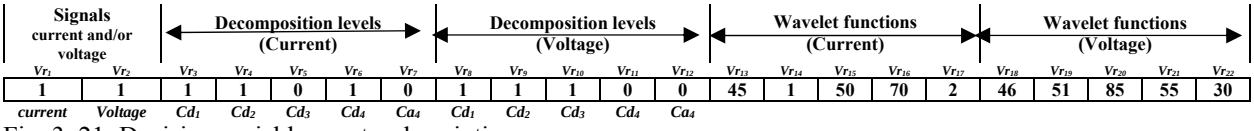


Fig. 3. 21: Decision variables vector description

Step 2: *Harmony memory initialization*

At the beginning, the HSA algorithm arbitrarily generates the initial values for every decision variable as follows:

$$y_x = y_{xmin} + rand_1(y_{xmax} - y_{xmin}) \tag{3.20}$$

Where *rand1* is an arbitrary number created using the uniform distribution between 0 and 1. The values of the decision variables are sorted in the Harmony Memory (HM) matrix as shown in (3.21), the number of columns N_y of the HM matrix is the number of decision variables, and the number of rows H_s of HM matrix is the harmony memory size, which is an input parameter to the algorithm.

$$\mathrm{HM} = \begin{bmatrix} \mathrm{y}_1^1 & .. & .. & \mathrm{y}_{N_y}^1 \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ \mathrm{y}_1^{H_s} & .. & .. & \mathrm{y}_{N_y}^{H_s} \end{bmatrix} \rightarrow \begin{bmatrix} \mathrm{f}_1(y) \\ .. \\ .. \\ \mathrm{f}_{H_s}(y) \end{bmatrix} \tag{3.21}$$

The values of the fitness function are computed using each row in HM and are then stored in a vector $[\mathrm{f}_1(y), .., \mathrm{f}_{H_s}(y)]^T$ as per (3.21). The main objective of the HSA is to search through multi-trials for the optimal values of decision variables by maximizing the value of fitness function. Hence, in each trial the HSA finds the highest value of the fitness function, and tries to replace it with another new one with a lower value.

Step 3: *Generation of a new harmony solution*

A new solution vector with new values for the decision variable values is generated as $y^{new} = [y_1^{new}, .., y_{N_y}^{new}]$ based on the harmony memory considering the rate HMC_r, the pitch adjusting rate PA_r, and the bandwidth (bw). Each value of the decision variable in the new vector y^{new} is generated according to (3.22).

$$y_1^{new} \leftarrow \begin{cases} y_1^{new} \in [y_1^1, .., y_1^{H_s}] & rand_2 \leq HMC_r \\ y_1^{new} \in [y_{1max}, y_{1max}] & rand_2 > HMC_r \end{cases} \tag{3.22}$$

Where *rand2* is an arbitrary number created using the uniform distribution between 0 and 1. The new decision variable value is adjusted according to the pitch adjusting rate value PA_r.

$$y_1^{new} \leftarrow \begin{cases} y_1^{new} \pm \mathrm{rand} \times \mathrm{bw} & rand_3 \leq \mathrm{PA}_r \\ y_1^{new} & rand_3 > \mathrm{PA}_r \end{cases} \tag{3.23}$$

Where *rand₃* is an arbitrary number created using the uniform distribution between 0 and 1. The HSA parameters PA_r and bw are updated in every iteration to improve the HSA performance using the following [36]:

$$\mathrm{PA}_r = \mathrm{PA}_{rmin} + [(\mathrm{PA}_{rmax} - \mathrm{PA}_{rmin})(Ir/MIr)] \tag{3.24}$$

$$bw = \mathrm{bw}_{max} \times e^{(\ln[(\mathrm{bw}_{min}/\mathrm{bw}_{max})/MIr]\times Ir)} \tag{3.25}$$

Where PA_{rmin} is the minimum pitch adjusting rate and PA_{rmax} is the maximum pitch adjusting rate, bw_{min} is the minimum bandwidth and bw_{max} is maximum bandwidth, Ir is the current iteration and MIr is the maximum iteration.

Step 4: *Harmony Memory Updating*

If the value of objective function for the new harmony vector is less than the worst member value, the HSA algorithm change (update) the HM matrix by replacing the worst harmony vector by the new one. Algorithm 2 outlines the implementation steps of HSA in detail.

Algorithm 2. Harmony Search Algorithm Implementation Procedures [11]

1: **Start**

Inputs: $y, f(y), N_y, H_s, HMC_r, PA_{rmax}, PA_{rmin}, bw_{max}, bw_{min}, , MIr, Ir, y_{max,}$,and y_{min}

2: **For** i =1:1: H_s **Do**

3: **For** j =1:1: N_y **Do**

4: Generate the HM Matrix

$$y_{i,j} = y_{jmin} + rand_1(y_{jmax} - y_{jmin})$$

5: **End For**

6: **End For**

7: C**ompute**: The fitness function *f(y)* using Equation (3.13) based on the chosen signals, wavelet decomposition levels, and wavelet functions for each HM row.

8: **Define:** The worst Harmony solution vector y_w

9: **While** (*Ir* < *MIr*) **Do**

10: Update the pitch adjusting rate and bandwidth

$$\mathrm{PA}_r = \mathrm{PA}_{rmin} + [(\mathrm{PA}_{rmax} - \mathrm{PA}_{rmin})(Ir/MIr)]$$

$$bw = \mathrm{bw}_{max} \times e^{(\ln[(\mathrm{bw}_{min}/\mathrm{bw}_{max})/MIr] \times Ir)}$$

12: **For** n=1:1: N_y **Do**

13: **If** $(rand \leq HMC_r)$**Then**

14: $y_1^{new} \epsilon [y_{1max}, y_{1max}]$

15: **If** $(rand \leq PA_r)$ **Then**

16: **If** $(rand \leq 0.5)$ **Then**

17: $y_n^{new} = y_n^{new} + \mathrm{rand} \times \mathrm{bw}$

18: **else**

19: $y_n^{new} = y_n^{new} - \mathrm{rand} \times \mathrm{bw}$

20: **End If**

21: **End If**

22: **else**

23: $y_n^{new} = y_n^{new} + \mathrm{rand} \times (y_{jmax} - y_{jmin})$

24: **End If**

25: **End For**

26: C**ompute**: The fitness function *f(y)* using Equation (3.18) based on the chosen signals, wavelet decomposition levels, and wavelet functions for the new harmony solution vector

27: **If** $(f(y^{new}) > f(y_w))$ **Then**

28: **Include** y^{new} into the harmony memory matrix

29: **Exclude** y_w from the harmony memory matrix

30: **End If**

31: **End While**

32: **Output:** The decision variable optimal values (i.e. type of signal, the best suitable wavelet function(s) and the optimal wavelet decomposition levels number

33: **End**

3.4 Machine Learning Techniques for Fault Classification

Machine learning [60] is a procedure of data analysis that automates systematic model structure using techniques that iteratively learn from data. The purpose of this work is to present the concept of the selection of the wavelet parameters to the fault diagnosis in the smart distribution systems and Micro-Grids and examine its effectiveness. These techniques include the Decision Tree (DT) or the K-Nearest Neighbor (K-NN) classifier, instead of other non-deterministic classification techniques like the Support Vector Machine (SVM) classifier, which uses optimization scheme or the Naïve Bayes (NB) classifier, which relies on probability theory.

The learning methods can be categorized into two main methods, the eager learning method and the lazy learning method. In the eager learning method, the inductive step involves the construction of a classification model using the labeled training data while, in the deductive step, the classifier model is applied to the labeled test data. The decision tree classifier is commonly described as an eager learner. On the other hand, in the lazy learning method, the process of model development is delayed and the classification process is performed by finding all the training instances that are relatively similar to the attribute test instance. Therefore, K-Nearest Neighbor is commonly described as a lazy learner.

In this thesis, two Machine Learning classifiers (i.e. the DT classifier as an eager learner and K-NN classifier as a lazy learner) are used to automate the fault classification process.

3.4.1 K-Nearest Neighbor Classifier

The K-Nearest Neighbor [60] is one of the most popular classification algorithms. The K-NN classifier has the dataset D which is formed of a_t features containing C cases, where every case is allocated a label Cl. The total number of classes is identified by m_c.

$$\mathrm{D} = \begin{bmatrix} \mathrm{D}_1^1 & .. & .. & \mathrm{D}_{a_t}^1 \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ \mathrm{D}_1^C & .. & .. & \mathrm{D}_{a_t}^C \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{bmatrix} \mathrm{Cl}_1 \\ .. \\ .. \\ \mathrm{Cl}_{m_c} \end{bmatrix} \tag{3.26}$$

The value of K determines the number of nearest neighbors to a new record. To predict the label of a new record, the K-NN classifier calculates the Euclidean distance (E_y) between the new record and the training dataset records with known labels [60].

$$\mathrm{E}_y = \sqrt{\sum_{z=1}^{a_t} (D_z^y - D_z')^2} \qquad y = 1, \dots, C \tag{3.27}$$

Where the Euclidean distance between the new record D' and every record in data set D^y at line y is E_y. The record labels that have the K[th] minimum value of Euclidean distance (nearest neighbor) are reserved to be the label of the new record. For multi-class problems, the records, which occur in the nearest neighbor follow a voting scheme (i.e. majority vote), in order to define the new record label.

$$\tau_n = arg \max_{v=Cl_1,\dots,Cl_{m_c}} \sum_{z=1}^{K} \xi(\Omega_v = C_z), \qquad \xi \begin{cases} 1 & v = C_z \\ 0 & v \neq C_z \end{cases} \tag{3.28}$$

Where the new record label is τ_n, the class labels set is Ω_v, the closest neighbors label is C_z, and $\xi(.)$is an index which is equal to 1 or 0 depending on Ω_v and C_z values. Fig. 3.22 shows an example for the K-NN classifier, two attributes with two class labels. To predict the new case with set (K=5).

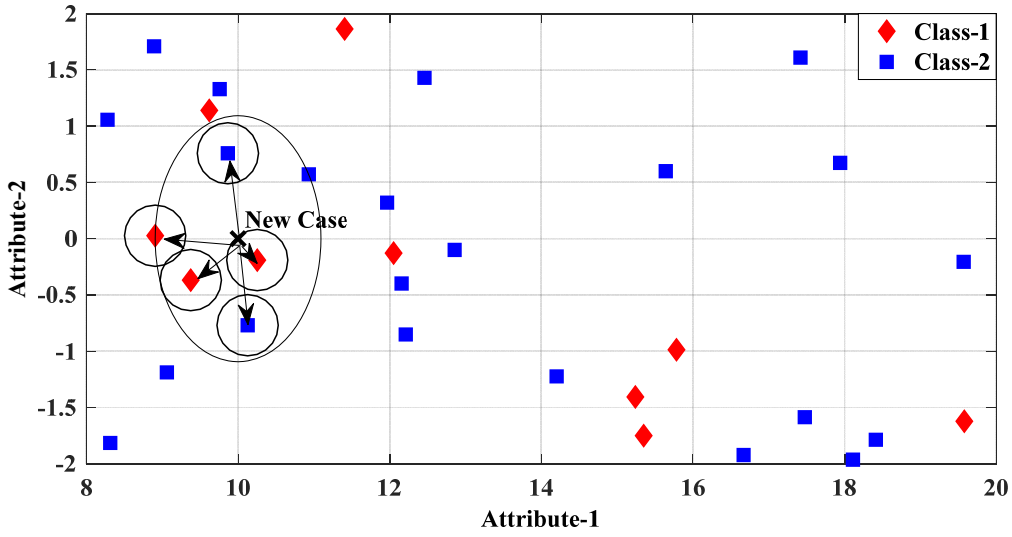


Fig. 3. 22: K-NN classifier procedures.

3.4.2 Decision Tree Classifier

The Decision Tree classifier [60] is usually considered as a simple classification technique, so it commonly applied in classification problems. The DT usually includes three types of nodes (i.e. root node, internal node, and leaf node) as shown in Fig. 3.23.

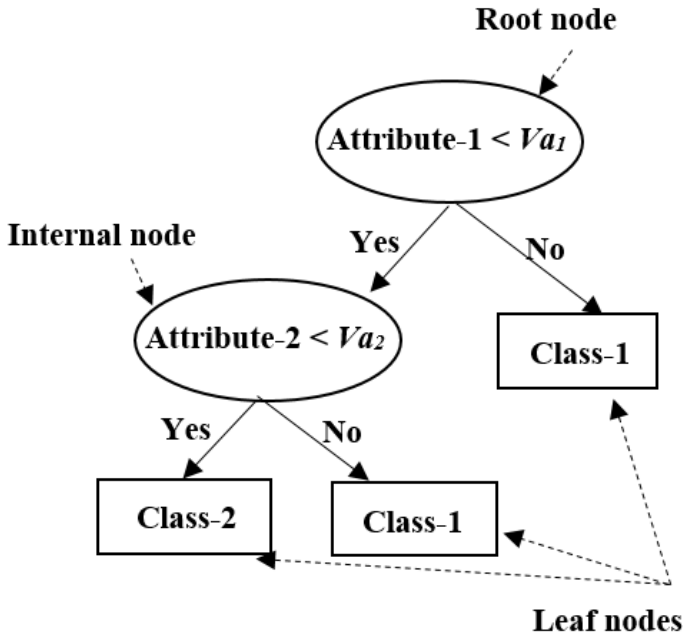


Fig. 3. 23: Decision Tree procedures.

In order to determine the best way to split the attributes, there are three impurity measures that can be used: the Gini index, the Entropy index, and the Classification Error index. In this thesis, the Gini Index, which is the most commonly used index used [61], was applied to determine the node impurity.

$$Gini(\gamma) = 1 - \sum_{\varepsilon=0}^{C_a-1} [p(\varepsilon|\gamma)]^2 \tag{3.29}$$

$$Entropy(\gamma) = -\sum_{\varepsilon=0}^{C_a-1} p(\varepsilon|\gamma) log_2 p(\varepsilon|\gamma) \tag{3.30}$$

$$Classification\ Error\ (\gamma) = 1 - \max_{\varepsilon}[p(\varepsilon|\gamma)] \tag{3.31}$$

Where $p(\varepsilon|\gamma)$ is the proportional frequency of class ε at a particular node γ.

3.5 Performance Evaluation of the Classifier

In the machine learning, the labeled data in the dataset are usually partitioned into two subsets: one set for training and another set for testing. The classification accuracy of the classifier strongly depends on the choice of the data used for training and those used for testing.

3.5.1 Monte Carlo Stratified Cross-Validation Algorithm

The cross-validation is a very common method used in the performance evaluation of the different classification techniques [62]. In the stratified cross validation method, the full dataset is divided firstly into *K*-subsets stratified and in equal size, which means each class in each subset is represented with almost equal ratio of each class in the full dataset. Each subset *K* is used in the testing process and the remaining subsets (*K*-1) are used in the training process as shown in Fig. 3.24.

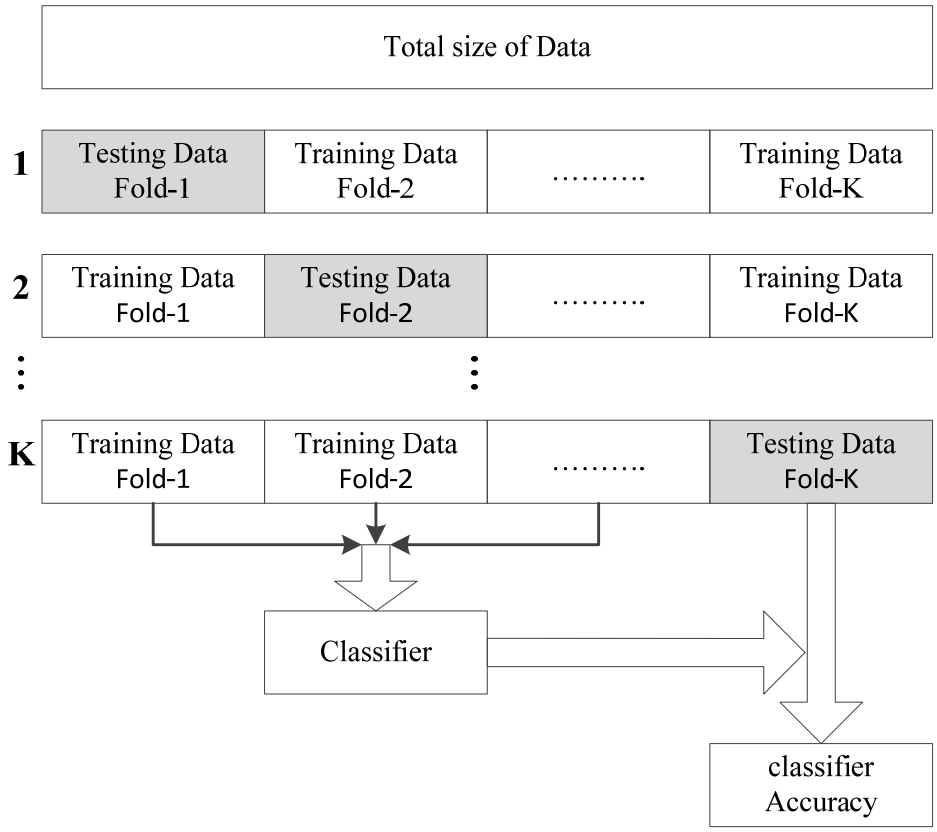


Fig. 3. 24: K-fold cross validation procedures.

The cross validation method relied on the randomness in the partitioning process, so the estimated classification accuracy by this method may be considered biased. The Monte Carlo method is used in this thesis to address the aforementioned issue (i.e. randomness). The random permutation of selected cases in each subset is applied using the Monte Carlo method, where the cross validation method is repeated in the *C* trials [63] and in each trial different random subsets are selected. Finally, the overall accuracy of the classifier is the mean value of all Monte Carlo trials. Algorithm 3 describes in details the procedures of Monte Carlo stratified cross-validation method.

In each trial of the K-fold trials, the number of cases incorrectly predicted (NK_{miss}) are recorded; the classification accuracy (Acc_K) of each trial k is then computed using the number of testing cases N_K.

$$Acc_K = (1 - \frac{NK_{miss}}{N_K}) \tag{3.32}$$

The total accuracy of the classification model Acc_t of the all k-fold trials is computed as in (3.33).

$$Acc_t = \frac{\sum_{j_o}^{K} Acc_K(j_o)}{K_f} \tag{3.33}$$

The overall accuracy of the classification model δ of the all Monte Carlo trials is computed as in (3.34).

$$\delta = \frac{\sum_{l_o}^{C} Acc_t(l_o)}{C} \tag{3.34}$$

3.5.2 Test of Significance

In order to estimating the confidence interval for the classification accuracies, the minimum and maximum bounds of the Confidence Interval CI are computed by means of the Z-test. The limit $Z_{\alpha/2}$ relies on the standard normal distribution at confidence level $CI(1-\alpha)$ with a value of $Z_{\alpha/2}$ =1.96 at α=0.05, CL=95% and N_t is the total number of cases.

$$CI = \frac{2 \times N_t \times \delta + Z_{\propto/2}^2 \pm Z_{\propto/2} \times \sqrt{Z_{\propto/2}^2 + 4 \times N_t \times \delta - 4 \times N_t \times \delta^2}}{2 \times (N_t + Z_{\propto/2}^2)} \tag{3.35}$$

Fig. 3.25 shows the overall structure of the proposed method. The method starts by applying Algorithms 1 and 2 to find the most suitable wavelet function(s) and the wavelet decomposition level(s). Then the Discrete Wavelet Transform was used to extract the signals' features. The Decision Tree and the K-Nearest Neighbor classifiers were used to automate the classification process, and finally Algorithm 3 was applied using the Monte Carlo cross validation method to evaluate the proposed method through the computing of the classifier accuracy.

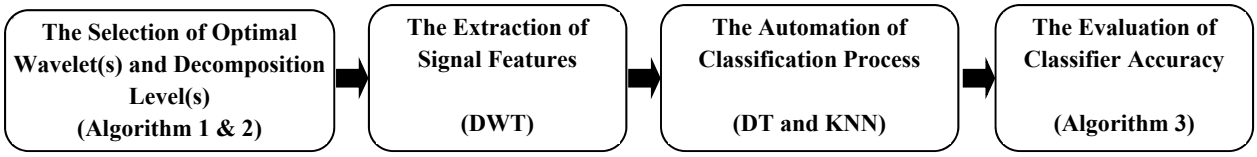


Fig. 3. 25: Overall proposed methodology structure

Algorithm 3. Monte Carlo Stratified Cross-Validation [12]

1: **Start**

Inputs: Full dataset, K=10, and C=1000

2: **For** i=1:1: C **Do** /* Monte Carlo Trials*/
3: **Divided:** Randomly the dataset into 10-Stratified subsets
4: **For** j=1:1: K **Do** /* 10-Folds cross validation*/
5: **Reserve:** One-subset for testing

6: **Reserve:** Remaining nine-subsets for training

7: **Perdict:** the classes of testing subset

8: **Calculate**: the classifier accuracy using

$$Acc_K = (1 - \frac{NK_{miss}}{N_K})$$

NK_{miss}: The number of cases incorrectly predicted

N_K: The number of cases in testing subset

9: **End For**

10: **Calculate:** The total accuracy of the classification model Acc_t using

$$Acc_t = \frac{\sum_{j_o}^{K} Acc_K(j_o)}{K}$$

11: **End For**

12: **Calculate:** The overall mean classification accuracy of classifier δ using

$$\delta = \frac{\sum_{l_o}^{C} Acc_t(l_o)}{C}$$

13: **End**

3.6 Summary

This chapter focused on the methodology of this thesis. The systematic procedures of the selection of the wavelet functions and decomposition levels was presented in this chapter. The performance of different transform analysis techniques was tested and it was found that the DWT outperformed other present transforms techniques (i.e. Short Time Fourier Transform, Continuous Wavelet Transform, Hilbert–Huang Transform, Stationary Wavelet Transform, and Wigner-Ville Distribution) in terms of the frequency-time representation, as well as the complexity time. It has been determined that the DWT is the most suitable transform to extract the hidden features in the transient signals needed for accurate fault diagnosis in the electric power system.

The influence of using different wavelet functions and the use of different wavelet decomposition levels in the wavelet transform analysis was examined and it was observed that the magnitude of the energy of the wavelet coefficients strongly depends on the choice of the wavelet functions used and the number of wavelet decomposition levels. Then, the presence of such differences might be considered as an indication of the discrepancies between the analyzed signal and the mother wavelet functions wave-shapes.

In order to solve the aforementioned problem of the selection of wavelet functions and the wavelet decomposition levels, the Smart Automated Fault-diagnosis Algorithm was proposed which finds the optimum combination of mother wavelets and the number of wavelet decomposition levels to help extracting the most important attributes from the signal needed for fault diagnosis in the electrical power system.

The performance of different optimizers was assessed and it was found that the HSA outperformed other existing optimizers. Hence, it has been determined that the HSA will be used throughout this thesis to identify the best combination of wavelet function(s) and the wavelet

decomposition level(s). The problem was formulated as an optimization problem in which the objective function is to maximize the variances of the distance vector, which leads to extracting the prominent features and maintaining good performance.

In order to examine the effectiveness of presenting the concept of the selection of the wavelet parameters to the fault diagnosis in the electrical power system, the machine learning algorithms presented in this thesis are described thoroughly within this chapter. Two machine learning classifiers (i.e. the DT classifier as an eager learner and K-NN classifier as a lazy learner) are used to automate the fault classification process. Also, the method of Monte Carlo Stratified Cross-Validation was presented as a means to evaluate the performance through calculating the classification accuracy.

Finally, the entire proposed approach is implemented in order to examine the approach as will be introduced in Chapter 4.

Chapter 4: Results and Evaluation

4.1 Introduction

In this chapter, the performance of the proposed automated wavelet-based fault detection and diagnosis (AWBFDD) technique is evaluated. The procedures used to implement the proposed approach will be explained, including the feature extraction process using the optimal wavelet function(s) and the wavelet decomposition levels as identified by HSA.

Two different test systems are considered in this work. The first test system is a 230 kV Transmission Line System (TLS) with both static and dynamic loads. In this work, a two-bus transmission system was used, in order to demonstrate the feasibility and to verify the concept of the AWBFDD approach. The second test system is a low voltage (0.480 kV) Micro-Grid Distribution System (MGDS) embedded with distributed energy resources, which enable the Micro-Grid to operate in two different operation modes (i.e. grid-connected and islanded modes). Both test systems are simulated in PSCAD/EMTDC software package, and then the dataset of several faulty cases with various operating conditions are generated. The proposed algorithm and the machine learning algorithms described in Chapter 3 are utilized in MATLAB software and then the classification accuracies are computed. Finally, a real-time experimental setup was implemented in order to evaluate the effectiveness of the proposed algorithm.

4.2 Transmission Line System

4.2.1 Test-System Description of Transmission Line System

A two-bus transmission line system, which is a real Taiwanese 230-kV system [64], as shown in Fig. 4.1 was used as a case study of a transmission system in this thesis. The system consists of a three-phase power source, and three phase transmission line with parameters outlined in Table

4.1, two 3-phase transformers, resistive loads as static loads and three phase Induction Motor (IM) as a dynamic load. Tables 4.2 - 4.5 summarize the transmission line test system parameters.

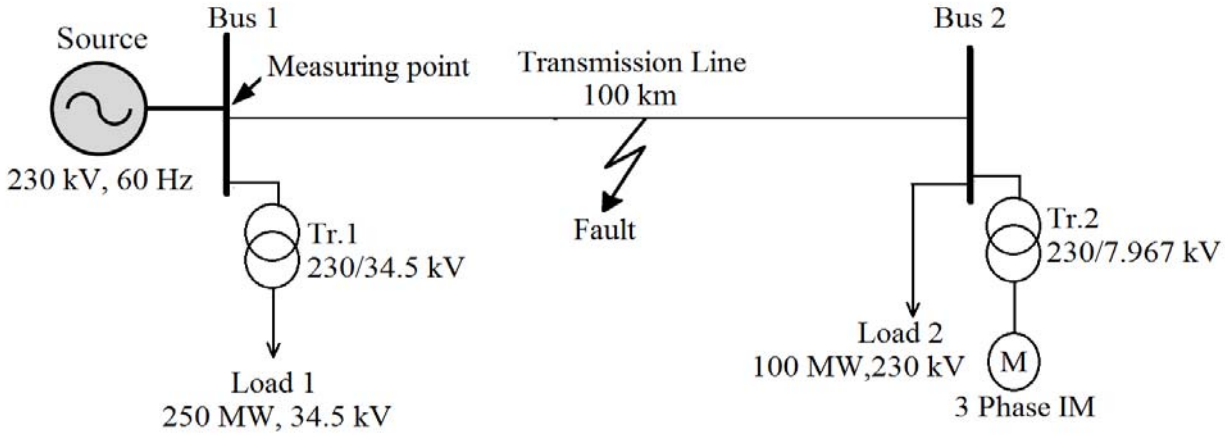


Fig. 4. 1: Single line diagram of transmission line system.

Table 4. 1: Transmission Line System: Lines Parameters.

Line Parameter	Value
Pos. seq. resistance	17.8×10e-6 Ω / m
Pos. seq. inductive reactance	313.88×10e-6 Ω / m
Pos. seq. capacitive susceptance	273.5 Ʊ.m
Zero seq. resistance	2.9522×10e-4 Ω / m
Zero seq. inductive reactance	0.0010 Ω / m
Zero seq. capacitive susceptance	414.1 Ʊ.m
Length	100,000 m

Table 4. 2: Transmission Line System: Source Parameters.

Power Source Parameter	Value
Rated Voltage	230 kV, 60 Hz
Rated Power	750 MVA
Short circuit level	30000 MVA
Stator resistance	0.2 Ω
Stator reactance	4.49 Ω

Table 4. 3: Transmission Line System: Transformer Parameters.

Transformer	Rated Power (MVA)	High-Voltage Side (kV)	Low-Voltage Side (kV)	X/R (%)	Z (%)
Tr.1	500	230 - D	34.5 -Gr. W	5	2
Tr.2	300	230 - D	7.967 - Gr. W	6	1

Table 4. 4: Transmission Line System: Load parameters.

Load	Model	Rated Voltage (kV)	Rated Power (MW)
Load1	Y-PQ	34.5	250
Load2	D-PQ	230	100

Table 4. 5: Transmission Line System: Induction Motor (IM) Parameters.

IM Parameter	value
Rated Voltage	7.967 kV
Rated current	15.59 kA
Stator resistance	0.019 Ω
Magnetizing reactance	1.138 Ω
Leakage reactance	0.013 Ω

4.2.2 Fault Cases Generation of Transmission Line System

The test system shown in Fig. 4.1 is modeled in the PSCAD/EMTDC software [65]. The current and voltage waveforms at Bus 1 (source bus) were recorded with a sampling frequency of 3.84 kHz (64 samples/cycle of power system frequency, 60Hz). This sampling rate represents the typical sampling rate used in most digital protective relays as reported by [2].

It is worth noting that the proposed approach is flexible in terms of being adaptable to different sampling rates. For example, in case of different sampling rates (i.e. 32 and 16

samples/cycle) the HSA will search for the best combination of wavelet decomposition levels so to maximize the variances of the distance vector, as described in Chapter 3.

The training and the testing datasets are generated following Table 4.6, where 11 fault types are modeled with 4 resistances of fault at 12 different locations. It is worth noting that all fault types under study are short circuit faults, because in the transmission system, the short circuit faults are an important class of faults with the most occurrence (high probability) faults as reported in [66]. This creates 528 records (i.e. 11 types × 12 locations × 4 resistances), which represents the total number of records in the dataset. Figs. 4.2 - 4.12 show the three-phase voltage and current signals at various fault types of the transmission line system.

Table 4. 6: List of parameters considered for all fault cases.

Fault type		Fault resistance	Fault Location	Cases
LG	AG	0.01, 0.1, 1, and 10 Ω	0, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 Km	48
	BG			48
	CG			48
LL	AB			48
	BC			48
	AC			48
LLG	ABG			48
	BCG			48
	ACG			48
LLL	ABC			48
LLLG	ABCG			48
Total	11	4	12	528

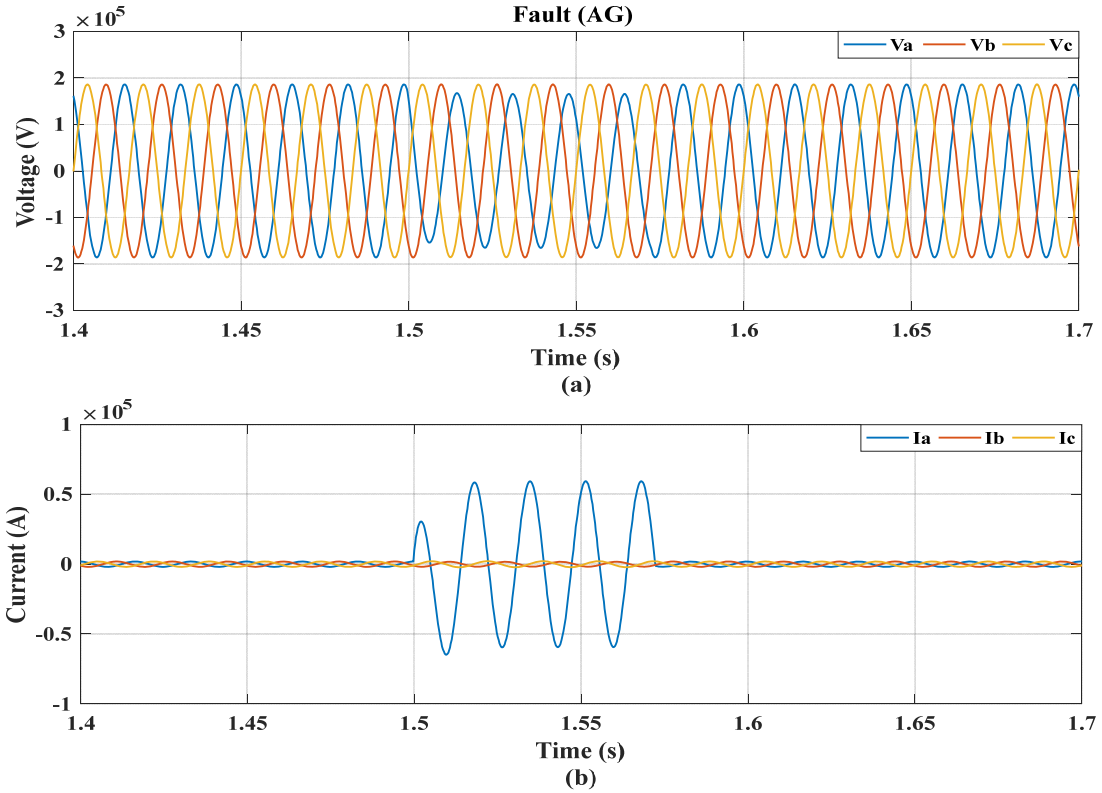


Fig. 4. 2: Three-phase voltage and current at AG fault for transmission line system

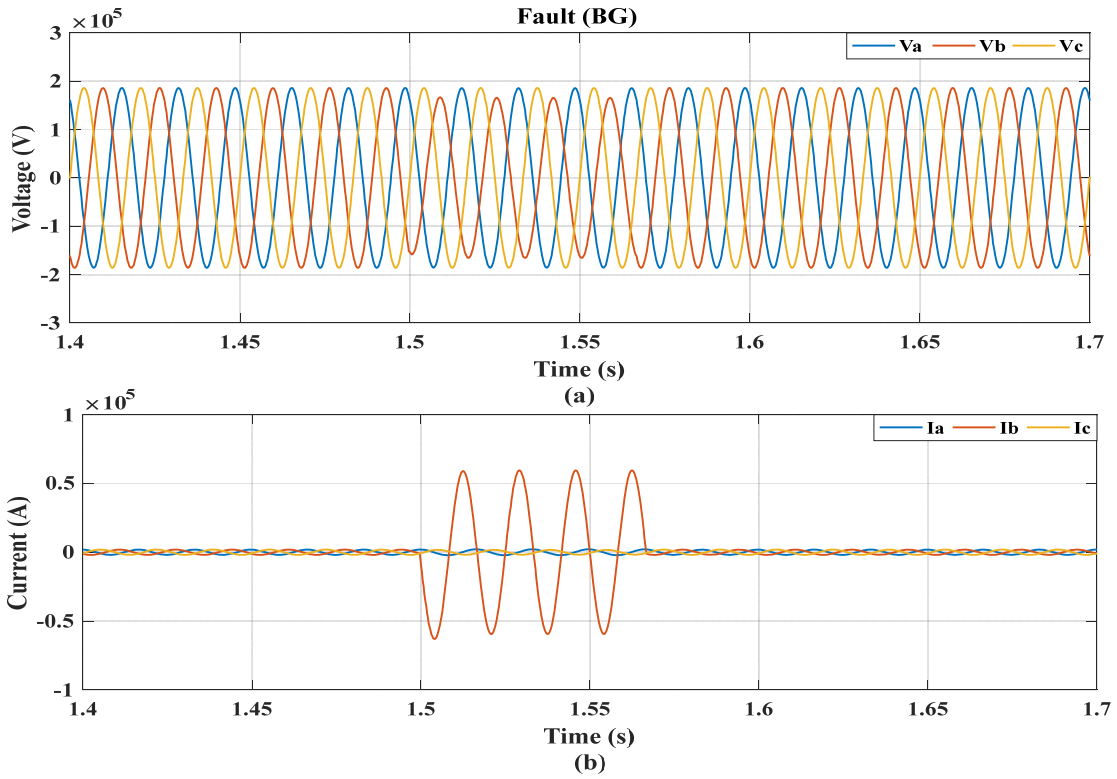


Fig. 4. 3: Three-phase voltage and current at BG fault for transmission line system

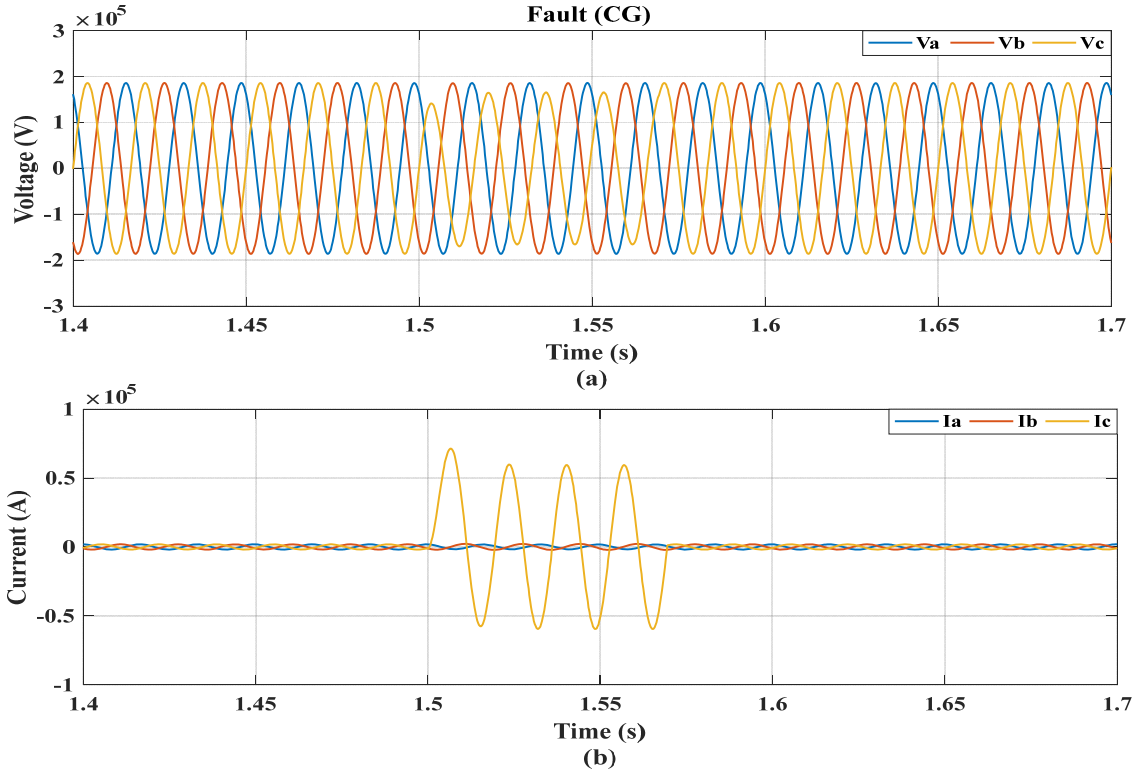


Fig. 4. 4: Three-phase voltage and current at CG fault for transmission line system.

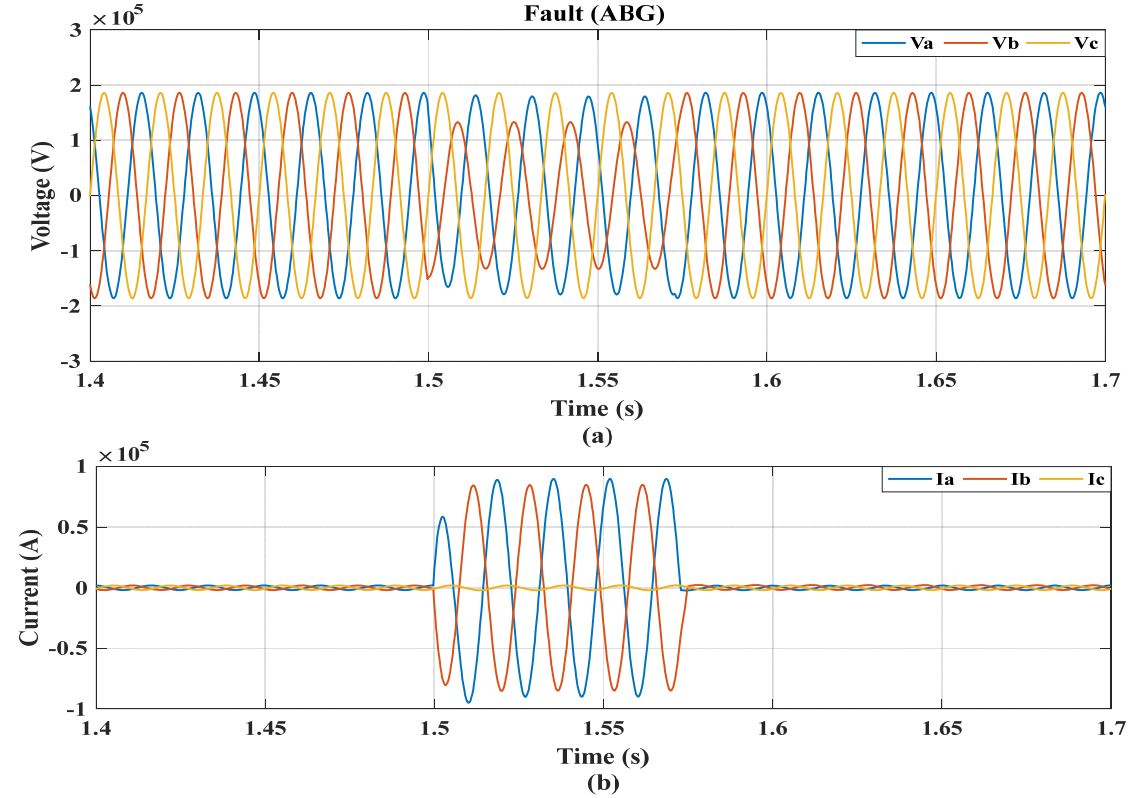


Fig. 4. 5: Three-phase voltage and current at ABG fault for transmission line system.

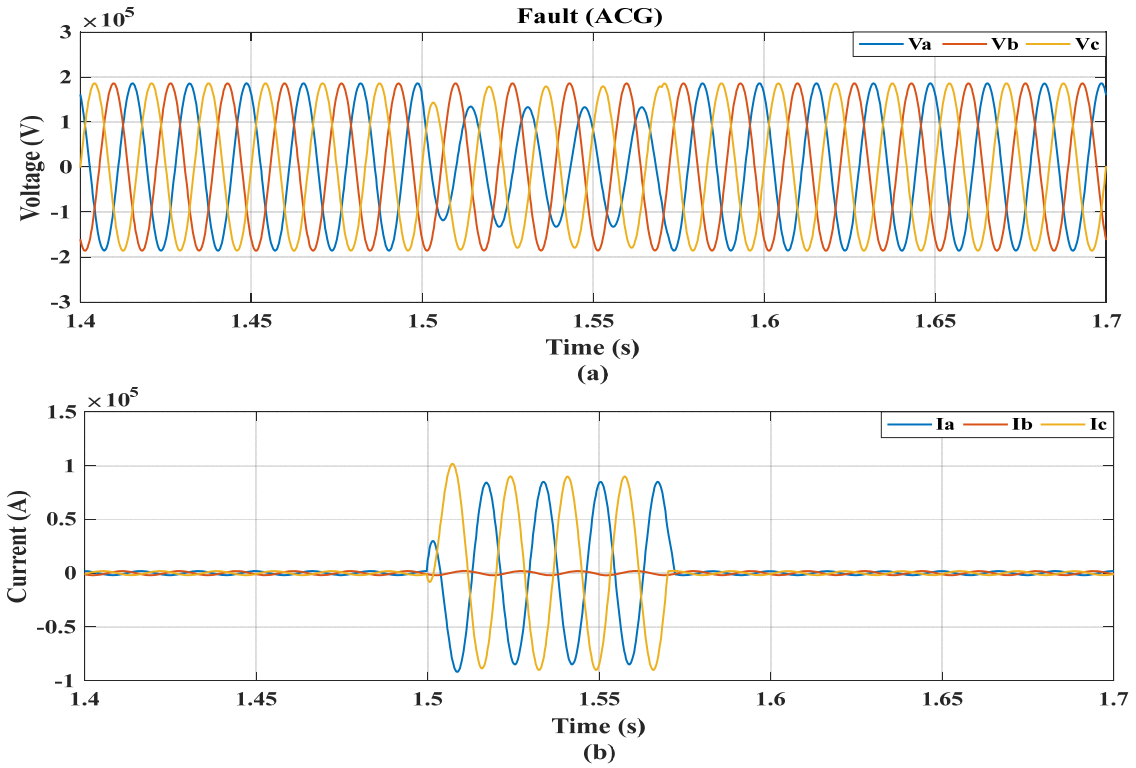


Fig. 4. 6: Three-phase voltage and current at ACG fault for transmission line system.

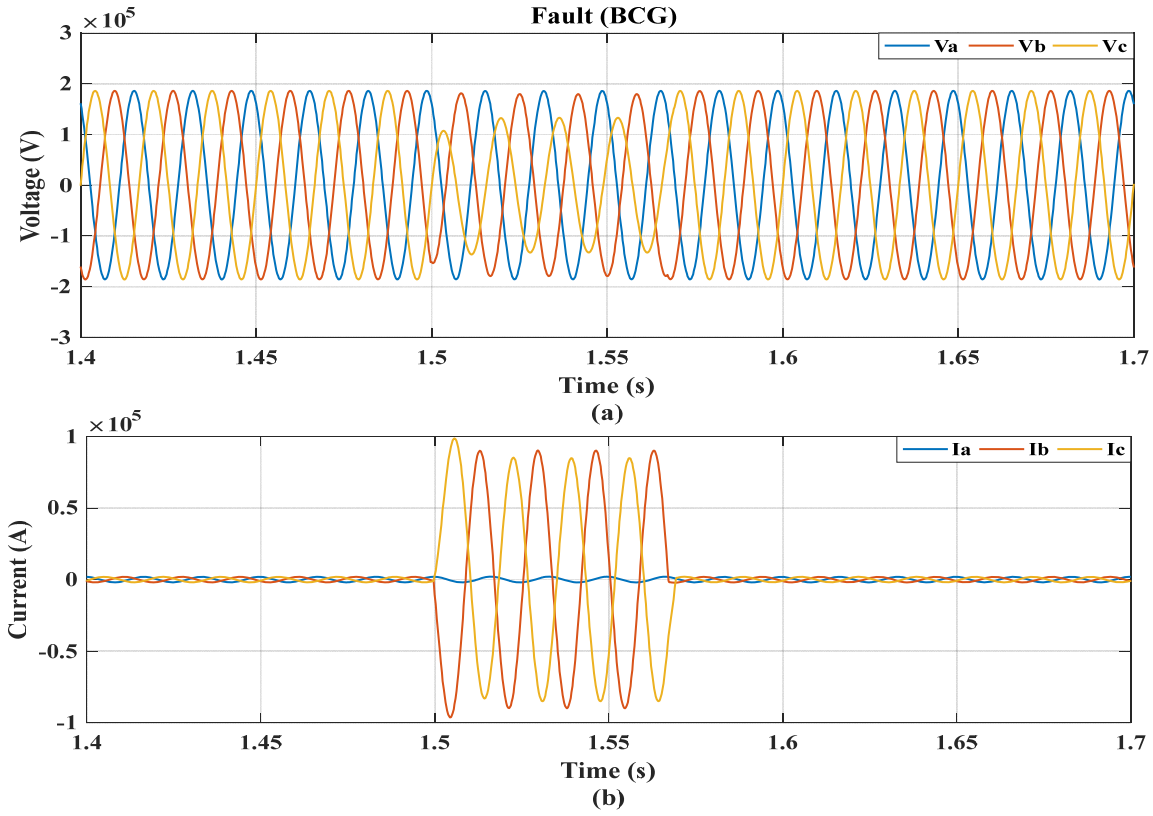


Fig. 4. 7: Three-phase voltage and current at BCG fault for transmission line system.

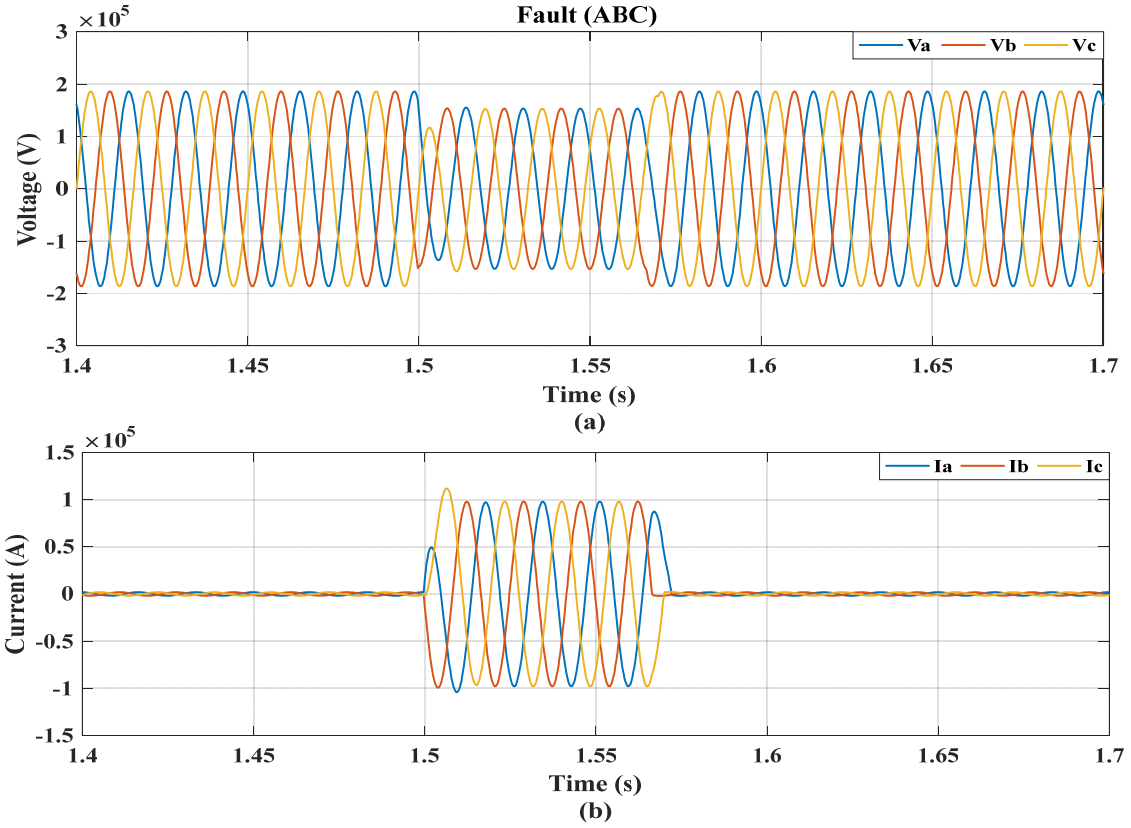


Fig. 4. 8: Three-phase voltage and current at ABC fault for transmission line system.

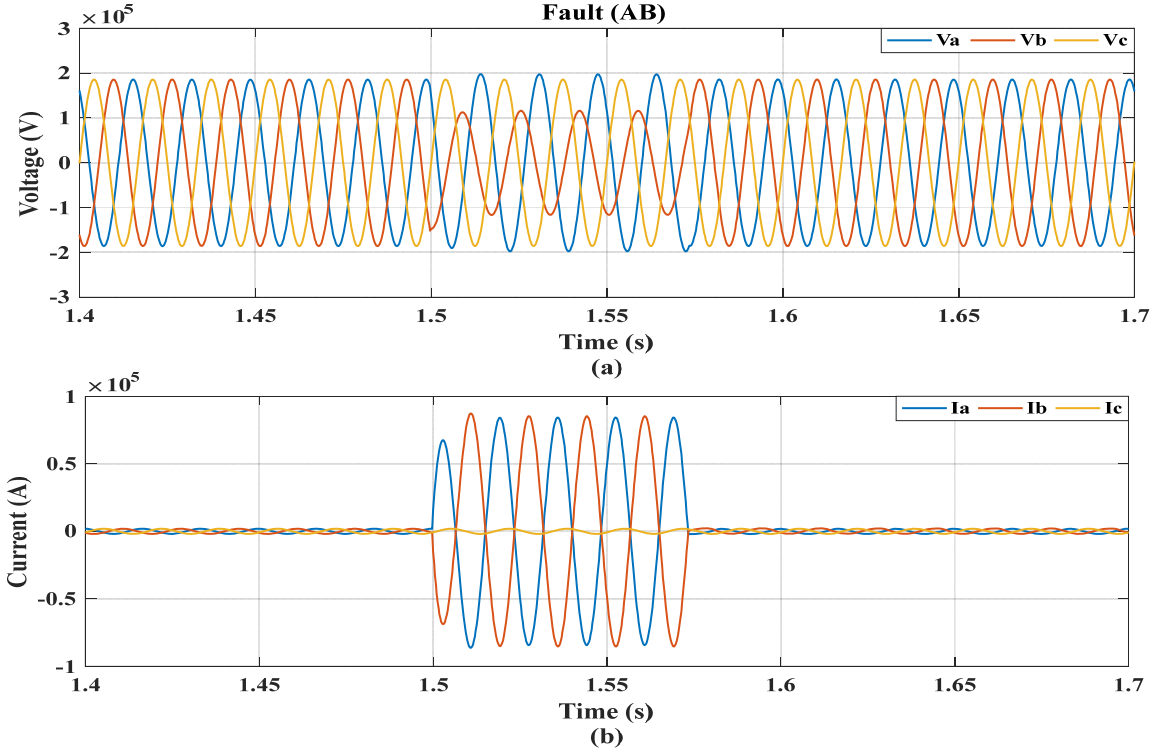


Fig. 4. 9: Three-phase voltage and current at AB fault for transmission line system.

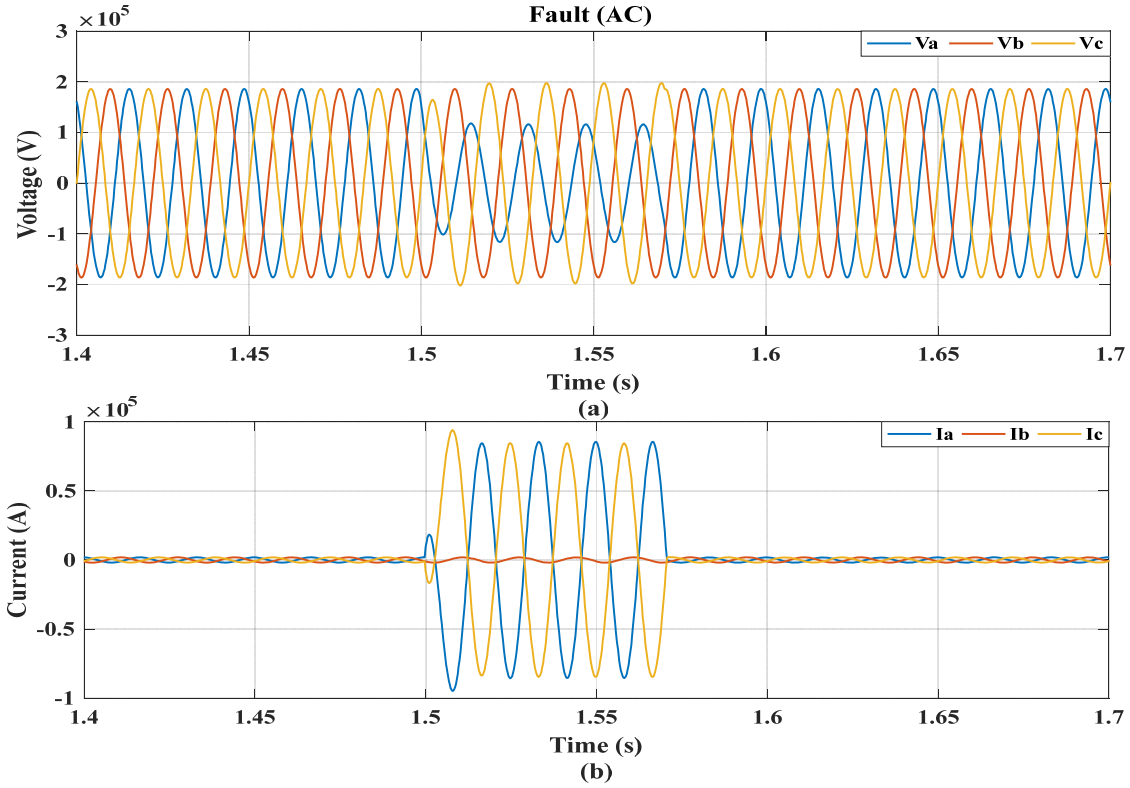


Fig. 4. 10: Three-phase voltage and current at AC fault for transmission line system.

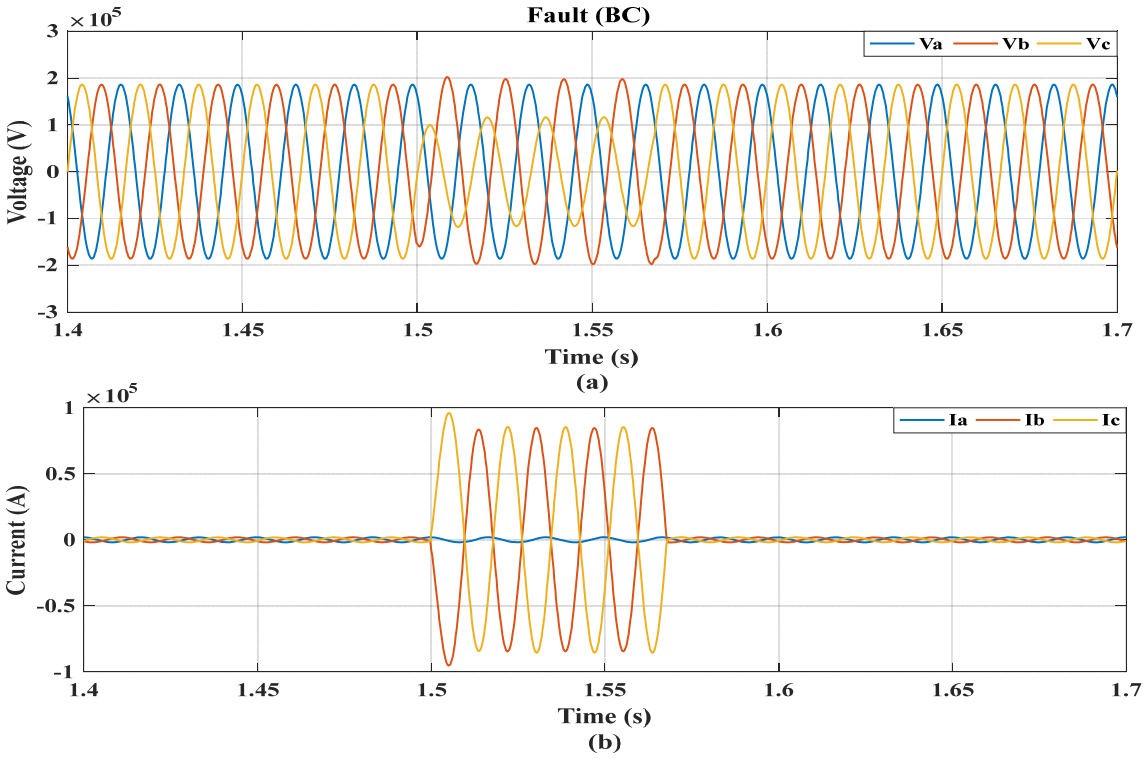


Fig. 4. 11: Three-phase voltage and current at BC fault for transmission line system.

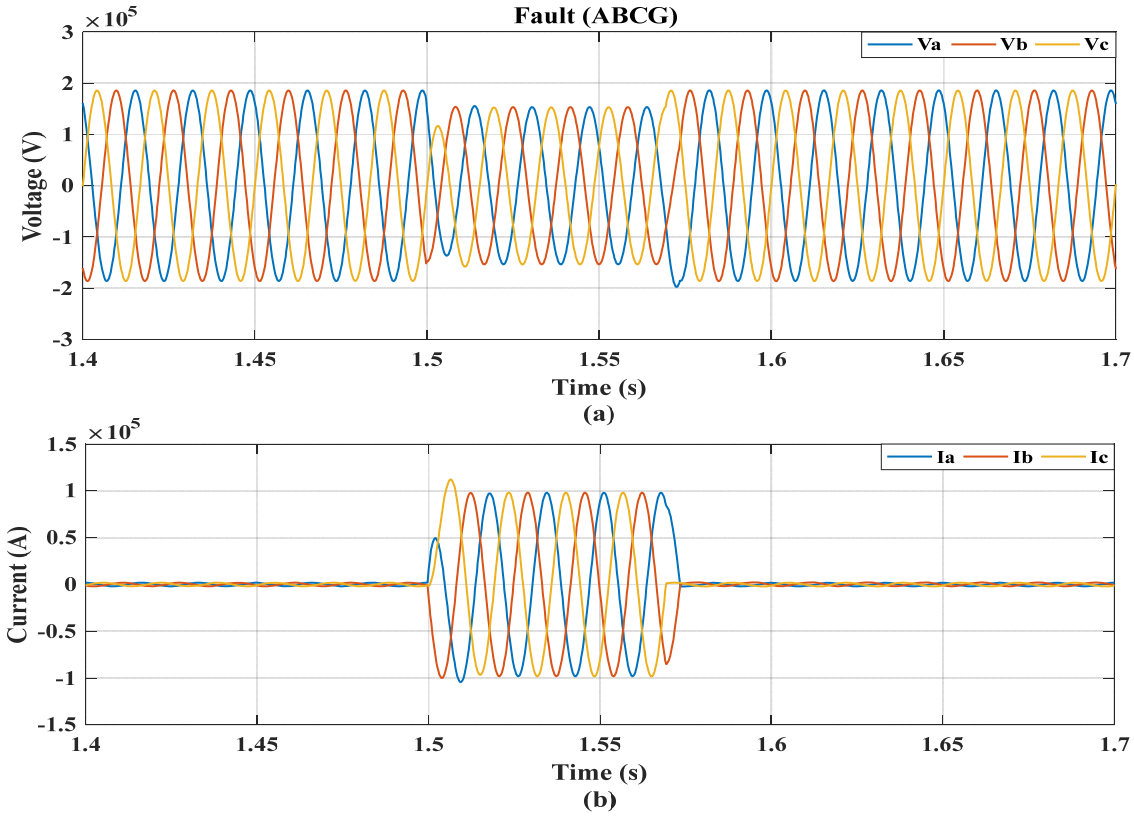


Fig. 4. 12: Three-phase voltage and current at ABCG fault for transmission line system.

4.2.3 Optimal Wavelet Combination in case of Transmission Line System

The harmony search algorithm is applied, as described in Chapter 3, to identify the most suitable wavelet function(s) and the optimal number of wavelet decomposition level(s). Table 4.7 outlines the parameter settings of the HSA used in this thesis, as suggested by Mahdavi et al., in [59].

Since the harmony search algorithm is a heuristic based technique, which means several runs may produce different results, the total number of trials is determined in this work to be 20 as it has been observed that any increase in the number of trials beyond this value does not provide much change in the HSA results.

Table 4. 7: Tuning parameters of Harmony Search Algorithm.

Parameter	value
Harmony Memory Size (H_s)	50
Harmony Memory Considering Rate (HMC_r)	0.95
Minimum Pitch Adjusting Rate (PA_{rmin})	0.35
Maximum Pitch Adjusting Rate (PA_{rmax})	0.99
Minimum Bandwidth (bw_{min})	1e-6
Maximum Bandwidth (bw_{max})	0.50
Maximum Iteration (MIr)	10,000

Table 4.8 shows the outcomes of implementing the HSA, where the HSA has identified the most suitable combination of decomposition levels, as well as, identified the most suitable wavelet functions for each identified wavelet decomposition level required for fault diagnosis application.

Table 4. 8: Optimal wavelet function and decomposition level on the transmission system.

Signal	Level	Wavelet Function	Occurrence (%)
Current	Cd_1	bior1.5	100
	Cd_2	bior1.5	100
	Cd_3	bior1.3	70
	Cd_4	db25	80
	Ca_4	sym4	75
Voltage	Cd_1	bior1.5	95
	Cd_2	bior1.5	95
	Cd_3	bior1.3	65
	Cd_4	db25	90
	Ca_4	db11	75

The selected wavelet functions and the wavelet decomposition levels, which has the most occurrence over the 20-trials, are listed in Table 4.8. The algorithm chooses four decomposition levels (i.e. Cd_1, Cd_2, Cd_3, Cd_4, and Ca_4) for current signal, and four decomposition levels (i.e.

Cd_1, Cd_2, Cd_3, Cd_4, and Ca_4) for voltage signal. In the case of the current signal, the HSA has identified the wavelet function (bior1.3) for decomposition level detail-3 (Cd_3) and the wavelet function (sym4) for decomposition level approximation-4 (Ca_4). On the other hand, in the case of the voltage signal, the HSA has identified the wavelet function (bior1.3) for decomposition level detail-3 (Cd_3), the wavelet function (db11) for decomposition level approximation-4 (Ca_4). Figs. 4.13 and 4.14 show the selected combination of the wavelet functions for the current and voltage signals in the transmission line test system.

4.2.4 Fault Classification Results of Transmission Line System

The Monte Carlo stratified k-fold cross validation, as described in Chapter 3, is applied in order to examine the performance of the proposed automated wavelet-based fault detection and diagnosis technique in transmission line test system.

The classes in each group stratified with the same proportion of the original data as shown in Table 4.9. In order to assess the performance of the proposed approach in the transmission line test system, the results of the classification accuracies in the case of the proposed AWBFDD technique using both DT and k-NN classifiers are evaluate and are listed in Tables 4.10 and 4.11.

Moreover, these results are compared to those obtained in the previous work (i.e. as in [26]-[28], [30], [31], [34], [36], and [38]). The Tables also show the misclassified records and the 95% Confidence Interval (*CI*) of the classification accuracies.

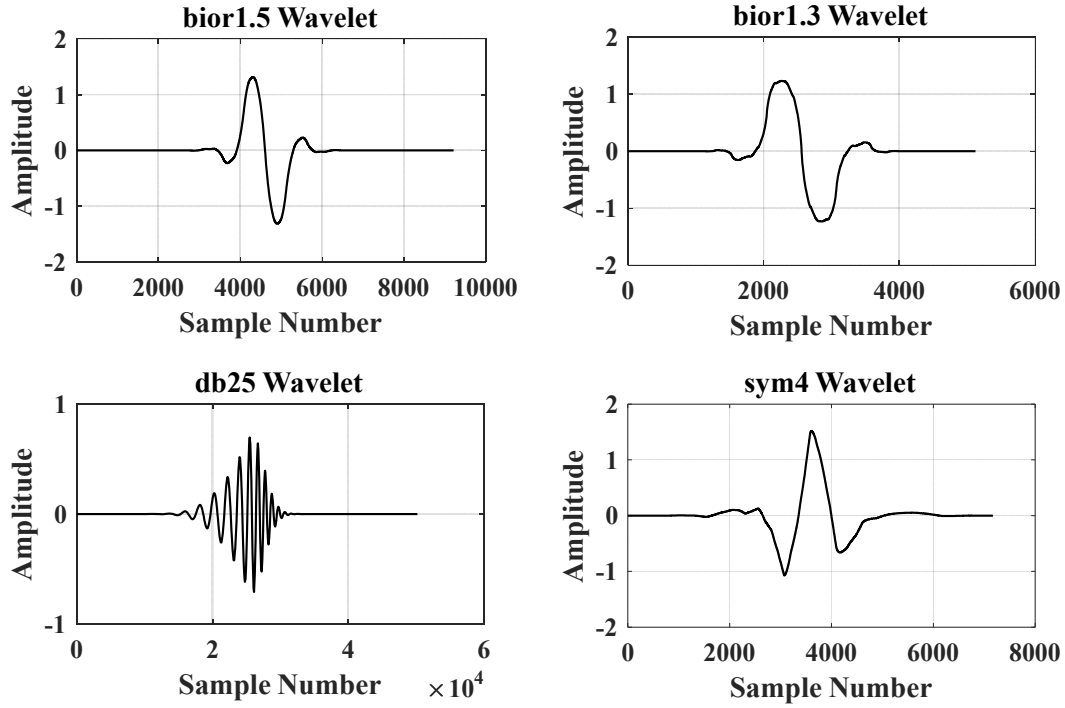


Fig. 4. 13: Selected combination of wavelet functions for current signal in transmission line system.

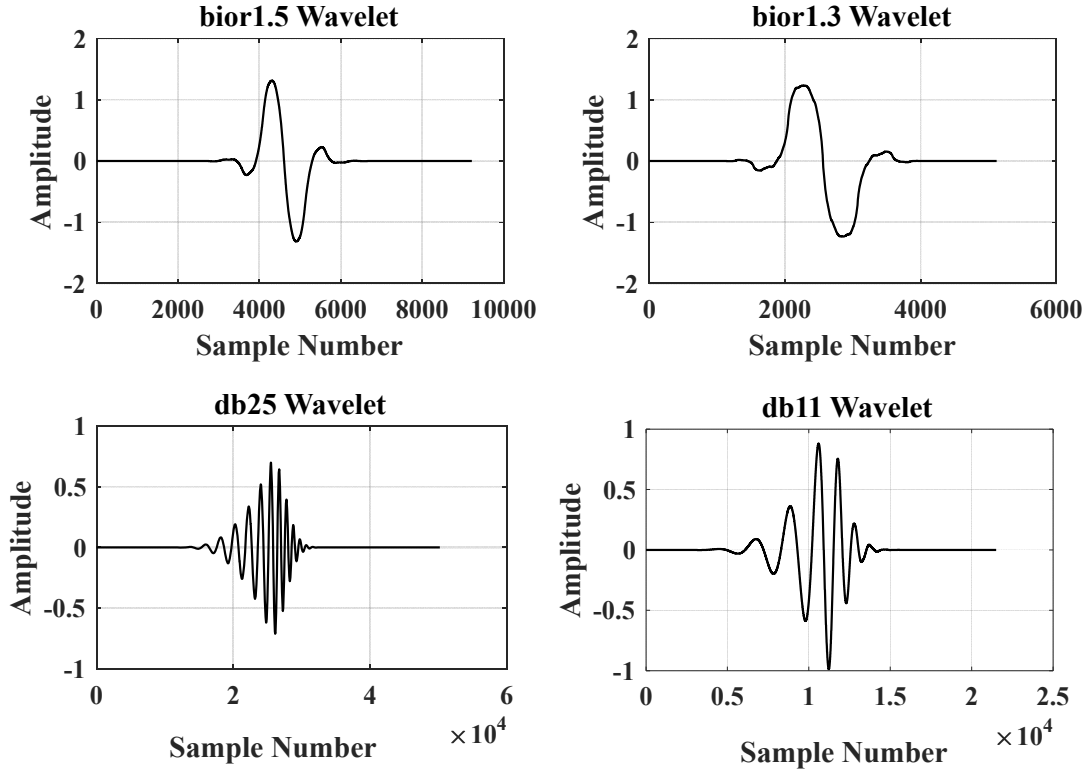


Fig. 4. 14: Selected combination of wavelet functions for current signal in transmission line system.

Table 4. 9: Class represented proportion of transmission line system.

Class	Number of records	Class proportion (%)
AG	48	9.09
BG	48	9.09
CG	48	9.09
AB	48	9.09
BC	48	9.09
AC	48	9.09
ABG	48	9.09
BCG	48	9.09
ACG	48	9.09
ABC(G)	48	18.18
Total	528	100

Table 4. 10: DT Classifier accuracy in transmission line system.

Ref.	Wavelet	Level	Signal	Misclassified records	Accuracy (%)	Confidence Intervals (%)
Proposed technique (AWBFDD)				0	100	99.00 - 100
[26]	db3	A1	I	9	98.24	96.80 - 99.10
[27]	sym2	D1,A1,A2,A3	I	8	98.43	97.00 - 99.20
[28]	db4	D1,D2,D3,D4	I& V	5	99.02	97.80 - 99.60
[30]	db4	D1	I	36	93.15	90.30 - 94.70
[31]	db8	D3	I	12	97.75	96.10 - 98.70
[34]	db8	A3	I	7	98.62	97.30 - 99.40
[36]	db4	D1	I & V	126	76.12	70.30 - 77.80
[38]	db1	D1	I & V	21	96.01	93.80 - 97.20

Table 4. 11: K-NN Classifier accuracy in transmission line system.

Ref.	Wavelet	Level	Signal	Misclassified records	Accuracy (%)	Confidence Intervals (%)
Proposed technique (AWBFDD)				0	100	99.00 - 100
[26]	db3	A1	I	7	98.71	97.30 - 99.40
[27]	sym2	D1,A1,A2,A3	I	2	99.55	98.60 - 99.90
[28]	db4	D1,D2,D3,D4	I& V	3	99.40	98.30 - 99.80
[30]	db4	D1	I	28	94.72	92.20 - 96.1
[31]	db8	D3	I	11	97.93	96.30 - 98.80
[34]	db8	A3	I	5	98.96	97.80 - 99.60
[36]	db4	D1	I & V	51	90.25	86.70 - 91.90
[38]	db1	D1	I & V	4	99.19	98.10 - 99.70

The results revealed that, in the case of DT, the minimum and maximum absolute difference in the classification accuracy between the proposed technique (HSA-based) and the approaches in previous work are 0.98% and 23.88% in the case of [28] and [36], respectively. It is clear that the use of the proposed approach based on HSA significantly improves the classification accuracy, compared to if an only one arbitrary wavelet function was used as the mother wavelet in feature extraction process. On the other hand, according to Table 4.11, in the case of K-NN, the minimum and maximum absolute differences in the classification accuracy between the proposed (HSA-based) and the approaches in the previous work are 0.45% and 9.75% in the case of [27] and [36], respectively. Also, the proposed HSA-based approach provides higher minimum and maximum values of the computed confidence interval as listed in Tables 4.10 and 4.11 compared to those reported in previous work, which shows the robustness of the proposed method.

Figs. 4.15 and 4.16 show the number of misclassified records for 1-trial of the 1,000 Monte Carlo trials with the DT and K-NN classifiers of 10-fold cross. Fig. 4.15 revealed that, in the case of DT, the minimum and maximum variances in the misclassified records using the approaches in

previous work are 0.50 and 7.82 in the case of [28] and [38], respectively. On the other hand, according to Fig. 4.16, in the case of K-NN, the minimum and maximum variances in the misclassified records using the approaches in previous work are 0.23 and 2.76 in the case of [26] and [38], respectively. The number of misclassified records using the proposed AWBFDD technique in both K-NN and DT classifier remain zero for all k-fold with variance zero. It can be observed that, the proposed technique is insensitive to the selection of k, where none of the faulted cases were misclassified for all k-fold.

Table 4.12 shows the confusion matrix between the actual and the predicted classes in the case of the proposed HSA-based approach. The results indicate that the proposed algorithm, which uses the HSA for the selection of optimal wavelet functions and levels, was able to correctly classify all fault cases, as listed in Table 4.6. The results revealed that the harmony search algorithm identified the optimal wavelet functions and was able to find the optimal number of the decomposition levels, while at the same time identifying the required signal in the analysis (i.e. voltage and current). Therefore, the discrete wavelet transform was successful in capturing the most prominent features from the signal using the identified wavelet functions and the wavelet decomposition levels by HSA technique. As a result, the purpose of the classifier, which is used to automate the fault classification process became easier, since the features of each fault type are became more descriptive/distinguish for each fault. Finally, it has been concluded that the use of only one wavelet may not be appropriate for fault classification problems, hereafter the use of a wavelets combination may be more suitable.

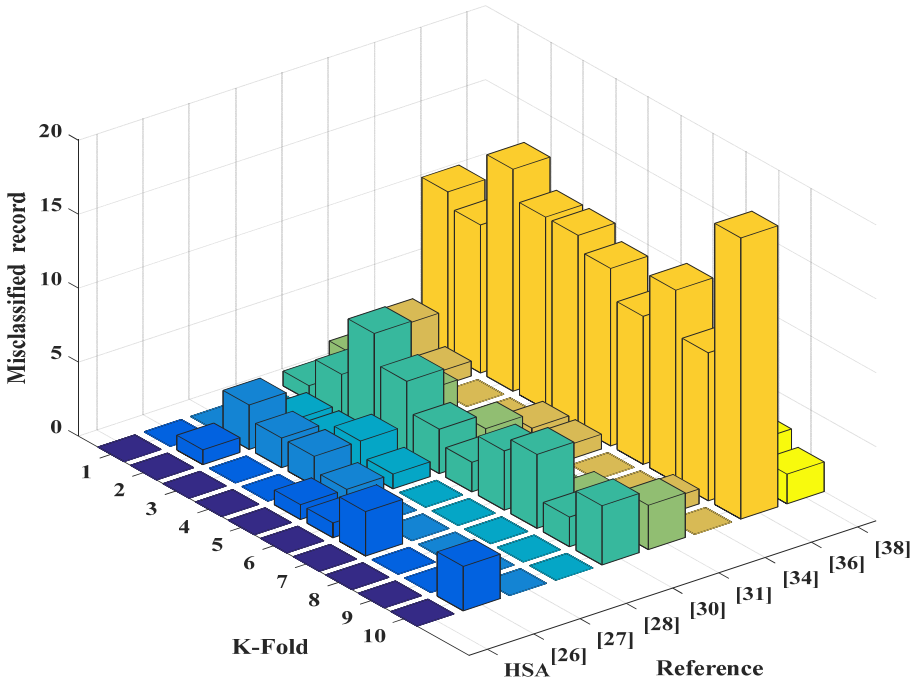


Fig. 4. 15: Misclassified records for 10-folds in case of 1-trial of DT on the transmission line system.

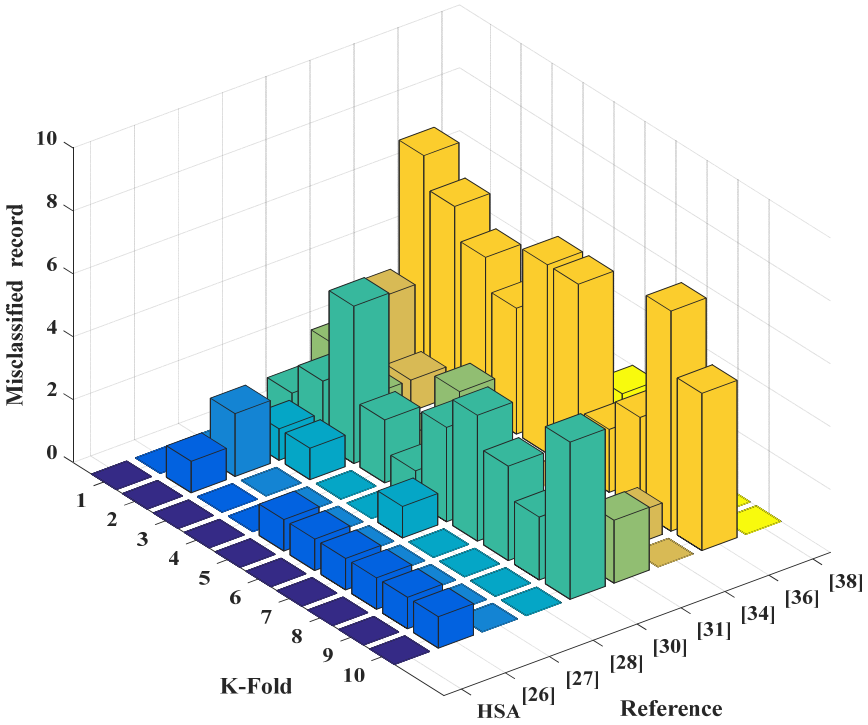


Fig. 4. 16: Misclassified records for 10-folds in case of 1-trial of K-NN on the transmission line system.

Table 4. 12: Confusion matrix in transmission line system.

		Predicted class									
		A-G	B-G	C-G	AB	BC	AC	AB-G	BC-G	AC-G	ABC(G)
Actual class	A-G	48	0	0	0	0	0	0	0	0	0
	B-G	0	48	0	0	0	0	0	0	0	0
	C-G	0	0	48	0	0	0	0	0	0	0
	AB	0	0	0	48	0	0	0	0	0	0
	BC	0	0	0	0	48	0	0	0	0	0
	AC	0	0	0	0	0	48	0	0	0	0
	AB-G	0	0	0	0	0	0	48	0	0	0
	BC-G	0	0	0	0	0	0	0	48	0	0
	AC-G	0	0	0	0	0	0	0	0	48	0
	ABC(G)	0	0	0	0	0	0	0	0	0	96

4.3 Micro-Grid Distribution System

4.3.1 Test-System Description of Micro-Grid Distribution System

The Consortium for Electric Reliability Technology Solutions (CERTS), which is outlined in [67] and [68], as shown in Fig. 4.17 is of a Micro-Grid distribution system (MGDS) and it was used as a test system in this thesis. The CERTS Micro-Grid system can be operated in two modes (i.e. Grid Connected (GC), or in Non Grid-Connected (NGC) islanded mode). The CERTS Micro-Grid is considered as a part of electric power distribution system, which is supplied from the secondary (low voltage side) of a three-phase distribution transformer rated 13.8/0.48 kV in the case of the grid connected mode or supplied from the distributed energy resources in the case of the islanded mode. As shown in Fig. 4.17, the CERTS Micro-Grid distribution system consists of three DERs, two solar photovoltaic sources (DER-PV1, DER-PV2) and one battery energy storage source (DER-Bt.S). Four loads (L3, L4, L5, and L6) are distributed along the system. Tables 4.13 - 4.16 summarize the CERTS Micro-Grid system parameters.

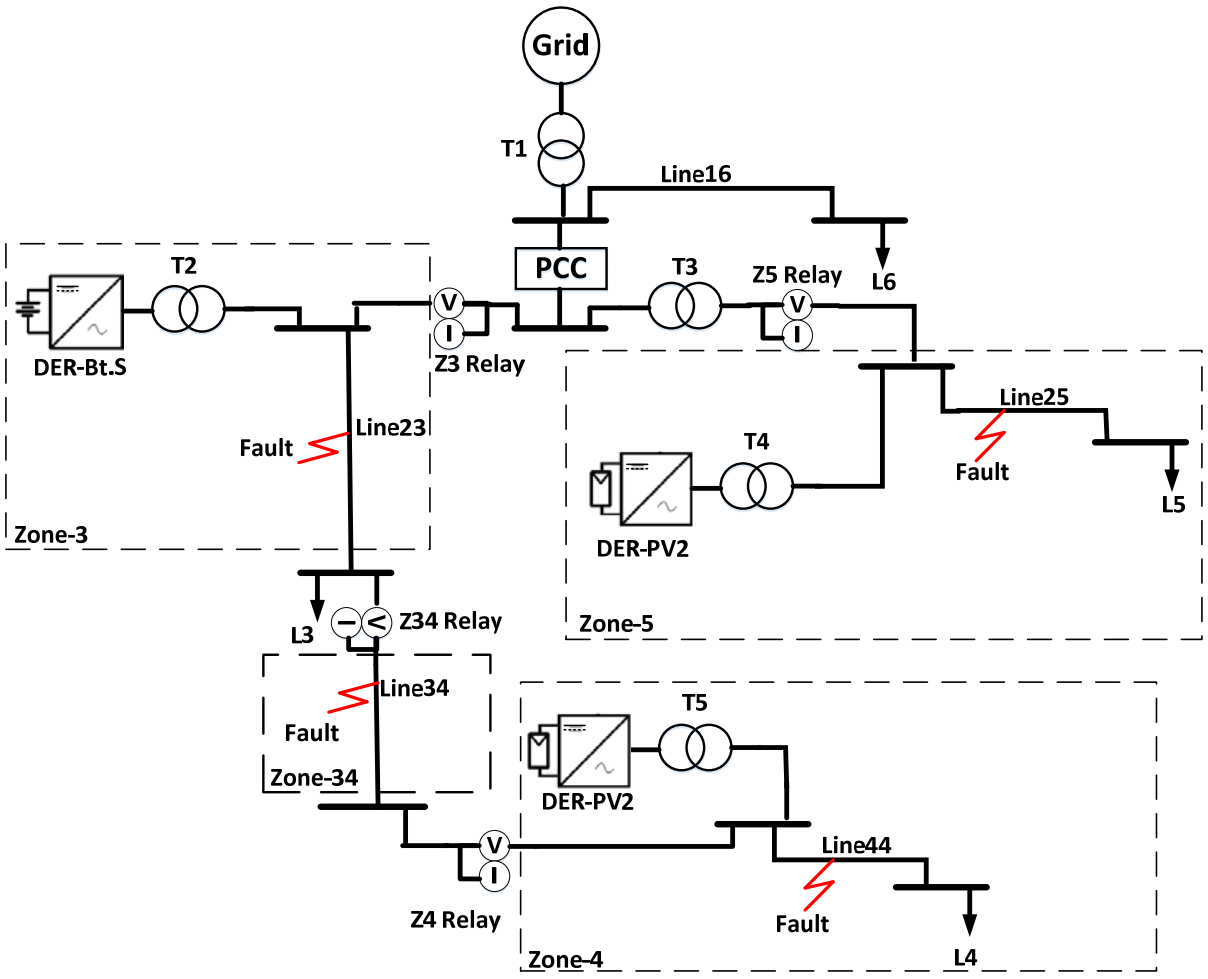


Fig. 4. 17: CERTS Micro-Grid Distribution System structure.

Table 4. 13: CERTS Micro-Grid Distribution System: Transformer parameters.

	Rated Power (kVA)	High-Voltage Side (kV)	Low-Voltage Side (kV)	X/R (%)	Z (%)
T1	2500	13.8 - D	0.48 -Gr. W	6	5
T2, T3, T4, and T5	500	0.48 - Gr.Wye	0.48 - Gr. W	5	1

Table 4. 14: CERTS Micro-Grid Distribution System: Load parameters.

Load	Model	Rated Real Power (kW)	Rated Reactive Power (kVAr)
L3, andL4	Y-PQ	90	45
L5	Y-PQ	90	-40
L6	Y-PQ	90	-20

Table 4. 15: CERTS Micro-Grid Distribution System: Lines parameters.

Line	Phasing	Conductor Size	Length (m)
Line16, Line25, Line23, and Line44	ABC-N	AWG# 2	68.58
Line34	ABC-N	AWG# 2/0	22.86

Table 4. 16: CERTS Micro-Grid Distribution System: DER parameters.

DER Parameter	value
DER-PV1, DER-PV1, and DER-PV1	100 kW, unity power factor (3-phase capacitor bank (15 kVA))

The switch at the Point of Common Coupling (PCC) is mainly used to change the mode of operation from grid connected mode to a non-grid connected (islanded) mode, or vise-versa. This switch is responsible for connecting/disconnecting the area, which includes the three DERs sources with the grid utility in the grid/non-grid connected modes. The battery storage source and the photovoltaic source are interfaced to the system through a Voltage Source Converter (VSC).

The voltage source converters are widely used nowadays in various fields. The gating signals that is applied to the switching devices are produced by the Pulse Width Modulation (PWM) technique. In the PWM technique, the gating signals are applied to turn on the converter with comparing a high frequency repeating signal (usually triangular) with a reference signal (usually the output of a controller). In this thesis, three different control schemes were used for voltage source converter:

1) ***Current-mode control with real/reactive power controller in dq-frame***: For Battery storage DER-Bt.S in both grid/islanding modes.

Essentially, most of the voltage source converters as shown in Fig. 4.18 are controlled in a way that provides real and reactive power control. Usually the VSC systems are sinusoidal three-

phase systems since the control of the real and reactive power should be independent from each other and this usually requires controlling all three phases instantly. The three-phase system should be transferred into a rotating single-phase system. This is performed by transforming all the parameters from the *abc* frame to the rotating *dq*-frame as described in Fig 4.19. The *dq*-frame was introduced as a solution to simplify the design of controllers where the signals are time varying.

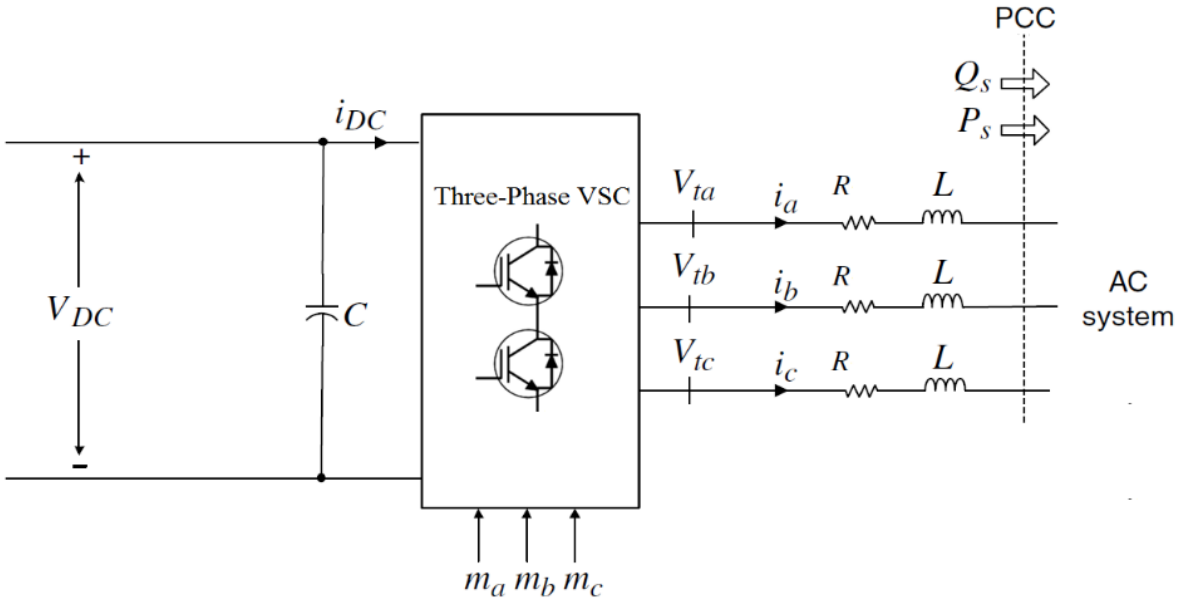


Fig. 4. 18: Three-Phase Voltage Source Converter (VSC) System [69].

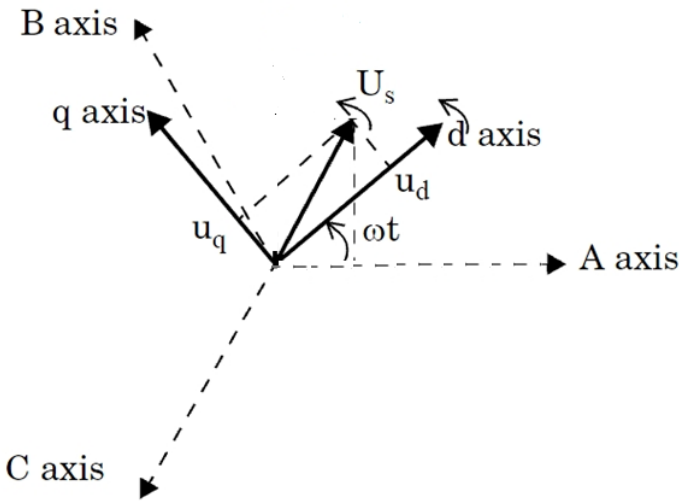


Fig. 4. 19: Vectors representation in d-q reference frame

In the current-mode control of the real/reactive power controller strategy, the real and reactive power in the *dq*-frame were controlled independently with the controlling of the currents in *dq*-frame. The voltage samples and the current samples were transformed to corresponding *dq0*-frame signals based on [69]:

$$\begin{bmatrix} U_d \\ U_q \\ U_0 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ 1/2 & 1/2 & 1/2 \end{bmatrix}\begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix} \tag{4.1}$$

The expressions of the real and reactive average power in the *dq*-frame are expressed as follows [69]:

$$P_s(t) = \frac{3}{2}\left(v_{sd}(t)\, i_d(t) + v_{sq}(t) i_q(t)\right) \tag{4.2}$$

$$Q_s(t) = \frac{3}{2}\left(-v_{sd}(t)\, i_q(t) + v_{sq}(t) i_d(t)\right) \tag{4.3}$$

Where, $v_{sq}(t)$ and $v_{sd}(t)$ are the dq-frame components of the AC-side voltages, and $i_q(t)$ and $i_d(t)$ are the dq-frame components of the AC-side currents. In the Equations (4.2) and (4.3), the real and reactive power are not decoupled since both current components are present in both equations. However, if $v_q(t)$ is kept zero by the Phase looked loop (PLL) in the steady-state condition, then the real and reactive power equations can be rewritten as follows [69]:

$$P_s(t) = \frac{3}{2}(v_{sd}(t) i_d(t)) \tag{4.4}$$

$$Q_s(t) = \frac{3}{2}\left(-v_{sd}(t) i_q(t)\right) \tag{4.5}$$

Therefore, based on (4.4) and (4.5), it is evident that by independently controlling i_d and i_q, the real and the reactive power $P_s(t)$ and $Q_s(t)$ can be controlled independently. Given the reference real and reactive powers, the currents $i_{dref}(t)$ and $i_{qref}(t)$ can be calculated according to the following equations [69]:

$$i_{dref}(t) = \frac{2}{3v_{sd}}\left(P_{sref}(t)\right) \tag{4.6}$$

$$i_{qref}(t) = -\frac{2}{3v_{sd}}\left(Q_{sref}(t)\right) \tag{4.7}$$

Since the d-component of the AC-side voltage v_{sd} in steady state is a DC variable. Furthermore, the d-q reference currents i_{dref} and i_{qref} are DC variables in case of the reference power signals P_{sref} and Q_{sref} are DC. Fig. 4.20 shows the full schematic diagram of a current-mode control technique of real/reactive power controller in dq-frame.

Firstly, the reference currents in dq-frame i_{dref} and i_{qref} are obtained from the reference powers according to the Equations (4.6) and (4.7). After obtaining the reference currents, they are then processed by the compensator, which is presented in [69]. Finally, the PWM generator was used to generate the firing signals of the VSC-IGBTs. Figs. 4.21 – 4.25 show the modeling of the battery storage source (DER-Bt.S.) system in the PSCAD software, with current-mode control of real/reactive power controller strategy.

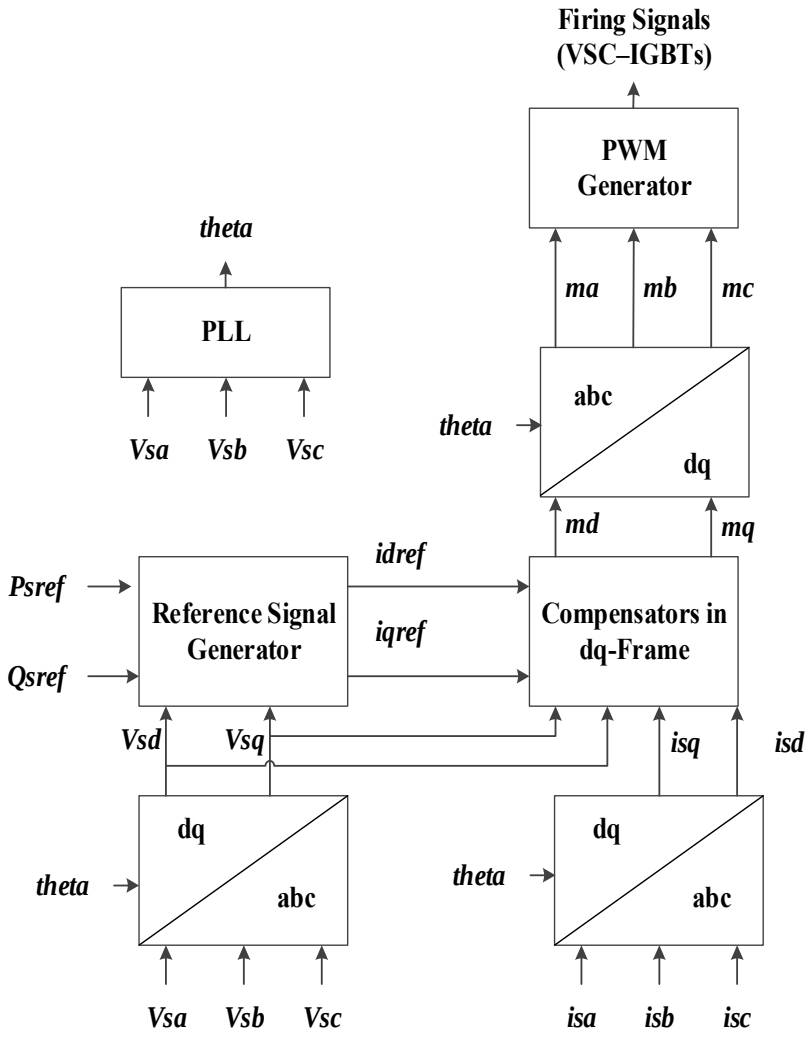


Fig. 4. 20: Schematic diagram of a current-mode control of real/reactive power controller in dq-frame.

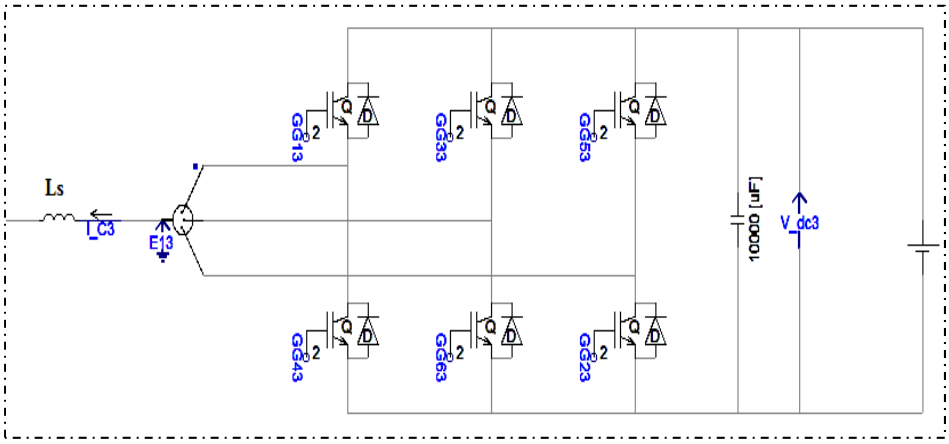


Fig. 4. 21: PSCAD model of the 3-phases VSC Block for Battery Storage source (DER-Bt.S.).

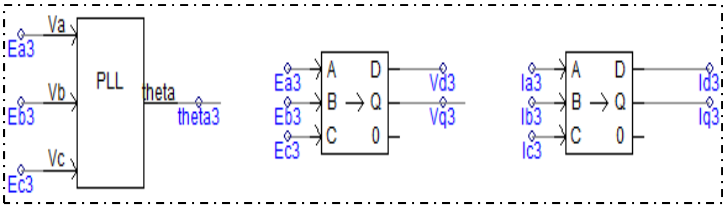


Fig. 4. 22: PSCAD model of the ABC-dq frame transformation Block for Battery Storage source (DER-Bt.S.).

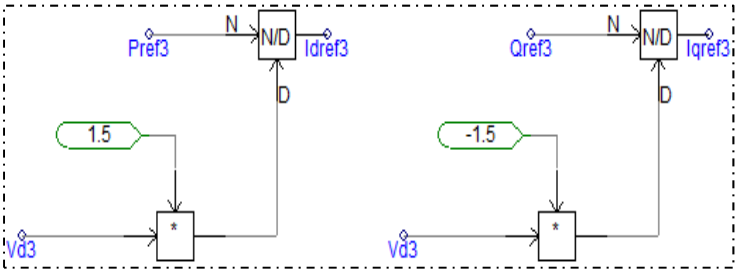


Fig. 4. 23: PSCAD model of reference signals generator block in d-q frame for Battery Storage source (DER-Bt.S.).

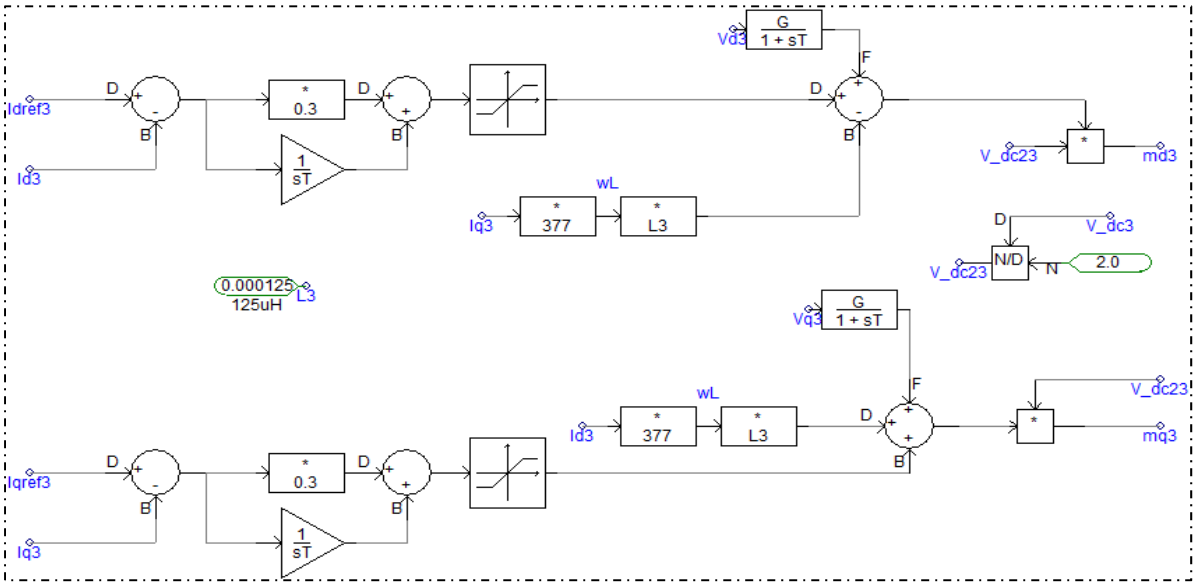


Fig. 4. 24: PSCAD model of the compensator block in d-q frame for Battery Storage source (DER-Bt.S.).

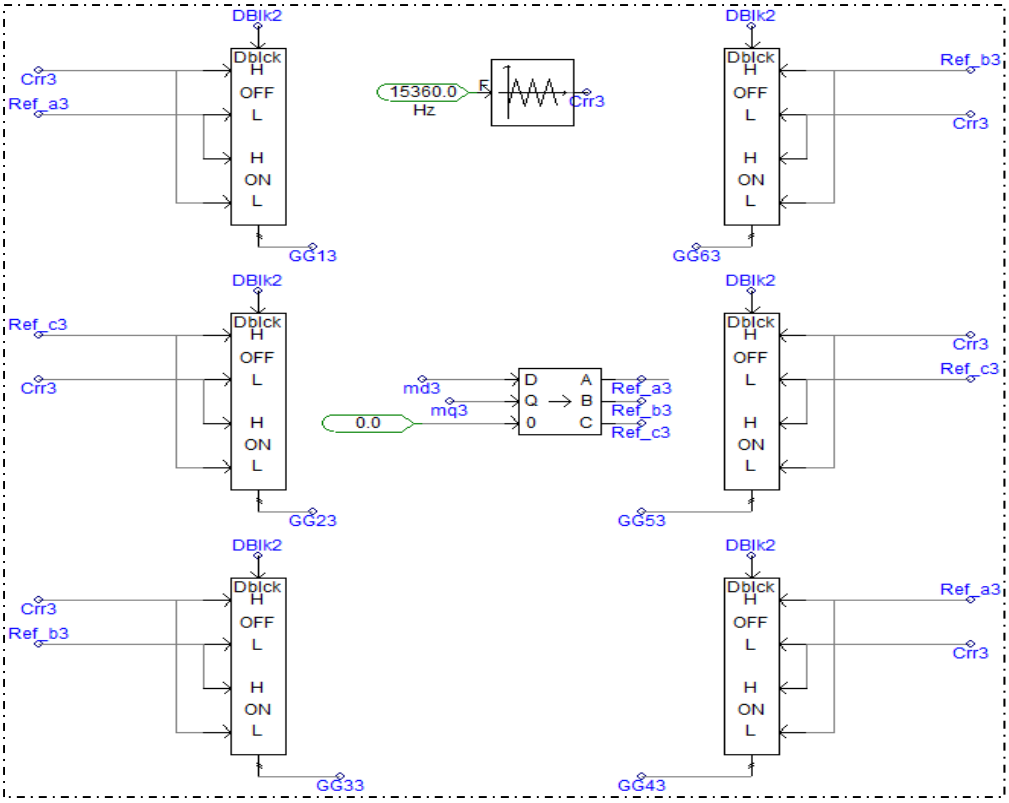


Fig. 4. 25: PSCAD model of the PWM Generator Block for Battery Storage source (DER-Bt.S.).

2) ***Modified current mode control with DC link voltage controller in dq-frame***: For DER-PV1 in grid mode only and for DER-PV2 in both grid/islanding modes.

The modified current mode control with DC link voltage controller in the *dq*-frame control strategy, which is presented in [70], is considered as a modified version of the current-mode control. This modified version is mainly applied for the sake of controlling the power factor of the solar photovoltaic system, as well as controlling the voltage of DC link in the converter, and the output power (i.e. real power) of the solar photovoltaic system. In addition, the DC link voltage-control strategy ensures the stability operation of the solar photovoltaic system and also ensures the safe operation of the voltage-source converter of the solar photovoltaic system.

Fig. 4.26 shows the full schematic diagram of the modified current-mode control with DC link voltage controller in dq-frame method. Firstly, i_{dref} and i_{qref} are obtained from the reference

reactive powers Q_{sref}, and the reference DC link voltage V_{dcref}. Following this step, the reference currents are then processed by the compensator as presented in [69].

Finally, the PWM generator was used to generate the firing signals of VSC-IGBTs. Figs. 4.27 – 4.31 show the modelling of the photovoltaic source-2 (DER-PV2) system in the PSCAD software, which were developed in this work, with the modified current-mode control with DC link voltage controller in dq-frame method.

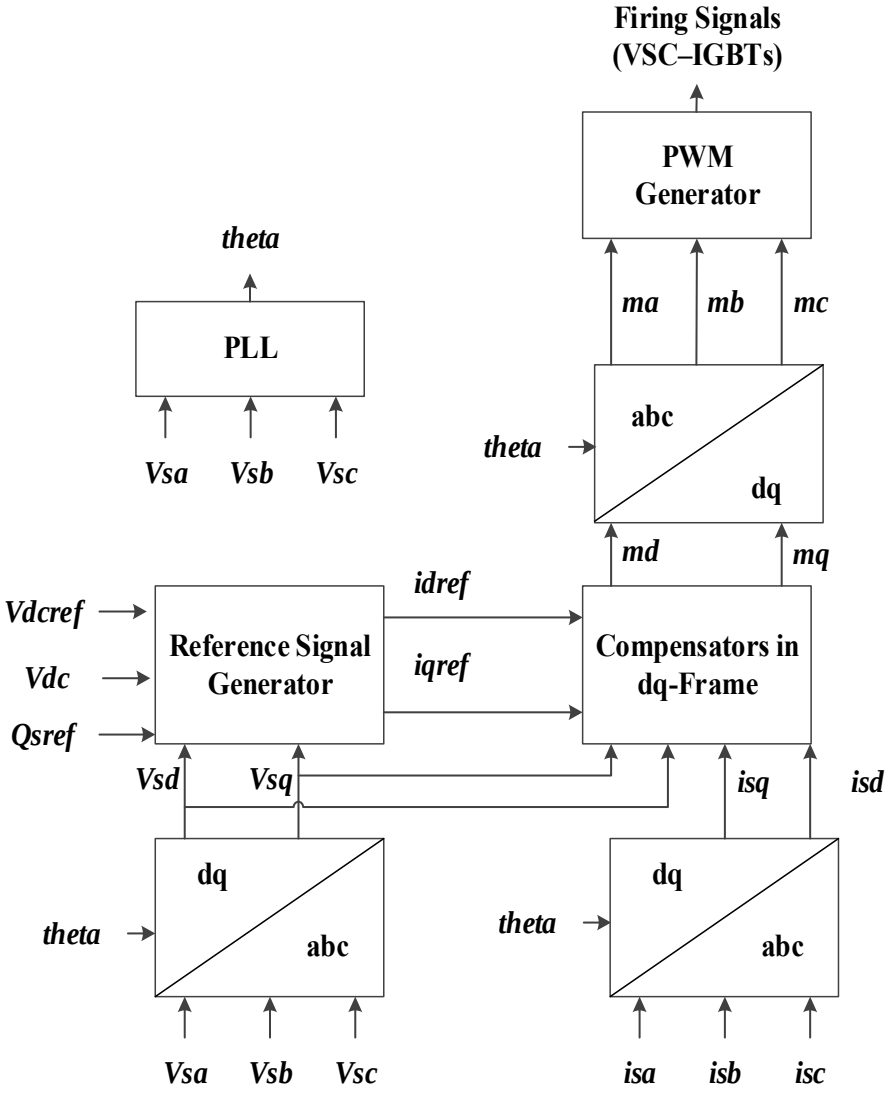


Fig. 4. 26: Schematic diagram of the modified current mode control with DC link voltage controller in dq-frame.

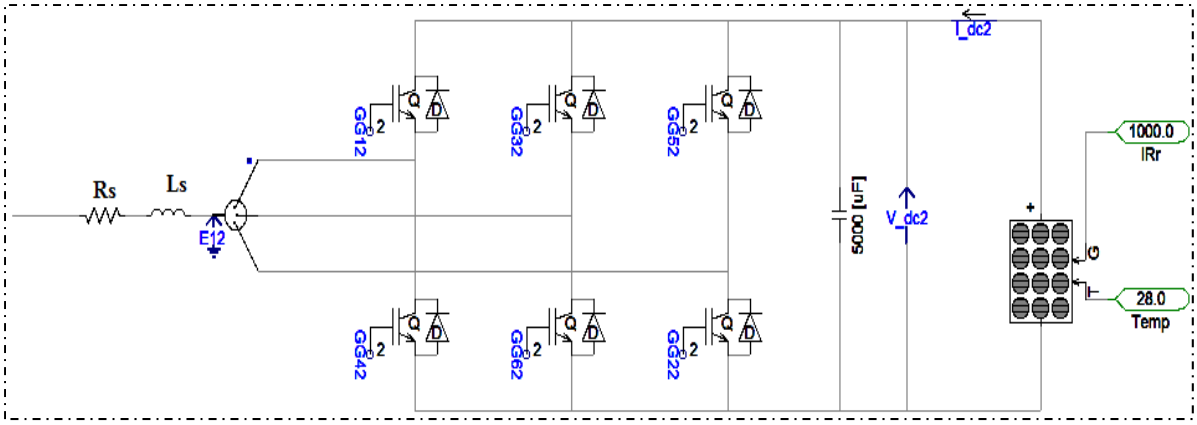


Fig. 4. 27: PSCAD model of the 3-phases VSC Block for solar photovoltaic source-2 (DER-PV2).

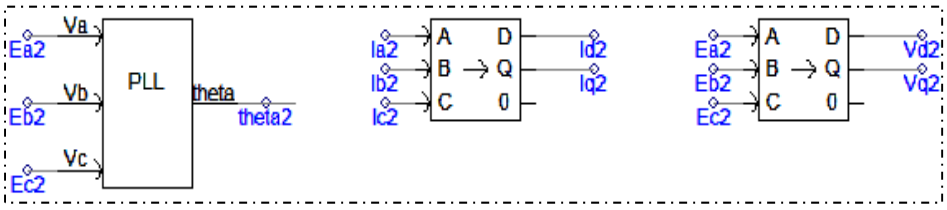


Fig. 4. 28: PSCAD model of the ABC-dq frame transformation Block for solar photovoltaic source-2 (DER-PV2).

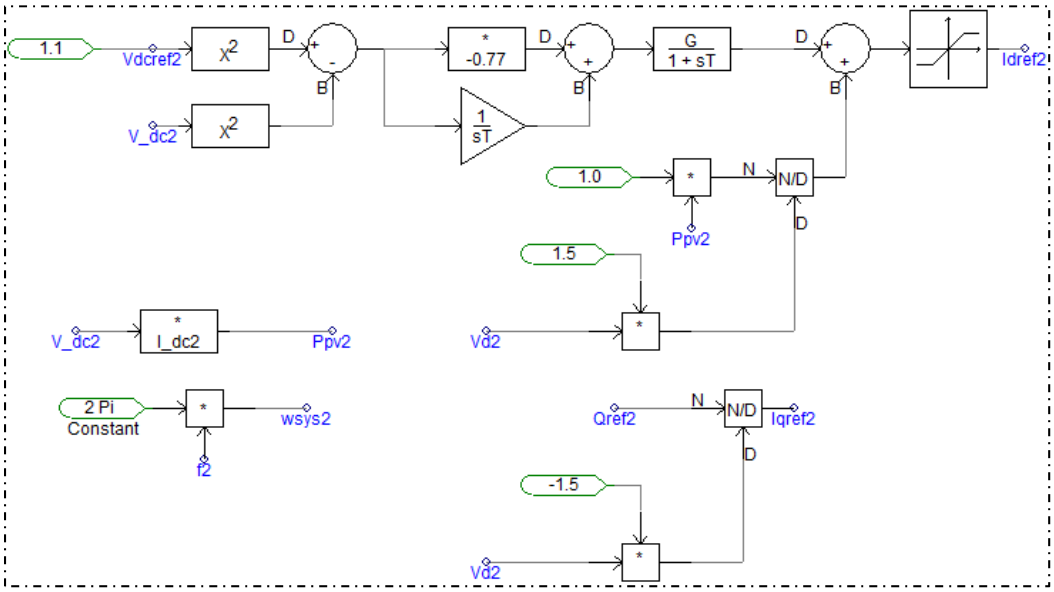


Fig. 4. 29: PSCAD model of the reference signals generator block in d-q frame for solar photovoltaic source-2 (DER-PV2).

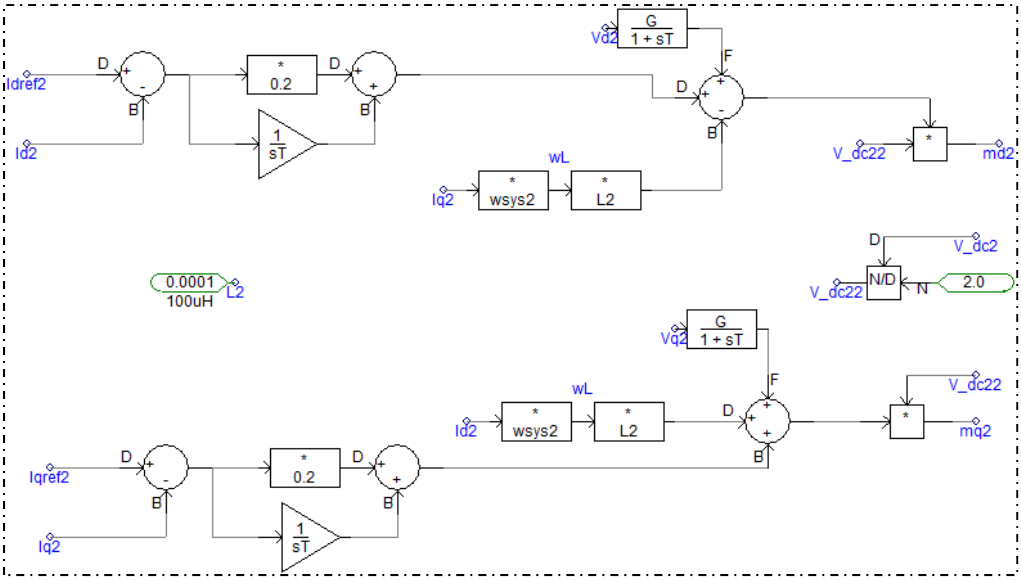


Fig. 4. 30: PSCAD model of the compensator block in d-q frame for solar photovoltaic source-2 (DER-PV2).

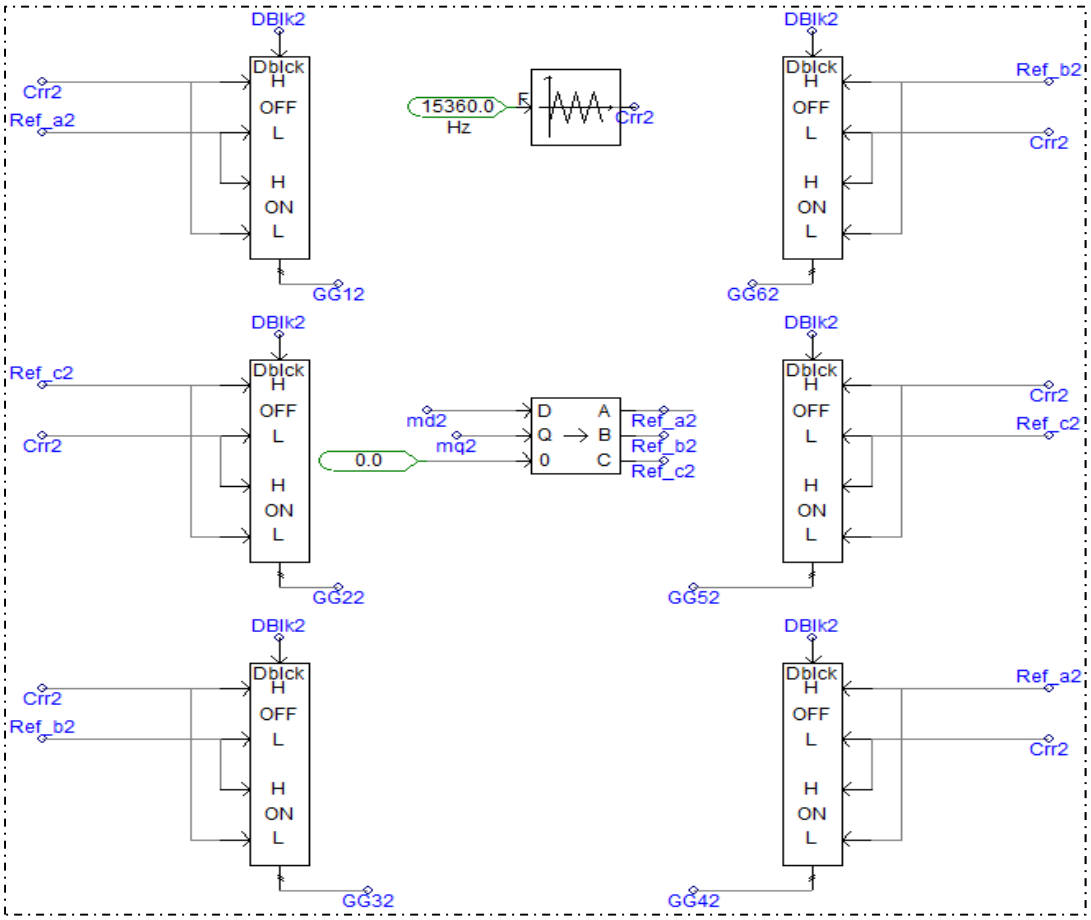


Fig. 4. 31: PSCAD model of the PWM Generator Block for solar photovoltaic source-2 (DER-PV2).

3) ***Frequency mode control in dq-frame***: For DER-PV1 in islanding mode only.

In the non-grid-connected (islanded) mode, the operating frequency and the voltage are not imposed by the AC system. Consequently, the operating frequency and the voltage need to be controlled by the VSC system, that control strategy class is denoted by controlled frequency VSC strategy. In this thesis, the solar Photovoltaic source-1 (DER-PV1) was controlled using two different strategy, the first strategy is the modified current-mode control in the case of grid-connected mode operation and the second strategy is the frequency mode control in the case of non-grid-connected mode operation. Fig 4.32 shows the supervisory control scheme for the operation in both grid-connected and non-grid-connected modes for the solar photovoltaic source-1 (DER-PV1). Fig. 4.33 shows the full schematic diagram of the frequency mode control in dq-frame strategy. The abc to dq-frame transformation were utilized using Equation (4.1), whereas the transformation angle (theta) is obtained by the voltage-controlled oscillator (VCO) with the input of the nominal power system frequency w_o.

In the frequency mode control in dq-frame strategy, the i_{dref} and i_{qref} are obtained from the reference voltage in dq-frame V_{sdref} and V_{sqref}. After obtaining the reference currents, they are then processed by the compensator, which is presented in [69]. Finally, the PWM generator was used to generate the firing signals of VSC-IGBTs. Figs. 4.34 – 4.38 shows the PSCAD models of the photovoltaic source-1 (DER-PV1) system with modified current-mode control with the DC link voltage controller in dq-frame method, which are developed in this work.

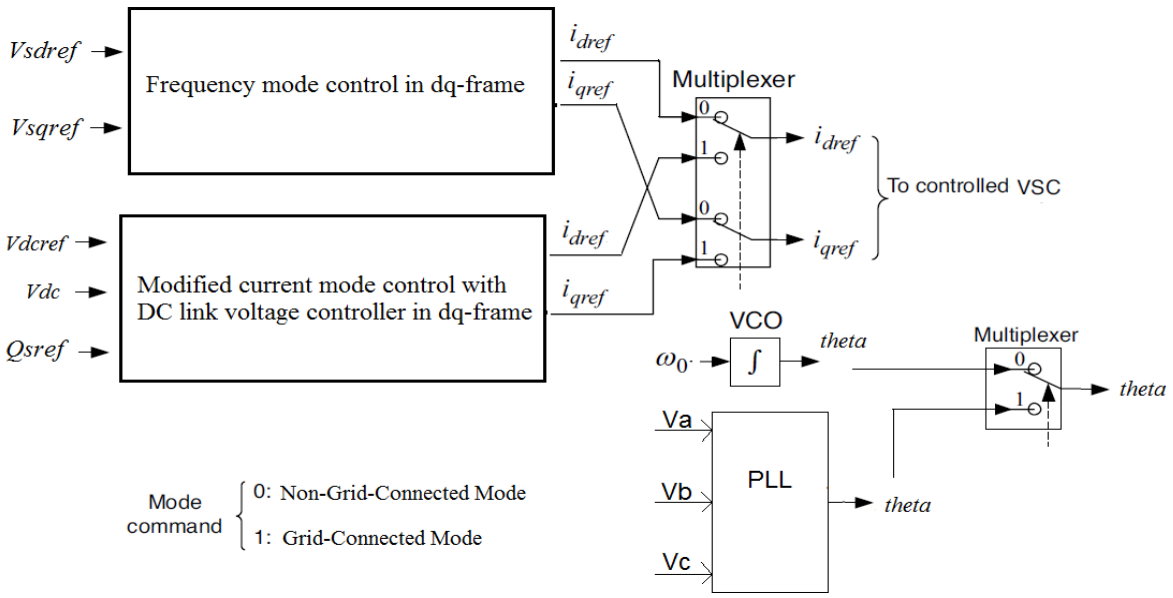


Fig. 4. 32: Supervisory control scheme in both grid-connected and non-grid-connected modes for solar photovoltaic source-1 (DER-PV1).

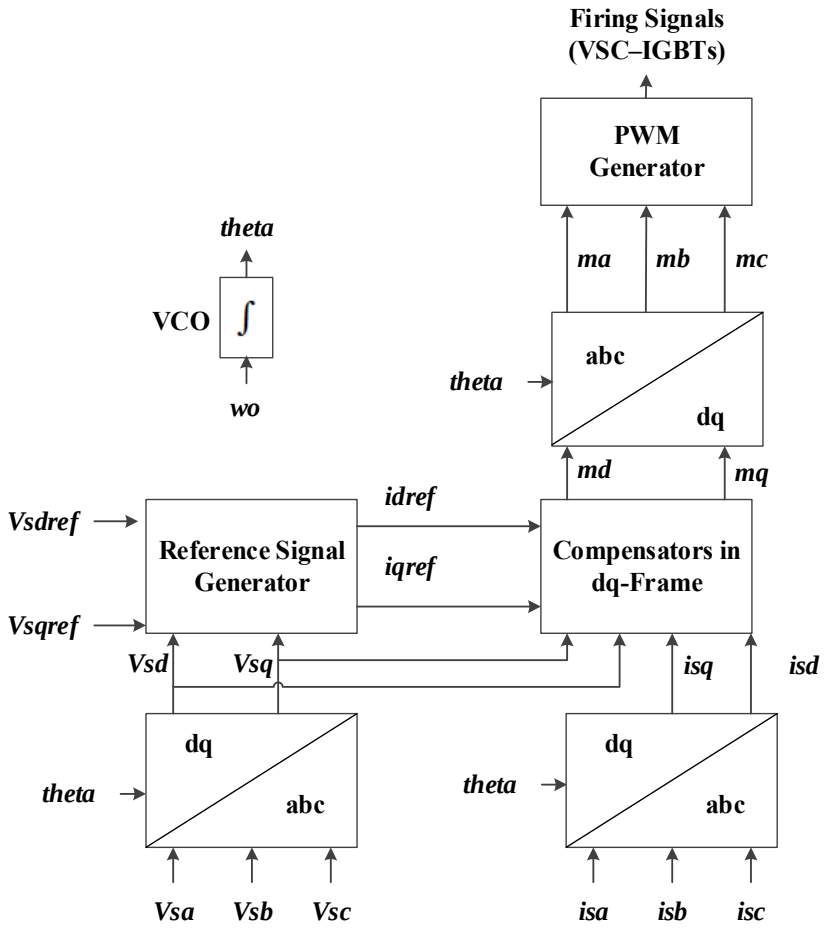


Fig. 4. 33: Schematic diagram of the frequency mode control in dq-frame.

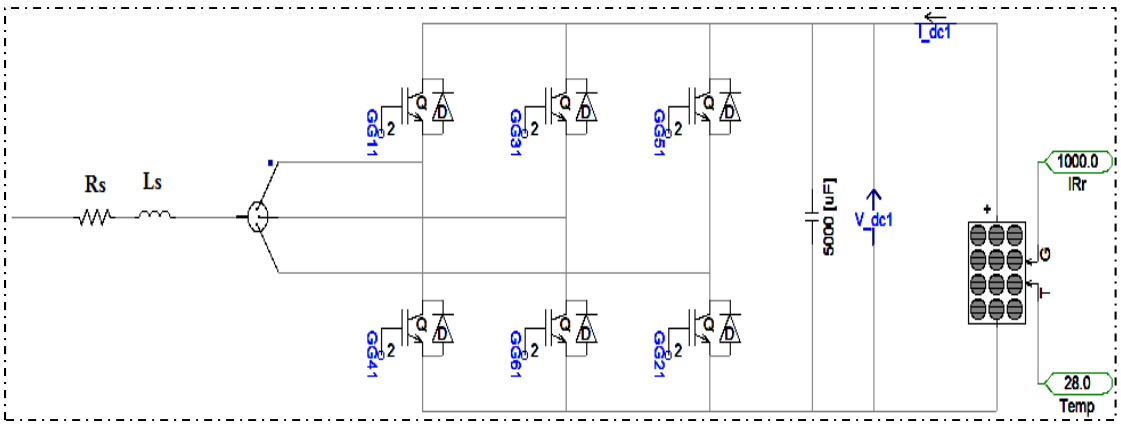


Fig. 4. 34: PSCAD model of the 3-phases VSC Block for solar photovoltaic source-1 (DER-PV1).

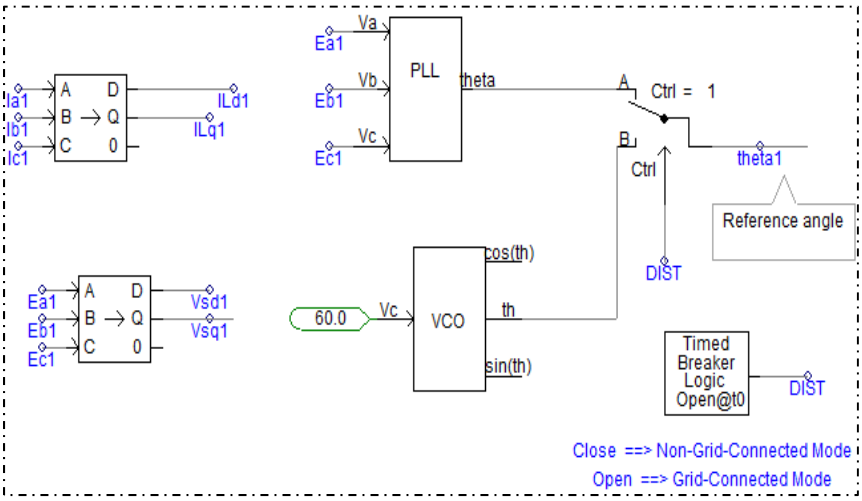
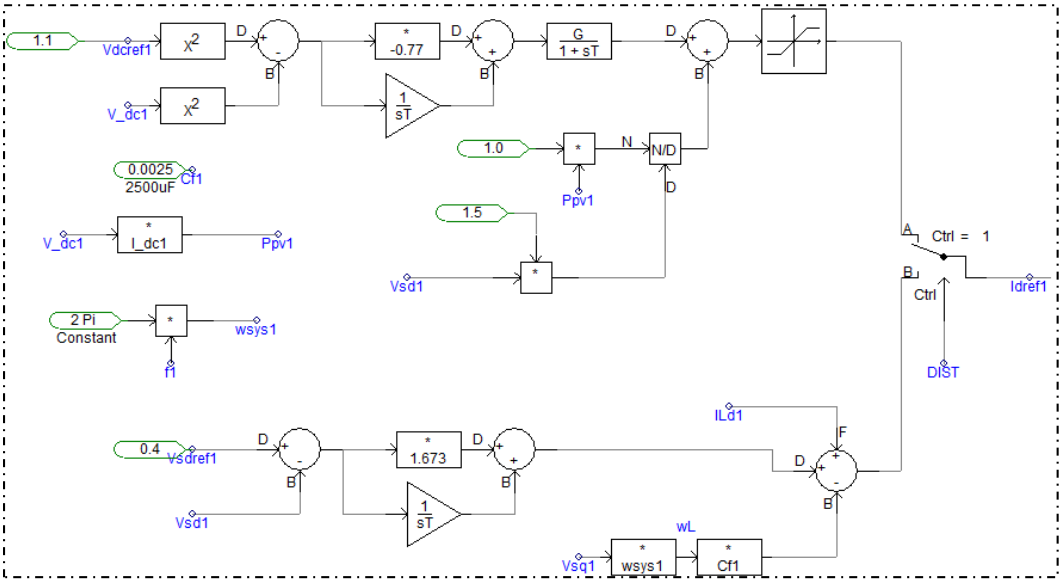
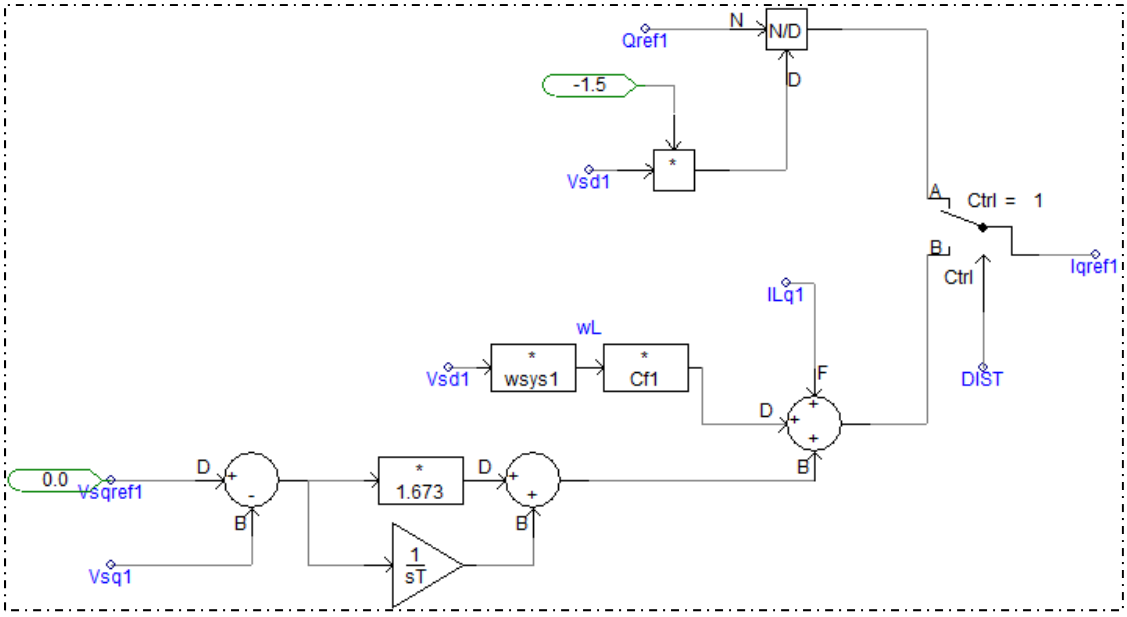


Fig. 4. 35: PSCAD model of the ABC-dq frame transformation block for solar photovoltaic source-1 (DER-PV1).



(a)



(b)

Fig. 4. 36: PSCAD model of the reference signal generator block in dq-frame for solar photovoltaic source-1 (DER-PV1) in both GC and NGC modes.

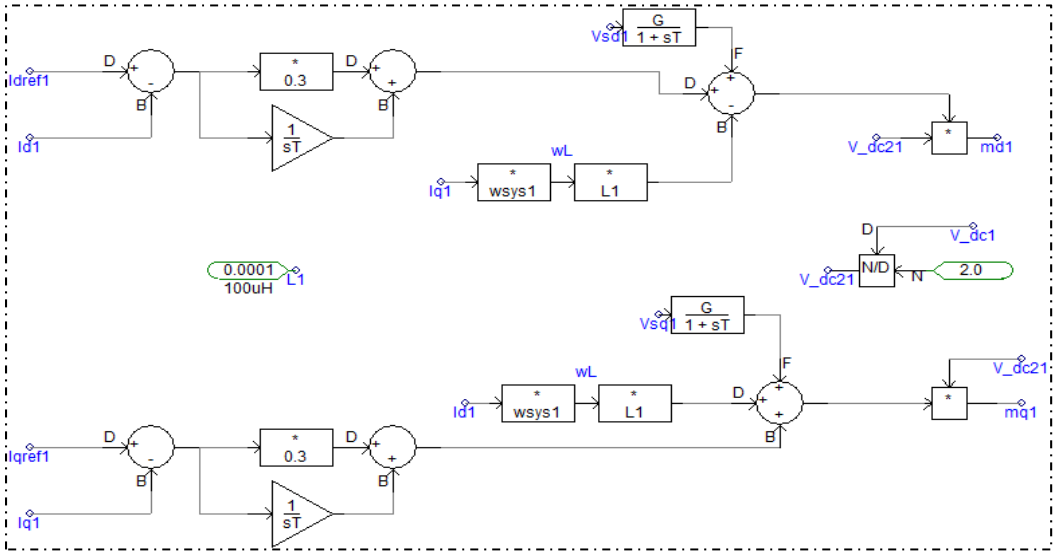


Fig. 4. 37: PSCAD model of the compensator block in dq-frame for solar photovoltaic source-1 (DER-PV1) in both GC and NGC modes.

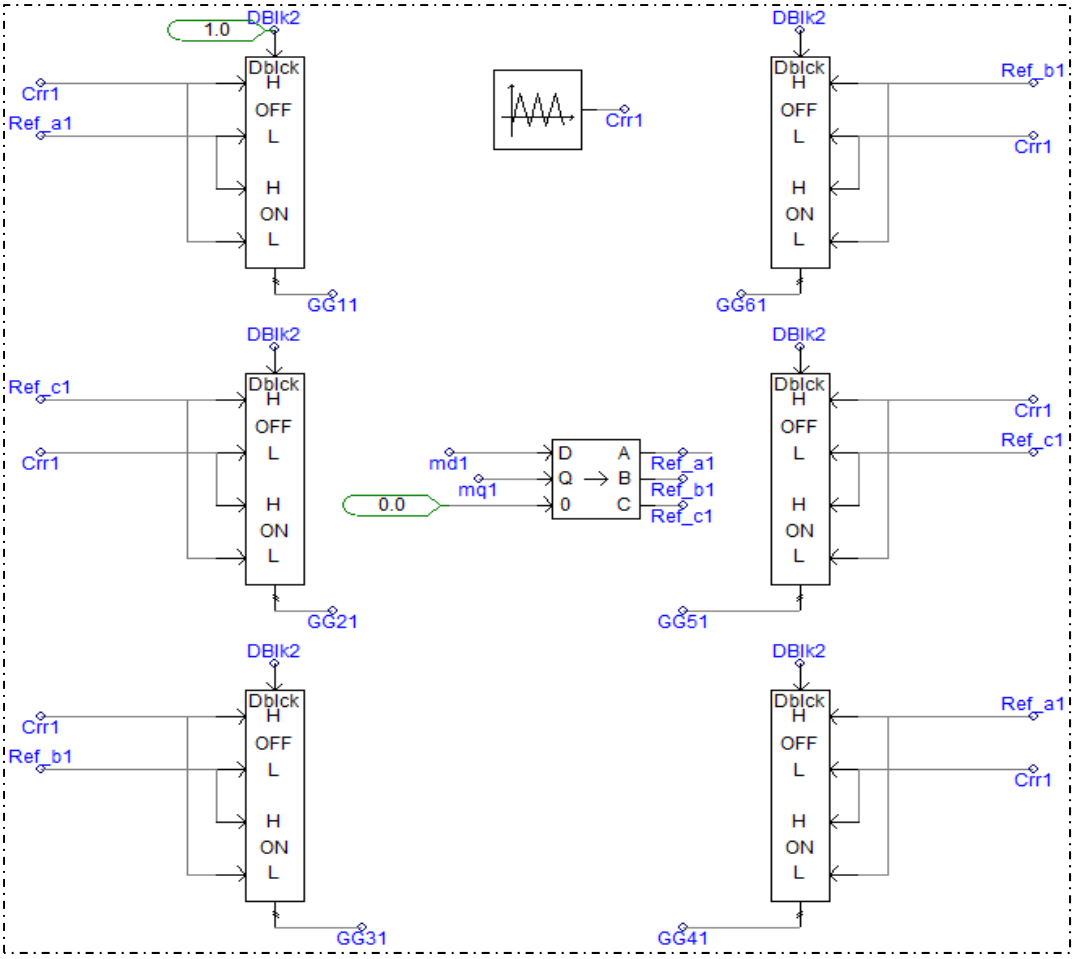


Fig. 4. 38: PSCAD model of the PWM Generator block for solar photovoltaic source-1 (DER-PV1).

4.3.2 Fault cases Generation for the Micro-Grid Distribution system

The CERTS Micro-Grid shown in Fig. 4.17 is modeled in the PSCAD/EMTDC software. The simulated system in PSCAD/EMTDC was used to generate the sampled voltages and currents signal (the sampled signals are recorded at each protection relay) needed in the fault diagnosis process in both modes of operation. The dataset of the fault cases is generated with different operating conditions as outlined in Table 4.17, with 11 different fault types considered including; 4 different fault resistances values; 2 modes of operation; and 4 different fault locations, which makes the full dataset consists of 352 records.

Table 4. 17: Dataset generation parameters during the fault for Micro-Grid distribution system.

Fault type		Fault resistance	Operation Mode	Fault Location	Cases
LG	AG	0.01, 0.5, 2 and 10 Ω	Islanded and grid connected	Z5, Z4, Z3, and Z34	32
	BG				32
	CG				32
LL	AB				32
	BC				32
	AC				32
LLG	ABG				32
	BCG				32
	ACG				32
LLL	ABC				32
LLLG	ABCG				32
Total	11	4	2	4	352

Figs. 4.39 – 4.54 show the voltage and current waveforms of all nodes of the system (e.g. grid, DER-PV1, DER-PV2, and DER-Bt.S) in grid connected and islanded mode, in the case of an AG fault that occurs in the midpoint of line (line25). The main differences in terms of duration and magnitude between the signals for two modes of operation can be seen in the figures. For instance, in the case of grid-connected mode, the magnitude of the current contributions during the fault from the DERs are very small compared to the current contribution form the grid source. Also, in the case of islanded mode, the fault current contribution from DERs did not reach the pick-up level for the protection device to detect the fault. These differences may be considered as a challenging problem for any fault detection approach.

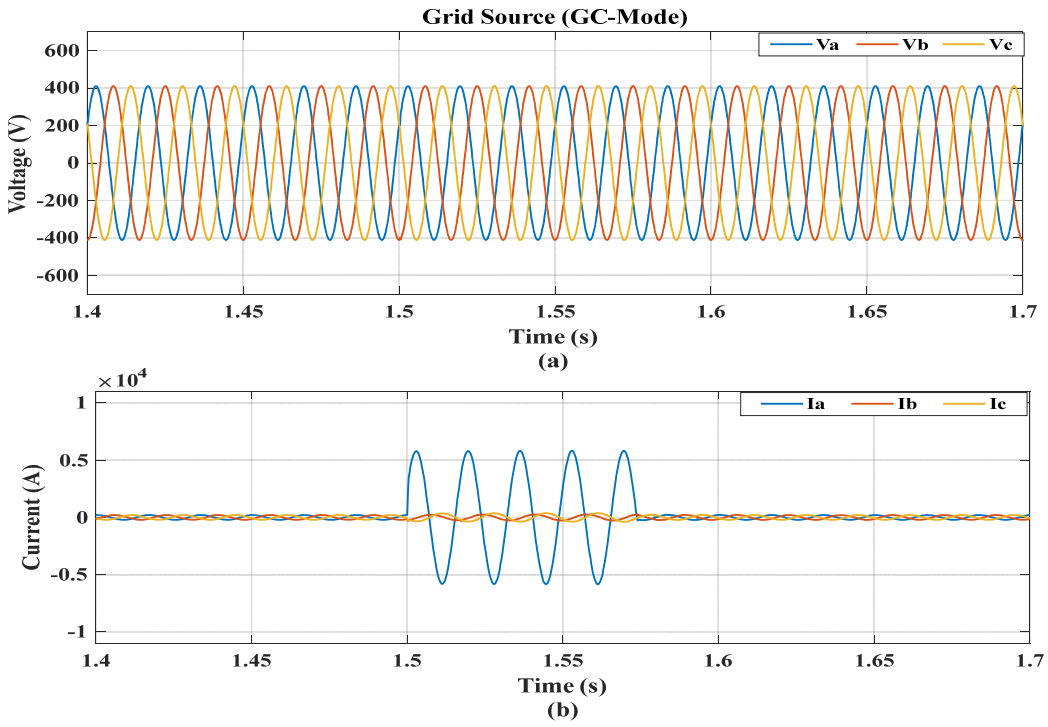


Fig. 4. 39: The voltage and current signals of grid source at AG fault in Z5 of Micro-Grid distribution system in GC mode.

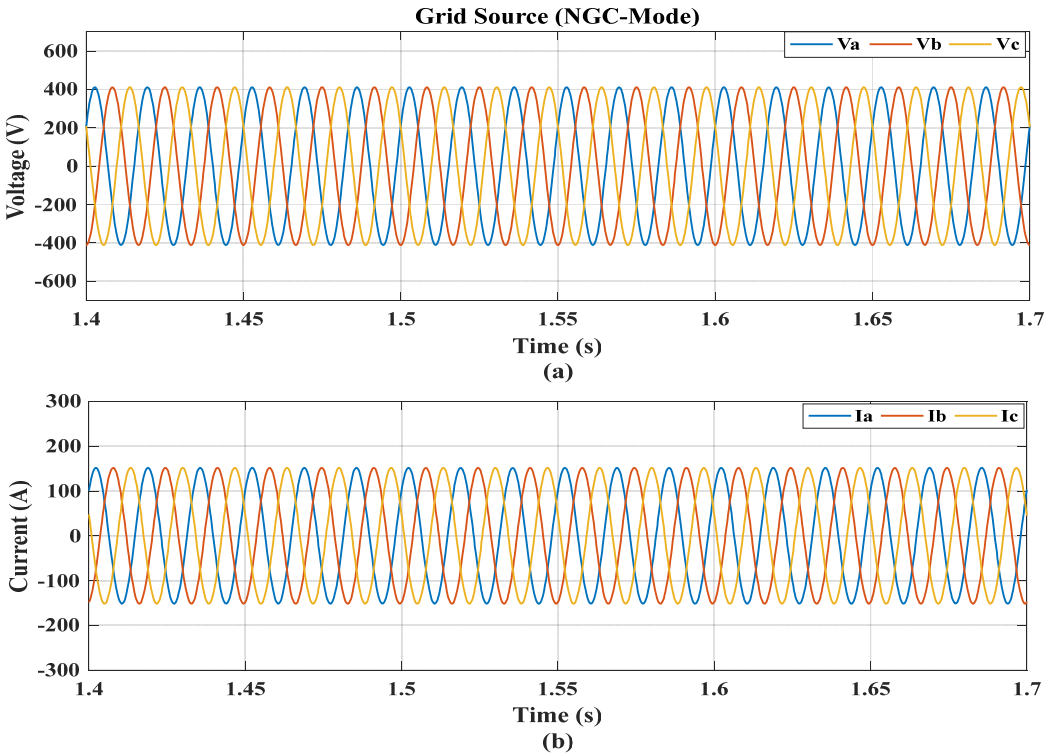


Fig. 4. 40: The voltage and current signals of grid source at AG fault in Z5 of Micro-Grid distribution system in NGC mode.

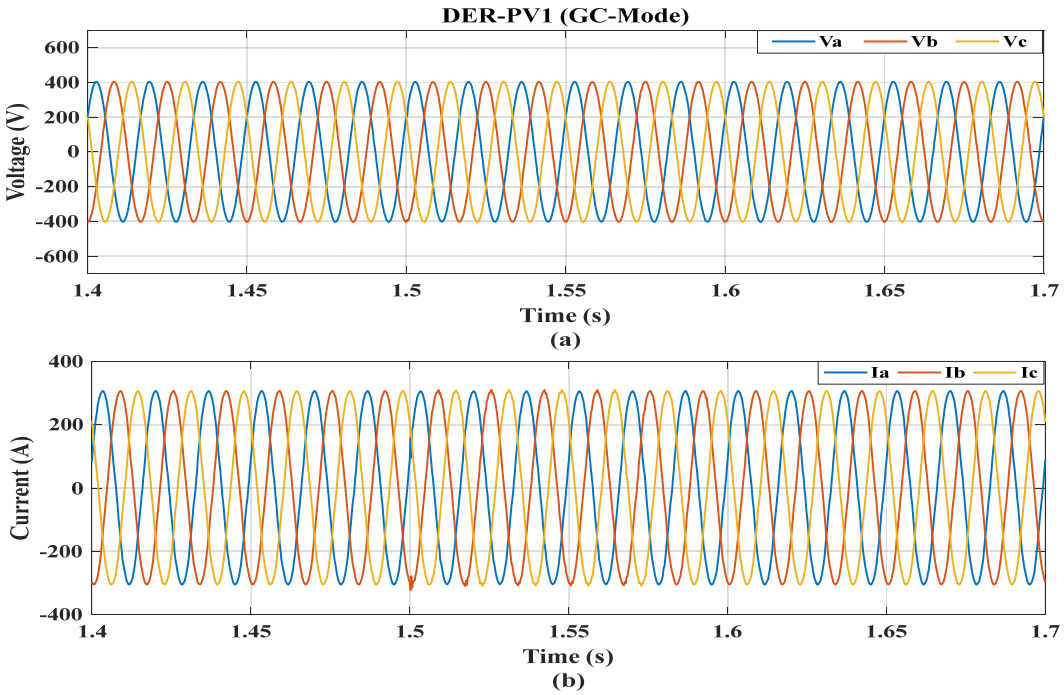


Fig. 4. 41: The voltage and current signals of DER-PV1 at AG fault in Z5 of Micro-Grid distribution system in GC mode.

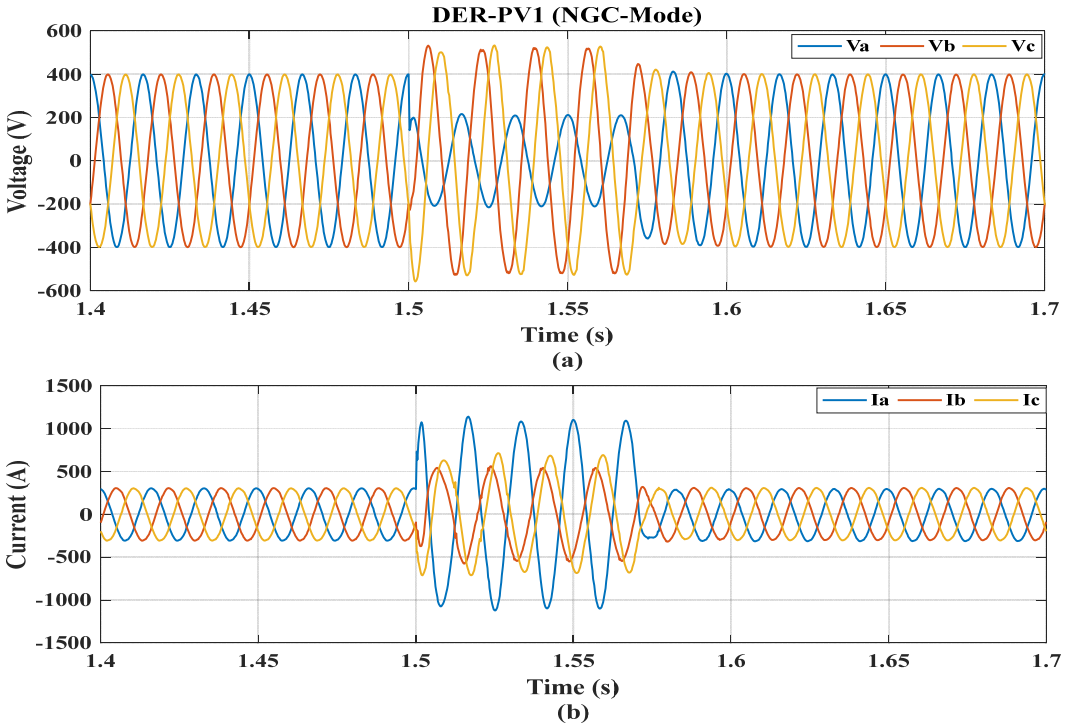


Fig. 4. 42: The voltage and current signals of DER-PV1 at AG fault in Z5 of Micro-Grid distribution system in NGC mode.

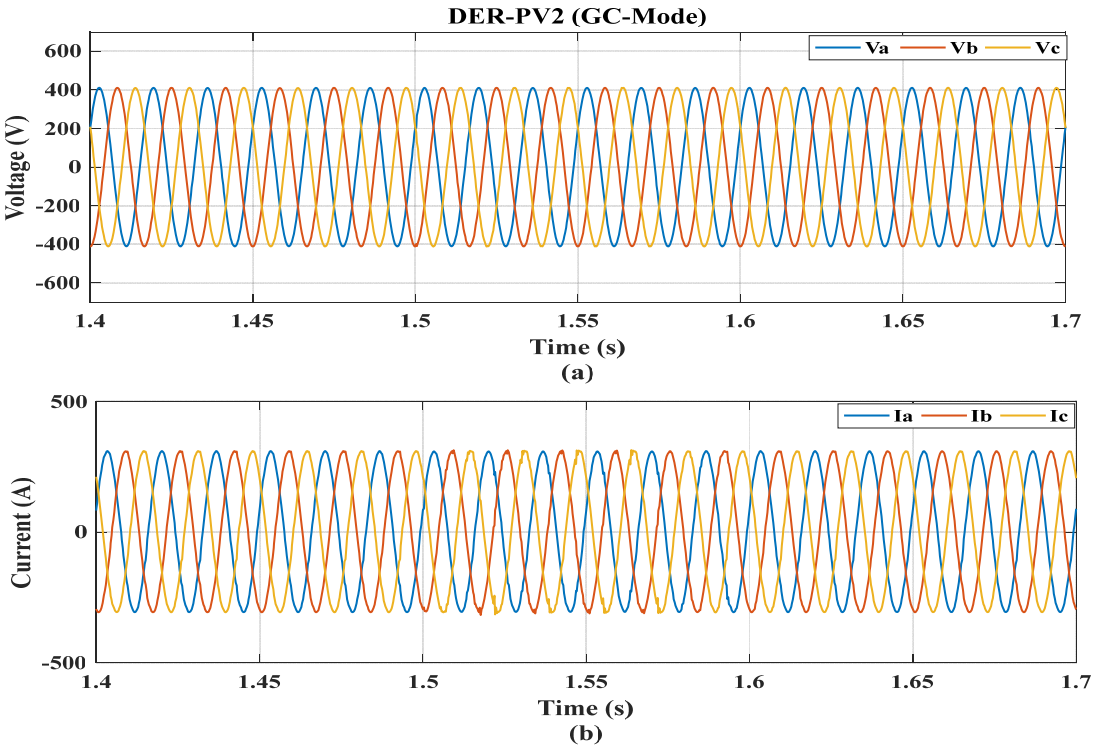


Fig. 4. 43: The voltage and current signals of DER-PV2 at AG fault in Z5 of Micro-Grid distribution system in GC mode.

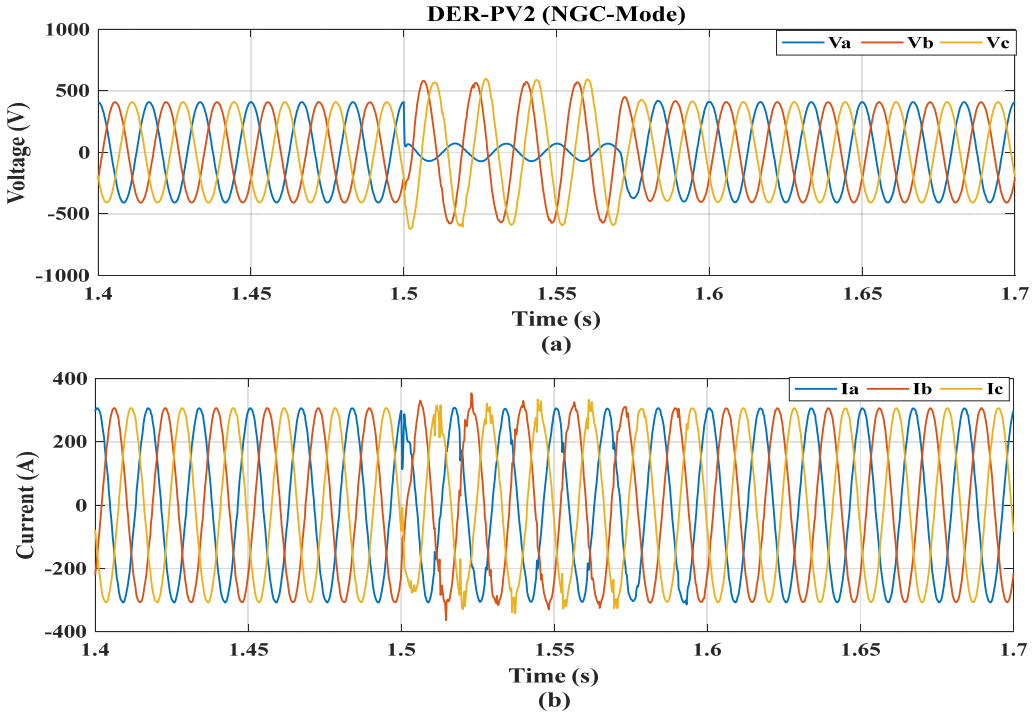


Fig. 4. 44: The voltage and current signals of DER-PV2 at AG fault in Z5 of Micro-Grid distribution system in NGC mode.

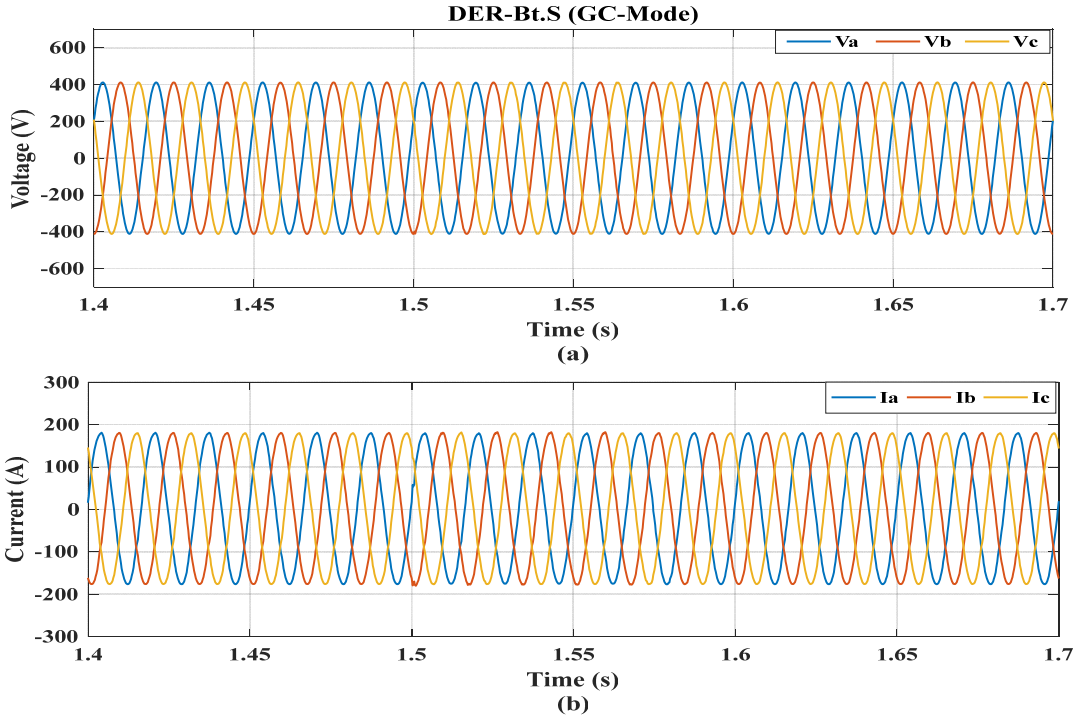


Fig. 4. 45: The voltage and current signals of DER-Bt.S at AG fault in Z5 of Micro-Grid distribution system in GC mode.

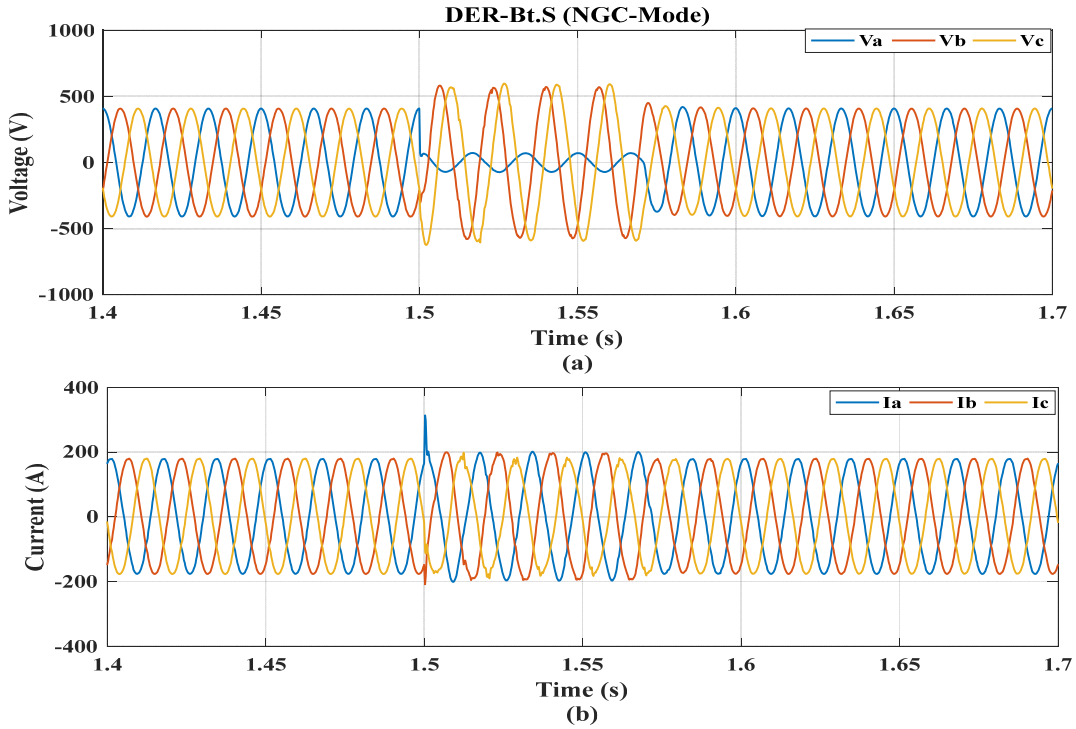


Fig. 4. 46: The voltage and current signals of DER-Bt.S at AG fault in Z5 of Micro-Grid distribution system in NGC mode.

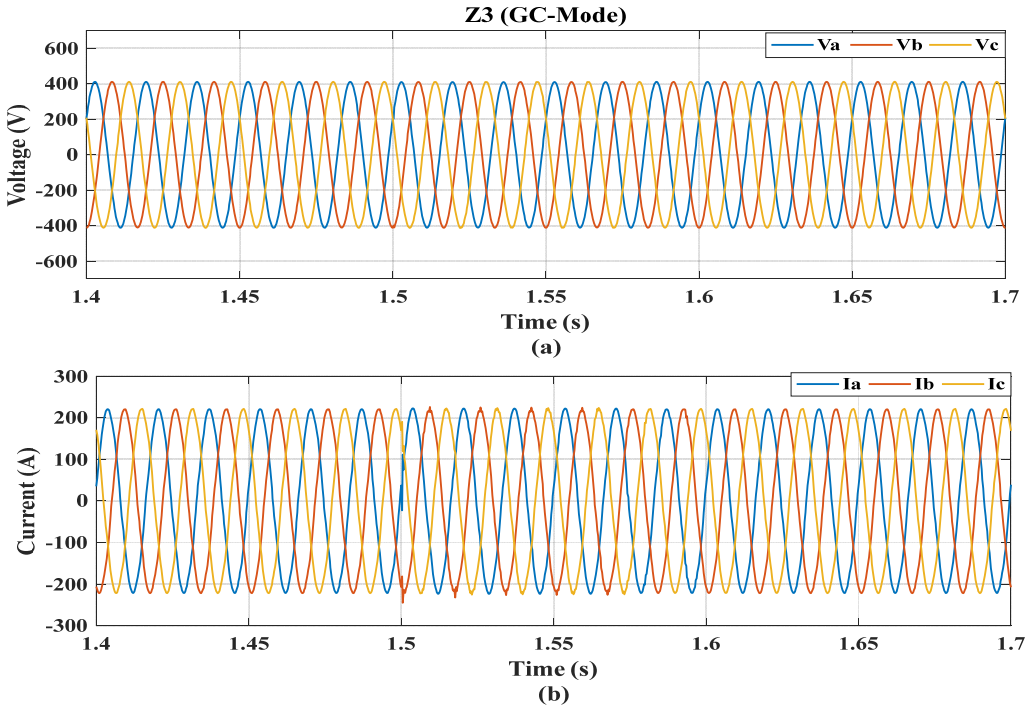


Fig. 4. 47: The voltage and current signals of Z3 relay at AG fault in Z5 of Micro-Grid distribution system in GC mode.

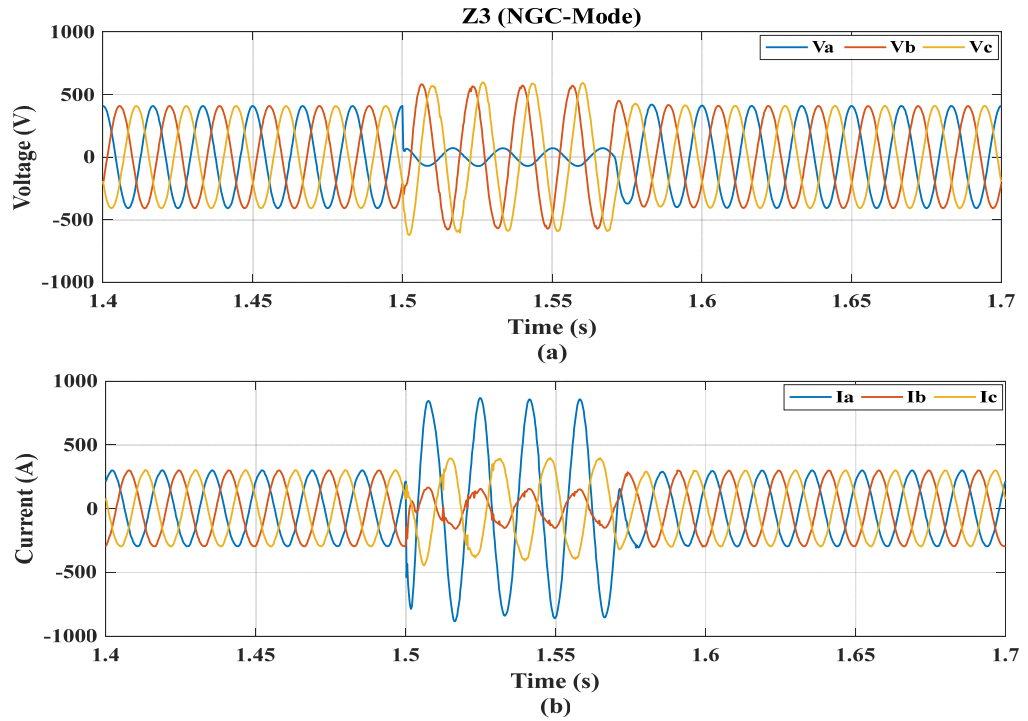


Fig. 4. 48: The voltage and current signals of Z3 relay at AG fault in Z5 of Micro-Grid distribution system in NGC mode.

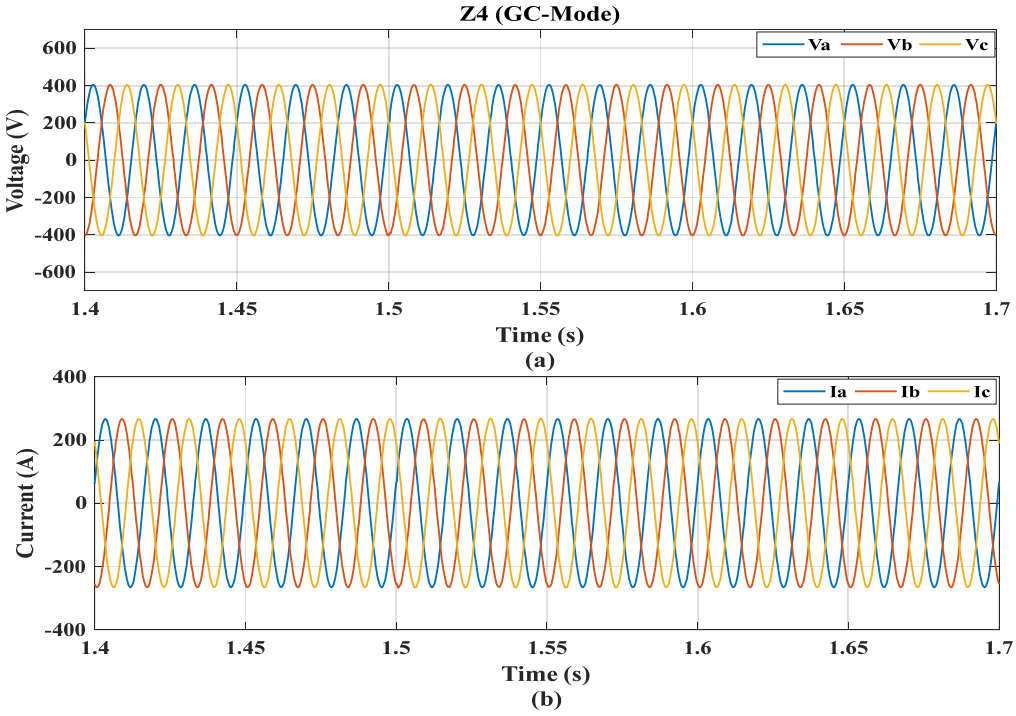


Fig. 4. 49: The voltage and current signals of Z4 relay at AG fault in Z5 of Micro-Grid distribution system in GC mode.

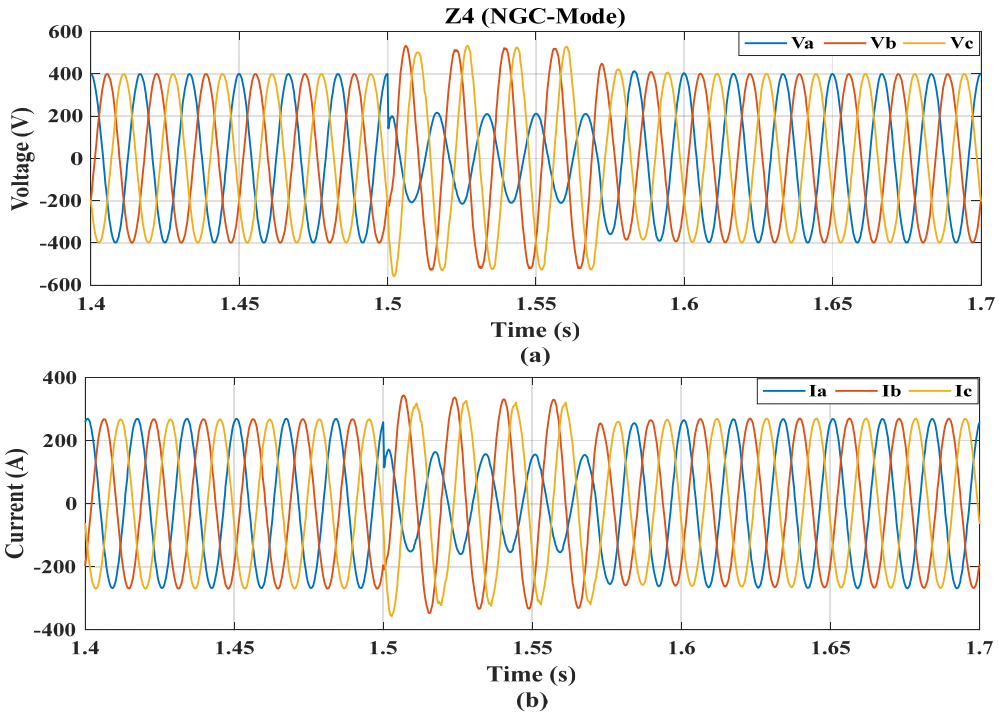


Fig. 4. 50: The voltage and current signals of Z4 relay at AG fault in Z5 of Micro-Grid distribution system in NGC mode

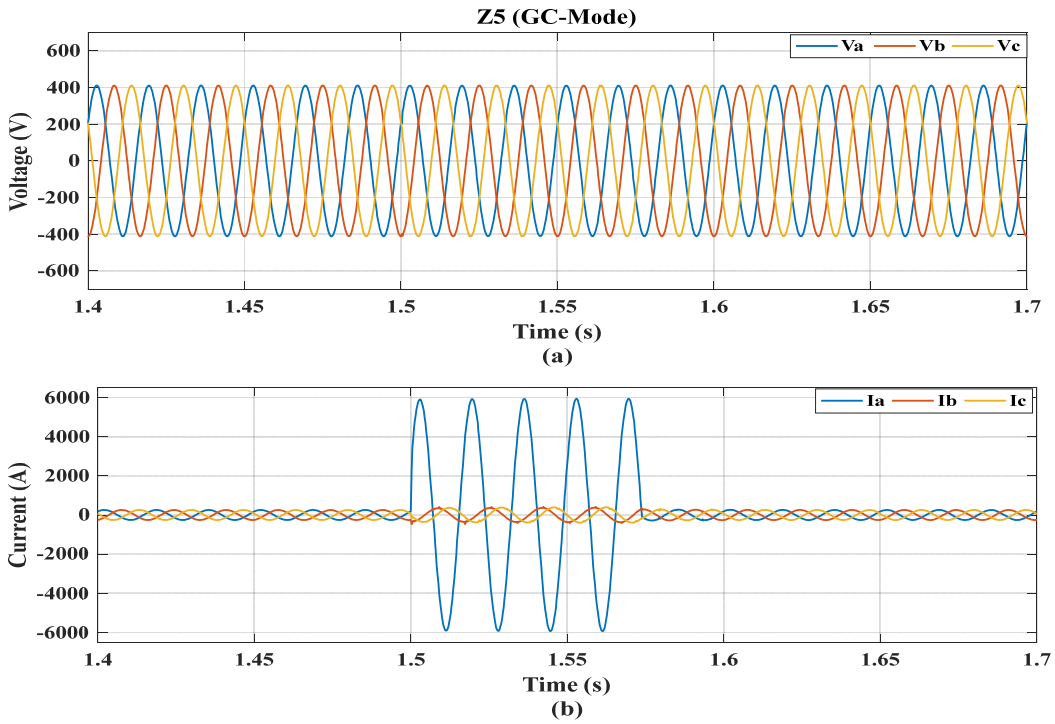


Fig. 4. 51: The voltage and current signals of Z5 relay at AG fault in Z5 of Micro-Grid distribution system in GC mode.

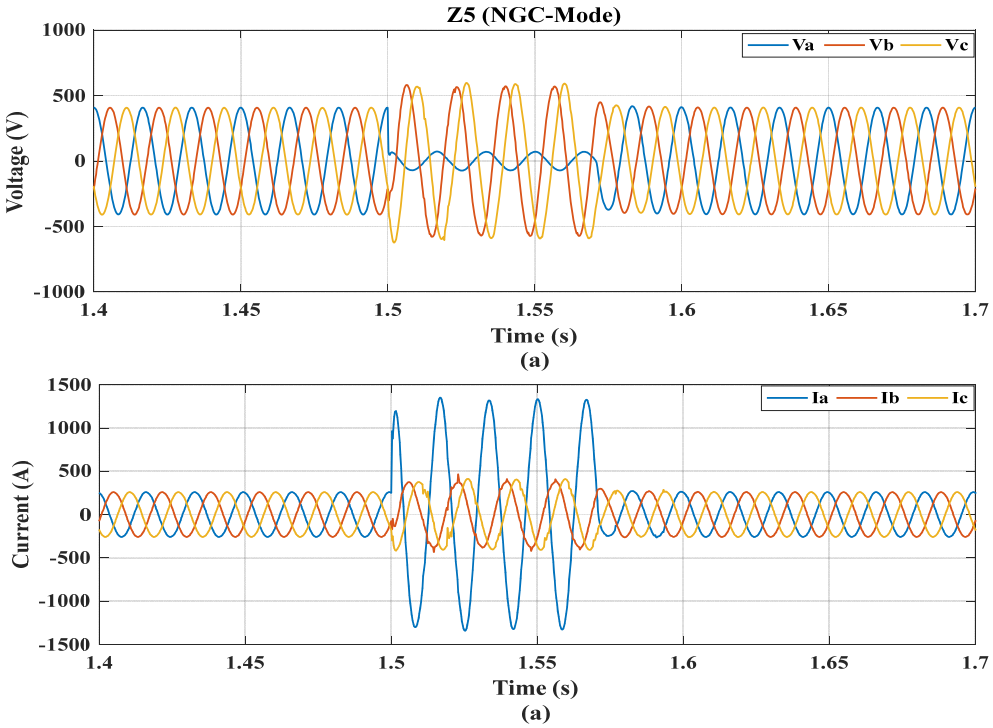


Fig. 4. 52: The voltage and current signals of Z5 relay at AG fault in Z5 of Micro-Grid distribution system in NGC mode

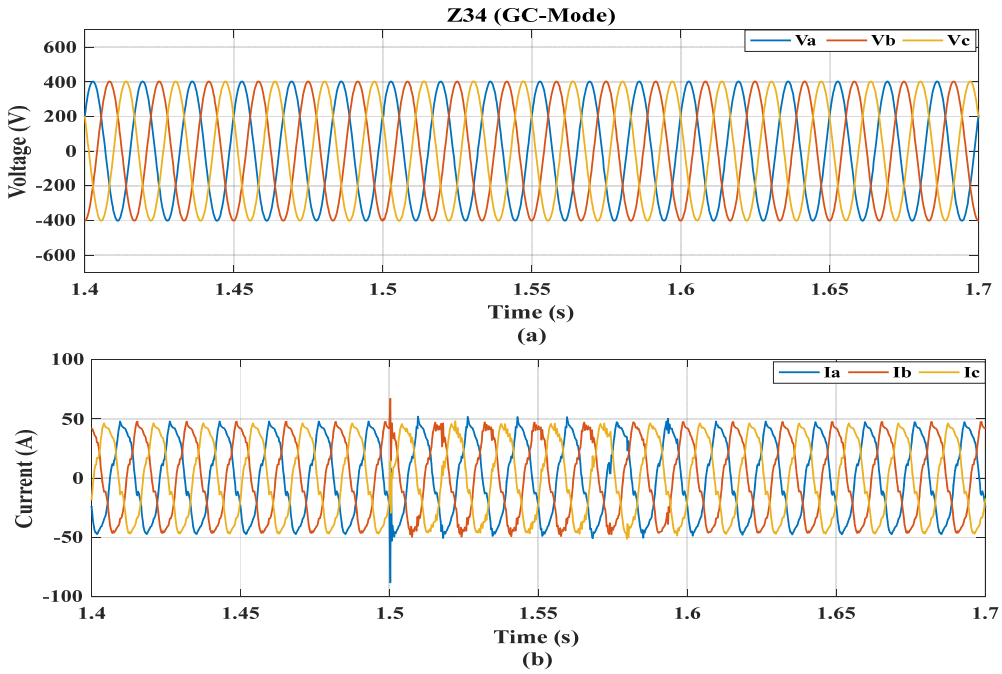


Fig. 4. 53: The voltage and current signals of Z34 relay at AG fault in Z5 of Micro-Grid distribution system in GC mode

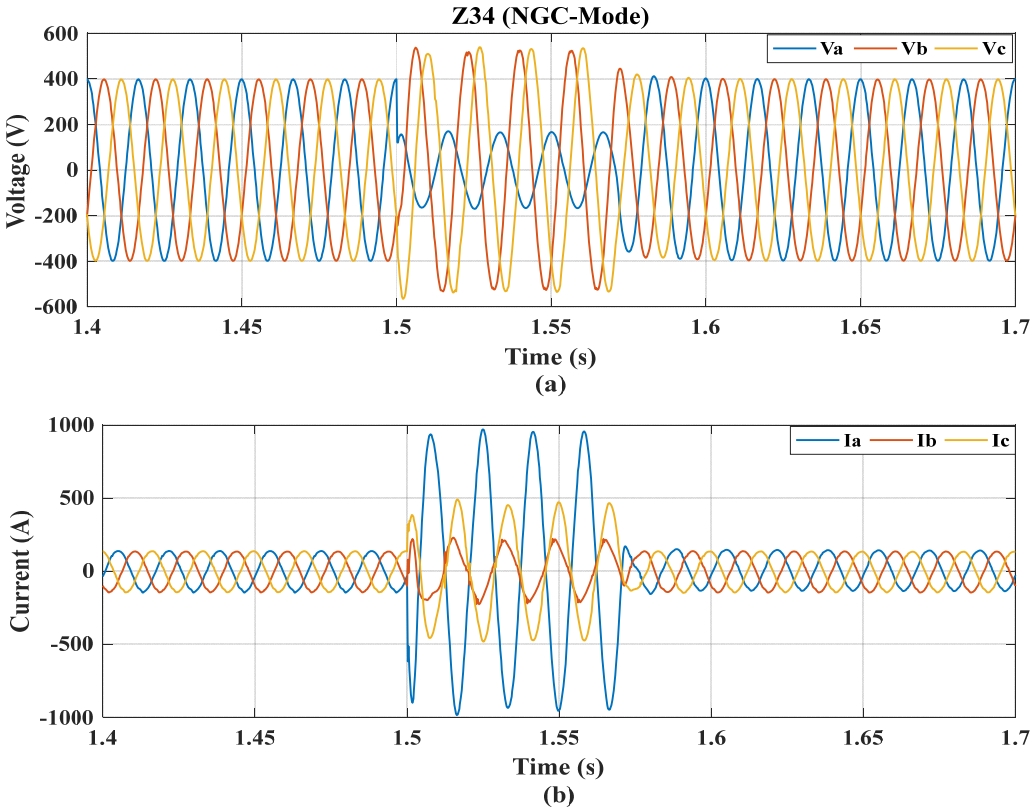


Fig. 4. 54: The voltage and current signals of Z34 relay at AG fault in Z5 of Micro-Grid distribution system in NGC mode

4.3.3 Optimal Wavelets Combination in case of Micro-Grid Distribution System

The harmony search algorithm with the settings parameter as listed in Table 4.7 is applied, as described in Chapter 3, to identify the most suitable wavelet function(s) and the optimal number of wavelet decomposition level(s) in the Micro-Grid distribution system.

Table 4.18 shows the outcomes of implementing the HSA (over the 20-trials), where the HSA has identified the most suitable combination of decomposition levels, as well as, identified the most suitable wavelet functions for each identified wavelet decomposition level required for fault diagnosis application. The selected wavelet functions and the wavelet decomposition levels, which has the most occurrence over the 20-trials, are listed in Table 4.18. The algorithm chooses four decomposition levels (i.e. Cd_1, Cd_2, Cd_3, Cd_4, and Ca_4) for the current signal, and three decomposition levels (i.e. Cd_1, Cd_2, and Cd_3) for the voltage signal.

In the case of the current signal, the HSA has identified the wavelet function (bior1.3) for decomposition level detail-1 (Cd_1) and the wavelet function (bior3.1) for decomposition level detail-3 (Cd_3).

On the other hand, in the case of the voltage signal, the HSA has identified the wavelet function (db41) for decomposition level detail-1 (Cd_1) and the wavelet function (sym9) for decomposition level detail-3 (Cd_3). Figs. 4.55 and 4.56 show the selected combination of the wavelet functions for the current and voltage signals in the Micro-Grid distribution test system.

It has been observed that when changing the configuration/topology of the studied system, the proposed AWBFDD technique was able to identify a new suitable wavelet functions and wavelet decomposition levels, which show the adaptability/flexibility of the proposed AWBFDD technique with the changing of configuration/topology of the studied system.

Table 4. 18: Optimal wavelet function and decomposition level on Micro-Grid distribution system.

Signal	Level	Wavelet Function	Occurrence (%)
Current	Cd_1	bior1.3	77
	Cd_2	bior3.5	70
	Cd_3	bior3.1	57
	Cd_4	db4	73
	Ca_4	db8	100
Voltage	Cd_1	db41	43
	Cd_2	bior1.5	63
	Cd_3	sym9	73
	Cd_4	-	0
	Ca_4	-	0

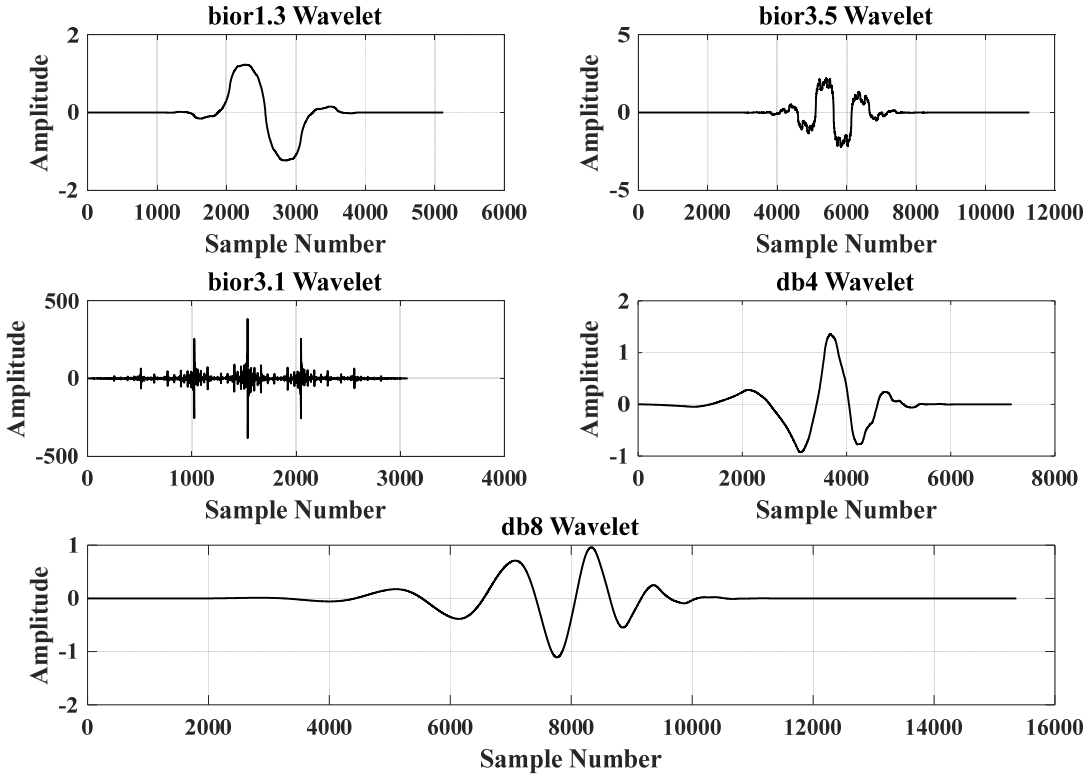


Fig. 4. 55: Selected wavelet function of voltage signal on the Micro-Grid distribution system.

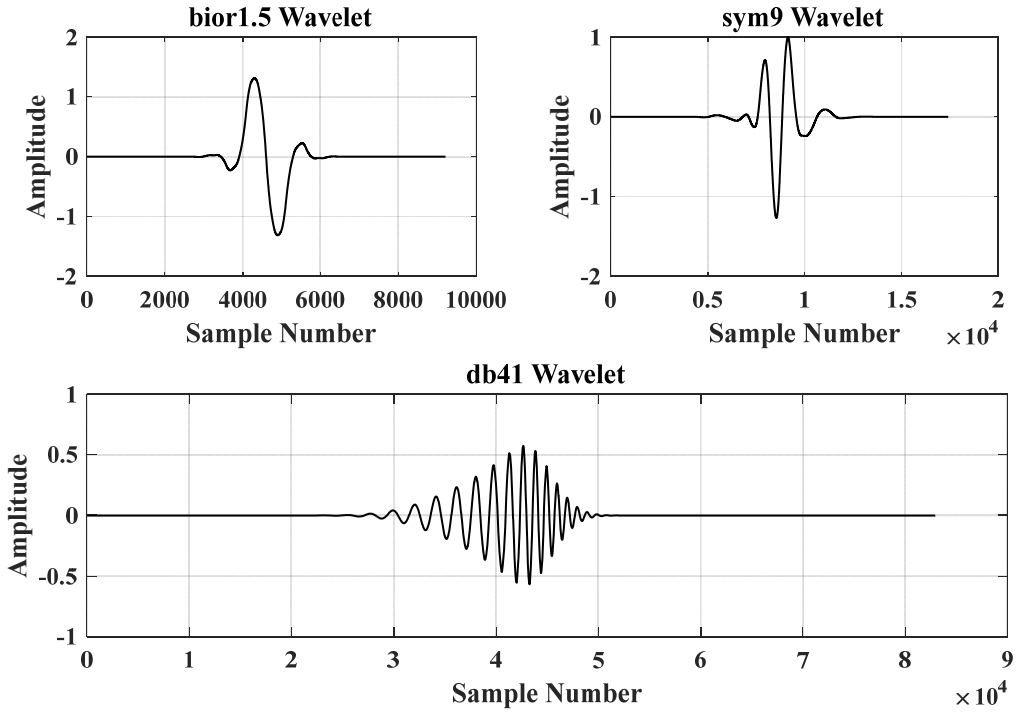


Fig. 4. 56: Selected wavelet function of current signal on the Micro-Grid distribution system.

4.3.4 Fault Classification Results of Micro-Grid Distribution System

The CERTS Micro-Grid distribution system is divided into four protective zones, as suggested by Lasseter et al., in [68], each equipped with a protection relay as shown in Fig. 4.17. At each protection relay point the current and voltage signals are sampled and recorded. The DWT is applied to the sampled signals using the identified wavelet combination by the HSA, where the analyzed window is two consecutive cycles, one cycle pre-fault (64 samples), and one cycle after the fault has occurred (64 samples).

The energy of the wavelet coefficient (i.e., Cd_1, Cd_2, Cd_3, Cd_4, and Ca_4) for the three-phase current signal and the energy of the wavelet coefficient (i.e., Cd_1, Cd_2, and Cd_3) for the three-phase voltage signal are calculated as in (3.8) – (3.11). The energy vector, which contains 24 elements (5 values for wavelet coefficient energies of current signal × 3 phase signals + 3 values for wavelet coefficient energies of voltage signal× 3 phase signals = 24), is used as the input

attributes to the classifier. Consequently, the dataset matrix dimension is 352 records × 24 attributes. Two different classifiers (i.e., DT and K-NN) are used in this thesis to automate the fault classification process in the CERT Micro-Grid distribution system. The Monte Carlo stratified cross validation is applied to estimate the classifiers performance, as explained in Chapter 3. For 1-trial of the 1,000 Monte Carlo trials, the classification accuracies for all protection zones with the DT and K-NN classifiers of 10-fold cross validation are represented by the box plots, as shown in Fig. 4.57.

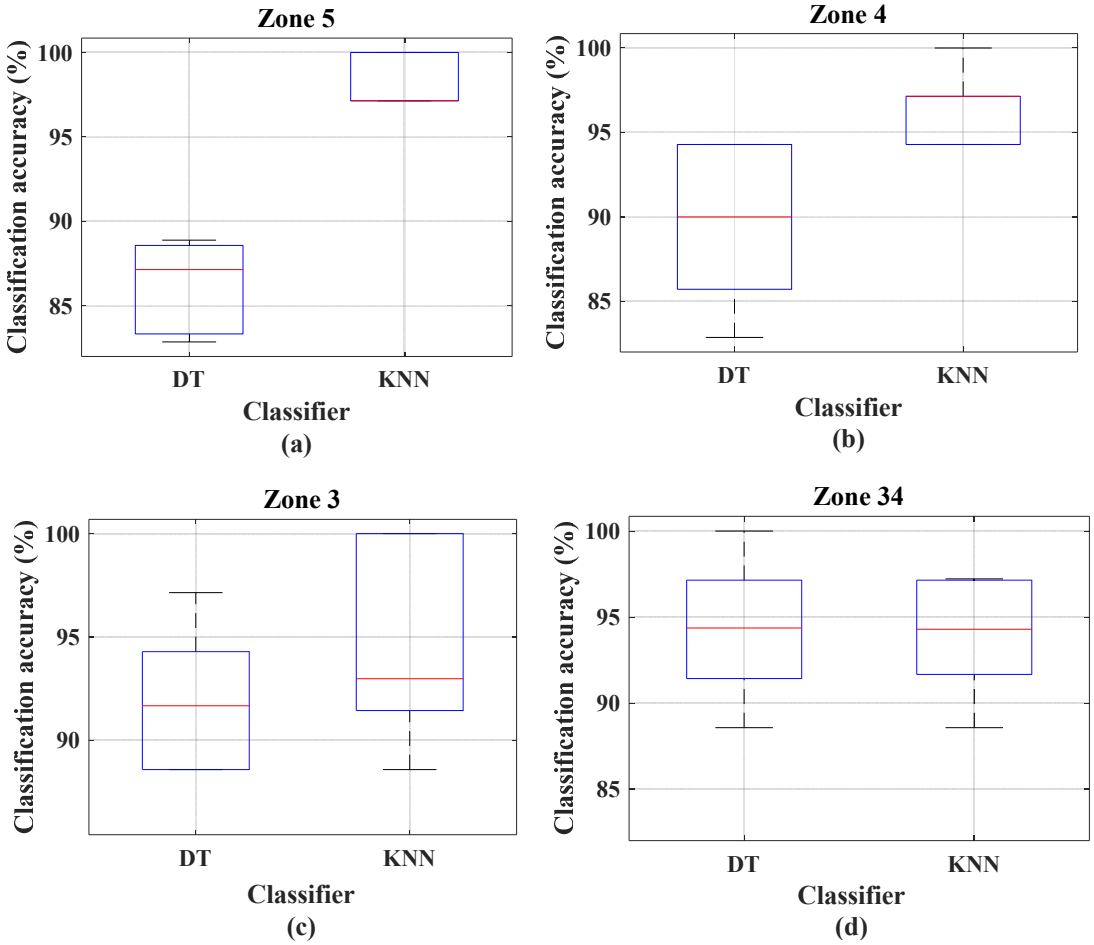


Fig. 4. 57: Box plot of classifier accuracy for 10-folds in case of 1-trial of the Micro-Grid distribution system.

Fig. 4.57 depicts the variance in the classification accuracies through 10 folds in one trial, where the maximum and minimum variances of classification accuracy are 27.26 % and 1.89 % in the case of zone34 and zone5, respectively. Consequently, the cross-validation has a large variance

estimation, which can be reduced by repeating the cross-validations. In this thesis, the 10-fold cross-validation is repeated for 1,000 trials. Algorithm 3, which outlines the Monte Carlo stratified cross validation, is applied to evaluate the classification accuracies over the 1,000-trials. Table 4.19 shows the classification accuracies for all protective zones, and the 95% Confidence Interval (*CI*) of the classification accuracy. The table shows the superiority of the K-NN classifier (highlighted in grey) over the DT in all the protection zones with higher lower and upper bounds of confidence interval. The new wavelet combinations are used to extract the prominent attributes holding the pattern of the current/voltage signals in each fault type, which means each group of class (i.e. fault types) is becoming more correlated together. In that sense, the role of K-NN classifier becomes easier according to the main concept of K-NN classifier, which mainly relies on the Euclidean distance (similarity) between the cases. The overall mean classification accuracies of each classifier are listed in Table 4.20.

Table 4. 19: Classifier accuracy for all classifiers in all protection zones on the Micro-Grid distribution system.

Protection zone	Classifier	Accuracy (%)	Confidence Intervals (%)
Zone5	DT	86.20	82.20 − 89.41
	K-NN	98.02	95.96 − 99.04
Zone4	DT	89.45	85.80 − 92.24
	K-NN	96.68	94.25 − 98.10
Zone3	DT	92.37	89.12 − 94.71
	K-NN	93.77	90.74 − 95.85
Zone34	DT	93.60	90.54 − 95.71
	K-NN	94.07	91.09 − 96.09

Table 4. 20: Overall classification accuracy on the Micro-Grid distribution system.

Classifier	Overall accuracy (%)	Confidence Intervals (%)
DT	90.40	86.88 – 93.06
K-NN	95.63	92.96 – 97.32

Table 4.21 lists the operating conditions and the results of the proposed HSA method and the other method (Mishra-Samantaray approach) used in [41] for fault classification in the CERTS Micro-Grid.

Table 4. 21: Classification results comparison of CERTS Micro-Grid distribution system.

	Proposed method	Ref. [41]	
Sampling rate	64 samples/cycle	111 samples/cycle	conventional schemes (over current relay)
Wavelet function	Wavelet combination (HSA)	db1 (Random)	
Number of classes	10 (i.e. A-G, B-G, C-G, AB, BC, AC, AB-G, BC-G, AC-G, ABC, and ABCG)	4 (i.e. LG, LL, LLG, LLLG)	
Accuracy method	Monte Carlo stratified cross validation	Random test group	
Classifier accuracy	95.63%	94%	56%

The variances between the accuracy of the proposed technique and the technique given in [41] reveals the ability of the proposed AWBFDD technique to outperform other approaches by recognizing the best wavelet combinations that hold the salient features for accurate fault classification of the Micro-Grid distribution system in both non-grid-connected and grid-

connected modes. Moreover, the proposed method takes into account all fault types (10-classes), and applies a systematic method (i.e. Monte Carlo stratified cross validation) to evaluate the classification accuracy. However, the work in [41] relied on a random test group, which causes the estimated accuracy to be unreliable. The performance of the classifier was highly affected by the composition of training and testing subsets. In the case of random selection, the training and testing dataset will be random subsets of the main dataset. Consequently, a class which is over-represented in one subset, will be under-represented in the other subset, and vice versa. The presented method in [41] distinguishes between only 4 classes, which limits its capability to distinguish between all fault types (i.e. 10 faults).

4.3.5 Immunity Testing

4.3.5.1 *Power Quality (PQ) disturbance*:

The performance of proposed technique was examined in the presence of Power Quality PQ disturbances in particular voltage magnitude variations and voltage frequency variations. A total of 20 cases considering the voltage variation and frequency variation were generated. According to [59], the ranges of under/over voltage and frequency were defined (over-voltage: 1.05 pu to 1.15 pu, under-voltage: 0.8 pu to 0.95 pu, Over-frequency: 60.1 Hz to 63 Hz, and under-frequency: 57 Hz to 59.9 Hz). The proposed AWBFDD technique is applied to the new cases of PQ disturbance and the classification accuracies of each classifier are listed in Table 4.22. The results indicate the capability of the proposed AWBFDD technique to accurately detect and classify faults in the presence of different PQ disturbances.

Table 4. 22: Overall classification accuracy with PQ disturbance cases on the Micro-Grid distribution system.

Classifier	Overall accuracy (%)	Confidence Intervals (%)
DT	90.01	86.43 - 92.72
K-NN	95.44	92.73 - 97.17

4.3.5.2 *Measurement Uncertainty (MU)*:

In order to further test the robustness of the proposed AWBFDD approach, the effect of measurement errors were taken into account in this thesis. A new dataset consisting of 60 cases is generated in the presence of measurement errors in the range ± 3% according to IEEE/ANSI Standard requirements for instrument transformers [71]. Table 4.23 shows the classification accuracies when considering the measurement errors, and it can be seen that the proposed approach is immune to such measurement uncertainty.

The results implies the ability of the proposed technique to accurately detect and classify faults in the presence of measurement uncertainties. The obtained results reveal the immunity of the proposed approach to such to PQ disturbances and to measurement uncertainty in the Micro-Grid system.

Table 4. 23: Overall classification accuracy with Measurement Uncertainty on the Micro-Grid distribution system.

Classifier	Overall accuracy (%)	Confidence Intervals (%)
DT	90.07	86.80- 92.59
K-NN	95.60	93.16 - 97.19

4.3.5.3 Effect of Fault Current Limiters (FCLs)

In the Micro-Grid distribution system, the presence of DERs may lead to a change in the fault current magnitudes in both modes of operations. Due to these changes in the fault current magnitudes, Fault Current Limiters (FCLs) were proposed in [72] and were used with the directional overcurrent relays in the system. To examine the effect of FCLs on the performance of the proposed AWBFDD technique, a detailed model of the FCL, which is based on [72], was incorporated to the CERTS Micro-Grid distribution system model developed in the PSCAD/EMTDC. A new dataset consisting of 24 cases is generated considering the presence of

the FCL in the system at different fault types, resistances, and locations. For example, Fig. 4.58 exemplifies the fault current in the case of AG fault, which occurs with/without FCL in the system. Table 4.24 shows the classification accuracy with the existing of FCL. The results imply the ability of the proposed technique to accurately detect and classify faults in the presence of FCL. The obtained results reveal the immunity of the proposed approach to the FCL influence in the Micro-Grid system.

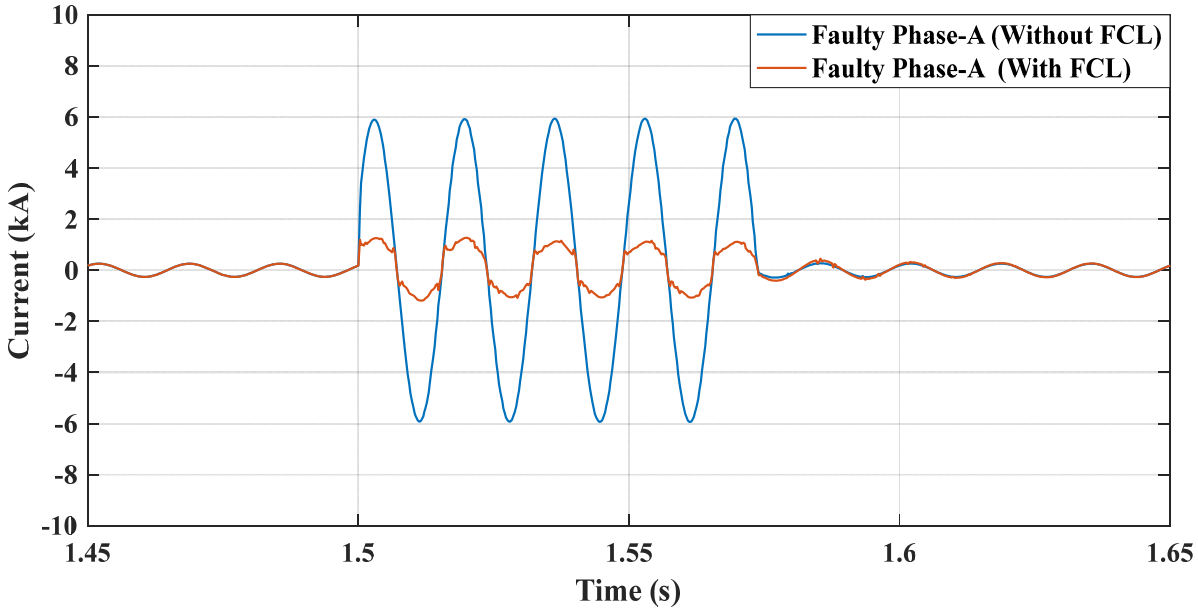


Fig. 4. 58: The current signal at AG fault with/without FCL of Micro-Grid distribution system.

Table 4. 24: Overall classification accuracy with FCL of Micro-Grid distribution system.

Classifier	Overall accuracy (%)	Confidence Intervals (%)
DT	90.27	86.85- 92.87
K-NN	95.30	92.66 - 97.01

4.4 Experimental Work

In order to verify the effectiveness of the proposed approach, the experimental set-up shown in Fig. 4.59 was used. A summary of specifications for each component can be found in the Appendix A. The experimental set-up allowed the emulation of actual faults in the power system. The following was the hardware description:

1) Arbitrary/Function generator: Tektronix AFG3022C, 25 MHz Sine Waveforms and 14-bits arbitrary waveform. The AGF is typically used with the supporting software, which allows importing any signal and playing it back in the real-time.

2) Data Acquisition Module (DAQ): Omega 1608FS-PLUS, 8 channel, 16-bit multifunction Universal Serial Bus (USB). It can sample up to eight channels at a maximum speed of 50 kHz per channel or four channels at 100 kHz while delivering 16-bit accuracy.

3) Personal Computer (PC): Intel(R) Xeon(R) CPU E5520 @ 2.27GHz, Dual core, 14 GB RAM.

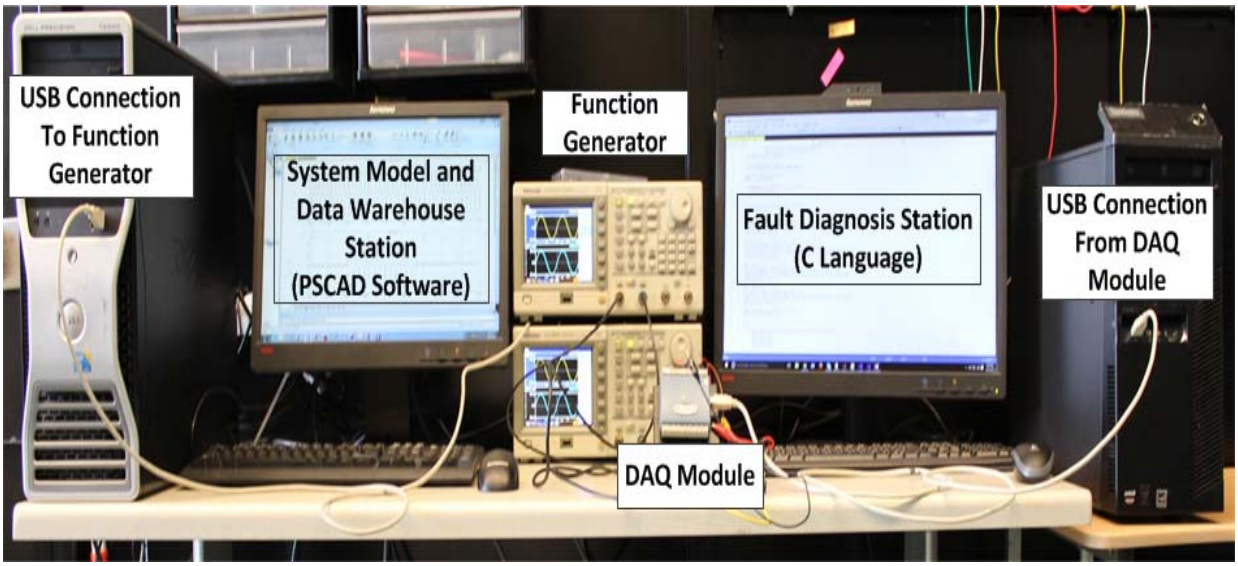


Fig. 4. 59: Real-time experimental setup.

Fig. 4.60 illustrates the schematic block diagram showing the procedures of the experimental execution. The PSCAD/EMTDC software was employed on the PC in order to generate new faulty cases and then the collected data were stored in as a Comma Separated Values (CSV) format. Through the Universal Serial Bus (USB) connection of the PC, the function generator reads the CSV files for the voltage and current, and then plays them back in real-time. The output signal of the function generator is supplied to the DAQ module as an input. The input signal to the DAQ module is then sampled by the DAQ at a sampling rate of 64 samples/cycle. The DAQ module then transfers the sampled data to the PC via USB.

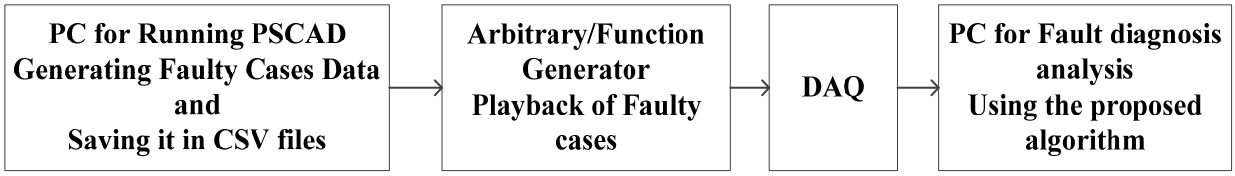


Fig. 4. 60: Schematic block diagram of the experimental setup.

The proposed algorithm for the fault diagnosis was implemented and written on the PC using the C-language, which is presented in detail in the Appendix B, which can be easily implemented on a microcontroller or FPGA. A new set of faulty cases were generated at various operating conditions is listed in Table 4.25.

The total number of cases is 20 and each of these cases were mimicked by the function generator and were transferred to the PC after the DAQ module had completed the sampling process. Table 4.26 shows the overall accuracy in the experimental framework. It is evident in the Table 4.26 that, the proposed AWBFDD algorithm is capable of successfully detecting and classifying the faults for each of the 20 cases. In the case of K-NN and DT classifiers, the proposed algorithm was able to detect and classify 20 cases correctly and accurately.

The results also show that K-NN and DT classifier providing 100% classification accuracy, which confirms the classification accuracy listed in Tables 4.10 and 4.11, and hence confirming the effectiveness of the proposed approach in real-time implementation.

Table 4. 25: Experimental operating conditions.

Case	Fault Location	Fault Resistance	Actual Fault Type	Prediction Fault Type (Experimental)	
				K-NN	DT
1			ABC-G	ABC-G	ABC-G
2			AB	AB	AB
3			BC	BC	BC
4	0 %	0.01 Ω	A-G	AG	AG
5			C-G	C-G	C-G
6			AB-G	AB-G	AB-G
7			AC-G	AC-G	AC-G
8			A-G	A-G	A-G
9			C-G	C-G	C-G
11	50 %	10 Ω	AB	AB	AB
12			AC	AC	AC
13			BC	BC	BC
14			A-G	A-G	A-G
15			B-G	B-G	A-G
16			C-G	C-G	C-G
17	100 %	100 Ω	ABC-G	ABC-G	ABC-G
18			AB	AB	AB
19			AC	AC	AC
20			BC	BC	BC

Table 4. 26: Overall experimental classification accuracy of transmission line system.

Classifier	Misclassified cases	Overall accuracy (%)
DT	0 out of 20	100
K-NN	0 out of 20	100

4.5 Summary

The proposed AWBFDD technique was implemented in this chapter, where a combination of harmony search algorithm, wavelet transform and the machine learning algorithms were utilized.

Firstly, a detailed description of the two test-bed systems (i.e. high voltage (230 kV) transmission line system and low voltage (0.480 kV) Micro-Grid distribution system) were discussed. In order to test the robustness of the proposed approach, a dataset was generated using the models of the two test systems, which were developed in PSCAD/EMTDC software. The data set includes a number of different cases in which the fault types, resistances, location were varied, with a total of 528, and 352 different cases for transmission line system, and Micro-Grid distribution system, respectively. The current and voltages waveforms from each of these cases are sampled at 64 sample/cycle (3.84 kHz) and are saved into a set of comma separated values (CSV) files.

The HSA was used to identify the most suitable combination of decomposition levels, as well as, to identify the most suitable wavelet functions required for fault diagnosis application in each test-bed system instead of using only one wavelet function, as in the previous work in the literature. Therefore, it has been concluded that, the proposed AWBFDD technique has identified different wavelet functions and different decomposition levels for the both test-bed systems, which show the adaptability/flexibility of the proposed algorithm with the changing of the studied system.

The transients of the generated dataset cases were imported into MATLAB where the sequence subtraction was performed and the DWT was applied to the current and voltage signals, using the identified wavelets combination (using the HSA optimizer) to compute the wavelet coefficient energies.

The energy of the wavelet coefficient was used as an input for two Machine Learning classifiers (DT, and K-NN) for which their performance were statistically evaluated. Both K-NN and DT machine learning classifiers were trained and tested using Monte Carlo Stratified Cross-Validation Algorithm on a dataset containing 528, and 352 cases for transmission line system, and Micro-Grid distribution system respectively, which representing different operating conditions.

The results show that the new optimal wavelet combination based on HSA is capable of providing significant enhancement in the classification accuracies in both test-bed systems. Moreover, the results show that, in the case of the new wavelet combination, the prominent attributes holding the patterns of the fault current/voltage signals are separately detected for all fault types. This is unlike the case of using only one wavelet, in which the wavelet coefficient contains the pattern of all single-phase-to-ground faults, which are represented by only one fault type as well as for other fault types (i.e. double-phase faults, and double-phase-to-ground faults).

In the case of transmission line test system, the proposed AWBFDD technique was found to provide a classification accuracy of 100%. The classification accuracies results revealed that the superiority of proposed AWBFDD technique compared to those obtained using the selected wavelet and level in the previous work (i.e. as in [26]-[28], [30], [31], [34], [36], and [38]).

On the other hand, in the case of Micro-Grid distribution test system, the proposed AWBFDD technique was found to provide classification accuracies of 95.63% in the case of K-

NN classifier. In addition, the results revealed the ability of the proposed AWBFDD technique to outperform the other approaches (i.e. as in [41]) in terms of the classification accuracy.

The proposed AWBFDD technique was examined in the presence of different PQ disturbances and measurement uncertainties, as well as the presence of fault current limiter. The results revealed the immunity of the proposed technique to such to PQ disturbances and to the measurement uncertainties in the Micro-Grid distribution system, in addition the immunity of the proposed technique to the FCL influence.

This chapter also discussed the real-time experimental set-up that was used to verify the effectiveness of the proposed AWBFDD technique while thoroughly defining the hardware used of each component. The experimental set-up framework was examined on a new set of faulty cases (i.e. 20 cases) with various operating conditions. The result of the experimental work (i.e. 100% classification accuracy) was confirmed the validity/effectiveness of the proposed approach in real-time implementation.

Chapter 5: Conclusion and Recommendations

5.1 Conclusion

The work presented in this thesis aims to study the time-frequency analysis techniques applied to the fault diagnosis application in the smart distribution systems and Micro-Grids. A state-of-the-art literature review was introduced in which the advantages and disadvantages of previously developed approaches were highlighted. The fault diagnosis methods are divided into two main groups, which were based on various information from the current/voltage signals: steady-state analysis and transient analysis. In the steady-state analysis, it was concluded that the approaches were insufficient in providing enough distinctive features. However, the transient signal analysis usually offers/extracts more distinctive features, which helps in the fault diagnosis application.

Therefore, the work developed in this thesis focused on the transient analysis, which is capable of extracting distinguishing features that could differentiate between various fault types. Through the previous work, the Fourier transform and wavelet transform were applied and it was decided which approach was capable of offering the most distinguishing features. The utilization of the Fourier transform offers features of the signals only in the frequency domain and consequently the attributes in the time domain are missed. Moreover, there are only two basic functions (i.e. sine and cosine), in the Fourier transform analysis, which make the transform not suitable for analyzing transients. As well, in the case of the short-time Fourier transform analysis, it suffers from the fixed window size, and for this reason, any attempt to enhance the frequency resolution comes at the expense of poor time resolution and vice versa.

As a result, it was concluded that the wavelet transform has more capabilities in extracting the features as it was capable of performing the analysis in the frequency and time domain.

Furthermore, the analyzed signal was broken down into two components, including both low and high frequency components, which provided variable window size, and possessed a large library of wavelet functions. Therefore, the work in this thesis focus on the time-frequency analysis by employing the wavelet transform with the identified wavelet functions and decomposition levels using the proposed automated wavelet-based fault detection and diagnosis technique.

This thesis presented a systematic method to identify the combination of the mother wavelets and the wavelet decomposition levels for fault diagnosis application in the smart distribution system Micro-Grids. The proposed automated wavelet-based fault detection and diagnosis technique utilized the harmony search algorithm to optimally identify the suitable wavelet function(s) and the number of wavelet decomposition level(s) that had better extraction of the hidden features in the analysis of the voltage/current signals instead of using only one wavelet function, as in the previous work in the literature. The discreet wavelet transform was then applied to the current and voltage signals, using the identified wavelets combination to compute the wavelet coefficient energies, which was then fed to the machine learning classifier.

Two different machine learning classifiers were implemented in order to automate the fault classification process while at the same time observing if different machine learning techniques influenced the classification accuracy. Specifically, the K-nearest neighbor was selected as a lazy learner while decision tree was chosen as an eager learner. The machine learning classification techniques, decision tree and K-nearest neighbor were utilized and tested using the k-fold cross validation, which is a method to split a dataset into training and testing sets. Monte Carlo trials were performed as a method to determine proper classification accuracies based on the randomness introduced by K-fold.

The proposed automated wavelet-based fault detection and diagnosis technique was tested through simulation by creating a dataset based on a test system in PSCAD/EMTDC with various fault cases including different fault types, fault resistances, and fault locations. Moreover, representing different operating conditions in both non-grid-connected and grid-connected operating modes. This resulted in a dataset consisting of 528 cases in the case of transmission line system, and 352 cases in the case of Micro-Grid distribution system, which was analyzed based on the identified wavelets by the harmony search algorithm. The energies of the identified decomposition levels were calculated in order to train/test the classifier (i.e. decision tree and K-nearest neighbor) model. The results show that the new wavelet combination based on the harmony search algorithm is capable of providing significant improvement in the classification accuracies on a testing set consisting of various fault cases with a various operating conditions.

It is concluded that, the utilization of the wavelet transform analysis using only one wavelet may not be appropriate. For example, the obtained absolute differences among the proposed automated wavelet-based fault detection and diagnosis technique accuracy and those employing only one wavelet (i.e. Daubechies of order-4 (db4)) as the mother wavelet was observed to increase up to 9% and 23% of the K-nearest neighbor and decision tree classifiers in the case of transmission line system.

On the other hand, in the case of Micro-Grid distribution system, the results show that the new wavelet combination based on the HSA is capable of providing significant improvement in the classification accuracies in both operating modes (non-grid-connected and grid-connected). Moreover, the results show that, in the case of the new wavelet combination, the prominent attributes holding the pattern of the fault current/voltage signals in both operating modes of Micro-Grid system are separately detected for all fault types. This is unlike one wavelet, in which the

wavelet coefficient contains the pattern of all single-phase-to-ground faults, which are represented by only one fault type as well as for other fault types (i.e. double-phase faults, and double-phase-to-ground faults). The classification accuracy using the proposed automated wavelet-based fault detection and diagnosis technique was 95.63% in the case of k-NN classifier when compared to only one wavelet (i.e. Daubechies of order-1 (db1)) of 94%, which is an increase of 1.63%.

In addition, it was found that different operating conditions, as well as, different topology/configurations, might affect the current levels, which may lead to failure of many protective approaches. However, the developed automated wavelet-based fault detection and diagnosis technique was able to overcome this problem, which has the flexibility and adaptability to these changes by applying the identified optimal wavelet functions and decomposition levels. For example, the proposed technique was immune to the power quality disturbances and the measurement uncertainty, furthermore it was immune to the changes due to the introduction of fault current limiter in the Micro-Grid distribution system. The classification accuracies obtained in the presence of power quality disturbances, measurement uncertainty, and fault current limiter were 95.44%, 95.60%, and 95.30% respectively.

Finally, an experimental study was implemented to verify the simulation utilizing hardware set-up in the smart grid laboratory at UOIT. The experimental set-up framework was assessed on a new set of faulty cases (i.e. 20 cases) with different operating conditions.

The result of the experimental work (100% classification accuracy) revealed that the proposed automated wavelet-based fault detection and diagnosis technique is capable of successfully detecting and classifying the faults for each of the 20 cases. The results also confirmed that the effectiveness of the proposed technique in real-time implementation.

5.2 Recommendations

When considering the analysis presented in this dissertation, the following recommendations are introduced; the simulation results of the work prove the importance of using the Wavelet Transform analysis technique in the fault diagnosing application as a powerful signal-processing tool. This thesis recommends the use of a combination of wavelet functions in the Wavelet Transform analysis based on a systematic method instead of using only one wavelet.

In order to have a flexible, robust, and reliable smart distribution system, this thesis also recommends that the proposed automated wavelet-based fault detection and diagnosis technique should be expanded to be considered as an application within the Distribution Management Systems (DMS). Consequently, the flexibility and adaptability of the proposed approach can help the operator in the DMS control center, with fault diagnosis by providing reliable and accurate results through determining the optimal wavelet functions and decomposition levels within the DMS integrated platform.

The proposed technique in this work can be useful in the early detection or anticipation of waveforms abnormality with the integration of a historical dataset.

5.3 Future Work

Through the completion of the work outlined in this thesis, more work can be applied to further improve upon the methods proposed. The proposed technique can be used in the detection of waveforms abnormality in the electrical power system and anticipate the failure of the electrical equipment before it occur. The required data in the analysis can be collected/stored using the power quality devices, which can be distributed in a different location in the real distribution system for a period of time.

Also, the proposed method of the selection of the best combination wavelets can be expanded to be used in a different analysis application not only in electric power system but also in another area, such as fault detection and classification in the electric rotating machinery, power electronics inverters, battery storage cells, and PV cells. However, the method was introduced as a technique for improving the classifiers performance.

References

[1] Nebraska Public Power District", online, available at: http://www.nppd.com/vegetation-management/faqs/.

[2] Blackburn, J. Lewis, and Thomas J. "Domin. Protective relaying: principles and applications," CRC press, 2015.

[3] Thomas MS, Macdonald JD (2015) Power System SCADA and smart grids, CRC Press, Taylor and Francis Group.

[4] Kuo, M.T. and Lu, S.D., 2013. Design and implementation of real-time intelligent control and structure based on multi-agent systems in microgrids. Energies, 6(11), pp.6045-6059.

[5] D.E. Olivares, A. Mehrizi-Sani, A.H. Etemadi, C.A. Canizares, R. Iravani, M. Kazerani, A.H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G.A. Jimenez-Estevez, N.D. Hatziargyriou, "Trends in Microgrid Control," *IEEE Trans. Smart Grid* , vol. 5, no. 4, pp. 1905-1919, July 2014.

[6] R. H. Lasseter, "MicroGrids," in Proc. IEEE Power Eng. Soc. Winter Meeting, Jan. 2002, vol. 1, pp. 305–308.

[7] S. A. Gopalan, V. Sreeram, and H. H.C. Iu, "A review of coordination strategies and protection schemes for microgrids," *Renew. Sustain. Energy Rev*., vol. 32, pp. 222- 228, April 2014.

[8] R. F. Arritt and R. C. Dugan, "Distribution system analysis and the future smart grid", *IEEE Trans. Ind. Appl*., vol. 47, no. 6, pp. 2343-350, Dec. 2011.

[9] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the art in research on microgrids: A review," *IEEE Access*., vol. 3, pp. 890–925, Jul. 2015.

[10] M. E. El-Hawary, Electrical power systems: Design and analysis" in IEEE Press Power Systems Eng. Ser, pp. 66-66, 1995, Wiley.

[11] T. S. Abdelgayed; W. G. Morsi; T. S. Sidhu, "A New Harmony Search Approach for Optimal Wavelets Applied to Fault Classification," in IEEE Transactions Smart Grid, doi: 10.1109/TSG.2016.2555141, April 2016.

[12] T. S. Abdelgayed; W. Morsi; T. Sidhu, "A New Approach for Fault Classification in Microgrids Using Optimal Wavelet Functions Matching Pursuit," in IEEE Transactions Smart Grid, doi: 10.1109/TSG.2017.2672881, February 2017.

[13] T. S. Abdelgayed; W. Morsi; T. Sidhu, " Fault Detection and Classification based on Co-Training of Semi-Supervised Machine Learning ," in IEEE Transactions on Industrial Electronics, doi: 10.1109/TIE.2017.2726961, July 2017.

[14] E. Casagrande, W. L. Woon, H. H. Zeineldin, and D. Svetinovic, "A differential sequence component protection scheme for microgrids with inverter-based distributed generators," IEEE Trans. Smart Grid, vol. 5, no. 1, pp. 29–37, Jan. 2014.

[15] A. Hooshyar; E. F. El-Saadany; M. Sanaye-Pasand, "Fault Type Classification in Microgrids Including Photovoltaic DGs," in IEEE Transactions on Smart Grid , vol.PP, no.99, pp.1-1.

[16] F. Costa, A. Monti, and S. Paiva, "Overcurrent Protection in Distribution Systems with Distributed Generation Based on the Real-Time Boundary Wavelet Transform", early access papers, IEEE Transaction on Power Delivery, Dec. 2015.

[17] M. A. Zamani, T. S. Sidhu, and A. Yazdani, "Investigations into the control and protection of an existing distribution network to operate as a microgrid: A case study," IEEE Trans. Ind. Electron., vol. 61, no. 4, pp. 1904–1915, Apr. 2014.

[18] M. A. Zamani, T. S. Sidhu, and A. Yazdani, “A protection strategy and microprocessor-based relay for low-voltage microgrids,” IEEE Trans. Power Del., vol. 26, no. 3, pp. 1873–1883, Jul. 2011.

[19] M. A. Zamani, A. Yazdani, and T. S. Sidhu, “A communication-assisted protection strategy for inverter-based medium-voltage microgrids,” IEEE Trans. Smart Grid, vol. 3, no. 4, pp. 2088–2099, Dec. 2012.

[20] A. Rahmati and R. Adhami, "A Fault Detection and Classification Technique Based on Sequential Components," *IEEE Trans. Ind. Appl.*, vol. 50, no. 6, pp.4202-4209, Dec. 2014

[21] S. Samanta, J. Bera, and G. Sarkar, “An approach for power system fault diagnosis using current samples towards smart grid operation,” in *Proc. Control, Instrumentation, Energy and Communication (CIEC).*, Calcutta, Feb. 2014, pp. 743-746.

[22] A. Yadav and A. Swetapadma, “Fault analysis in three phase transmission lines using k-nearest neighbor algorithm,” in *Proc. Advances in Electronics, Computers and Communications (ICAECC)*, Bangalore, Oct. 2014, pp. 1-5.

[23] P. Gopakumar, M.J.B. Reddy and D.K. Mohanta, “Adaptive fault identification and classification methodology for smart power grids using synchronous phasor angle measurements,” *IET Gener. Transm. Distrib.*, vol. 9, no. 2, pp. 133-145, Jan. 2015.

[24] M. Ben Hessine, H. Jouini and S. Chebbi, “Fault detection and classification approaches in transmission lines using artificial neural networks,” in *Proc. 17th IEEE. Mediterranean Electrotechnical Conference (MELECON)*, Beirut, April 2014, pp. 515-519.

[25] L.L. Zhang, M.S. Li, T.Y. Ji, Q.H. Wu, L. Jiang and J.P. Zhan, “Morphology Singular Entropy-Based Phase Selector Using Short Data Window for Transmission Lines,” *IEEE Trans. Power Del.*, vol. 29, no. 5, pp. 2162-2171, Oct. 2014.

[26] O.A.S. Youssef, "Fault classification based on wavelet transforms," in Proc. *IEEE/PES Transmission and Distribution Conference and Exposition.*, Atlanta, GA, Nov. 2001, pp. 531-536.

[27] M. Solanki and Y.H. Song, "Transient protection of EHV transmission line using discrete wavelet analysis," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, July 2003, vol. 3, pp. 1868–1873.

[28] He Zhengyou, Fu Ling, Lin Sheng and Bo Zhiqian,, "Fault Detection and Classification in EHV Transmission Line Based on Wavelet Singular Entropy," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2156-2163*,* Oct. 2010.

[29] D. Guillen, G. Idarraga-Ospina, A. Zamora, M.R.A. Paternina and J.M. Ramirez, "Fault detection and classification in transmission line using the Euclidian Norm of the total WSE," in *Proc. IEEE PES Transmission & Distribution Conference and Exposition - Latin America (PES T&D-LA).*, Medellin, Sept. 2014, pp. 1-6.

[30] K.M. Silva, B.A. Souza and N.S.D. Brito, "Fault detection and classification in transmission lines based on wavelet transform and ANN," *IEEE Trans. Power Del.,* vol. 21, no. 4, pp. 2058-2063, Oct. 2006.

[31] D. Chanda, N. K. Kishore and A. K. Sinha, "Application of wavelet multiresolution analysis for classification of faults on transmission lines," in *Proc. IEEE Conf. Convergent Technologies for Asia-Pacific Region*, vol. 4, Oct. 2003, pp. 1464–1469.

[32] M. Jaya Bharata Reddy, D. Venkata Rajesh and D.K. Mohanta, "Robust transmission line fault classification using wavelet multi-resolution analysis," *Computers & Electrical Engineering*, vol. 39, no. 4, pp. 1219-1247*,* May 2013.

[33] K. H. Kashyap and U. J. Shenoy, "Classification of power system faults using wavelet transforms and probabilistic neural networks," in Proc. *of the 2003 International Symposium on Circuits and Systems (ISCAS '03).*, Bangkok, May 2003, pp. 423–426.

[34] O.A.S. Youssef, "Combined fuzzy-logic wavelet-based fault classification technique for power system relaying," *IEEE Trans. Power Del.*, vol. 19, no. 2, pp. 582-589*,* April 2004.

[35] J. Upendar, C.P. Gupta, G.K. Singh and G. Ramakrishna, "PSO and ANN-based fault classification for protective relaying," *IET Gener. Transm. Distrib.*, vol. 4, no. 10, pp. 1197-1212, October 2010.

[36] H. Livani and C.Y. Evrenosoglu, "A fault classification method in power systems using DWT and SVM classifier," in *Proc. IEEE PES Transmission and Distribution Conference and Exposition (T&D)*, Orlando, FL, May 2012, pp. 1-5.

[37] Jianyi Chen and R.K. Aggarwal, "A new approach to EHV transmission line fault classification and fault detection based on the wavelet transform and artificial intelligence," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, San Diego, CA., July 2012, pp. 1–8.

[38] F.B. Costa, A.H.P. Sobrinho, M. Ansaldi and M.A.D. Almeida, "The effects of the mother wavelet for transmission line fault detection and classification," *in Proc. 3rd Int. Youth Conf. on Energetics (IYCE)*., Leiria, Jul. 2011, pp. 1-6.

[39] M. Geethanjali and K.S. Priya, "Combined wavelet transforms and neural network (WNN) based fault detection and classification in transmission lines," in *Proc. International Conference on Control, Automation, Communication and Energy Conservation (INCACEC 2009),* Perundurai, Tamilnadu, June 2009, pp. 1-7.

[40] S. Jana and A. De, "Transmission line fault detection and classification using wavelet analysis," in *India Conference (INDICON)*, Mumbai, Dec. 2013, pp. 1-6.

[41] D. P. Mishra; S. R. Samantaray; G. Joos, "A Combined Wavelet and Data-Mining Based Intelligent Protection Scheme for Microgrid," in IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. pp. 2295-2304, Sept. 2016.

[42] S. Alshareef, S. Talwar and W. G. Morsi, "A New Approach Based on Wavelet Design and Machine Learning for Islanding Detection of Distributed Generation," in *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1575-1583, July 2014.

[43] W. G. Morsi, S. Alshareef, and S. Talwar, "A smart multi-purpose monitoring system using wavelet design and machine learning for smart grid applications," U.S. Patent 14 532 621, Nov. 2014.

[44] S. Alshareef and W. G. Morsi, "Application of wavelet-based ensemble tree classifier for non-intrusive load monitoring," *2015 IEEE Electrical Power and Energy Conference (EPEC)*, London, ON, 2015, pp. 397-401.

[45] J. M. Gillis, S. M. Alshareef and W. G. Morsi, "Nonintrusive Load Monitoring Using Wavelet Design and Machine Learning," in *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 320-328, Jan. 2016.

[46] J. Gillis and W. G. Morsi, "Non-intrusive load monitoring using orthogonal wavelet analysis," *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Vancouver, BC, 2016, pp. 1-5.

[47] J. M. Gillis; W. G. Morsi, "Non-Intrusive Load Monitoring Using Semi-Supervised Machine Learning and Wavelet Design," in IEEE Transactions on Smart Grid , vol.PP, no.99, pp.1-8

[48] Chung, Jefferson, "Machine learning classification techniques for non-intrusive load monitoring," master's thesis, UOIT, 2016, http://hdl.handle.net/10155/735.

[49] Gillis, Jessie Michael, "Time-frequency analysis techniques for non-intrusive load monitoring," master's thesis, UOIT, 2016, http://hdl.handle.net/10155/744.

[50] R. Yan and R. X. Gao, "Hilbert–Huang transform-based vibration signal analysis for machine health monitoring," IEEE Trans. Instrum. Meas., vol. 55, no. 6, pp. 2320–2329, Dec. 2006.

[51] H. Li, Z. Haiqi, and T. Liwei, "Wigner–Ville distribution based on EMD for faults diagnosis of bearing," in Fuzzy Systems and Knowledge Discovery. Heidelberg, Germany: Springer, 2006, pp. 803–812.

[52] W. G. Morsi and M. E. EL-Hawary, "A new perspective for the IEEE Standard 1459-2000 via stationary wavelet transform in the presence of non-stationary power quality disturbance," IEEE Trans. Power Del., vol. 23, no. 4, pp. 2356–2365, Oct. 2008.

[53] M. A. Pinsky, Introduction to Fourier Analysis and Wavelets, Providence, Rhode Island: American Mathematical Society, 2002.

[54] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 7, pp. 674 -693, July 1989.

[55] Mladen Kezunovic, Jinfeng Ren, and Saeed Lotfifard, Design, Modeling and Evaluation of Protective Relays for Power Systems, 1st ed. Switzerland, Springer International Publishing, , 2016, DOI 10.1007/978-3-319-20919-7.

[56] Z. W. Geem, "Harmony search algorithm for solving sudoku," in *Proc. Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, vol. 4692, DOI: 10.1007/978-3-540-74819-9_46, pp. 371–378, 2007.

[57] Parizad, A. Khazali, A. Kalantar, M. Application of HSA and GA in optimal placement of FACTS devices considering voltage stability and losses", Electric Power and Energy Conversion Systems, 2009. EPECS '09. International Conference on 10-12 Nov. 2009.

[58] K. S. Lee and Z. W. Geem, "A new metaheuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Comput. Methods Appl. Mech. Eng.*, vol. 194, DOI 10.1016/j.cma.2004.09.007, no. 36–38, pp. 3902–3933, Sep. 2004.

[59] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput*, vol. 188, no. 2, pp. 1567-1579*,* May 2007.

[60] P.-N. Tan, M. Steinbach and V. Kumar *Introduction to Data Mining*. Boston, MA: Addison-Wesley, 2005.

[61] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen*, Classification and Regression Trees*, 1st ed. London, U.K.: Chapman and Hall/ CRC, 1984.

[62] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005.

[63] Q.-S. Xu and Y.-Z. Liang, "Monte carlo cross validation", Chemometrics and Intelligent Laboratory Systems, vol. 56, no. 1, pp. 1- 11, April 2001.

[64] J. A. Jiang, C. L. Chuang, Y. C. Wang, C. H. Hung, J. Y. Wang, C. H. Lee, and Y. T. Hsiao, "Hybrid framework for fault detection, classification, and location Part I: Concept, structure, and methodology," IEEE Trans. Power Del., vol. 26, DOI 10.1109/PESMG.2013.6672597, no. 3, pp. 1988–1997, Jul. 2011.

[65] *"*PSCAD/EMTDC *User's Guide,"* HVDC Research Centre, Winnipeg, MB, Canada, Apr. 2005.

[66] P. M. Anderson, “Analysis of Faulted Power Systems,” in Analytical simplification. New York: IEEE Press, 1995.

[67] E. Alegria, T. Brown, E. Minear, and R. H. Lasseter, “CERTS microgrid demonstration with large-scale energy storage and renewable generation,” IEEE Trans. Smart Grid, vol. 5, no. 2, pp. 937–943, Mar. 2014.

[68] R. H. Lasseter, J. H. Eto, B. Schenkman, J. Stevens, H. Vollkommer, D. Klapp et al., “CERTS microgrid laboratory test bed,” IEEE Trans. Power Del., vol. 26, no. 1, pp. 325–332, Jan. 2011.

[69] A. Yazdani, and R. Iravani, Voltage-sourced converters in power systems: modeling, control, and applications. John Wiley & Sons, Mar. 2010.

[70] A. Yazdani and P. Dash, “A control methodology and characterization of dynamics for a photovoltaic (PV) system interfaced with a distribution network,” *IEEE Trans. Power Del*., vol. 24, no. 3, pp. 1538– 1555, Jul. 2009.

[71] IEEE (1986) Standard requirements for instrument transformers, IEEE/ANSI C57.13

[72] W. El-Khattam and T. S. Sidhu, “Restoration of directional overcurrent relay coordination in distributed generation systems utilizing fault current limiter,” IEEE Trans. Power Del., vol. 23, no. 2, pp. 576–585, Apr. 2008.

Appendix A

A.1 Hardware Component Specifications

A.1.1 Arbitrary/Function generator AFG: Tektronix AFG3022C

Table A.1 outline the specification of the function generator

Table A.1: Tektronix AFG3022C: specification

Features & Benefits
10 MHz, 25 MHz, 50 MHz, 100 MHz, or 240 MHz Sine Waveforms
14 bits, 250 MS/s, 1 GS/s, or 2 GS/s Arbitrary Waveforms
Amplitude up to 20 Vp-p into 50 Ω Loads
5.6 in. Color TFT LCD Display for Full Confidence in Settings and Waveform Shape
Multilanguage and Intuitive Operation Saves Setup Time
Pulse Waveform with Variable Edge Times
AM, FM, PM, FSK, PWM
Sweep and Burst
Dual-channel Models Save Cost and Bench Space
USB Connector on Front Panel for Waveform Storage on Memory Device
USB, GPIB, and LAN
LabVIEW and LabWindows/IVI-C Drivers

A.1.2 Omega 1608FS-PLUS Data Acquisition Module

Table A.2 outline the specification of the data acquisition module.

Table A.2: Omega 1608FS-PLUS: specification

Analog inputs	8 single-ended
A/D Sampling Rate (Continuous Scan to Computer Memory)	100kS/s maximum for any channel
Channel-Gain Current	Up to 8, ordered elements
Digital Output Current	±2.4 mA per pin
Calibration	Factory calibration only
External Clock Input	100 kHz maximum
Trigger Sensitivity	Edge or level sensitive
Counter Input	1 MHz input frequency maximum
Range	Accuracy
±10V	5.66mV
±5V	2.98mV
±2V	1.31mV

Appendix B

B.1 C-Language Code of proposed technique in the experimental work

C-Language Code

```
/*ULAI03.C*************************************************************

File:                         ULAI03.C

Library Call Demonstrated:  cbAInScan(), BACKGROUND mode

Purpose:                      Scans a range of A/D Input Channels
                              in the background.

Demonstration:                Displays the analog input on one channel.
                              Calls cbGetStatus to determine the status
                              of the background operation. Updates the
                              display until a key is pressed.

Other Library Calls:          cbGetStatus()
                              cbStopBackground()
                              cbErrHandling()

Special Requirements:         Board 0 must have an A/D converter.
                              Analog signals on an input channel.


Copyright (c) 1995-2002, Measurement Computing Corp.
All Rights Reserved.
**********************************************************************/
#define _CRT_SECURE_NO_WARNINGS
#define _VC_NODEFAULTLIB
#define numrows(x) (sizeof(x)/sizeof(x[0]))

/* Include files */
#include <windows.h>
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <math.h>
#include <inttypes.h>
#include <time.h>
#include "..\headers\cbw.h"
#include "..\headers\wavelib.h"
#include "..\headers\getRealTime.h"

/* Function Identifiers */
int KNN_Classifier(double unknown_load[], double known_cases[][16], int
known_cases_length);
int DT_Classifier(double unknown_load[]);
```

```
int main () {

    int i, j, x;
    int prevIndex;
    int SamplesCycle = 64; /* Sample rate in samples per cycle per channel
*/
    int SamplesCycleAllChannel = SamplesCycle * 7; /* Sample rate in samples
per cycle all channel */
    bool first_window = false;
    bool classified = false;
    float ch0, ch1, ch2, ch3, ch4, ch5, ch6;
    // double threshold = 20;
    double threshold = 5;
    double Ia, Ib, Ic, In, Va, Vb, Vc;
    double Moving_Window[256][7];
    double average, std_deviation, sum , sum1;
    double variable_holder;
    FILE *fp,*tempf;

    double start_ms, end_ms; /* start and end time of the application after
threshold has been past */
    double *Ia_signal,*Ib_signal,*Ic_signal, *In_signal, *Va_signal,
*Vb_signal, *Vc_signal; /* Post trigger 1 cycle memory space */
    // double *detail_1, *detail_2, *detail_3, *detail_4, *approx_4; /*
Memory space to hold the details and approximation of DWT */
    int start_index_D1, start_index_D2, start_index_D3, start_index_D4;
    int length_D1, length_D2, length_D3, length_D4, length_A4;

    /* Discrete Wavelet Transform variables */
    wave_object obj;
    wt_object wt_bior15, wt_bior13, wt_db25, wt_sym4;
    int wave_level = 4;
    char *name;

    /* Variable Declarations */
    int BoardNum = 0;
    int ULStat = 0;
    int LowChan = 0;
    int HighChan = 6;
    int Gain = BIP5VOLTS;
    short Status = 0;
    long CurCount;
    long CurIndex;
    int Count = SamplesCycle * 60 * 7;
    long Rate = SamplesCycle * 60;
    unsigned Options;
    HANDLE MemHandle = 0;
    WORD *ADData;
    float   RevLevel = (float)CURRENTREVNUM;
    int  ADRes;



    /* Declare UL Revision Level */
    ULStat = cbDeclareRevision(&RevLevel);
```

```
    /* Initiate error handling
        Parameters:
            PRINTALL :all warnings and errors encountered will be printed
            DONTSTOP :program will continue even if error occurs.
                      Note that STOPALL and STOPFATAL are only effective in
                      Windows applications, not Console applications. */
    cbErrHandling (PRINTALL, DONTSTOP);

    /* Get the resolution of A/D */
    cbGetConfig(BOARDINFO, BoardNum, 0, BIADRES, &ADRes);

    MemHandle = cbWinBufAlloc(Count);
    ADData = (WORD*) MemHandle;

    if (!MemHandle) { /* Make sure it is a valid pointer */
        printf("\nout of memory\n");
        exit(1);
    }

    /* Initializing Discrete Wavelet Transform variables */
    int wave_length = 20;

    name = "bior1.5";
    obj = wave_init(name);// Initialize the wavelet
    wt_bior15 = wt_init(obj, "dwt", wave_length, wave_level);// Initialize
the wavelet transform object
    setDWTExtension(wt_bior15, "sym");// Options are "per" and "sym".
Symmetric is the default option
    setWTConv(wt_bior15, "direct");

    name = "bior1.3";
    obj = wave_init(name);// Initialize the wavelet
    wt_bior13 = wt_init(obj, "dwt", wave_length, wave_level);// Initialize
the wavelet transform object
    setDWTExtension(wt_bior13, "sym");// Options are "per" and "sym".
Symmetric is the default option
    setWTConv(wt_bior13, "direct");

    name = "db11";
    obj = wave_init(name);// Initialize the wavelet
    wt_db25 = wt_init(obj, "dwt", wave_length, wave_level);// Initialize the
wavelet transform object
    setDWTExtension(wt_db25, "sym");// Options are "per" and "sym".
Symmetric is the default option
    setWTConv(wt_db25, "direct");

    name = "sym4";
    obj = wave_init(name);// Initialize the wavelet
    wt_sym4 = wt_init(obj, "dwt", wave_length, wave_level);// Initialize the
wavelet transform object
    setDWTExtension(wt_sym4, "sym");// Options are "per" and "sym".
Symmetric is the default option
    setWTConv(wt_sym4, "direct");
```

```
    tempf = fopen("KNN Set/Quarter_Cycle_Faults.csv", "r");
    double known_cases[475][16];

    i = 0;
    do {

fscanf(tempf,"%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%lf,%l
f\n",
            &known_cases[i][0],
            &known_cases[i][1],
            &known_cases[i][2],
            &known_cases[i][3],
            &known_cases[i][4],
            &known_cases[i][5],
            &known_cases[i][6],
            &known_cases[i][7],
            &known_cases[i][8],
            &known_cases[i][9],
            &known_cases[i][10],
            &known_cases[i][11],
            &known_cases[i][12],
            &known_cases[i][13],
            &known_cases[i][14],
            &known_cases[i][15]);
        ++i;
    } while (i < numrows(known_cases));
    fclose(tempf);

    /* Allocating memory for the cycle after the trigger has been hit */
    Ia_signal = (double*)malloc(sizeof(double)* wave_length);
    Ib_signal = (double*)malloc(sizeof(double)* wave_length);
    Ic_signal = (double*)malloc(sizeof(double)* wave_length);
    In_signal = (double*)malloc(sizeof(double)* wave_length);
    Va_signal = (double*)malloc(sizeof(double)* wave_length);
    Vb_signal = (double*)malloc(sizeof(double)* wave_length);
    Vc_signal = (double*)malloc(sizeof(double)* wave_length);

    /* Hardcoded DWT length for the deatils and approximations.
        Hardcoding allows the program to operate faster and allocate the
appropriate memory earlier */
    length_D1 = 12;
    length_D2 = 10;
    length_D3 = 6;
    length_D4 = 20;
    length_A4 = 7;

    start_index_D1 = 37;
    start_index_D2 = 27;
    start_index_D3 = 10;
    start_index_D4 = 20;

    // detail_1 = (double*)malloc(sizeof(double)* length_D1);
    // detail_2 = (double*)malloc(sizeof(double)* length_D2);
    // detail_3 = (double*)malloc(sizeof(double)* length_D3);
    // detail_4 = (double*)malloc(sizeof(double)* length_D4);
    // approx_4 = (double*)malloc(sizeof(double)* length_A4);
```

```
    /* set up the display screen */
    system("cls");
    printf ("Relay Test Program for 20 Samples of 64 Samples per Cycle\n");
    printf ("Press q to manually quit.\n\n");

    /* Collect the values with cbAInScan() in BACKGROUND mode
        Parameters:
            BoardNum    :the number used by CB.CFG to describe this board
            LowChan     :low channel of the scan
            HighChan    :high channel of the scan
            Count       :the total number of A/D samples to collect
            Rate        :sample rate in samples per second
            Gain        :the gain for the board
            ADData[]    :the array for the collected data values
            Options :data collection options */
    Options = CONTINUOUS + BACKGROUND;
    ULStat = cbAInScan (BoardNum, LowChan, HighChan, Count, &Rate, Gain,
MemHandle, Options);

    /*
    *   Since C is so fast the program actually runs a few thousand times
before the A2D can initialize.
    *   This Do While loop waits for the A2D to initialize before actually
running the core of the program
    */
    do {
        ULStat = cbGetStatus (BoardNum, &Status, &CurCount, &CurIndex,
AIFUNCTION);
    } while(CurCount==0);

    printf ("A2D has been initialized and sampling has started.\n");

    while (true) {

        if (_kbhit()) {
            switch(_getch()) {
                case 113:
                    ULStat = cbStopBackground (BoardNum,AIFUNCTION);
                    cbWinBufFree(MemHandle);
                    // free(detail_1);
                    // free(detail_2);
                    // free(detail_3);
                    // free(detail_4);
                    // free(approx_4);
                    free(Ia_signal);
                    free(Ib_signal);
                    free(Ic_signal);
                    free(In_signal);
                    free(Va_signal);
                    free(Vb_signal);
                    free(Vc_signal);
                    exit(1);
                    break;
            }
```

```
        }

        /* Check the status of the current background operation
        Parameters:
            BoardNum  :the number used by CB.CFG to describe this board
            Status    :current status of the operation (IDLE or RUNNING)
            CurCount  :current number of samples collected
            CurIndex  :index to the last data value transferred
            FunctionType: A/D operation (AIFUNCTIOM)*/
        ULStat = cbGetStatus (BoardNum, &Status, &CurCount, &CurIndex,
AIFUNCTION);

        /* Check if there are data in the buffer */
        if (first_window) {
            if (CurIndex < prevIndex)
                CurIndex += Count;

            i = prevIndex;
            while (i < CurIndex) {
                cbToEngUnits(BoardNum, Gain, ADData[i%Count    ], &ch0);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 1], &ch1);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 2], &ch2);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 3], &ch3);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 4], &ch4);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 5], &ch5);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 6], &ch6);

                Ia = ((ch0 - 0.0015) * 175);
                Ib = ((ch1 - 0.0015) * 175);
                Ic = ((ch2 - 0.0015) * 175);
                In = ((ch3 - 0.0015) * 175);
                Va = (ch4 * 39);
                Vb = (ch5 * 39);
                Vc = (ch6 * 39);

                /*
                    Threshold Checker
                */
                if ((fabs(Ia -
Moving_Window[(i%SamplesCycleAllChannel)/7][0]) > threshold ||
                     fabs(Ib -
Moving_Window[(i%SamplesCycleAllChannel)/7][1]) > threshold ||
                     fabs(Ic -
Moving_Window[(i%SamplesCycleAllChannel)/7][2]) > threshold ||
                     fabs(In -
Moving_Window[(i%SamplesCycleAllChannel)/7][3]) > threshold ) &&
!classified) {
                    do {
                        ULStat = cbGetStatus (BoardNum, &Status, &CurCount,
&CurIndex, AIFUNCTION);

                        if (CurIndex < i)
                            CurIndex += Count;

                        if ((CurIndex - i) >= (140)) {
```

```
                                // fp = fopen("A2D_Output.csv", "a+");
                                j = i;
                                while (j < i+(140)) {
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
], &ch0);
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
+ 1], &ch1);
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
+ 2], &ch2);
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
+ 3], &ch3);
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
+ 4], &ch4);
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
+ 5], &ch5);
                                    cbToEngUnits(BoardNum, Gain, ADData[j%Count
+ 6], &ch6);

                                    Ia = ((ch0 - 0.0015) * 100) -
Moving_Window[(j%SamplesCycleAllChannel)/7][0];
                                    Ib = ((ch1 - 0.0015) * 100) -
Moving_Window[(j%SamplesCycleAllChannel)/7][1];
                                    Ic = ((ch2 - 0.0015) * 100) -
Moving_Window[(j%SamplesCycleAllChannel)/7][2];
                                    In = ((ch3 - 0.0015) * 100) -
Moving_Window[(j%SamplesCycleAllChannel)/7][3];
                                    Va = (ch4 * 39) -
Moving_Window[(j%SamplesCycleAllChannel)/7][4];
                                    Vb = (ch5 * 39) -
Moving_Window[(j%SamplesCycleAllChannel)/7][5];
                                    Vc = (ch6 * 39) -
Moving_Window[(j%SamplesCycleAllChannel)/7][6];

                                    // fprintf(fp, "%G,%G,%G,%G,%G,%G,%G\n",
Ia,Ib,Ic,In,Va,Vb,Vc);

                                    Ia_signal[(j%i)/7] = Ia;
                                    Ib_signal[(j%i)/7] = Ib;
                                    Ic_signal[(j%i)/7] = Ic;
                                    In_signal[(j%i)/7] = In;
                                    Va_signal[(j%i)/7] = Va;
                                    Vb_signal[(j%i)/7] = Vb;
                                    Vc_signal[(j%i)/7] = Vc;

                                    j += 7;
                                }
                                // fclose(fp);

                                start_ms = getRealTime();

                                double load_energy[15] = {0};

                                j = 0;
                                do {
```

```
                                switch(j) {
                                    case 0:
                                        dwt(wt_bior15, Ia_signal);// Perform
DWT
                                        dwt(wt_bior13, Ia_signal);// Perform
DWT
                                        dwt(wt_db25, Ia_signal);// Perform
DWT
                                        dwt(wt_sym4, Ia_signal);// Perform
DWT
                                        break;
                                    case 1:
                                        dwt(wt_bior15, Ib_signal);// Perform
DWT
                                        dwt(wt_bior13, Ib_signal);// Perform
DWT
                                        dwt(wt_db25, Ib_signal);// Perform
DWT
                                        dwt(wt_sym4, Ib_signal);// Perform
DWT
                                        break;
                                    case 2:
                                        dwt(wt_bior15, Ic_signal);// Perform
DWT
                                        dwt(wt_bior13, Ic_signal);// Perform
DWT
                                        dwt(wt_db25, Ic_signal);// Perform
DWT
                                        dwt(wt_sym4, Ic_signal);// Perform
DWT
                                        break;
                                }

                                x = 0;
                                do {
                                    variable_holder = wt_sym4->output[x];
                                    load_energy[12 + j] += variable_holder *
variable_holder;
                                    ++x;
                                } while (x < wt_sym4->length[0]);

                                x = 0;
                                do {
                                    variable_holder = wt_db25-
>output[x+start_index_D4];
                                    load_energy[9 + j] += variable_holder *
variable_holder;
                                    ++x;
                                } while (x < wt_db25->length[1]);

                                x = 0;
                                do {
                                    variable_holder = wt_bior13-
>output[x+start_index_D3];
                                    load_energy[6 + j] += variable_holder *
variable_holder;
```

```
                                    ++x;
                                } while (x < wt_bior13->length[2]);

                                x = 0;
                                do {
                                    variable_holder = wt_bior15-
>output[x+start_index_D2];
                                    load_energy[3 + j] += variable_holder *
variable_holder;
                                    ++x;
                                } while (x < wt_bior15->length[3]);

                                x = 0;
                                do {
                                    variable_holder = wt_bior15-
>output[x+start_index_D1];
                                    load_energy[0 + j] += variable_holder *
variable_holder;
                                    ++x;
                                } while (x < wt_bior15->length[4]);
                                ++j;
                            } while(j < 3);

                            sum = 0;
                            sum1 = 0;
                            j = 0;
                            do
                            {
                                sum = sum + load_energy[j];
                                ++j;
                            } while (j < 15);
                            average = sum / 15.0;

                            j = 0;
                            do
                            {
                                sum1 = sum1 + pow((load_energy[j] -
average), 2);
                                ++j;
                            } while (j < 15);
                            std_deviation = sqrt(sum1 / 15.0);

                            j = 0;
                            do
                            {
                                load_energy[j] = (load_energy[j] - average)
/ std_deviation;
                                ++j;
                            } while (j < 15);

                            /* Classifiers */
                            int KNN_predicted_class, DT_predicted_class;
                            KNN_predicted_class = 0;
                            DT_predicted_class = 0;
```

```
                            KNN_predicted_class =
KNN_Classifier(load_energy, known_cases, numrows(known_cases));
                            DT_predicted_class = DT_Classifier(load_energy);

                            end_ms = getRealTime();

                            switch(KNN_predicted_class) {
                                case 1:
                                    printf ("\tThe KNN predicted class is:
A-G\n");
                                    break;
                                case 2:
                                    printf ("\tThe KNN predicted class is:
B-G\n");
                                    break;
                                case 3:
                                    printf ("\tThe KNN predicted class is:
C-G\n");
                                    break;
                                case 4:
                                    printf ("\tThe KNN predicted class is:
AB-G\n");
                                    break;
                                case 5:
                                    printf ("\tThe KNN predicted class is:
AC-G\n");
                                    break;
                                case 6:
                                    printf ("\tThe KNN predicted class is:
BC-G\n");
                                    break;
                                case 7:
                                    printf ("\tThe KNN predicted class is:
ABC-G\n");
                                    break;
                                case 8:
                                    printf ("\tThe KNN predicted class is:
AB\n");
                                    break;
                                case 9:
                                    printf ("\tThe KNN predicted class is:
AC\n");
                                    break;
                                case 10:
                                    printf ("\tThe KNN predicted class is:
BC\n");
                                    break;
                                default:
                                    printf ("\tThe KNN couldn't
predict.\n");
                                    break;
                            }

                            switch(DT_predicted_class) {
                                case 1:
```

```
                                    printf ("\tThe DT predicted class is: A-
G\n");
                                    break;
                                case 2:
                                    printf ("\tThe DT predicted class is: B-
G\n");
                                    break;
                                case 3:
                                    printf ("\tThe DT predicted class is: C-
G\n");
                                    break;
                                case 4:
                                    printf ("\tThe DT predicted class is:
AB-G\n");
                                    break;
                                case 5:
                                    printf ("\tThe DT predicted class is:
AC-G\n");
                                    break;
                                case 6:
                                    printf ("\tThe DT predicted class is:
BC-G\n");
                                    break;
                                case 7:
                                    printf ("\tThe DT predicted class is:
ABC-G\n");
                                    break;
                                case 8:
                                    printf ("\tThe DT predicted class is:
AB\n");
                                    break;
                                case 9:
                                    printf ("\tThe DT predicted class is:
AC\n");
                                    break;
                                case 10:
                                    printf ("\tThe DT predicted class is:
BC\n");
                                    break;
                                default:
                                    printf ("\tThe DT couldn't predict.\n");
                                    break;
                            }

                            double prediction_time = (end_ms-start_ms +
((1.0/60.0) * wave_length/SamplesCycle)) * 1000;
                            printf ("\tThe prediction time is: %G\n",
prediction_time);

                            double test_ms1, test_ms2;
                            test_ms1 = getRealTime();
                            Sleep(1);
                            test_ms2 = getRealTime();
                            double accuracy_ms = (test_ms2 - test_ms1) *
1000;
```

```
                            tempf = fopen("20_Samples_Energy.csv", "a+");

fprintf(tempf,"%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%G,%i,%i,%G,%G\n",
                                load_energy[0], load_energy[1],
load_energy[2], load_energy[3], load_energy[4],
                                load_energy[5], load_energy[6],
load_energy[7], load_energy[8], load_energy[9],
                                load_energy[10], load_energy[11],
load_energy[12], load_energy[13], load_energy[14],
                               KNN_predicted_class, DT_predicted_class,
prediction_time, accuracy_ms);
                            fclose(tempf);

                            ULStat = cbStopBackground (BoardNum,AIFUNCTION);
                            cbWinBufFree(MemHandle);
                            // free(detail_1);
                            // free(detail_2);
                            // free(detail_3);
                            // free(detail_4);
                            // free(approx_4);
                            free(Ia_signal);
                            free(Ib_signal);
                            free(Ic_signal);
                            free(In_signal);
                            free(Va_signal);
                            free(Vb_signal);
                            free(Vc_signal);
                            exit(1);

                            // first_window = false;
                            // classified = true;
                            // i = CurIndex;
                        }
                    } while(!classified);
                }

                Moving_Window[(i%SamplesCycleAllChannel)/7][0] = Ia;
                Moving_Window[(i%SamplesCycleAllChannel)/7][1] = Ib;
                Moving_Window[(i%SamplesCycleAllChannel)/7][2] = Ic;
                Moving_Window[(i%SamplesCycleAllChannel)/7][3] = In;
                Moving_Window[(i%SamplesCycleAllChannel)/7][4] = Va;
                Moving_Window[(i%SamplesCycleAllChannel)/7][5] = Vb;
                Moving_Window[(i%SamplesCycleAllChannel)/7][6] = Vc;

                i += 7;
            }
            prevIndex = CurIndex%Count;
        } else if (CurCount > SamplesCycleAllChannel && !first_window) {
            /* Save the first 256 sample points inside the moving window */
            if (CurIndex < SamplesCycleAllChannel)
                CurIndex += Count;

            i = CurIndex - SamplesCycleAllChannel;
            while (i < CurIndex) {
                cbToEngUnits(BoardNum, Gain, ADData[i%Count    ], &ch0);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 1], &ch1);
```

```
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 2], &ch2);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 3], &ch3);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 4], &ch4);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 5], &ch5);
                cbToEngUnits(BoardNum, Gain, ADData[i%Count + 6], &ch6);

                Moving_Window[(i%SamplesCycleAllChannel)/7][0] = (ch0 -
0.0015) * 175,
                Moving_Window[(i%SamplesCycleAllChannel)/7][1] = (ch1 -
0.0015) * 175;
                Moving_Window[(i%SamplesCycleAllChannel)/7][2] = (ch2 -
0.0015) * 175;
                Moving_Window[(i%SamplesCycleAllChannel)/7][3] = (ch3 -
0.0015) * 175;
                Moving_Window[(i%SamplesCycleAllChannel)/7][4] = ch4 * 39;
                Moving_Window[(i%SamplesCycleAllChannel)/7][5] = ch5 * 39;
                Moving_Window[(i%SamplesCycleAllChannel)/7][6] = ch6 * 39;

                i += 7;
            }
            first_window = true;
            classified = false;
            prevIndex = CurIndex%Count;
            printf("Moving window has been initialized.\n");
        }
    }
    ULStat = cbStopBackground (BoardNum,AIFUNCTION);
    cbWinBufFree(MemHandle);
    // free(detail_1);
    // free(detail_2);
    // free(detail_3);
    // free(detail_4);
    // free(approx_4);
    free(Ia_signal);
    free(Ib_signal);
    free(Ic_signal);
    free(In_signal);
    free(Va_signal);
    free(Vb_signal);
    free(Vc_signal);
    return 1;
}

int KNN_Classifier(double unknown_load[], double known_cases[][16], int
known_cases_length) {
    /* READ IN FILE OF ALL CLASSES */
    /* KNOWN LOAD VARIABLE NAME: known_cases */
    int i = 0;
    double predicted_class;
    double min_distance;

    do {
        double distance = sqrt ( pow (unknown_load[0]-known_cases[i][0],2) +
                                 pow (unknown_load[1]-known_cases[i][1],2) +
                                 pow (unknown_load[2]-known_cases[i][2],2) +
                                 pow (unknown_load[3]-known_cases[i][3],2) +
```

```
                                    pow (unknown_load[4]-known_cases[i][4],2) +
                                    pow (unknown_load[5]-known_cases[i][5],2) +
                                    pow (unknown_load[6]-known_cases[i][6],2) +
                                    pow (unknown_load[7]-known_cases[i][7],2) +
                                    pow (unknown_load[8]-known_cases[i][8],2) +
                                    pow (unknown_load[9]-known_cases[i][9],2) +
                                    pow (unknown_load[10]-known_cases[i][10],2)
+
                                    pow (unknown_load[11]-known_cases[i][11],2)
+
                                    pow (unknown_load[12]-known_cases[i][12],2)
+
                                    pow (unknown_load[13]-known_cases[i][13],2)
+
                                    pow (unknown_load[14]-
known_cases[i][14],2));
        if (i==0) {
            min_distance = distance;
            predicted_class = known_cases[i][15];
        } else if (distance < min_distance) {
            min_distance = distance;
            predicted_class = known_cases[i][15];
        }
        i++;
    } while (i < known_cases_length);
    return (int)predicted_class;
}

int DT_Classifier(double unknown_load[]) {
    if (unknown_load[6] < -0.422354) {
        if (unknown_load[12] < 2.04664) {
            return 7;
        } else if (unknown_load[12] >= 2.04664) {
            if (unknown_load[3] < -0.441323) {
                return 9;
            } else if (unknown_load[3] >= -0.441323) {
                return 8;
            }
        }
    } else if (unknown_load[6] >= -0.422354) {
        if (unknown_load[10] < -0.302983) {
            if (unknown_load[3] < -0.311361) {
                return 5;
            } else if (unknown_load[3] >= -0.311361) {
                return 1;
            }
        } else if (unknown_load[10] >= -0.302983) {
            if (unknown_load[4] < -0.278922) {
                if (unknown_load[0] < -0.331429) {
                    if (unknown_load[9] < -0.315011) {
                        if (unknown_load[4] < -0.408916) {
                            return 10;
                        } else if (unknown_load[4] >= -0.408916) {
                            if (unknown_load[0] < -0.407283) {
                                if (unknown_load[1] < -0.418293) {
                                    return 10;
```

```
                                } else if (unknown_load[1] >= -0.418293) {
                                    return 6;
                                }
                            } else if (unknown_load[0] >= -0.407283) {
                                return 10;
                            }
                        }
                    } else if (unknown_load[9] >= -0.315011) {
                        if (unknown_load[12] < 2.3447) {
                            return 4;
                        } else if (unknown_load[12] >= 2.3447) {
                            if (unknown_load[0] < -0.435209) {
                                return 4;
                            } else if (unknown_load[0] >= -0.435209) {
                                return 8;
                            }
                        }
                    }
                } else if (unknown_load[0] >= -0.331429) {
                    return 3;
                }
            } else if (unknown_load[4] >= -0.278922) {
                return 2;
            }
        }
    }
    return 0;
}
```