A Comparative Analysis of Machine Learning Algorithms for Fault Detection and Classification in Microgrids

By

Charles Munyi Gathiani

Thesis
Submitted to Flinders University
for the degree of

Master of Engineering, Electrical and Electronics

Flinders University
27.05.2024

TABLE OF CONTENTS

ABSTRACT

Optimal operation and control of microgrids depends on the accuracy of the fault detection and classification capabilities, which allows for quick fault identification, isolation, and recovery. Due to a reliance on large fault currents and the dynamic nature of microgrids, there is a need for the development of new fault detection techniques.

This study investigates and proposes a machine learning-based microgrid fault detection scheme for high precision using Bayesian regularization algorithm. The proposed machine learning method extracts its learning features from the point of common coupling of the distributed energy resources and the main grid using the discrete wavelet transform. Under different fault and microgrid operating conditions, the learning features extracted were the three-phase measurements of the voltage magnitude, three-phase measurements of the current magnitude, fault impedance, zero sequence voltage values, zero sequence current values, and frequency. The Discrete Wavelet Transform was used to extract the learning features and then decompose them into the time-frequency characteristics. The eight extracted features were then applied as the input variables for purposes of machine learning.

To investigate and validate the performance and effectiveness of the detection and classification model, the results were compared to other training algorithms for accuracy, selectivity, and sensitivity. The results of the simulations were compared to the Levenberg Marquardt training algorithm. The simulation results clearly indicate that the Bayesian Regularization algorithm provides more accurate detection and classification of faults while guaranteeing better response to changes in input variables resulting from microgrid operating conditions. The Bayesian Regularization algorithm did not experience overfitting and provided accurate results even with an introduction of new variables. Although the Bayesian Regularization algorithm provided accurate results and the best response, it had a longer processing time which may not be suitable for use in time-constrained operations.

DECLARATION

I certify that this thesis:

1. does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university.
2. and the research within will not be submitted for any other future degree or diploma without the permission of Flinders University; and
3. to the best of my knowledge and belief, does not contain any material previously published or written by another person except where due reference is made in the text.

Signature of student....Charles Munyi............

Print name of student: Charles Munyi Gathiani

Date: 27/05/2024

I certify that I have read this thesis. In my opinion it is/is not (please circle) fully adequate, in scope and in quality, as a thesis for the degree of Master of Electrical Engineering. Furthermore, I confirm that I have provided feedback on this thesis and the student has implemented it minimally/partially/fully (please circle).

Signature of Principal Supervisor..

Print name of Principal Supervisor....Amin Mahmoudi....

Date....27 May 2024....

ACKNOWLEDGEMENTS

I would like to acknowledge and express huge gratitude to the individuals whose support and contributions have been hugely instrumental toward the successful completion of this research. First and foremost, I extend my heartfelt gratitude to my primary supervisor Dr. Amin Mahmoudi for his guidance, invaluable insight, and continuous feedback throughout the project. His unwavering commitment has greatly enriched the quality of this work.

I am also indebted to my College of Engineering peers for their constructive feedback and thoughtful suggestions that enhanced the depth and rigor of this study. Special thanks to Flinders University for providing access to lab facilities and resources that played a huge role in executing this research.

Lastly, I am grateful for my family and wife for their encouragement, understanding and unwavering support. Their belief in my abilities pushed me through all the highs and lows of my research journey.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Background

The increased push toward the minimization of carbon emissions has resulted in a rapid adoption of renewable energy sources and the deployment of Distributed Energy Resources (DER). This transformation in the energy market has contributed to the development of microgrids. A microgrid is a localised group of distributed energy resources, loads, and energy storage systems within specific boundaries, like remote mining sites, cities, and campuses, as shown in Figure 1.

Figure removed due to copyright restriction

Figure 1: Layout of a Microgrid with Distributed Energy Resources, Loads, and Energy Storage Systems [1].

Microgrids predominantly feature as an attractive way of electricity delivery to final consumers. The advancement of hardware and software technologies is driving the microgrid market through a cost-effective approach thanks to the commoditization of battery storage and solar PV. With governments continuing to invest in microgrids, the annual spending is expected to reach $25.9 billion by 2030, with a cumulative spending expected to hit $110.5 billion globally as the Asia Pacific region continued to be the market leader as shown in Figure 2. The Australian government is bolstering its plans to transform Australia into a renewable energy superpower with a $40 billion commitment to renewable energies in the 2023-2024 budget [2].

Figure removed due to copyright restriction

Figure 2: Microgrid Capacity and Spending by Region Between 2021 to 2030 [3].

The continued penetration of renewable energy as an alternative power source has sparked an increased adoption of microgrids, which are characterized by Distributed Energy Resources (DER), energy storage systems, and loads. The efficient operation and management of microgrids has been underpinned by the integration of sophisticated control algorithms and modern power electronics.

For resilience and reliability, microgrids must be secured and protected in both islanded and grid operating modes. The adoption of microgrids has been largely limited by the challenge of protection. Microgrid protection is difficult to implement in inverter-based microgrids because inverters do not provide large fault currents during faults. This is in contrast to traditional distribution power protection system have been designed for radial flows and that depend on large fault currents from induction and synchronous machines. In inverter-based systems, fault currents look similar to motor inrush or start currents and classic overcurrent protection schemes may not detect faults.

1.2 Problem Statement

While microgrids have come off as a secure, reliable, and cost-effective energy supply option, their inherent configuration continues to pose challenges in protection. The high penetration of distributed energy resources such as inverter interfaced distributed generators (IIDGs) and asynchronous-based distributed generators (ASDG) makes the protection and control of microgrids challenging Dehghanian et al. [4]. The different operating modes, which are grid-connected and islanded mode, also add to the problem of protection in microgrids. These challenges are attributed to the peculiar characteristics associated with microgrids, including: (i) variations in fault current (Due to grid-connected and islanded operating modes, weather conditions, e.g., irradiance, and DER operating modes), (ii) bi-directional power flow, (iii) limited short-circuit capacity, and (iv) low inertia resulting in critical frequency abnormalities in islanded mode. Microgrids also rely on inverter-based resources where inverters don't dynamically behave in the same manner as induction/synchronous machines since their fault current are not based on the electromagnetic characteristics of traditional machines Maheswari et al. [5]. These varying characteristics call for the development of a robust fault detection and classification framework.

A relevant research area is the detection and classification of faults in microgrids using machine learning.

1.3 Objectives

- To develop a fault detection and classification framework for microgrids based on machine learning.
- To model various microgrid operating parameters, operating conditions and key fault parameters to obtain a nearly realistic machine learning training dataset.
- To apply machine learning in the prediction and classification of three phase microgrid faults.
- To compare and assess the performance and effectiveness of two machine learning training algorithms (Bayesian Regularization and Levenberg Marquardt).
- To test and validate the machine learning algorithms using a standard IEC microgrid.

1.4 Methodology

The research is primarily performed using the MATLAB/Simulink simulation software. The simulation of three phase microgrid faults is performed by the development of 12 individual microgrid models to represent each fault scenario. Different microgrid operating and load conditions are also simulated in order to achieve a variety of realistic microgrid operating conditions. Different operating conditions are achieved by changing the solar irradiance and wind power generation profiles. Different load conditions are achieved by using constant load as well as dynamic loads. MATLAB scripts are used to read microgrid fault data and the Discrete Wavelet Transform is used to perform signal processing of eight extracted features, phase voltages, phase currents, zero sequence voltage, and zero sequence current. These extracted features are used as the inputs for training the machine learning algorithm. A performance analysis and assessment is performed on the machine learning models, which are then tested on a standard IEC microgrid.

1.5 Limitations

The machine learning models and techniques proposed in this study was based on simulated data of real-world microgrid systems. While these models are useful in performing various analyses, they cannot be an actual substitute for real-world, measured data. Besides, there is a possible shortcoming of not being able to generate a simulation data set that exhaustively and truly represents all possible real-world scenarios. Besides, since machine learning methods may perform poorly when they encounter new cases or rare occurrences not learned in the training, simulated data may be inadequate for training commercialized machine learning models for real world systems.

Machine learning models, particularly deep learning models are quite prone to overfitting. This occurs when a model performs well on training data but responds poorly to new and unseen data. In this work, the Levenberg Marquardt training algorithm was found to suffer from overfitting, hence the use of a regularisation algorithm to mitigate overfitting.

CHAPTER 2

Literature Review

2.1 Introduction

Microgrids, which are characterized by the integration of DERs, loads, and energy storage systems are increasingly being adopted for the potential they bear in enhancing efficiency, reliability, and sustainability of overall power systems. However, their dynamic and intermittent nature presents huge challenges in terms of protection, fault detection, and classification. Conventional protection and detection schemes designed for traditional grids with large fault currents and unidirectional flow of power are often insufficient for microgrids which may exhibit low fault currents, bidirectional power flow, and operate in either islanded or grid connected mode. This literature will review the present state of research into machine learning algorithms for fault detection in microgrids, highlighting key methodologies, comparative studies, challenges, and any advancements.

2.2 Machine Learning

The use of machine learning in power systems has gained traction because of its ability to work with huge data sets and identify complex operational patterns. Machine learning is by large an umbrella term referring to a wide range of algorithms that make intelligent predictions based on a specific data set Nichols et al. [6]. With regard to microgrids, machine learning schemes are utilized to help enhance fault detection and classification by processing the diverse data that is generated from various measuring points and devices. Machine learning algorithms are largely classified into supervised, unsupervised, and reinforcement learning.

- **Supervised learning** trains a machine learning model based on labelled data where the inputs and expected outputs are known. Some common supervised learning ML algorithms include Decision trees, Support Vector Machine (SVM), Neural Networks, and Random Forests. Models developed from these algorithms learn from historical fault data and make relevant predictions based on new and unseen data. More popular supervised learning tasks are classification, which separates data and regression, which fits data Han [7].
- **Unsupervised learning** uses unlabelled data to find hidden patterns or intrinsic structures in the input data without human interference. With this concept, popular learning tasks include density estimation, clustering, feature learning, anomaly detection, etc.
- **Reinforcement learning** technique lets machines and software agents evaluate the optimal behaviour in a specific environment or context to improve efficiency. This learning uses a

penalty-reward approach to reward desired outcomes and penalize undesired outcomes Sarker et al. [8].

2.3 Feature Extraction Techniques

The effective detection of faults in microgrids heavily relies on the accurate extraction of key fault signatures and features from raw data. These fault features include current and voltage measurements, frequency, impedance, and the various sequence components. The Discrete Wavelet Transform (DWT) is a powerful feature extraction technique that decomposes signals into their time-frequency components to capture the transient behaviour characteristics of different faults in microgrids.

2.4 Machine Learning Algorithms for Fault Detection

Multiple machine learning algorithms have been explored and investigated for the detection and classification of faults in microgrids. This section will review the more commonly used ones, including their potential merits and limitations.

- **Support Vector Machines (SVM):** SVMs have proven to be effective for classification problems that are binary and of multiple classes. These algorithms work to identify the optimal hyperplane that distinguishes various classes in the features. SVMs have demonstrated great robustness and high accuracy in the detection of various fault types in microgrids.
- **Random Forests and Decision Trees:** The Random Forest machine learning method builds on the idea of decision trees by combining a forest of decision trees working together into a single model. Each decision tree that is within the model makes a prediction and the results from the majority of trees become the result of the algorithm. The number and complexity of trees can be altered to test the combination that provides the best results [9].
- **Neural Networks (NN):** The neural network algorithm is modelled based on the communication of neurons in the brain, allowing it to detect relationships and patterns within datasets. This algorithm is based on an input layer, hidden layers, and an output layer. Neural networks were used for this study.

2.5 Comparative Studies

Multiple studies have explored the use of machine learning and analysed the performance of different schemes in the detection and classification of faults in microgrids.

In [9], Deb & Jain investigate the detection and classification of faults in low-voltage microgrids using a hybrid machine learning approach that combined the cosine k-Nearest Neighbour and Bagged Ensemble Learner (BEL) algorithms. Local current and voltage measurements are used by BEL to detect faults while C-kNN performs classification as either pole-to-pole or pole-to-ground. While this method showed high accuracy in fault detection and a promise of efficiency in real-time applications, its effectiveness for larger, complex systems still needs further validation.

Ahmadi et al. [10] introduced a novel method of detecting high impedance faults in power grids using Support Vector Machines (SVMs). The study focused on the identification of faults such as single line and double line in distributed generation system. Notable improvement in fault detection was observed through the differentiation of fault conditions from other fault-like phenomena while providing an enhanced response time when compared to methods like the Wavelet Transformation (WT) method. While this study showed the effectiveness of SVM in differentiating fault and non-fault scenarios while considering variations in fault impedance, changing in microgrid operating conditions were not considered.

The study by Ananth et al. [11] focused on the use of Generalized Regression Neural Networks to estimate the dynamic fault currents in microgrids under different operational conditions. The approach employed adopted well to different fault scenarios induced by the microgrid operating modes (islanded or grid-connected) and weather conditions that could affect power generation. The dynamic fault conditions influenced by different operating modes were considered for fault currents alone and no other parameters were put to test.

Kanojia & Shah [12] presented a fault detection method using the Wavelet theory in grid-tied solar PV and battery-based AC microgrid. Particle Swarm Optimization for MPPT was proposed in the study to help improve power quality problems during faults. While real-time microgrid operating conditions with varying solar irradiances were studies for quick and accurate fault detection, different impedance levels were not considered for the various fault conditions.

A method for diagnosing faults in islanded microgrids using the Support Vector Machine (SVM) and the Wavelet sliding window energy was presented by Han et al. [13]. By addressing the challenges of limited fault information and short fault dynamics, this study enhanced feature extraction by processing wavelet coefficients with the help of sliding windows. Transient characteristics were analysed, and wavelet energy used as a fault feature vector to establish fast and accurate local microgrid fault diagnosis. While a higher diagnostic accuracy was achieved with incredibly short diagnosis times, only islanded operating mode of the microgrid was considered.

Sarangi et al. [14] presented an intelligent microgrid protection scheme using the Fast Fourier Discrete Transform to extract differential and spectral energy features resulting from microgrid fault scenarios. Key features like minimum and maximum differential energy were calculated and used as inputs to a hybrid optimization algorithm that combined Kernel Extreme Learning Machine (KELM) and Particle Swarm Optimization (PSO) for accurate fault detection. Innovative signal processing was used as the primary method of fault detection.

Sahoo & Samal [15] introduced a novel machine learning approach for online fault detection, classification, and prediction. Different fault types, including line to line and line to ground were detected and classified using a Deep Neural Network (DNN). Instantaneous fault parameters like voltage and current are processed in real time while data analysis was performed using techniques like SVM, Naive Bayes and logistic regression. While this study compared multiple machine learning methods, the consideration of varying fault impedance levels was not considered.

2.6 Gap Statement

Although numerous papers have tried to explore fault detection in microgrids, several gaps are still evident in most of the studies undertaken. A key shortcoming of most studies is the lack of consideration of various parameters that are pertinent in the operation of microgrids. The consideration of a combination of operational parameters like current, voltage, impedance, and frequency are crucial in the analysis of faults in microgrids. Accurate fault detection and analysis requires an analysis of the dynamic operating nature of microgrids. As a result, the impact of different microgrid operating conditions like changing solar irradiance and varying loads is not adequately investigated. Most of the studies in the literature explored consider stable microgrid operating conditions. The combination of various microgrid operational parameters and operating conditions in the investigation of faults can help engineers create resilient protection systems.

CHAPTER 3

Methodology

3.1 Microgrid Model Design and Development

Building machine learning models for fault detection and classification in microgrids requires the generation of data to train and test the models. A more reliable way to obtain this data is by collecting field measurements from microgrid systems. However, this presents the risk of damage on microgrid equipment by short circuit currents and voltages due to intentional faults. As a result, the required dataset was produced through a numerical simulation of microgrids on MATLAB/Simulink. Different microgrid operating events are considered to ensure the machine learning algorithm works on a large dataset for reliability and to address the research gaps identified where studies focus on few parameters. The microgrid operating events are classified into no fault events (to represent normal operating conditions) and fault events. The fault event and no-fault event groups are summarized in Table 1 and Table 2.

TABLE 1: Table Showing the Simulation Parameters, Fault Events, and the Number of Counts for Each When Fault Events Were Considered.

Parameters	Events	Variations
Type of Fault	Single line to ground: A to ground, B to ground and C to ground Double line to ground fault: AB to ground, AC to ground, and BC to ground Double line fault: Double line AB, Double line AC, Double line BC. Triple line fault: Triple line ABC and triple line ABC to ground fault.	11 combinations
Faulted lines	Lines A, Line B, and Line C	Multiple variations
Fault distance (in kilometres)	0 – 50 kms	Step size of 1 kilometre (50 counts)
Fault resistance	From 0 to R_{max} Ω	High impedance faults and low impedance faults
Operating mode	Islanded mode and Grid-connected mode	2 variations
Operating conditions	Different solar irradiance profiles	Multiple solar irradiance variations.
Loading conditions	Both constant and variable loads	Multiple load variations

TABLE 2: Table Showing the Simulation Conditions for No Fault Events.

Parameters	Non-Fault Events	Variations
Operating modes	Islanded mode and grid-connected mode	2
Loading conditions	Constant load and variable loads	Step size of 1 kilometre (50 counts)
Operating conditions	Multiple solar irradiance profiles	Different fault resistances
Loading conditions	Both constant and variable loads	Multiple load variations
Distributed energy resources penetration level	0 – 100%	Multiple variations of penetration levels.

3.2 Data Collection

The voltage magnitude, current magnitude, and sequence components from the 6.6 kV point of common coupling bus were collected. The data sets were collected to cover a wide range of fault conditions, such as high fault impedance, and microgrid operating conditions (varying loads and changing irradiances on the PV module). As a result, 8 input features are extracted from the fault data obtained, which are the 3 phase voltages, 3 phase currents, the zero-sequence voltage, and the zero-sequence current measurement.

3.3 Data Pre-processing and Feature Extraction

A crucial step in the automated identification of relationships and patterns in a huge dataset is the extraction of features that can be used for model building. A feature can be defined as a property that is derived from the raw data with the aim of deriving a suitable representation. As a result, feature extraction helps preserve discriminatory information and separate the factors of variation that are relevant to the learning task Goodfellow et al. [16]. The discrete wavelet transform is also used to extract the time-frequency domain characteristics from the phase voltage, phase current, and zero sequence component data and analyse the transients in the fault signals. The fault signals contain high frequency components that that are analysed for accurate representation of the fault characteristics.

Due to the presence of noise, inconsistencies and outliers, data pre-processing was done to clean the data and remove any noise because of frequencies resulting from incipient faults. Normalization and scaling were also done to ensure the data was in a consistent format and scale.

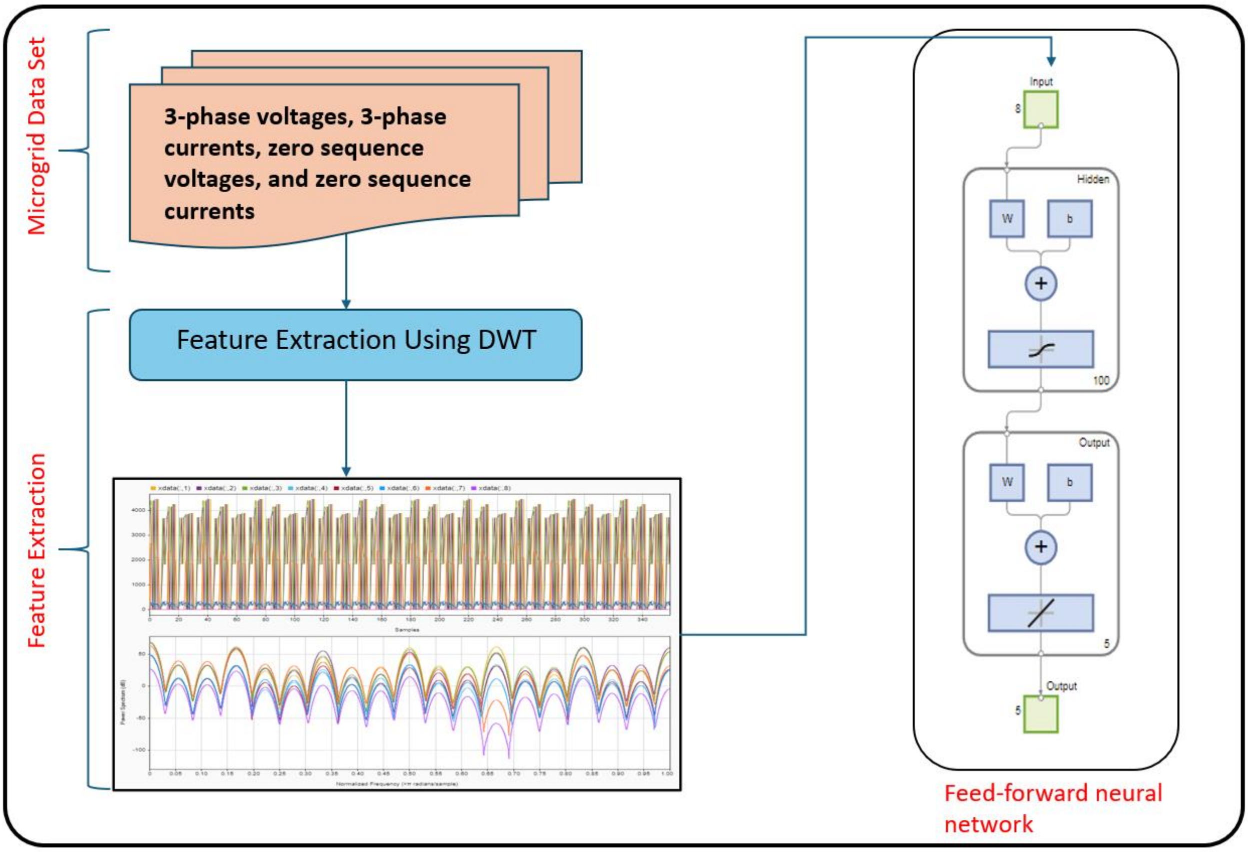


Figure 3: The General Architecture of Feature Extraction from the Fault Signals. The Extracted Features Are Then Used as the Inputs to The Training Network.

3.3.1 Discrete Wavelet Transform

A discrete signal is transformed into its discrete wavelet equivalent using the Discrete Wavelet Transform (DWT), which provides a time-scale representation of a signal in in time-frequency characteristics. The Wavelet Transform (WT) is ideal in the analysis of nonperiodic signals since it located different frequency spectrum components of a signal over time. The criteria for using the discrete wavelet are that is should be oscillatory in nature, decay quickly to zero, and have a zero average value.

In fault detection, the DWT provides a sparse representation of the transients in the fault signals. The WT of a continuous signal $x(t)$ is expressed as:

$$WT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right) dt \qquad (1)$$

Where, a is the scaling constant, b the shifting constant, and ψ is the wavelet function.

As a result, the DWT is expressed as:

$$DWT(m,k) = \frac{1}{\sqrt{a_0^m}} \sum_n x(n) g\left(\frac{k - n a_0^m}{a_0^m}\right) \qquad (2)$$

Where x(n) is described as the mother wavelet and a and b are a function of an integer parameter m; $a = a_0^m$ and $b = na_0^m$

The DWT utilizes multiresolution analysis which breaks down the original signal into the approximate and detailed (low-frequency and high-frequency) components with different resolution scales with the help of a high pass and low pass filter. Both high-frequency and low frequency signals undergo further decomposition through a multistage filter bank as shown in Figure 3.

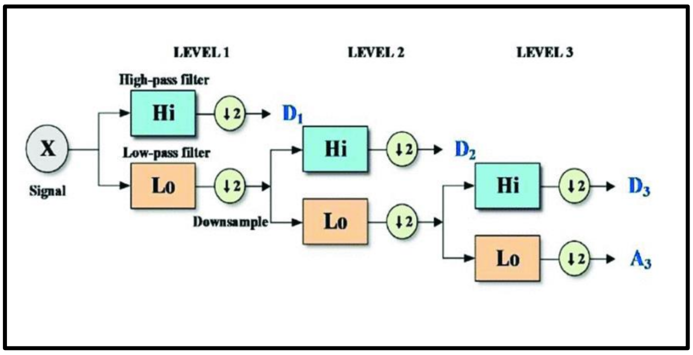


Figure 4: Signal Decomposition Tree Through a Filter bank Using the Discrete Wavelet Transform.

The extraction of features using the Discrete Wavelet Transform into the time-frequency characteristics is shown in Appendix 1.

In this work, the Discrete Wavelet Transform (DWT) is used to convert the time domain fault signals into wavelets (small waves) that have different frequency bands. The decomposition tree in Figure 4 comprises two filters at each stage, a high pass filter (Hi) and a low pass filter (Lo) to divide the signal into varying frequency bands. The Discrete Wavelet Transform of the signal at each stage separates the fault signatures into low frequency components A_3,Approximation Coefficient and high frequency components D_3, known as the Detailed Coefficient Orlando et al. [17]. The raw fault signatures obtained as shown in Appendix 1 are decomposed and divided into 8 levels of segmentation, which represent the 8 parameters that are needed to analyse the nature of the fault events. The 8 features extracted from the raw fault data are then used for machine learning training.

3.4 Machine Learning Training and Model Building

During machine learning model building, the extracted features are used by the learning algorithm to identify relationships and patterns that are relevant to the respective learning task. Each family of learning algorithms uses different mechanisms for purposes of analytical model building. In classification models, the learning algorithms learn by splitting the data records into increasingly homogenous portions Janiesch et al. [18]. In this work, the learning algorithm was trained using the recorded fault voltages, currents and zero sequence components collected from the simulation of different microgrid operating conditions. The training data is composed of eight inputs, which include the fault voltages and currents of each of the three phases and the zero sequence currents and voltages. For normal, no-fault operating conditions, the target output is set to 0 while a target output of 1 is used to allocate the fault conditions. In this case, a total of 6,000 observations are made for the 8 different fault conditions. In this work, training and model building was done using the MATLAB Neural Network Fitting and Neural Network Pattern Recognition tools. Two training algorithms were then incorporated into these tools, i.e. Bayesian Regularization and Levenberg Marquardt.

In this paper, the multilayered feedforward neural network is trained with the Bayesian Regularization algorithm to accurately detect and classify three phase faults. Network training in Bayesian Regularization algorithm uses Jacobian calculations and training continues until the network makes good generalizations or until any further training doesn't result in improvements in the network's generalization. Through Bayesian Regularization, both overtraining and overfitting are avoided since the network trains on weights or effective network parameters while ignoring any irrelevant parameters. The training objective function used by Bayesian Regularization is shown in equation 1, with S_w being the sum of squared weights and S_e the sum of the squared network errors.

$$F(w) = \alpha S_w + \beta S_e$$

During training, the combination of the squared network errors and weights is reduced until an optimal combination is achieved, that allows the neural network to achieve a good generalization. Training stops when a good generalization is achieved. A simplified machine learning script used in this work is shown in Appendix 3.

3.5 Performance Assessment

For an assessment of the machine learning model quality, different aspects are considered, including the performance, computational resources, and the interpretability. Performance-based metrics are used to make an evaluation of how the model accurately satisfies the objective specified by the learning task. Regression models can be evaluated by measuring estimation errors like the root mean square error (RMSE) or mean absolute percentage error (MAPE). To assess the performance of the training algorithms, critical aspects like the confusion matrix, performance validation plot, and the mean square error (through the regression plot) are generated and analysed to establish the relationship between the real and predicted values. To validate the suitability of machine learning prediction models, a comparison was made to models with a different training algorithm.

3.6 Comparison of Training Algorithms

To confirm and validate the effectiveness of machine learning for fault detection in microgrids, the outcomes of both training algorithms (Levenberg Marquardt and Bayesian Regularization) are compared in terms of their achieved performances. The comparison of the training algorithms is done by comparing the regression and mean square error as well as the performance validation plots of the trained models. A comparison of these performances is shown in Figure 5 and Figure 6. This comparison of both training algorithms seeks to provide comparative insights into the accuracy and effectiveness of both algorithms and provide a clear picture of the best performing model for purposes of fault detection in microgrids.

CHAPTER 4

Results and Discussion

Within the proposed machine learning framework for detecting faults in microgrids, the performance and effectiveness of the algorithms was analysed, and the developed machine learning model tested. The performance of the algorithms with respect to their training efficiency was assessed using performance validation plots, regression plots and a confusion matrix.

Performance Validation Plot

The variation of the Mean Square Error versus the number of epochs for training, validation and testing of the data is observed as shown in Figure 5. The mean Square Error values are observed to decrease gradually to indicate better achieved accuracy for the training. The gradual decrease in the MSE values is due to the continuous update of the weights after every epoch. The errors start high then continue to decrease until no further training continues, or the model stops generalizing any further.

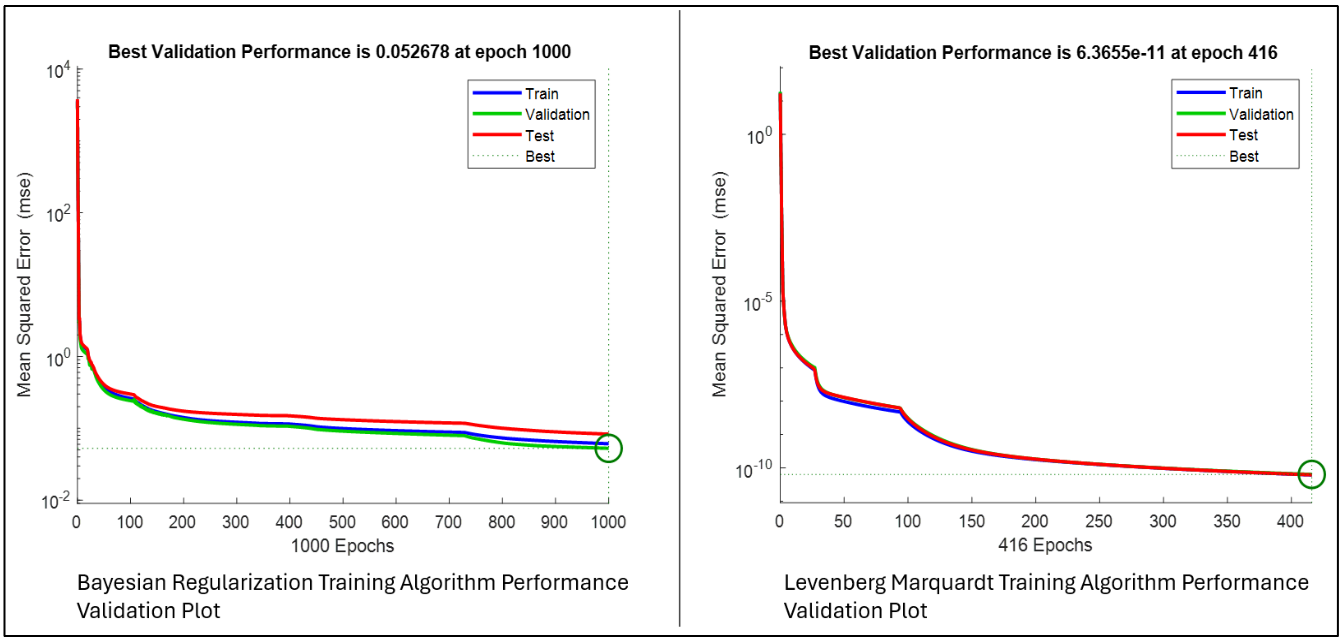
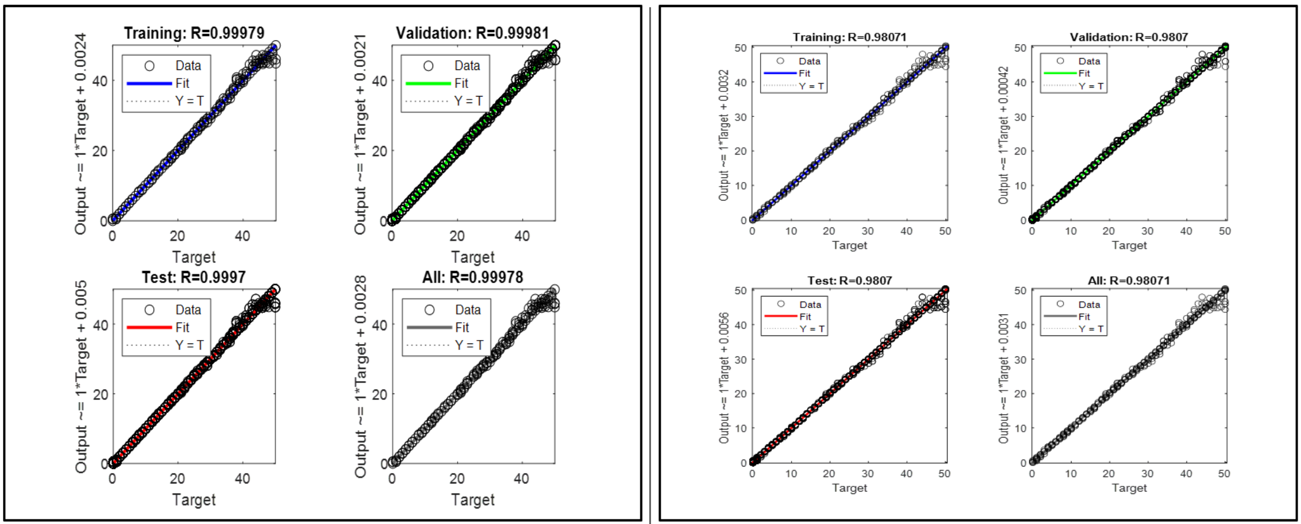


Figure 5: The Performance Validation Plots for the Bayesian Regularization and Levenberg Marquardt Algorithms. The Bayesian Regularised Algorithms Takes 1000 Epochs (Iterations) to Converge While the Levenberg Marquardt Algorithm Converged in 416 Iterations.

Regression Plot

The regression plots in machine learning provide a visual representation of the relationship between the predicted and actual values. They assess a machine learning model's performance on how the predictions align with the actual outcomes. The data follows closely along the diagonal line, which indicates a higher accuracy of the learning model. The Bayesian Regularization algorithm has a regression value of 0.99979 while Levenberg Marquardt achieves a regression of 0.98071. The Bayesian Regularised model achieves better accuracy than the Levenberg Marquardt algorithm.

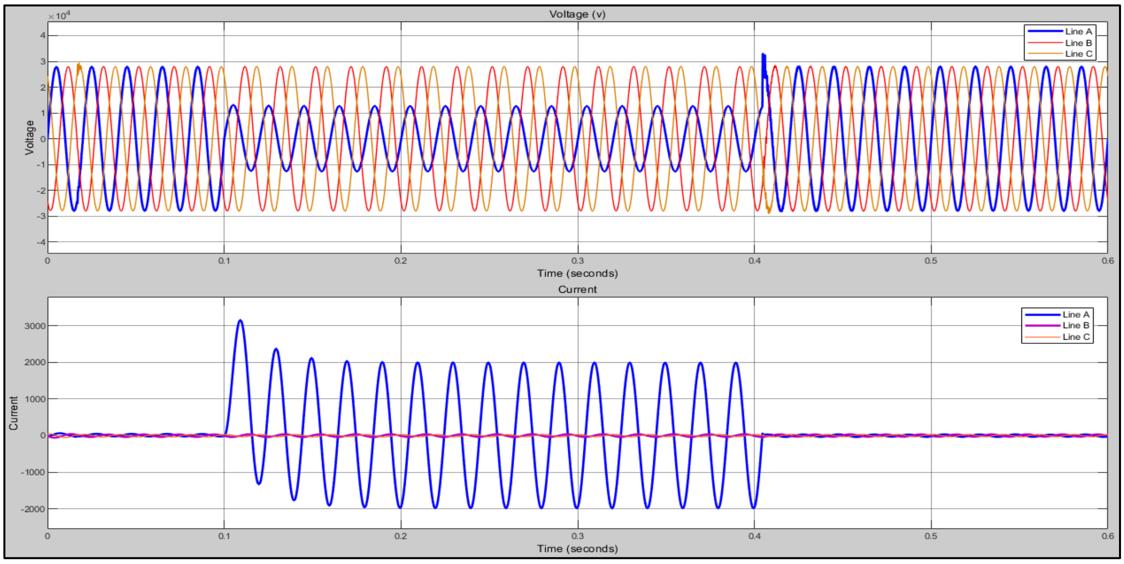


Bayesian Regularization Training Algorithm Regression Plot

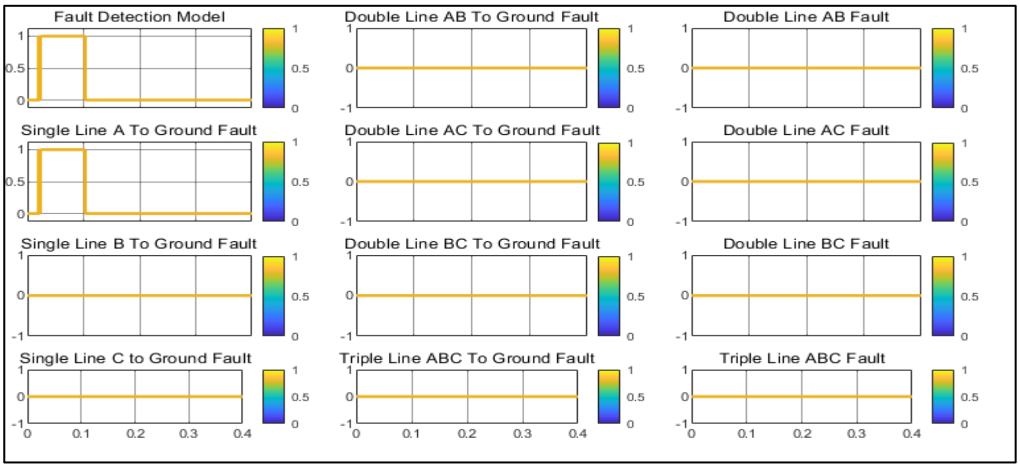
Levenberg Marquardt Training Algorithm Regression Plot

Figure 6: Regression Plots for the Bayesian Regularization Trained Algorithm and the Levenberg Marquardt Training Algorithm.

Algorithms in Action: Single Line A to Ground Fault



Bayesian Regularization Algorithm Learning Outcome



Levenberg Marquardt Algorithm Learning Outcome

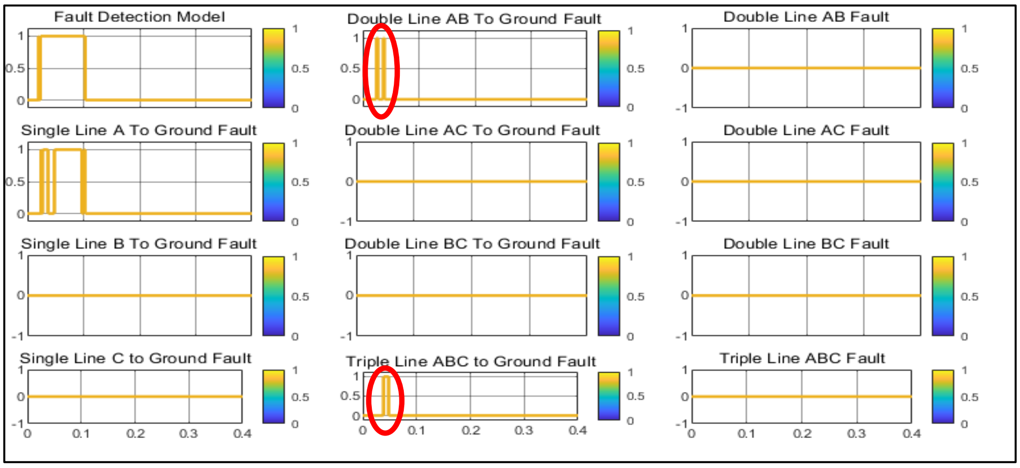


Figure 7: A Comparison of the Fault Detection and Classification Results for Bayesian Regularization and Levenberg Marquardt Algorithms. Both Algorithms Exhibit Accurate Fault Detection and Classification Abilities for Single Line A to Ground Faults but Levenberg Marquardt Algorithm-Trained Network Suffers from Overfitting and Detects Undesired Noises (Circled in Red).

For the single line A to ground fault occurring in the microgrid, the fault parameters were depicted by a high phase current and reduced phase voltage in the affected phases. With different fault events simulated, the dataset obtained comprised different fault signature patterns that were distinct to each three-phase fault. These patterns were then used to train and develop a machine learning model based on the Bayesian Regularization and Levenberg Marquardt algorithms. The machine learning model was then tested using a standard IEC microgrid for different three phase faults. From Figure 7, different machine learning model outcomes were obtained for the different training algorithms used. The machine learning model outcomes are displayed in a visual dashboard with a heatmap color-coded in blue, green, and yellow to show the extent of the fault. Blue and green are in the no-fault region (Logic 0) while yellow is the fault region (Logic 1). The x-axis of the individual plots represents time while the y-axis is encoded in Logic 0 or Logic 1 to represent the presence of a fault or no-fault. During a fault event, the fault detection model jumps from Logic 0 to Logic 1 between the timeframe of 0.02 seconds and 1 second. In the same instance, the classification model would simultaneously work to decipher the type of fault and make the classification. In this case, for the presence of a single Line A to ground fault, the fault detection model decodes the fault signature and uses Logic 1 for the period of the fault. The classification model then performs a single-handed classification to intelligently distinguish the fault from all others.

Algorithms in Action: Double Line AB Fault

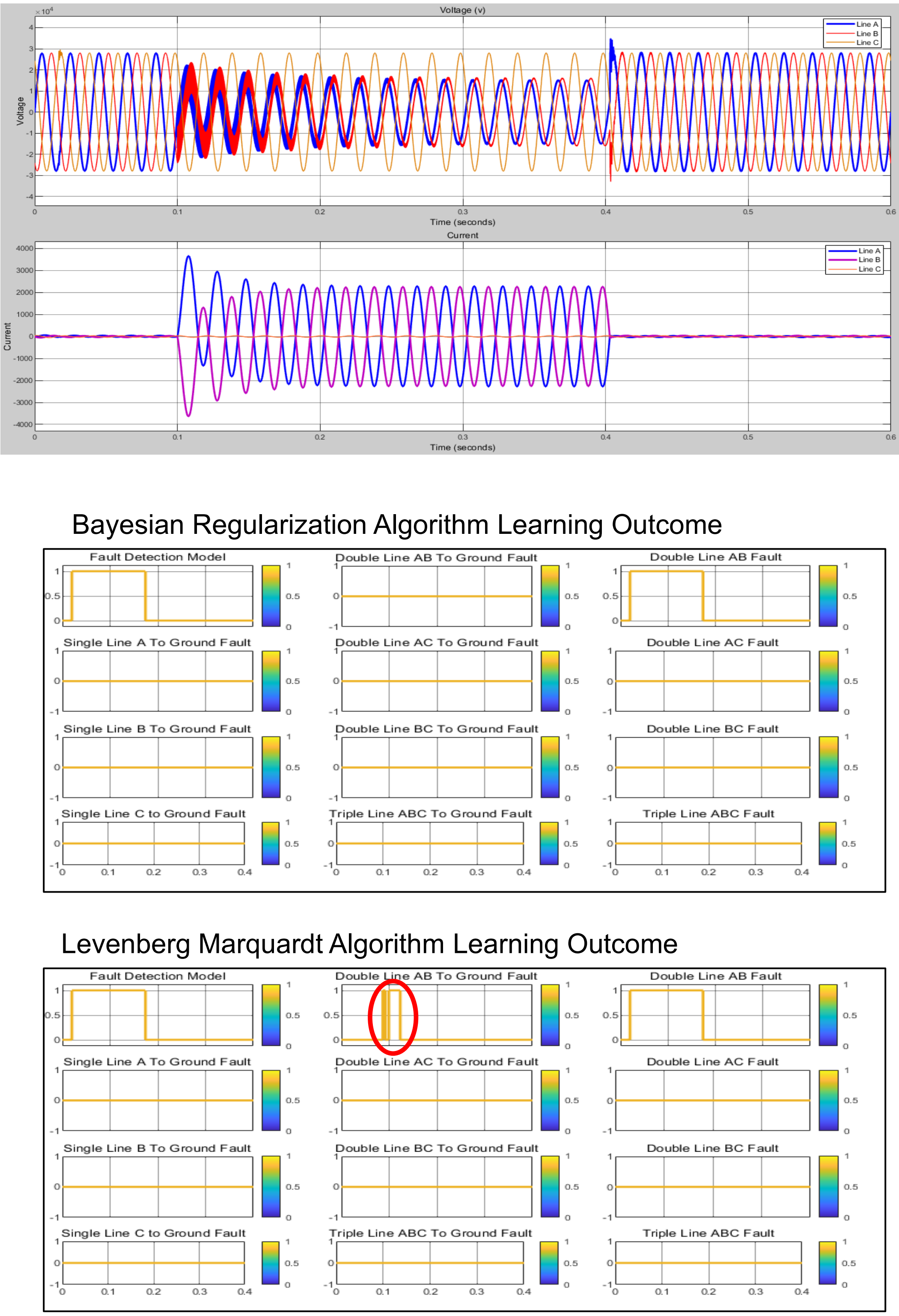
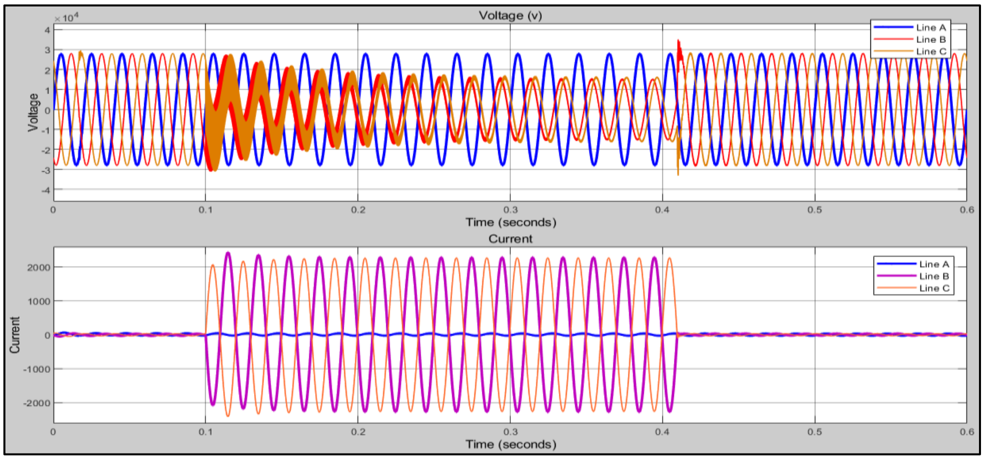


Figure 8: A Comparison of the Fault Detection and Classification Results for Bayesian Regularization and Levenberg Marquardt Algorithms. Both Algorithms Exhibit Accurate Fault Detection and Classification Abilities for Double Line AB Faults but the Levenberg Marquardt Algorithm-Trained Network Suffers from Overfitting and Detects Undesired Noises (Circled in Red).

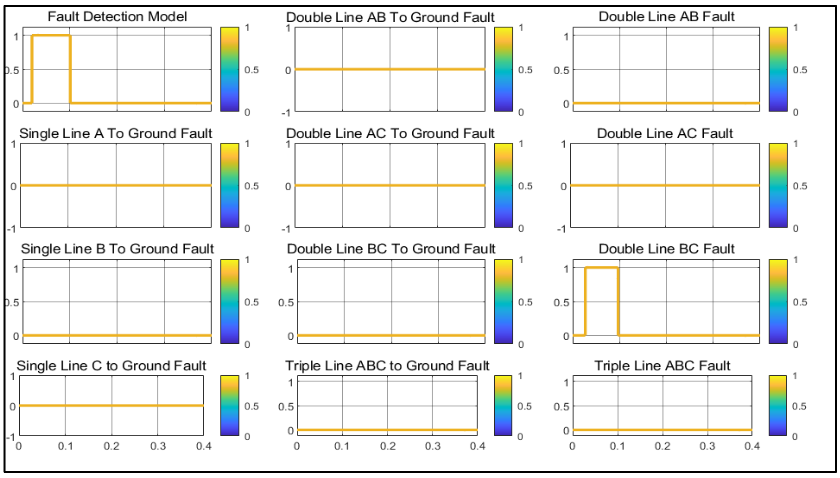
The IEC microgrid under study is then subjected to a double line AB fault and the instantaneous voltage, current and fault signatures obtained from the microgrid's point of common coupling. From the results obtained in Figure 8, the phase voltage for the two faulted phases reduced to nearly 1800 V while the fault current increases drastically for the affected phases. The output of the fault detection machine learning models accurately determines the presence of a fault condition by switching between Logic 0 to Logic 1 for the fault event period, between 0.02s and 0.1s. When a presence of a fault is detected, the classification model extracts the fault signal features from the point of common coupling to accurately classify the type of fault. From Figure 8, the outputs from both machine learning models clearly indicate and predict the presence of a fault as a double line fault by switching from Logic 0 to Logic 1 and the non-faulted phases remaining at Logic 0 all through.

A comparison of both detection and classification schemes shows a stack contrast with respect to the detection of undesired noise. Both the Bayesian Regularization and Levenberg Marquardt algorithms accurately detect and classify the double line AB fault. However, the Levenberg-trained models is observed to detect undesired noises, as circles in red. These are undesired fault signatures detected, and which could present false signals. The detection of the undesired noises in the fault signals is attributed to the nature of the Levenberg Marquardt algorithm which tends to have a limitation with accuracy when it is presented with a new event.

Algorithms in Action: Double Line BC Fault



Bayesian Regularization Algorithm Learning Outcome



Levenberg Marquardt Algorithm Learning Outcome

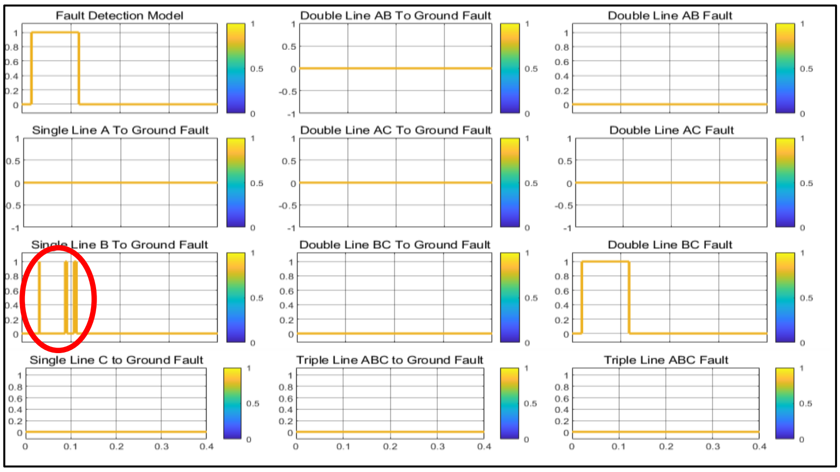
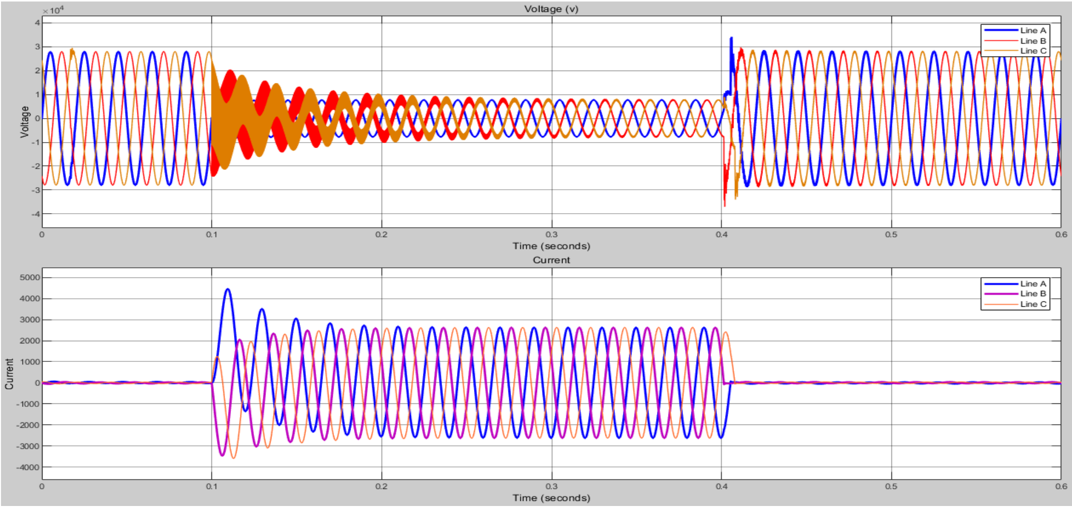


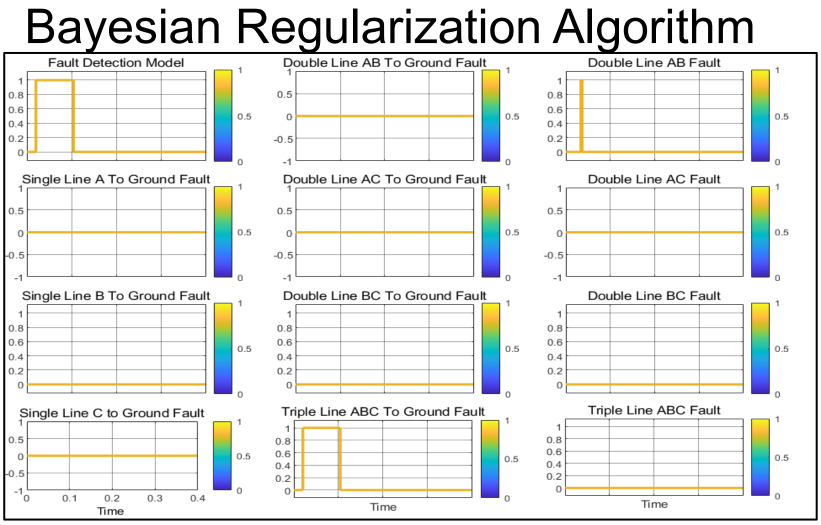
Figure 9: A Comparison of the Fault Detection and Classification Results for Bayesian Regularization and Levenberg Marquardt Algorithms. Both Algorithms Exhibit Accurate Fault Detection and Classification Abilities for Double Line BC Faults but the Levenberg Marquardt Algorithm-Trained Network Suffers from Overfitting and Detects Undesired Noises (Circled in Red).

The IEC microgrid under study was then subjected to a double line BC fault condition. The instantaneous voltage and current waveforms recorded from the point of common coupling of the microgrid show a decrease in the voltage waveform for the affected phases while the current for the affected phases increased. The machine learning model trained using the fault signal parameters and features extracted from the point of common coupling was then implemented on the test IEC microgrid to examine its ability to detect and classify the fault based on the extracted fault signatures. From Figure 9, different machine learning model outcomes were obtained for the different training algorithms used. The machine learning model outcomes are displayed in a visual dashboard with a heatmap color-coded in blue, green, and yellow to show the extent of the fault. Blue and green are in the no-fault region (Logic 0) while yellow is the fault region (Logic 1). The x-axis of the individual plots represents time while the y-axis is encoded in Logic 0 or Logic 1 to represent the presence of a fault or no-fault. During the double line BC fault event, the fault detection model accurately identifies the presence of a fault by switching from Logic 0 to Logic 1 between the timeframe of 0.02 seconds and 1 second. In the same instance, the classification model simultaneously works to decipher the type of fault and make the classification. In this case, for the presence of a double Line BC fault, the fault detection model decodes the fault signature and uses Logic 1 for the period of the fault. The classification model then performs a single-handed classification to intelligently distinguish the nature of the three-phase fault from all others.

A comparison of both detection and classification schemes shows a stack contrast with respect to the accuracy of their fault classification capabilities. Both the Bayesian Regularization and Levenberg Marquardt algorithms accurately detect and classify the double line BC fault. However, the Levenberg-trained models is observed to detect undesired noises, as circles in red (for the fault single Line B to ground). These are undesired fault signatures detected, and which could present false signals. The detection of the undesired noises in the fault signals is attributed to the nature of the Levenberg Marquardt algorithm which tends to have a limitation with accuracy when it is presented with a new event. This characteristic of the machine learning algorithm is known as overfitting, where a machine learning algorithm learns properly but fails to make accurate predictions when presented with a new data set.

Algorithms in Action: Triple Line to Ground Fault





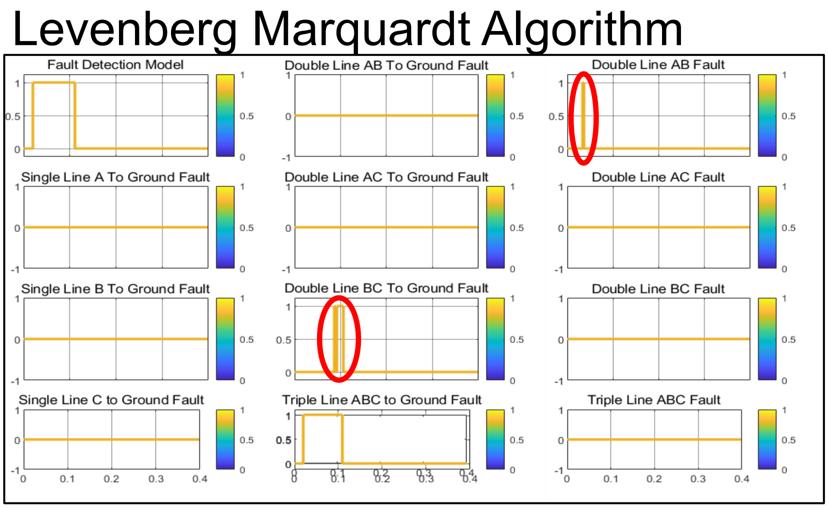


Figure 10: A Comparison of the Fault Detection and Classification Results for Bayesian Regularization and Levenberg Marquardt Algorithms. Both Algorithms Exhibit Accurate Fault Detection and Classification Abilities for Triple Line To Ground Faults but the Levenberg Marquardt Algorithm-Trained Network Suffers from Overfitting and Detects Undesired Noises (Circled in Red).

To further confirm and validate the accuracy of machine learning in the detection and classification of three-phase faults, the IEC microgrid under study was then subjected to a triple line ABC to ground fault condition. The instantaneous voltage and current waveforms recorded from the point of common coupling of the microgrid show a decrease in the voltage waveform for the affected phases while the current for the affected phases increased. The machine learning model trained using the fault signal parameters and features extracted from the point of common coupling was then implemented on the test IEC microgrid to examine its ability to detect and classify the fault based on the extracted fault signatures.

From Figure 10, different machine learning model outcomes were obtained for the different training algorithms used. The machine learning model outcomes are displayed in a visual dashboard with a heatmap color-coded in blue, green, and yellow to show the extent of the fault. Blue and green are in the no-fault region (Logic 0) while yellow is the fault region (Logic 1). The x-axis of the individual plots represents time while the y-axis is encoded in Logic 0 or Logic 1 to represent the presence of a fault or no-fault. During the triple line ABC to ground fault event, the fault detection model accurately identifies the presence of a fault by switching from Logic 0 to Logic 1 between the timeframe of 0.02 seconds and 1 second. In the same instance, the classification model simultaneously works to decipher the type of fault and make the classification. In this case, for the presence of a triple Line ABC to ground fault, the fault detection model decodes the fault signature and uses Logic 1 for the period of the fault. The classification model then performs a single-handed classification to intelligently distinguish the nature of the three-phase fault from all others.

A comparison of both detection and classification schemes shows a stack contrast with respect to the accuracy of their fault classification capabilities. Both the Bayesian Regularization and Levenberg Marquardt algorithms accurately detect and classify the triple line ABC to ground fault. However, the Levenberg-trained models is observed to detect undesired noises, as circles in red (for the fault double Line BC to ground). These are undesired fault signatures detected, and which could present false signals. The detection of the undesired noises in the fault signals is attributed to the nature of the Levenberg Marquardt algorithm which tends to have a limitation with accuracy when it is presented with a new event. This characteristic of the machine learning algorithm is known as overfitting, where a machine learning algorithm learns properly but fails to make accurate predictions when presented with a new data set.

Key Findings

Comparative analysis of proposed machine learning algorithms for fault detection and classification.

From the results shown in Figure 7, Figure 8, Figure 9 and Figure 10, the accurate prediction and classification abilities of the Bayesian Regularization and Levenberg Marquardt algorithms are compared. Different types of three phase faults were introduced and the two trained learning models were used to examine how accurately they would detect and classify the fault. As observed, both the Bayesian Regularization and Levenberg Marquardt training algorithms accurately detected the single line to ground fault. The presence of the fault is confirmed by the fault detection line moving from Logic 0 to Logic 1 for the period of the fault (from 0.02 seconds to 0.1 seconds). Both algorithms also accurately classify the three-phase fault as a single line to ground fault as displayed on the output windows.

It was also observed that some undesired spikes and noises were detected in other fault detection display outputs, as highlighted in red. The undesired outliers were witnessed in the detection and classification scheme trained using the Levenberg Marquardt algorithm. These undesired fault detection samples indicated the algorithms susceptibility to overfitting and false positives when tested using new variables.

Bias and Drift in Data

In analytical model building, the awareness of the presence of cognitive biases that were introduced into the machine learning models is also considered. These are biases that the model may heavily adopt to an extent that the models exhibit the same induced tendencies present in the data or even amplify them Fuchs, D [19]. Cognitive biases are illogical beliefs or inferences that people may adopt as a result of flawed reporting of facts or flawed decision heuristics Howard et al. [20] In this work, the presence of biases in the results was witnessed in the machine learning models that were trained using the Levenberg Algorithm. These biases presented themselves in the form of undesired noises detected by the learning model.

Overfitting

Overfitting is a common issue in machine learning, where the machine learning model learns too complex and fails to fit the training data well. This often results in a poor performance on new, unseen data. In this work, the machine learning model trained using the Levenberg Marquardt algorithm suffered from overfitting. This was evident due to the model’s ability to learn to recognize outliers or noise in the training dataset as important analytical features whereas they were not. This is an undesired characteristic in machine learning that could result in false positives.

CHAPTER 5

Conclusion and Future Work

Conclusion

The objective of this work was to develop a machine-learning based fault detection and classification scheme for microgrids. Three-phase fault scenarios were obtained under different microgrid operating conditions, and the discrete wavelet transform was used to extract the time-frequency characteristics of the raw data. A total of 8 distinct features were extracted from the data set and used as the input variables for the purpose of machine learning. The Bayesian Regularization algorithm was used for model training and its performance assessed based on the regression value and the mean square error. The performance, fault detection accuracy, and prediction capabilities of the Bayesian Regularization algorithm were compared to the Levenberg Marquardt training algorithm for the same input variables and hidden layers.

The detection and predictive performance of both training algorithms was analysed, and the Bayesian Regularization training algorithm was found to provide better accuracy and performance. The error rate displayed by the Bayesian Regularization algorithm was low and the algorithm was found to make accurate predictions and better generalizations even with the introduction of new distinct variables. From the comparative analysis, it can be concluded that the Bayesian Regularization training algorithm provides better machine learning performance for fault detection and classification in microgrids and is a suitable choice where high accuracy is needed. However, the training algorithm is time-consuming and may not be appropriate for use where training speed is a huge constraint. The Levenberg Marquardt algorithm would be a better choice for time-constrained training situations.

The findings of this work demonstrated the huge potential for the use of machine learning in microgrids for monitoring and predicting the presence of faults. This opens up the opportunity for the development of automated monitoring systems and decision support tools that can help engineers in predicting and detecting fault events in microgrid systems. Two machine learning algorithms have been used to assess the effectiveness of their predictive capabilities. Evaluating their performance, it was discovered that both algorithms used in this study achieved performance assessment values that were above the optimal threshold. Notably, the Bayesian Regularization algorithm emerged as the most accurate model for the detection and classification of faults.

The practical significance of the study conducted in this paper lies in its ability to aid in the early detection of faults in microgrids, provide a more targeted prevention strategy, and guide more precise interventions. These implications have a potential to immensely enhance the fault diagnosis outcomes for microgrids by effectively addressing the challenge of accurate detection due to the dynamic operating nature of microgrids. As a result, this study introduces new perspectives to the field of fault detection in microgrids through its hybrid innovative approach, incorporation of a wide range of parameters, the development of accurate prediction models, and proposal for individualized implementation.

By incorporating different machine learning algorithms, this study provides a comprehensive and robust investigation of the analytical and predictive capabilities of machine learning to reveal hidden patterns and relationships in data that may not be easily identified through conventional statistical methods.

Future Works

The application of machine learning techniques and models in the detection and classification of faults in microgrids paves the way for the adoption of data-driven methods in decentralized energy systems. While this work leverages the implementation of machine learning models, some key areas not addressed by this work could provide a direction for future work. There areas are as follows:

Data management

The adoption of data driven methods and techniques presents the challenge of efficiently managing transmission of operation and fault data from smart and intelligent devices. While existing techniques provide approaches for maximizing network traffic, more research needs to be done on how rapid configuration modifications can impact the integrity and accuracy of the data Gorban and Andrei [21].

In addition, microgrids of the future will depend on robust communication network for efficient operation, optimal use, and protection. Due to the extensive use of communication networks, there is a need to perform more research into the effect of communication latencies and noise on overall microgrid protection.

Transfer Learning

Training machine learning models for fault detection and classification in power systems relies on enormous simulation data. The generation of such data points may be difficult to achieve with real-world power systems. As a result, machine learning training based on simulation data may lead to a deterioration in accuracy of detection. As a result, transfer learning could be explored as a possible solution to this problem. Transfer learning is an improved way of learning a new task through knowledge transfer from a related task that has already occurred. Transfer learning is a two-step approach; the first is training a model based on a source task and data set followed by the transfer of the learned knowledge and features to a new model to help with training for a new data set Fatemeh et al. [22]. Transfer learning has been found to enhance the learning speed of models and reduce the amount of training data needed while providing notable improvements in accuracy.

IoT and Edge Computing Integration

The continued proliferation of Internet of Things (IoT) devices and an adoption of edge computing provides more opportunities for enhancement of fault detection systems for microgrids. IoT can help process data closer to its source, hence reducing problems with latency Kumar & Singh [23]. Besides, IoT devices can provide a rich and comprehensive data set for the training and validation of machine learning models.

REFERENCES

[1] ARENA (2020) *Fringe of grid mines embrace renewables*, *Australian Renewable Energy Agency*. Available at: https://arena.gov.au/blog/hybrid-renewables-mines/ (Accessed: 08 May 2024).

[2] *Building a clean energy future – budget*. *Budget.gov.au*. Available at: https://budget.gov.au/content/factsheets/download/factsheet_clean_energy-20230510.pdf (Accessed: 08 May 2024).

[3] *Executive summary: Market data: Energy as a service microgrids* (no date) *Guidehouse Insights*. Available at: https://guidehouseinsights.com/-/media/project/navigant-research/navigant-research-executive-summaries/2021/2q-2021/market-data-energy-as-a-service-microgrids-executive-summary.pdf (Accessed: 08 May 2024).

[4] Dehghanian, P., Wang, B. and Tasdighi, M. (2019) 'New protection schemes in smarter power grids with higher penetration of renewable energy systems', Pathways to a Smarter Power System, pp. 317–342. doi:10.1016/b978-0-08-102592-5.00011-9.

[5] Maheswari, K.L., Sathya, B. and Jeylani, A.M. (2021) 'Mitigating measures to address challenges of renewable integration—forecasting, scheduling, dispatch, balancing, monitoring, and control', Integration of Renewable Energy Sources with Smart Grid, pp. 281–304. doi:10.1002/9781119751908.ch13.

[6] Nichols, J.A., Herbert Chan, H.W. and Baker, M.A. (2018) 'Machine learning: Applications of artificial intelligence to imaging and diagnosis', *Biophysical Reviews*, 11(1), pp. 111–118. doi:10.1007/s12551-018-0449-9.

[7] Han, J. (2023) *Data Mining Concepts and Techniques*. Cambridge, MA, United States: Elsevier.

[8] Sarker, I.H. *et al.* (2020) 'Cybersecurity Data Science: An Overview From Machine Learning Perspective', *Journal of Big Data*, 7(1) pp.1-29. doi:10.1186/s40537-020-00318-5.

[9] Valsdóttir, V. *et al.* (2023) 'Comparative study of machine learning methods for modeling associations between risk factors and future dementia cases', *GeroScience*, 46(1), pp. 737–750. doi:10.1007/s11357-023-01040-9.

[10] Deb, A. and Jain, A.K. (2024) ‘An effective data-driven machine learning hybrid approach for fault detection and classification in a standalone low-voltage DC microgrid’, *Electrical Engineering* [Preprint]. doi:10.1007/s00202-024-02334-7.

[11] Ahmadi, A., Aghajari, E. and Zangeneh, M. (2022) ‘High-impedance fault detection in power distribution grid systems based on support vector machine approach’, *Electrical Engineering*, 104(5), pp. 3659–3672. doi:10.1007/s00202-022-01544-1.

[12] Ananth, A., Mohan Babu, P. and Vaithilingam, C. (2022) ‘Estimation of dynamic fault currents of microgrids using generalized regression neural network’, Lecture Notes in Networks and Systems, pp. 139–146. doi:10.1007/978-981-16-9756-2_13.

[13] Kanojia, S.S. and Shah, A. (2024) ‘Fault detection at PCC using wavelet theory in grid-tied solar PV battery-based AC Microgrid’, Lecture Notes in Electrical Engineering, pp. 253–271. doi:10.1007/978-981-99-8289-9_19.

[14] Han, H. et al. (2022) ‘Fault diagnosis method of islanded DC Microgrid based on Wavelet Sliding Window Energy’, Lecture Notes in Electrical Engineering, pp. 825–834. doi:10.1007/978-981-19-1532-1_88.

[15] Sarangi, S., Sahu, B.K. and Rout, P.K. (2022) ‘An advanced fault detection technique for DG integrated microgrid using fast fourier discrete orthonormal Stockwell transform-based hybrid optimized kernel extreme learning machine’, Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 46(2), pp. 329–351. doi:10.1007/s40998-022-00481-w.

[16] Sahoo, A.K. and Samal, S.K. (2022) ‘Online fault detection and classification of 3-phase long transmission line using machine learning model’, Multiscale and Multidisciplinary Modeling, Experiments and Design, 6(1), pp. 135–146. doi:10.1007/s41939-022-00132-x.

[17] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. The MIT Press. (Goodfellow)

[18] Orlando Oliveira M, Horacio Reversat J and Alberto Reynoso L (2019) Wavelet Transform Analysis to Applications in Electric Power Systems. Wavelet Transform and Complexity. IntechOpen. Available at: http://dx.doi.org/10.5772/intechopen.85274.

[19] Janiesch, C., Zschech, P. and Heinrich, K. (2021) ‘Machine learning and deep learning’, Electronic Markets, 31(3), pp. 685–695. doi:10.1007/s12525-021-00475-2.

[20] Fuchs, D. J. (2018). The dangers of human-like Bias in machine-learning algorithms. *Missouri S&T's Peer to Peer, 2*(1), 15.

[21] Howard, A., Zhang, C., & Horvitz, E. (2017). Addressing bias in machine learning algorithms: A pilot study on emotion recognition for intelligent systems. *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 1–7. https://doi.org/10.1109/ARSO.2017.8025197.

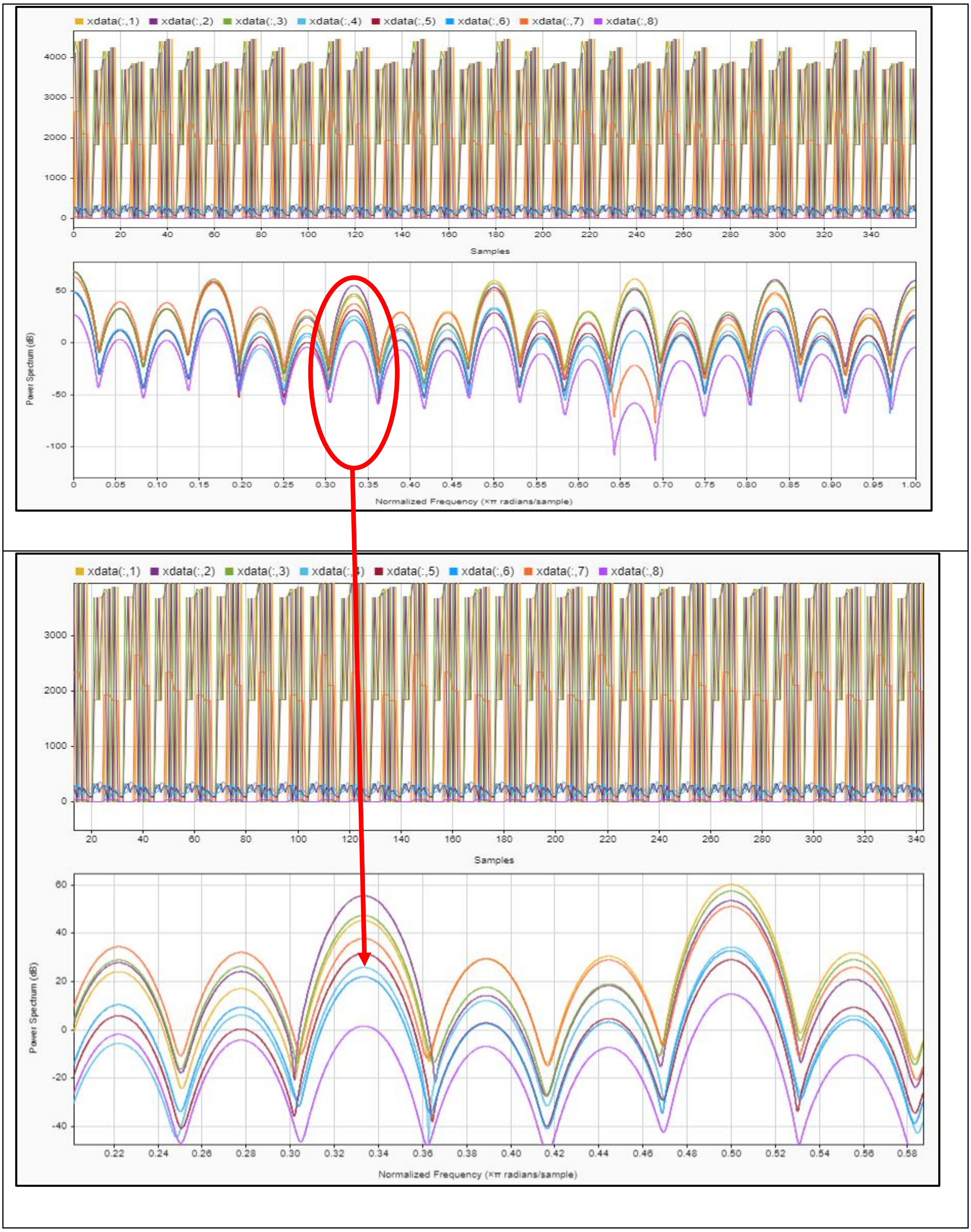
[22] Gorban, Alexander N and Andrei Yu. Zinovyev. "Handbook of Research on Machine Learning Applications and Trends : Algorithms , Methods , and Techniques." (2009).

[23] Shakiba, Fatemeh Mohammadi et al. "Transfer Learning for Fault Diagnosis of Transmission Lines." *ArXiv* abs/2201.08018 (2022): n. pag.

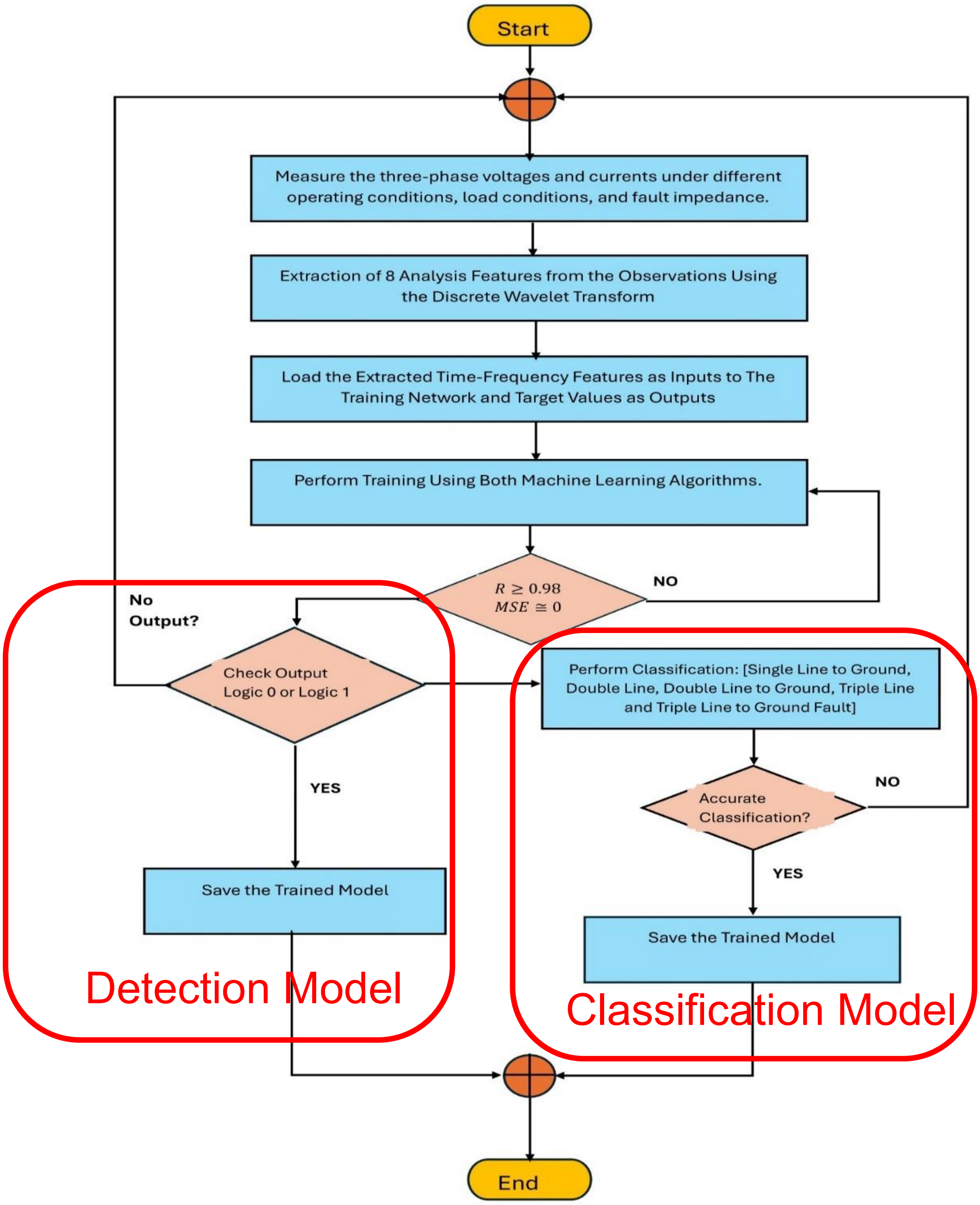
[24] Kumar, A. and Singh, D. (2023) 'Detection and prevention of ddos attacks on edge computing of IOT devices through reinforcement learning', International Journal of Information Technology, 16(3), pp. 1365–1376. doi:10.1007/s41870-023-01508-z.

APPENDICES

APPENDIX 1: FEATURE EXTRACTION IN ACTION FROM MICROGRID FAULT SIGNAL SIGNATURES.



APPENDIX 2: SIMPLIFIED MACHINE LEARNING PROCESS



APPENDIX 3: MACHINE LEARNING TRAINING SCRIPT

```
% This script assumes these variables are defined:
%
%   xdata - input data.
%   tdata - target data.

x = xdata';
t = tdata';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 100;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

```
% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse';  % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```