QATAR UNIVERSITY

COLLEGE OF ENGINEERING

ENHANCING FAULT DETECTION AND LOCALIZATION IN TRANSMISSION LINES USING ARTIFICIAL NEURAL NETWORKS AND LEARNING ALGORITHMS: A COMPREHENSIVE ANALYSIS

BY

FADEL QRAITA T S AL ANZI

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Electrical Engineering

January 2025

COMMITTEE PAGE

The members of the Committee approve the Thesis of
FADEL AL-ANZI defended on 01/06/2024.

S M Muyeen
Thesis/Dissertation Supervisor

Atif Iqbal
Committee Member

Maher Abdelkhalek Azzouz
Committee Member

Balanthi Beig
Committee Member

Approved:

Dean, College of Engineering

ABSTRACT

AL-ANZI, FADEL, QRAITA., Masters: January : 2025,
Masters of Science in Electrical Engineering

Title: Enhancing Fault Detection and Localization in Transmission Lines Using Artificial Neural Networks and Learning Algorithms: A Comprehensive Analysis

Supervisor of Thesis: Advisor’s S. M. Muyeen.

Accurately detecting and pinpointing faults in power cable circuits presents a substantial challenge in ensuring the reliability and efficiency of electrical systems. The growing expansion of underground cable networks, along with the increasing complexity of modern power grids, has heightened the challenge of accurately locating faults. This challenge is further compounded by the diverse range of fault types encountered, including short circuits, insulation breakdowns, and intermittent faults, which add complexity to the localization process. This research explores the effectiveness of artificial neural networks (ANNs) in fault detection, classification, and localization. A total of 16 examples were studied by examining different power system topologies, using multiple backpropagations learning algorithms (e.g., LM (Levenberg-Marquardt) and SCG (Scaled Conjugate Gradient), and utilizing data from both endpoints of a transmission line. The findings show that using the LM learning algorithm and combining input from both terminals improves fault detection and localization in the majority of instances. Although the SCG algorithm exhibits faster convergence in fault location, both algorithms yield comparable results in fault identification and classification only. Integrating data from the remote end as additional inputs to ANNs enhances the fault localization process by offering a more comprehensive dataset for analysis.

DEDICATION

I dedicate this thesis to my beloved family, whose steadfast love, support, and encouragement have been the foundation of my academic journey.

ACKNOWLEDGMENTS

I wish to convey my profound gratitude to my supervisors, Prof. Muyeen & Dr. Hassan, for their invaluable guidance, support, and encouragement throughout this research journey. Their expertise, patience, and unwavering dedication have played a crucial role in shaping this thesis and fostering my development as a researcher. I also extend my sincere thanks to the staff and faculty of Qatar University for providing the resources, facilities, and academic environment that enabled the successful completion of this work. I am deeply appreciative of my colleagues and friends for their camaraderie, encouragement, and unwavering support during challenging times, as well as their enthusiasm in discussing the ideas and concepts that form the foundation of this thesis. Their collective contributions have been immensely valuable, and I am truly thankful for their support throughout this entire endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

The electric power system is inherently complex, consisting of numerous interconnected components including generation, transmission, protection, and distribution. Each stage of this intricate network plays a crucial role in maintaining a reliable supply of electricity.

Within this framework, the importance of a robust protection system cannot be overstated. With the presence of costly equipment susceptible to damage during faults induced by high currents, the need for a swift and precise protection mechanism becomes paramount. Such faults can result in significant expenses and downtime if left unchecked. Therefore, an efficient protection system is indispensable in safeguarding the integrity and functionality of power system infrastructure [1].

In essence, the protection system assumes a pivotal role within the broader context of the power system, serving as a critical safeguard against potential disruptions and ensuring the sustained operation of essential equipment.

Transmission lines are vital for transporting electrical power from generation sources to end consumers, using either overhead lines or underground cables to facilitate this process. Electric power transmission via underground cables offers numerous advantages over overhead lines, prompting its preference in many cases. These advantages include:

- Underground cables significantly reduce visual impact compared to overhead lines as they are concealed beneath the ground and not visible.
- They can be easily routed in urban areas since they do not require the safety clearance space needed by overhead lines to mitigate the effects of magnetic fields.
- Underground cables offer greater stability during extreme weather conditions like rain and storms, which often disrupt overhead lines and cause power outages.

- They incur lower maintenance costs, as they are less susceptible to damage from external factors such as falling trees and tall vehicles.

Nevertheless, both underground cables and overhead lines are susceptible to different types of faults within the power system [1].

In power systems, faults are generally categorized into two main types: symmetrical and unsymmetrical faults. Symmetrical faults consist of a single category, specifically the three-phase-to-ground (3LG) fault. In contrast, unsymmetrical faults include three different types: single-line-to-ground (1LG), line-to-line (LL), and double-line-to-ground (2LG) faults. Among these, single-line-to-ground faults are the most prevalent, while three-phase faults, although less frequent, tend to be more severe. To safeguard transmission underground cables from various faults, it is essential to establish a rapid and accurate protection system that can effectively identify fault locations [2].

Upon detection and isolation of a fault on an underground cable by the protection system, the subsequent stage involves determining the fault's location. Fault locator techniques serve as pivotal tools within underground transmission systems, facilitating the precise identification of fault locations along the entire cable length. The utilization of fault locators offers numerous advantages, notably reducing the costs associated with extensive excavation and investigation to pinpoint the fault location. Additionally, fault locators contribute to minimizing outage durations, thereby expediting cable restoration and averting potential single feed source situations for loads, consequently mitigating load losses within the electricity network [3], [4].

As seen in Figure 1, a variety of approaches can be used to precisely localize the fault; the most popular ones are impedance-based, knowledge-based , traveling wave, and hybrid techniques. Impedance-based methods, which encompass both oneended and two-ended approaches, involve calculating line impedance by measuring voltage and current. The resulting impedance magnitude is then used to pinpoint the fault location, considering factors such as line length and impedance. Conversely, traveling wave methods depend on the timing of wave reception at both ends of the line to ascertain the fault location, using the characteristics of voltage or current waves. Knowledge-based methods can be further categorized into artificial intelligence, distributed devices, and hybrid methods. Among these, artificial intelligence-based approaches are generally considered the most accurate. Hybrid methods employ recorded post-fault data, including voltage and current information, to identify the location of faults. Comparative analysis of these methods reveals that knowledge-based techniques employing artificial intelligence offer the highest accuracy in fault localization [5].

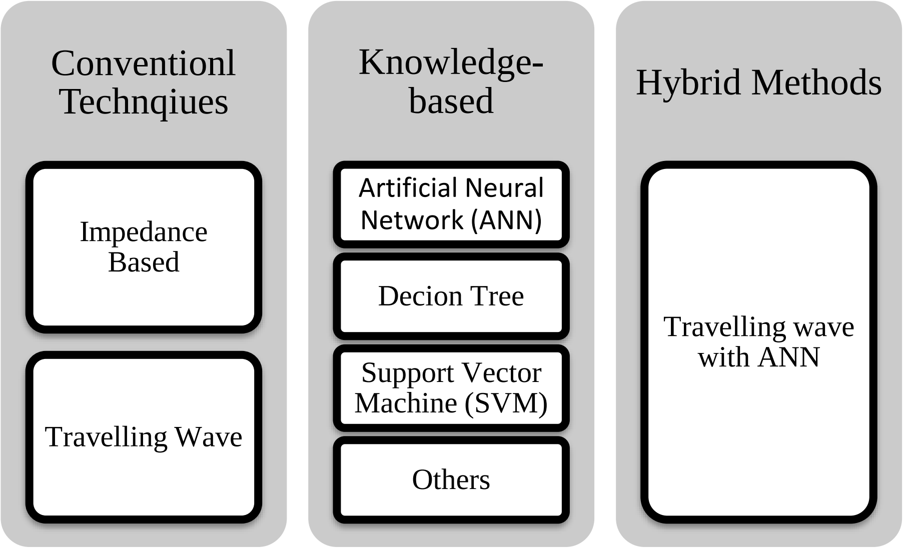


Figure 1. Summary of fault localization techniques.

The power grid serves as the backbone of various critical functions, including communication networks, modern transportation systems, and the provision of drinking water, all of which significantly contribute to enhancing people's quality of life. However, the landscape of power grid operation and control has become increasingly uncertain, owing to the substantial increase in renewable energy adoption and the implementation of demand response measures. Consequently, power system operation and planning have evolved into highly complex endeavors. To tackle these challenges, extensive efforts have been directed towards bolstering resilience and devising more precise solutions by leveraging artificial intelligence (AI) in conjunction with real-time data obtained from electronic devices. This collective endeavor has resulted in an in-depth investigation of machine learning techniques in the operation and planning of power systems, concentrating on several critical areas [6]:

1- Grid Management: AI-driven optimization enhances power grid operation and control, improving efficiency and reliability through demand prediction, fluctuation management, and issue identification.

2- Predictive Maintenance: AI algorithms analyze sensor data to forecast equipment failures, facilitating proactive maintenance and minimizing downtime in power generation and distribution.

3- Energy Forecasting: Artificial intelligence enhances the precision of forecasting energy consumption trends, facilitating improved resource allocation, energy trading, and the optimal use of renewable energy sources.

4- Smart Grids: Artificial intelligence facilitates real-time monitoring, control, and automation within smart grid technologies, promoting efficient energy distribution, demand response, and the seamless integration of renewable energy sources.

5- Energy Storage Optimization: AI algorithms optimize energy storage system cycles, enhancing efficiency and contributing to grid stability.

6- Cybersecurity: AI enhances power system cybersecurity by detecting anomalies, identifying threats, and responding to cyber-attacks in real-time.

7- Demand Response: AI analyzes historical consumption patterns to predict and manage peak demand, allowing for efficient resource allocation and reduced strain on the power grid during peak periods.

8- Renewable Energy Integration: Artificial intelligence addresses the variability of renewable energy sources by forecasting generation patterns and enhancing their integration into the power grid.

9- Optimal Power Flow: AI algorithms optimize power distribution in networks, considering factors such as generation capacity, transmission constraints, and demand to maximize efficiency.

10- Fault Detection and Diagnostics: AI rapidly identifies faults in the power system, enabling operators to isolate and address issues promptly, thereby reducing outage durations and enhancing system reliability.

These applications underscore the crucial role of artificial intelligence in rendering power systems more resilient, efficient, and capable of navigating the challenges presented by evolving energy landscapes.

The electric power system is a complex network requiring reliable protection to safeguard against faults and ensure uninterrupted electricity supply. Transmission lines, including overhead and underground cables, play a critical role in transporting power, with underground cables offering advantages such as reduced visual impact, urban adaptability, and resilience to weather. However, both types are prone to faults, which are classified as symmetrical or unsymmetrical, with single-line-to-ground faults being the most common. Rapid fault detection and precise localization are vital to minimize downtime and repair costs, leveraging techniques like impedance-based, traveling wave, and knowledge-based methods, with artificial intelligence (AI) proving the most accurate. AI is also transforming power systems by optimizing grid management, predictive maintenance, energy forecasting, smart grids, energy storage, cybersecurity, and renewable energy integration. These advancements enhance power system resilience, efficiency, and adaptability, addressing challenges such as renewable energy variability and evolving grid demands.

1.2 OBJECTIVES

Artificial Neural Networks (ANNs) offer several advantages over traditional fault localization methods. Unlike conventional techniques, ANNs can adaptively learn and generalize from input-output relationships, making them more robust in handling complex and nonlinear systems. Additionally, ANNs can process large volumes of data simultaneously, allowing for efficient analysis of diverse fault scenarios. Moreover, ANNs have the capability to self-organize and self-adapt, enabling them to adapt to changing system conditions without explicit reprogramming.

Moreover, Artificial Neural Networks (ANNs) offer several advantages over other artificial intelligence methods when applied to fault localization problems in power cable circuits. ANNs excel at learning complex patterns and relationships from data. In fault localization, power cable circuits exhibit nonlinear behavior, and faults can manifest in various forms (such as short circuits, insulation breakdowns, and intermittent faults). ANNs can effectively capture these intricate patterns and relationships, enabling accurate fault identification and localization. Overall, the inherent capabilities of ANNs in learning complex patterns, adapting to changing environments, and processing multimodal data make them well-suited for fault localization tasks in power cable circuits, offering advantages over other artificial intelligence methods in terms of efficiency, accuracy, and sturdiness.

The aims of employing artificial neural networks (ANNs) for fault localization are as follows:

1. Fault Identification: Develop ANN models capable of accurately identifying different fault types, such as short circuits, insulation

breakdowns, and intermittent faults, based on input data collected from power cable circuits.

2. Fault Classification: Train ANNs to classify identified faults into distinct categories, enabling rapid categorization and prioritization of fault responses.
3. Fault Localization: Deploy artificial neural network (ANN) algorithms to accurately pinpoint faults in power cable circuits, identifying the precise location of the fault along the cable's length.
4. Testing ANNs Across Multiple Power System Models: Through systematic evaluation of artificial neural networks (ANNs) across various power system configurations, this study aims to offer in-depth insights into the effectiveness and applicability of ANN-based methods for fault identification, classification, and localization in a range of real-world scenarios. The results of this research will aid in the development of more robust and reliable fault detection systems, ultimately enhancing the resilience and efficiency of electrical distribution networks.
5. Integration of Remote End Data: Explore methods to incorporate data from both terminals of the transmission cable line, including remote end data, to test whether it will improve fault localization performance and robustness.

6. Learning algorithms: This study intends to examine the effectiveness of Artificial Neural Networks (ANNs) by employing two different learning algorithms: Levenberg-Marquardt (LM) and Scaled Conjugate Gradient

(SCG). The focus is on their application in fault identification, classification, and localization tasks within power cable circuits.

By accomplishing these goals, the use of ANNs in fault localization seeks to improve the reliability, efficiency, and resilience of electrical power systems, ultimately minimizing downtime and enhancing overall system performance.

1.3 PROBLEM STATEMENT

Fault location in power cables can be a challenging task due to various factors and complexities associated with the cable systems. Some of the difficulties encountered in fault location in power cables include:

- Accessibility: Cables are often buried underground or installed in locations that are not easily accessible. This lack of accessibility makes it is difficult to physically access the fault location for inspection and repair.
- Length of the Cable: Power cable systems can extend over vast distances, complicating the task of accurately identifying the exact fault location, particularly when a fault can arise at any point along the cable's length.
- Various Types of Faults: Cable faults can manifest in different forms, such as insulation breakdown, conductor damage, partial discharges, or intermittent faults. Identifying and locating these different types of faults accurately can be complex.
- Interference from Surroundings: Environmental factors, such as moisture, temperature changes, nearby electromagnetic fields, or external disturbances, can affect the behavior of the cable and complicate fault location efforts.
- High Voltage and Currents: Power cables carry high voltages and currents.

During a fault, these high voltages and currents can result in significant energy releases or damage, making the fault location hazardous and potentially unsafe for personnel.

- Cable Sheath and Insulation: The cable's protective sheath and insulation can obscure the fault location or cause misleading readings during fault detection due to their dielectric properties.
- Limited Test Access: Testing and diagnostic equipment may have limited access points along the cable, making it challenging to perform accurate measurements and diagnostics at specific locations.
- Time Constraints: In situations where power interruptions need to be minimized, there might be time constraints for fault location and repair, adding pressure to locate the fault swiftly and accurately.
- Complexity of Cable Networks: In complex cable networks with multiple branches, connections, and configurations, identifying the exact location of a fault and isolating it can be intricate and time-consuming.

Addressing these challenges often involves using specialized fault location methods and equipment, such as Time Domain Reflectometry (TDR), cable route tracers, acoustic methods, and advanced diagnostic technologies, to overcome these difficulties and accurately locate faults in power cables. However, these traditional techniques can be expensive, time-consuming, and sometimes less effective in handling complex fault scenarios, such as those involving high resistance faults or intermittent issues. This is where Artificial Neural Networks (ANNs) offer a significant advantage. ANNs can process large volumes of data efficiently, learn complex patterns in fault signals, and adapt to nonlinear and diverse fault behaviors. By leveraging ANN-based models, the process of fault detection and localization becomes faster, more accurate, and cost-effective, reducing downtime and maintenance efforts. Moreover, ANNs can integrate data from both local and remote ends of the transmission line, providing enhanced reliability and precision in fault localization compared to traditional methods.

CHAPTER 2: LITERATURE REVIEW

The next section will explore the use of AI for detecting, identifying, and locating faults in transmission lines, where the majority of these incidents occur. Various methodologies have been developed to assess the nature and distance of faults from specific measurement points. While many of these methods provide distinct advantages, they also come with complexities or cost considerations that can hinder their implementation. Furthermore, machine learning techniques for fault identification and localization can be categorized into two primary groups: the types of algorithms used and the parameters selected for feature extraction. Diverse algorithms, such as artificial neural networks (ANNs), support vector machines (SVMs), and decision trees (DTs), have been utilized to detect faults and ascertain their locations. Researchers have also created custom features by processing input data with techniques like fast Fourier transform (FFT), discrete wavelet transform (DWT), continuous wavelet transform (CWT), and statistical measures such as maximum, minimum, mean, and standard deviation (SD) to train the algorithms efficiently. As a result, the application of machine learning in fault localization offers a wide range of possibilities, with ongoing research efforts aimed at addressing the challenges faced by traditional fault location methods. The following discussion will concentrate on recent methodologies, examining the techniques used and the inputs leveraged for feature extraction and model training.

The processes for training, testing, and evaluating the intelligent fault locator are based on a multilayer perceptron feed-forward neural network that utilizes the backpropagation algorithm. The Mean Square Error (MSE) metric is employed to assess the performance of both the fault detector/classifier and the fault locator. The results reveal a validation performance (MSE) of 2.36×10^{-9} for the fault detector/classifier and $2.179x10^{-5}$ for the fault locator [7]. This system effectively demonstrates its ability to detect faults, classify different fault types, and accurately identify fault locations.

A new integrated technique has been created that combines protective relaying with an advanced support vector machine algorithm to detect faults and accurately locate them within extensive transmission lines. This innovative approach effectively identifies and categorizes various types of symmetrical and asymmetrical faults. The method involves using voltage and current data from both ends of the transmission line to train the model, which is then utilized for the relay system. While the maximum average accuracy attained by the developed algorithms reaches 95.8%, which is deemed satisfactory, it is essential to note that this proposed method is specifically designed for detecting and classifying faults within the transmission line [8].

This study offers a comprehensive comparative analysis centered on performing protective relaying tasks, particularly focusing on fault classification and location estimation [9]. Utilizing methods such as discrete wavelet transformation, Artificial Neural Networks (ANN), and Decision Trees (DT), the research leverages spectral energy derived from wavelet-transformed voltages and currents as inputs for both ANNs and DTs. The study successfully identifies all ten fault types, accurately locates faults, and determines their zones. It underscores the strengths of Decision Trees in rapid fault classification while noting their limitations in precise fault location determination. In contrast, the ANN demonstrates superior capabilities in fault location and zone identification, despite a slower inference speed [9]. Consequently, the article recommends employing the Decision Tree for fault classification and the ANN for tasks related to location and zone identification.

In [10], The article underscores the effectiveness of Artificial Neural Networks (ANNs) in fault identification, highlighting factors influencing their performance: available data for training and time invested in developing ANN structures. Investigating varied measurement setups in transmission lines, the research explored diverse ANN structures and assessed the benefit of using multiple ANNs. While introducing alternative fault analysis methods, it concluded that no single neural network universally solves all problems. The optimal ANN structure varies based on available measurements, requiring extensive investigation. The study stressed that in order to ensure confidence in defect diagnosis, ANN designers must have a thorough understanding of certain issues in order to implement them effectively.

An integrated approach that employs Wavelet transform, GoogLeNet, and Convolutional Neural Networks (CNN) was introduced to effectively identify, classify, and accurately locate electrical faults on transmission lines [11]. The process begins with the application of discrete wavelet transform and Multi-Resolution Analysis (MRA) to extract pre-cycle and post-cycle fault signals from the sending end. These signals are decomposed into detail and approximate coefficients using the Daubechies (db4) wavelet. Subsequently, the wavelet transform feature components are computed and converted into RGB images (224x224x3) for classification by GoogLeNet, followed by localization using a five-layer CNN. The method's performance is evaluated through MATLAB simulations on a 220 km, 220 kV transmission line within a four-bus power system, encompassing various fault types, resistances ranging from 0 Ω to 15 Ω, and load variations of ± 5%. The results demonstrate that the proposed technique achieves reliable, efficient, and rapid detection, classification, and localization of faults.

This article [12] introduces an innovative fault location algorithm that considers a strategic placement of meters in distribution systems. Demonstrated on the IEEE 13 and 34 node test feeders, this method effectively pinpoints faults using a limited number of meters, circumventing the challenge of multiple estimations common in model-based approaches. Emphasizing meter placement for fault detection enhances location accuracy by employing a more substantial number of measurement points. The method offers insights into the accuracy achieved with varying meter quantities, aiding electric companies in assessing the cost-effectiveness of equipping their networks accordingly. This approach is adaptable to radial distribution networks with distributed generation, facilitating fault analysis amid frequent topology changes.

Previous artificial intelligence methods for identifying faults in transmission lines heavily relied on extracting fault signal characteristics, demanding substantial expertise. These approaches are sensitive to line parameters and are limited in their application across different transmission lines. To overcome these limitations, this research introduces a novel technique, a dual-ended fault localization model merging Convolutional Neural Network (CNN), Maximum Mean Discrepancy (MMD), and Long Short-Term Memory (LSTM). Initially, MMD categorizes distinct transmission lines, enabling the creation of tailored dual-ended CNN-LSTM models suited for comparable lines. These models independently capture fault features comprehensively and adjust combination model weights using the Q-learning algorithm. This approach effectively estimates fault distances, showcasing the strong adaptability of the CNN-LSTM dual-ended combined model across lines with diverse parameters [13].

A deep learning-based Long Short-Term Memory (LSTM) network is utilized for fault detection and classification, while a combination of ANN and LSTM is employed for accurately identifying fault locations within a Microgrid in [14]. The input signals consist of three-phase voltages, three-phase currents, and zero-sequence voltage and current, which are provided to both networks. A thorough analysis across various fault types reveals that the proposed LSTM-based approach demonstrates enhanced accuracy and effectiveness in fault classification and detection. The system's implementation on a real-time simulation platform successfully showcases its capabilities in fault detection, classification, and location identification within the Microgrid.

Innovative nonlinear methods have been developed to accurately identify ground fault locations within the power distribution network by utilizing voltage phasor measurements collected from various network buses via the Digital-PMU phased distribution unit [15]. The first approach employs genetic optimization algorithms and particle swarm optimization for nonlinear modeling, enabling the determination of fault positions along the distribution line under different fault scenarios, including single-phase, two-phase, and three-phase faults. The second technique introduces neural fuzzy network training, utilizing a variety of phasor measurement devices and relying exclusively on phase information derived from the network bus voltage. To validate the effectiveness of these algorithms, a 9-bus system with differing line lengths and characteristics was constructed using MATLAB software. Following the induction of single-phase, two-phase, and three-phase faults, the results demonstrated that the algorithms successfully localized faults in the shortest possible time.

An innovative ANN method for detecting and locating faults in Extra High Voltage (EHV) transmission networks by comparing measured fault distance values with standard or target values. The ANN model undergoes multiple iterations to achieve accurate fault distance estimation. Initially, the untrained ANN results in slower execution, but as it gets trained, it speeds up. Error between predicted and standard values is reduced using backpropagation, employing Gradient Descent and Chaining methods. It's a new, practical, straightforward, precise, and cost-efficient approach [16].

In [17], focuses primarily on selecting effective deep neural networks (DNN-based) methods to develop an application serving as a diagnostic tool for intelligent Low Voltage Distribution Grids (LVDGs). The presented method application's foundation relies on CWT and CNNs models, capable of managing substantial data from measuring units and accurately identifying fault patterns. The implementation of the Tree-Structured Parzen Estimator (TPE) algorithm facilitates the exploration of optimal hyperparameters related to various components of the fault diagnosis tool, including faulty feeder, branch, class, and distance models. This approach achieved an accuracy of 91.4% in fault detection, 93.77% in correct branch identification, 94.93% in fault type classification, and a root mean square error (RMSE) value of 2.45% for location calculations.

A data-fusion model incorporating multiple algorithms that leverage the location outcomes generated by existing algorithms was introduced in [18]. Initially, it examines the complementarity among four chosen algorithms. Subsequently, it explains the potential for enhancing fault location precision through a model of data-fusion and outlines the model's comprehensive structure. The data-fusion model is built utilizing an artificial neural network, celebrated for its excellent ability to fit data effectively. The model's performance is assessed based on multiple parameters, including fault inception angles, fault resistances, line characteristics, and fault positions, while remaining unaffected by factors such as distributed generation and sampling rate.

A technique utilizing neuro-fuzzy system combined with discrete wavelet transform (DWT) to identify and classify high-impedance faults occurring in the distribution network [19]. MATLAB software is utilized to construct the network, conducting various fault simulations (high impedance, symmetrical, and unsymmetrical faults) to assess the success of the neruro-fuzzy classifier method. The neuro-fuzzy classifier is trained using features extracted from the three-phase fault current signal (standard deviation values) via the DWT technique across various fault cases with differing fault resistance values in the system. The proposed method demonstrates a 100% success rate in identifying and classifying high impedance faults.

A discrete wavelet transform used to identify faults within the transmission network by extracting fault characteristics from zero-sequence currents, which are then employed to train an ANN. Using the post-fault data is the core of this approach acquired from both line ends received by the relay [20]. Utilizing Fortescue’s transform, the zero-sequence current from both terminals is computed, and by wavelet transformation, high-frequency stored information yields horizontal components in the zero-sequence current. Evaluating the energy stored in these components, alongside extracting their maximum scales, unveils fault-specific features beneficial for neural network training. Simulation outcomes highlight the dependence of maximum scales and stored energy on fault resistance, type, angle, and location, emphasizing the importance of representative training data for accurate diagnosis by the neural network.

The technique is evaluated on transmission networks with different line lengths, highlighting the algorithm's effectiveness in estimating both fault distance and type under a range of conditions.

The machine learning and advanced signal processing used to detect, classify, and localize faults in a radial distribution grid. By analyzing three-phase current signals and extracting features via discrete wavelet transform (DWT) and statistical methods, the data is fed into models like ANN, Multilayer Perceptron (MLP), and Extreme Learning Machine (ELM) [21]. These models are trained on various fault scenarios, including load and resistance variations. The evaluation, based on performance metrics like MAPE, RMSE, and R^2, shows that MLP and ELM offer high accuracy and comparable fault localization capabilities.

A fast and precise method for detecting, classifying, and locating faults in transmission lines using one-dimensional convolutional neural networks (1D-CNNs) were introduced in [22]. This approach integrates feature extraction and fault classification into a single learning process, eliminating the need for separate feature extraction. The model uses three-phase voltage and current signals from one end of the transmission line as input and is tested on a 132kV transmission line using Matlab Simulink. Compared to conventional neural network and fuzzy neural network methods, the 1D-CNN demonstrates superior accuracy and speed in fault detection and classification.

The proposed method utilizes wavelet transform to decompose fault-generated signals into frequency components, which are then fed into a multilayer perceptron neural network using backpropagation. This approach distinguishes fault locations by focusing on transient signals during fault conditions. Testing shows the method accurately identifies 70.59% of faults, demonstrating its effectiveness in detecting and pinpointing faults in transmission lines using boundary protection. The study also concludes that neural networks provide more precise fault identification and classification compared to conventional systems. [23].

In [24], explores an enhanced protection approach using artificial neural networks within the 330kV Nigerian network simulated in MATLAB. The method uses voltage and current signals from measured faults, processed through discrete Fourier transform via fast Fourier transform, as inputs for a neural network. The network successfully handles fault detection, classification, and location across the entire protected line. Compared to traditional protection methods, the neural network shows faster operation, completing its task in 5.15 milliseconds, significantly quicker than the 0.350 seconds needed by standard numerical relays.

An artificial neural network (ANN) method using a regularized radial basis function (RRBF) to diagnose and locate faults in the IEEE 13-bus active distribution network (ADN) was presented in this article [25]. The approach focuses on analyzing phase angle features derived from fault voltage and current signals, utilizing synchronized amplitude and phase angle data for diagnosis. The research assesses the accuracy and precision of fault location, considering various input signals, fault positions, and resistances. Simulation results show that using the phase angle feature enhances location accuracy, contributing to improved stability in ADN systems.

This thesis explores The implementation of an artificial neural network (ANN) algorithm was explored in a numerical relay for power system protection in [26] . The three-phase current and voltage data used in a feedforward neural network, along with a backpropagation algorithm, is utilized to detect, classify, and locate faults [27]. The study investigates different hidden layer configurations to optimize the neural network's performance. Simulation results demonstrate the ANN's ability to accurately identify and categorize faults, as well as pinpoint their locations, under multiple fault scenarios in transmission lines. The proposed method proves to be highly effective in enhancing fault detection and classification.

Two approaches for fault classification are presented: one using a single ANN and the other with an integrated ANN structure. A comparison is made between the two to identify the better-performing classifier. Additionally, the article outlines the design of an integrated ANN-based fault locator, comparing the accuracy of three different fault locators [28]. The ANNs use pre- and post-fault samples of three-phase currents and voltages for fault detection. Simulations across various fault conditions demonstrate the effectiveness of the proposed methods in accurate fault classification and location.

In [29], introduces an advanced synthetic framework designed for the rapid identification and localization of short circuit faults occurring in transmission lines. The proposed algorithm distinguishes short circuit faults by analyzing the voltage waveform and three-phase current measurements during fault incidents on power transmission lines. It utilizes values obtained from wavelet transforms of three-phase voltages and currents for the classification of faults process. Following the classification process, a fault localization algorithm is activated. The approach integrates various methods like SVM and multilevel wavelet transform to determine fault classification and precise location in real-time. The paper elucidates the core concept of this framework, introducing a pattern recognition method employing wavelet transforms, statistical processing techniques, neural networks (NN), and a collective decision-making mechanism. Preprocessing and training of support vector machines utilize voltage and recorded current values from the moment of the fault occurrence to a quarter of the post-fault duration.

A novel method for fault location that leverages high-frequency noise patterns generated by switching transients from photovoltaic converters and associated cables has been introduced in [30]. By applying the discrete wavelet transform, the monitored signal is decomposed to extract distinctive features from the high-frequency noise. Norm values of the waveform across various frequency bands yield unique characteristics linked to different fault locations, which are then compiled into feature vectors for pattern recognition. A three-layer feedforward artificial neural network (ANN) classifier is employed to automate the classification of fault locations based on these extracted features. Evaluations demonstrate the effectiveness of the proposed fault locator across varying system parameters, revealing that the method delivers accurate and reliable performance even amidst noisy measurements and fluctuating operational conditions.

An Artificial Neural Network (ANN) were examine to detect, classify, and locate faults in power transmission lines using MATLAB by applied the method to a 3-Bus power system [31]. The feedforward network utilizes a backpropagation algorithm employing the Levenberg Marquardt optimization technique. During fault conditions and normal operations, the method takes RMS values of three-phase voltages, currents, and zero-sequence current and voltage as inputs. Overall, both ANN models exhibit satisfactory and acceptable performance based on evaluation criteria.

A two-phase fault detection system tailored for medium-voltage direct current (MVDC) networks, utilizing wavelet analysis techniques is used with ANN . The proposed Grid Transient Classifier (GTC) continuously monitors the energy levels of wavelet decomposition from both current and voltage signals to identify a range of transient disturbances, including load variations, various fault types, voltage dips, and switching activities. Following this, an artificial neural network (ANN) classifier is employed to determine the specific type of disturbance identified by the GTC. Additionally, the Active Grid Impedance Estimator (AGIE) introduces a signal comprising two fundamental frequencies to evaluate grid impedance under new conditions from a designated measurement point. This information aids in determining the fault type, location, and severity. Hardware-in-the-loop (HIL) results from a case study conducted on a shipboard grid validate the real-time applicability of this method, confirming its efficacy [32].

The application of Artificial Neural Networks (ANN) for detecting and locating faults in power distribution lines is highlighting the technology's promise for improving power system management. In [33], [34] and [35] focus on enhancing the accuracy and efficiency of fault identification, aiming to develop a faster and more precise approach for fault detection. Using MATLAB and the Levenberg-Marquardt algorithm, the research created and trained an ANN model with various configurations of hidden layers and neurons. The results demonstrated the ANN model's exceptional accuracy in recognizing and locating faults, outperforming traditional fault detection methods in both speed and precision. These findings underscore the significant potential of ANN for fault detection and localization in power systems.

Table 1 shown below illustrates the summary of the literature review.

Table 1: Summary of Literature Review

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[7]	2019	multilayer Perceptron feed-forward neural network	Three-phase voltage and current before and during fault	10 types of faults	MATLAB Simulink	MSE and Accuracy	The accuracy was 100% for fault classifier and 2.179x10^{-5} for fault locator
[8]	2023	SVM	Three-phase voltage, Three-phase current and time and frequency parameters	10 types of faults	MATLAB Simulink	Accuracy	The average accuracy was 98%
[9]	2022	DT & ANN	Three-phase voltage and current after applying DWT	10 types of faults	MATLAB Simulink	Recall and actual percentage error	ANN has better performance comparing with DT

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[10]	2022	ANN	Phasor voltage and current	10 types of faults	MATLAB Simulink	MAE	The paper focused on choosing best ANN configuration in localizing and classifying faults in transmission lines
[11]	2022	GoogLeNet & CNN	WT of current and voltage signals	10 types of faults	MATLAB Simulink	MAE	Combine image processing with CNN
[12]							
[13]	2021	CNN-LSTM	Three-phase voltage and current	LG, LLG, LLL	MATLAB Simulink	RMSE, MAE & MAPE	The paper focused on fault localization performance and generalization performance
[14]	2023	LSTM and Feed-Forward Neural Network	Three-phase voltage and current	10 types of faults	MATLAB Simulink & OPAL-RT digital simulator.	Graphical representation	Comparison between LSTM and ANN

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[15]	2023	Neural Fuzzy Network	voltage phasor measurements	Ground faults in the distribution network	MATLAB Simulink	Accuracy	Nonlinear techniques are introduced for locating ground faults in the power distribution network using voltage phasor measurements from various points in the network.
[16]	2023	ANN-based	Three-phase voltage and current	10 types of faults	MATLAB Simulink	Percentage Error	The maximum percentage error was 1.28%

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[17]	2023	CWT & CNN	Three-phase voltage and current	10 types of faults	MATLAB Simulink	Mean Absolute Error (MAE) & Root Mean Square Error (RMSE)	The results show high effectiveness, delivering accurate fault diagnosis to system operators with 91.4% accuracy in fault detection, 93.77% in branch identification, 94.93% in fault type classification, and a 2.45% RMSE for location calculation

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[18]	2023	Data-fusion model based on ANN	Three-phase voltage and current after FFT	10 types of faults	PSCAD/EMTDC	Mean Absolute Error (MAE)	The developed model demonstrates strong generalization capabilities and maintains its performance regardless of distributed generation or sampling rate
[19]	2018	Adaptive Neuro-fuzzy Inference System	Three-Phase Current Signals	10 types of faults	MATLAB Simulink	Success rate and Discrimination rate	The paper focused on HIF using artificial intelligent
[20]	2019	ANN-based	Wavelet characteristic of zero sequence current from both ends	10 types of faults	MATLAB Simulink	Absolute error	Comparison between proposed method with impedance based
[21]	2020	ANN-based, MLP and ELM	Wavelet of decomposition of three-phase current of sending end	10 types of faults	MATLAB Simulink	MAPE, RMSE, (WIA) and NSEC	Comparison between three different ANNs

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[22]	2021	One dimensional convolutional neural network (1D-CNNs)	The three-phase voltages and currents signal of one end	10 types of faults	MATLAB Simulink	Accuracy	Comparison between proposed method with conventional ANN and fuzzy neural
[23]	2021	A Multilayer Perceptron Neural Network with Backpropagation	Wavelet of decomposition of three-phase current of sending end	10 types of faults	MATLAB Simulink	Accuracy	The study reached 70.6% of accuracy
[24]	2017	ANN-based	Three-phase voltage and current after FFT	10 types of faults	MATLAB Simulink	Absolute error	This paper describes artificial neural networks algorithm for fault classification and location
[25]	2018	The artificial neural network (ANN) based on regularized radial basis function (RRBF)	Three-phase voltage and current	10 types of faults	MATLAB Simulink	Accuracy	The RRBF method is applicable to diagnose and locate fault at a high precision

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[26]	2022	ANN-based	Three-phase voltage and current	10 types of faults	MATLAB Simulink	MSE	This paper describes artificial neural networks algorithm for fault classification and location
[27]	2023	ANN based	Three-phase voltage and current	10 types of faults	MATLAB Simulink	Absolute error	The accuracy of fault locator reached upto 98.33%
[29]	2017	SVM and Support Vector Regressions (SVR)	Wavelet of decomposition of three-phase current and voltage	10 types of faults	MATLAB Simulink	Percentage error	The paper used SVR and SVM for fault classification and location
[30]	2018	Wavelets and ANN	Wavelet of decomposition of three-phase current and voltage	10 types of faults	MATLAB Simulink	Accuracy	Using Wavelets and ANN to locate fault on ungrounded PV system
[31]	2022	ANN based	Three-phase Voltage, current, zero sequence current phase and voltage	10 types of faults	MATLAB Simulink	Percentage error	The used ANN to detect, classify and locate fault in the transmission line

Ref#	Year	AI type	Input Features	Fault Type	Data Source	Performance parameters	Key Finding
[32]	2022	ANN based	Wavelet of decomposition of phase current and voltage	DC faults	MATLAB Simulink	Accuracy	The accuracy of proposed method reached upto 99.5 %
[33]	2023	ANN based	The voltages and currents of three-phases, the voltage and current of zero sequences, and the angle of the negative voltage sequence component	10 types of faults	MATLAB Simulink	Percentage error	The used ANN to detect, classify and locate fault in the transmission line by introducing new inputs
[34]	2022	ANN based	Three-phase Voltage, current	10 types of faults	MATLAB Simulink	Accuracy	Simulate distance and differential protection systems
[35]	2023	ANN based	Three-phase Voltage, current	10 types of faults	MATLAB Simulink	Accuracy	Estimate fault in power cable circuit

CHAPTER 3: SYSTEM CONFIGURATION AND FAULT CALCULATION

This chapter will explore the development and illustration of the power system used for the generation of training, testing, and validation datasets. It will comprehensively discuss various fault types that will be encompassed within the study. Furthermore, it will interpret symmetrical components and the calculation of sequence impedance, aiming to facilitate the computation of short-circuit currents during unsymmetrical faults. Moreover, it will expound upon the per-unit system and RMS values, helping in the creation of datasets utilized for training the Artificial Neural Network (ANN). Lastly, a detailed presentation regarding fault calculation for all fault types will be provided.

3.1 SYSTEM CONFIGURATION

Conventional transmission lines primarily comprise power cables or overhead lines (OHLs) that facilitate the transmission of electrical power between substations. The system, which serves as the foundation for generating all trained datasets, is visually represented in Figure 2. This schematic demonstrates the interconnection of two substations using two parallel power cables. One of these substations is linked to the electrical grid, while the other interfaces with a designated load bus.

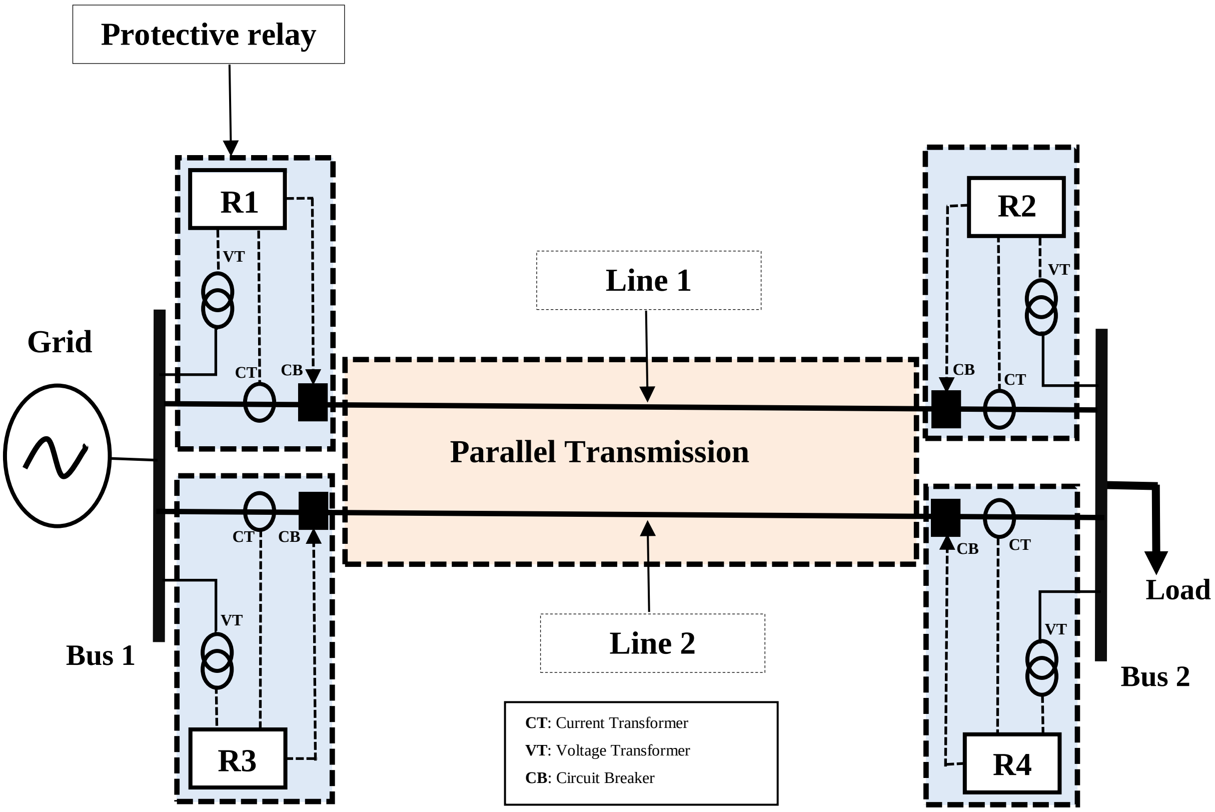


Figure 2. Interconnection of two substations by parallel power cables

Throughout this research endeavor, a diverse array of network scenarios will be meticulously examined and detailed as follows:

1- A single-line configuration with a generation source located at the sending end.

In this scenario, one of the lines within the network is deliberately switched off, while the other line maintains its operational status, and the sending end linked to the grid. The configuration of this scenario is shown in Figure 3.

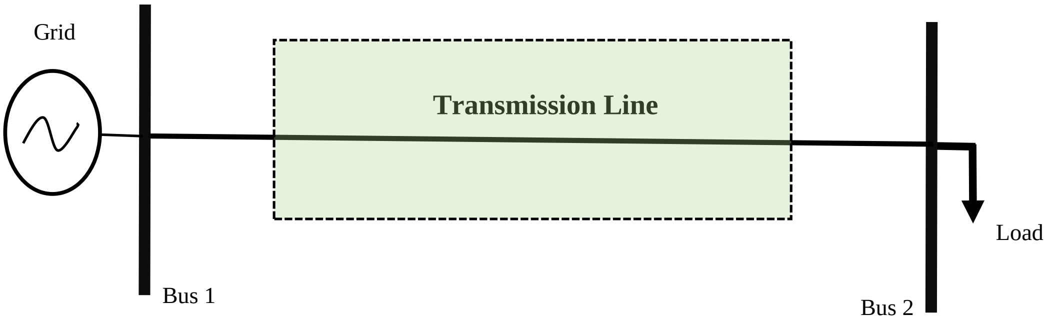


Figure 3. A single line configuration with a generation source located at the sending end

2- Parallel lines configuration with a single generation source at the sending end.

This configuration portrays the simultaneous connection of two parallel lines interlinking the two substations and the sending end connected to the grid. Figure 4 shows the configuration of this scenario.

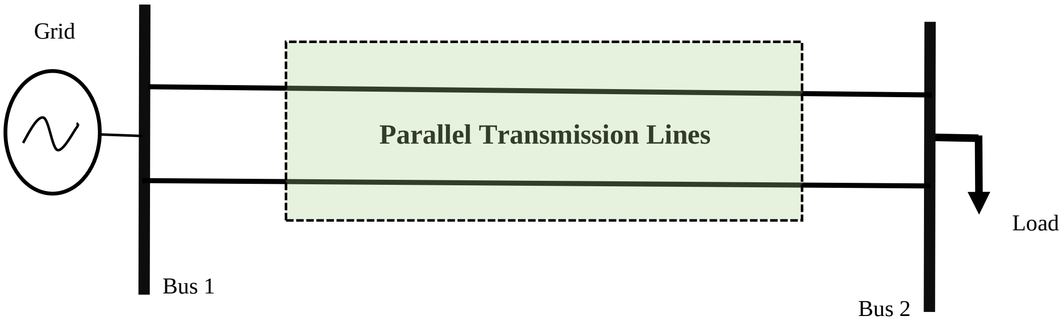


Figure 4. Parallel lines configuration with a single generation source at the sending end

3- A single line configuration with generation sources located at both the sending and receiving ends.

In this context, a comparative evaluation is being carried out between the substations. While one substation remains connected to the primary grid, the other generates power autonomously; however, only one line is operational. Both substations serve distinct loads, and this scenario delves into the operational dynamics of these paired configurations. Figure 5 shows the configuration of this scenario.

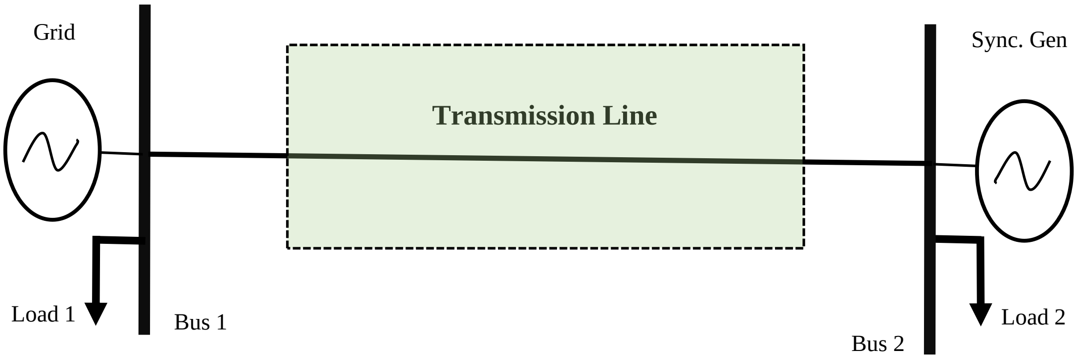


Figure 5. A single line configuration with generation sources located at both the sending and receiving ends

4- Parallel lines configuration with generation sources situated at both the sending and receiving ends.

Here, a comparative analysis is conducted between the substations. One substation is connected to the primary grid, while the other independently generates power. Both substations cater to specific loads, and this scenario examines the dynamics of these dual configurations with parallel lines. Figure 6 shows the configuration of this scenario.

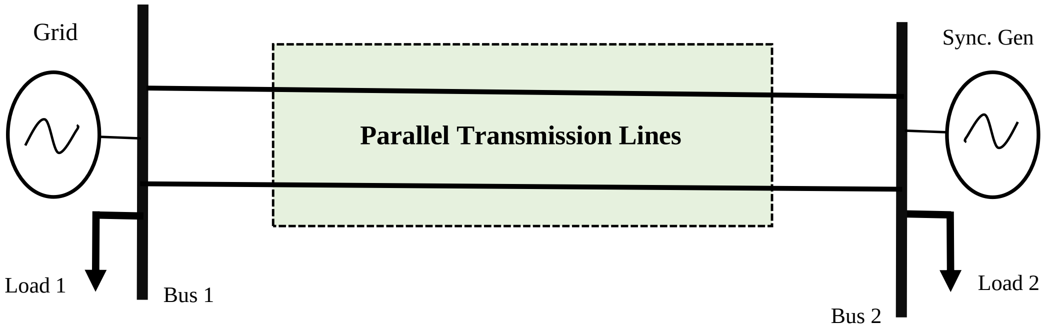


Figure 6. Parallel lines configuration with generation sources situated at both the sending and receiving ends

The investigation into these diverse network scenarios seeks to explore the intricate behaviors and responses of the transmission lines during faults under varying conditions. By thoroughly analyzing each of these scenarios, this research aims to ascertain the system's resilience, efficiency, and functionality of ANN in distinct operational states. Additionally, insights derived from these multifaceted scenarios will provide valuable information for optimizing the operational performance and reliability of ANN, particularly concerning fault tolerance and stability.

The system model in all scenarios relies on specific component parameters, detailed comprehensively in Table 2, for accurate system representation and simulation.

Table 2. Parameters Of All Components Used In The Model

Components	Value
Grid	Voltage: 66 KV Short circuit level: 50 MVA Configuration: Y grounded
Power Cable -1	Line Length: = 10 km R_1 = $12.73 \text{x} 10^{-2}$ ohms/km X_1 = $93.37 \text{x} 10^{-3}$ ohms/km R_0 = $38.64 \text{x} 10^{-2}$ ohms/km X_0 = $41.264 \text{x} 10^{-4}$ ohms/km
Power Cable -2	Line Length = 10 km R_1 = $12.73 \text{x} 10^{-2}$ ohms/km X_1 = $93.37 \text{x} 10^{-3}$ ohms/km R_0 = $38.64 \text{x} 10^{-2}$ ohms/km X_0 = $41.264 \text{x} 10^{-4}$ ohms/km
Sync. Generation	Voltage: 66 KV Apparent power: 10 MVA
Load-1	Active power: 3.6 MW Reactive power: 2.7 MVAR
Load-2	Active power: 3.6 MW Reactive power: 2.7 MVAR

3.2 CLASSIFICATION OF POWER CABLE FAULTS

Power cable faults can arise due to several reasons, such as insulation breakdown, joint failures, or external factors like accidental damage during excavation. Understanding fault types is crucial for effective system management. Faults are generally categorized as symmetrical and unsymmetrical. Symmetrical faults encompass three-phase faults and three-phase-to-ground faults. On the other hand, unsymmetrical faults include line-to-ground (1LG), line-to-line (LL), and line-to-line-to-ground (2LG) faults. Symmetrical faults are relatively easier to analyze since they adhere to the laws governing electric circuits. Conversely, unsymmetrical faults pose complexities in calculating fault currents and voltages, requiring alternative analytical approaches. The subsequent section delves into symmetrical components, pivotal for comprehending and assessing unsymmetrical faults in power cable systems. Understanding these fault types is critical for establishing robust fault detection and mitigation strategies within power cable networks. Figure 7 visually represents the various fault types discussed previously, encompassing both symmetrical and unsymmetrical faults found in power cable systems.

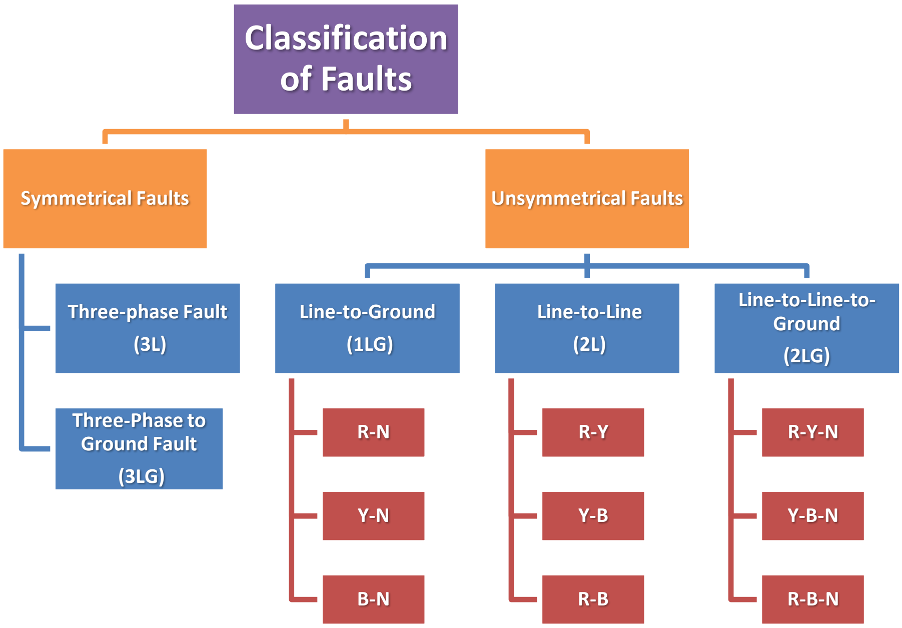


Figure 7. Classification of faults in transmission lines

3.3 SYMMETRICAL COMPONENTS

Symmetrical components, introduced by C.L. Fortescue in 1918, stand as a powerful analytical tool employed for scrutinizing unbalanced three-phase systems in electrical engineering. This method's core principle revolves around the transformation of a complex three-phase system into three individual and decoupled systems known as sequence networks. The purpose of these sequence networks lies in dividing and understanding the behavior of the system in an unbalanced state. These sequence networks, comprising zero, positive, and negative sequences, are instrumental in comprehending and analyzing the intricate aspects of unbalanced systems. Figure 8 shows the symmetrical components for three-phase system.

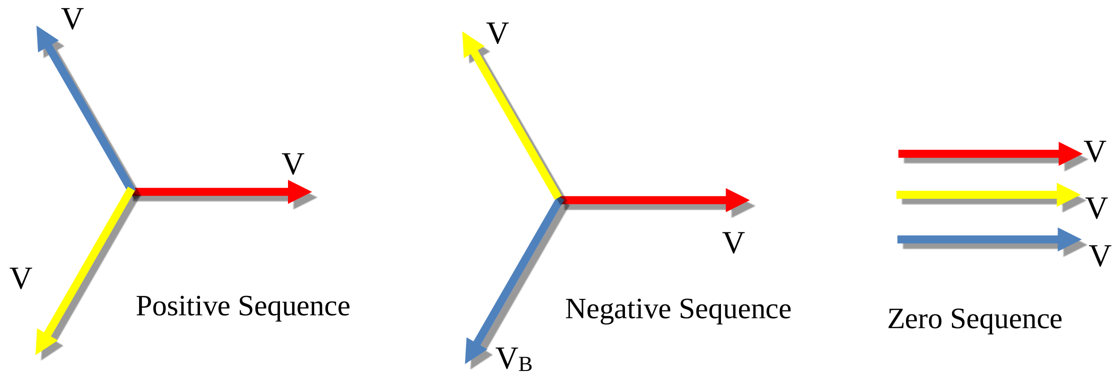


Figure 8. Symmetrical components of three-phase voltage

The transformation process involves segregating the original system into three distinct components: the positive sequence, negative sequence, and zero sequence networks. The zero sequence network deals with components that have symmetrical amplitudes and are in phase across all three-phases. This sequence characterizes phenomena such as ground faults, offering insights into fault currents and system stability in such occurrences. Contrarily, the positive sequence network comprises elements with equal magnitudes and phase shifts. This sequence primarily characterizes balanced operation, depicting the standard conditions where the system functions under balanced load conditions and normal operations. The negative sequence network reflects conditions where there are imbalances or negative sequence components within the system. This sequence often signifies asymmetrical conditions, usually caused by unsymmetrical loads or other disruptive factors, showcasing deviations from the system's ideal balanced state. The analysis of these sequence networks grants a complete understanding of the behavior of an unbalanced system during various operational scenarios. By isolating and examining these distinct sequences, engineers gain insights into fault types, system stability, and responses to asymmetrical conditions. Moreover, symmetrical components offer a methodical approach to understanding system behavior during faults and deviations from normal conditions. For instance, during a fault event, the zero sequence network aids in comprehending the ground fault and assessing the system's response to it. The negative sequence network facilitates the identification and analysis of unbalanced system elements and their effects on overall system behavior, while the positive sequence network helps assess the system's performance under balanced conditions.

The calculation of sequence voltages involves specific mathematical transformations applied to the original three-phase voltage system. The positive sequence voltage, **V_1** which represents the balanced components of the system, can be determined by taking the average of the three-phase voltages:

$$V_1 = \frac{V_a + a^2 V_b + a V_c}{3} \tag{3.3.1}$$

Where:

- V_a, V_b and V_c are the phase voltages of the three-phases A, B, and C, respectively.
- $a = 1\angle 120$ represents the phase shift in a three-phase system.

The negative sequence voltage, V_2, which highlights the asymmetrical components in the system, is calculated similarly:

$$V_2 = \frac{V_a + a V_b + a^2 V_c}{3} \tag{3.3.2}$$

Meanwhile, the zero-sequence voltage Vo signifies the symmetrical components or the ground fault. It is computed by taking the average of the three-phase voltages:

$$V_o = \frac{V_a + V_b + V_c}{3} \tag{3.3.3}$$

Therefore, the sequence voltage can be calculated as in (3.3.4):

$$\begin{bmatrix} V_1 \\ V_2 \\ V_o \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & a & a^2 \\ 1 & a^2 & a \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{3.3.4}$$

Similarly, the calculation of sequence currents involves transforming the original three-phase current system into positive, negative, and zero sequence components by using following (3.3.5)

$$\begin{bmatrix} I_1 \\ I_2 \\ I_o \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & a & a^2 \\ 1 & a^2 & a \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \tag{3.3.5}$$

These calculations provide understandings into the characteristics and behavior of the system under balanced and unbalanced conditions, aiding in fault analysis and system stability assessments.

Each component within an electrical network possesses its unique sequence impedances. These sequence impedances, including positive, negative, and zero sequence impedances, characterize the behavior of components within the network concerning unbalanced conditions or faults. The sequence impedances enable the analysis of asymmetrical conditions and fault scenarios, allowing for a more comprehensive understanding of the system's response to such events. By having distinct sequence impedances for each component, the network's behavior during unbalanced operations or fault occurrences can be accurately evaluated and addressed to maintain system reliability and stability.

Typically, the sequence impedances of electrical components are supplied by the manufacturer. These values are acquired through specialized testing procedures conducted by the manufacturer.

3.4. SINGLE LINE TO GROUND FAULT

Let's consider a scenario as shown in Figure 9, where a ground fault occurs in phase A (typically denoted as A, B, and C instead of R, Y, and B, respectively. E_a, E_b, and E_c represent the per-phase voltage at the generator terminal, denoted in bold letters to signify vector form.

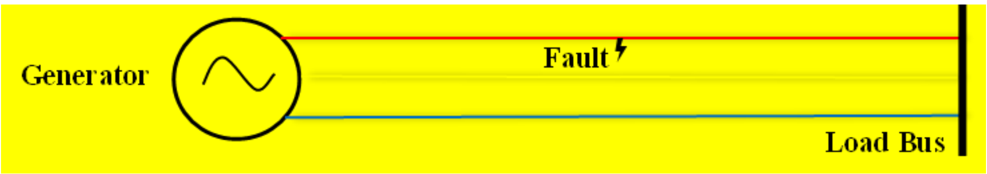


Figure 9. Single phase to ground fault in phase A

As a result of the ground fault in phase A, the voltage at the fault location drops to zero, and consequently, the current flowing through phases B and C also can be consider zero compared to fault current. This circumstance leads to the expression:

$V_A = 0$ $\quad\quad$ $I_B = 0$ $\quad\quad$ $I_C = 0$

Hence, based on the symmetrical components calculation; by using (3.3.5) can express this as:

$$I_1 = \frac{I_A}{3} \qquad I_2 = \frac{I_A}{3} \qquad I_0 = \frac{I_A}{3}$$

From the formulations depicting the positive, negative, and zero sequence components of current in faulty phase A, it's evident that all sequence currents share equivalent magnitudes and phases. Hence, in a single line to ground fault in the affected phase, the expression can be represented as:

$$I_{a0} = I_{a1} = I_{a2} = \frac{I_A}{3} \qquad (3.4.1)$$

It is crucial to note that the fault current flows through the grounded neutral. Without a grounded neutral, no-fault current would exist. To calculate the fault current, Kirchhof's voltage law will be utilized. As the fault current solely flows through the faulted phase A, our focus will be on determining I_a. then voltage at generator's terminal E_a is:

$$E_a = V_A + I_{a0}Z_{a0} + I_{a1}Z_{a1} + I_{a2}Z_{a2} \qquad (3.4.2)$$

But,

$$V_A = 0 \text{ and } I_{a0} = I_{a1} = I_{a2} = \frac{I_A}{3} \qquad (3.4.3)$$

Then,

$$E_a = \frac{I_A Z_{a0} + I_A Z_{a1} + I_A Z_{a2}}{3} \qquad (3.4.4)$$

Then,

$$I_A=\frac{3E_a}{Z_{a0}+Z_{a1}+Z_{a2}} \tag{3.4.5}$$

The preceding fault current expression indicates that for the single line to ground fault, the positive, negative, and zero sequence impedances are linked in series. This arrangement can be represented by the following equivalent circuit in Figure 10:

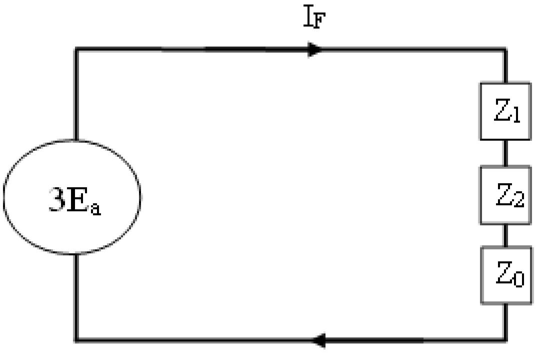


Figure 10. Sequence impedance for single phase to ground fault

The fault current equation is established under the assumption of a solidly grounded neutral system. If the system's neutral is grounded with a finite resistance, denoted as $\mathbf{Z_f}$, then the fault current can be expressed as:

$$I_A=\frac{3E_a}{Z_{a0}+Z_{a1}+Z_{a2}+3Z_f} \tag{3.4.6}$$

3.5. LINE TO LINE FAULT

A line-to-line fault, also known as a phase-to-phase fault, occurs when two conductors, such as phases R and Y or Y and B, contact each other. This fault condition results in a direct short circuit between these conductors. Figure 11 shows a typical Y-

B fault.

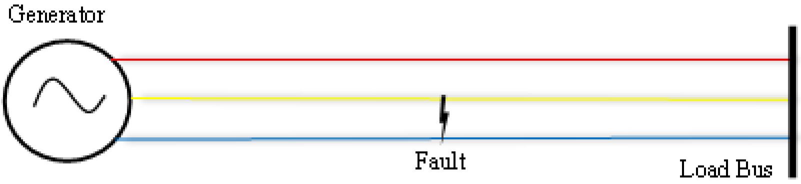


Figure 11. LL fault yellow to blue

$$I_a = 0 \tag{3.5.1}$$

$$I_c = -I_b \tag{3.5.2}$$

$$V_{bg} - V_{cg} = Z_F I_b \tag{3.5.3}$$

From above equation we can write positive, negative and zero sequence current as:

$$\begin{bmatrix} I_0 \\ I_1 \\ I_2 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & a & a^2 \\ 1 & a^2 & a \end{bmatrix}\begin{bmatrix} 0 \\ I_b \\ -I_b \end{bmatrix} \tag{3.5.4}$$

And sequence voltages can be written as:

$$(V_0 + a^2 V_1 + aV_2) - (V_0 + aV_1 + a^2 V_2) = Z_F(I_0 + a^2 I_1 + aI_2) \tag{3.5.5}$$

By simplifying (3.5.5)

$$V_1 - V_2 = Z_F I_1 \qquad (3.5.6)$$

By translating the fault conditions from the phase domain to the sequence domain, the resulting fault current can be expressed as:

$$I_1=-I_2=\frac{V_F}{(Z_1+Z_2+Z_F)} \qquad (3.5.7)$$

By employing equations (3.5.4) through (3.5.7), the sequence network can be constructed as depicted in Figure 12.

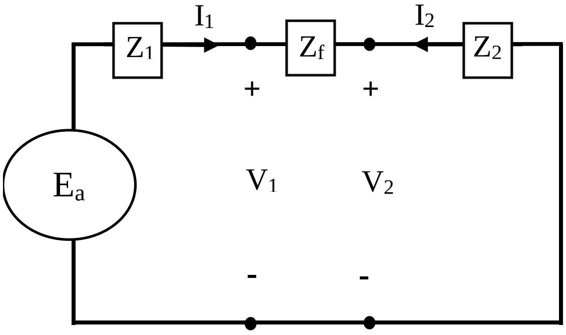


Figure 12. Sequence impedance of LL fault

3.6 DOUBLE LINE-TO-GROUND FAULT

A double line-to-ground fault occurs when two phases are connected to the ground through a fault impedance, as depicted in Figure 13.

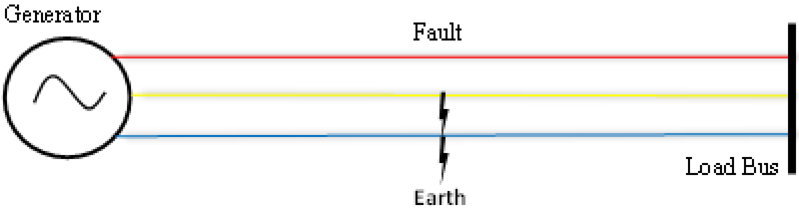


Figure 13. 2LG fault

I_a=0 (3.6.1)

V_{cg}=V_{bg} (3.6.2)

V_{bg}=Z_F(I_b+I_c) (3.6.3)

Converting the system into a sequence network through the application of symmetrical components will define the fault conditions.

I_0+I_1+I_2=0 (3.6.4)

V_0-V_1=(3Z_F)I_0 (3.6.5)

$V_1 = V_2$ (3.6.6)

Equations (3.5.4) through (3.5.6) are met when the zero, positive, and negative sequence networks are linked in parallel, as illustrated in Figure 14.

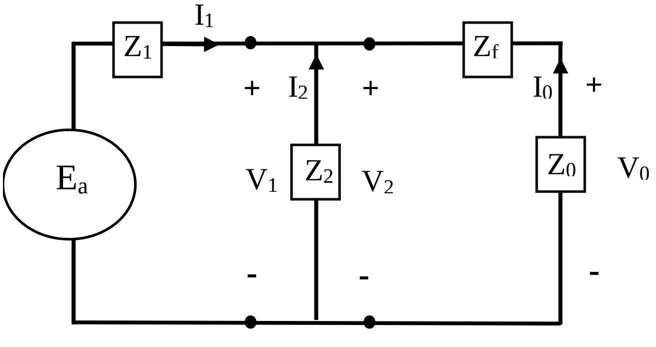


Figure 14. Sequence impedance of 2LG Fault

So, the sequence currents will be:

$$I_1 = \frac{V_F}{Z_1 + \left[\frac{Z_2(Z_0+3Z_F)}{Z_2+Z_0+3Z_F}\right]} \qquad (3.6.7)$$

$$I_2 = (-I_1)\left(\frac{Z_0+3Z_F}{Z_2+Z_0+3Z_F}\right) \qquad (3.6.8)$$

$$I_0 = (-I_1)\left(\frac{Z_2}{Z_2+Z_0+3Z_F}\right) \qquad (3.6.9)$$

3.7 SYMMETRICAL FAULT

Symmetrical faults occur when all three-phases are interconnected. Such faults are relatively rare, accounting for only 2%-5% of all faults [29]. Despite their infrequency, symmetrical faults pose the most severe threat as they typically occur in balanced networks. Thevenin's method is commonly employed to address symmetrical faults, involving the calculation of Thevenin impedance from the faulted point and the subsequent determination of fault current. Figure 15 shows the symmetrical fault.

$$I_f = \frac{V_F}{Z_{Th}} \qquad (3.6.1)$$

Where:

V_f and Z_{Th} are the pre-fault voltage and Thevenin impedance observed from the faulted point.

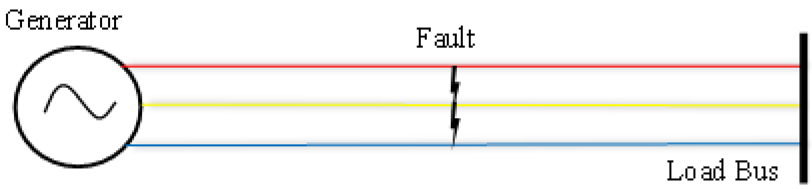


Figure 15. Symmetrical fault (3LG)

3.8 PER UNIT SYSTEM

In the context of power systems, "per unit" (pu) is a method of normalizing and expressing electrical quantities in a relative manner rather than using absolute values. This method is commonly used in power system analysis and design to simplify calculations and analysis, especially when dealing with complex interconnected systems.

The per unit system is used to express the magnitudes of various electrical quantities such as voltage, current, power, impedance, and other parameters in a normalized form relative to a base value. This base value is typically chosen as the rated or nominal value of the quantity. The formula to convert absolute values to per unit values is:

$$Per\ Unit\ Value = \frac{Actual\ Value}{Base\ Value} \qquad (3.8.1)$$

For example, consider a power system rated at 50 MVA and 66 KV on the side of the transmission line. If we choose the base values as 50 MVA and 66 kV, then the per unit values can be calculated:

If the actual voltage on the primary side is 132 kV:

$$Per\ Unit\ Voltage = \frac{66}{66} = 1\ pu \quad (3.8.2)$$

Similarly, current and impedance can also be expressed in per-unit values using their respective base value as follows:

$$I_{base} = \frac{S_{base}}{V_{base}} \quad (3.8.3)$$

$$Z_{base} = \frac{V_{base}^2}{S_{base}} \quad (3.8.4)$$

Therefore, the current base and impedance base is as follow:

$$I_{base} = \frac{50 \times 10^6}{66 \times 10^3} = 757.76\ A \quad (3.8.5)$$

$$Z_{base} = \frac{(66 \times 10^3)^2}{50 \times 10^6} = 87.12\ \Omega \quad (3.8.6)$$

The calculation of transmission line per-unit impedance relies on the base impedance, as illustrated in equation 3.8.6. The per-unit impedance values for transmission lines are detailed in Table 3.

Table 3. Impedance Pu For Transmission Lines

Line from bus-to bus	Impedance (pu)
Line 1	$1.461 + j1.0172 \times 10^{-3}$
Line 2	$1.461 + j1.072 \times 10^{-3}$

The per-unit system simplifies calculations in power system analysis, especially in scenarios involving multiple transformers, generators, transmission lines, and loads with different ratings. It helps standardize values across the system and makes it easier to compare and analyze various components regardless of their absolute ratings.

3.9 ROOT-MEAN-SQUARE (RMS)

RMS, in the context of a power system, usually refers to Root Mean Square, which is a method of measuring the effective value of an alternating current (AC) or voltage. RMS values are commonly used to quantify the equivalent DC value of an AC waveform, making it easier to compare and calculate power in AC systems.

In a power system, RMS (Root Mean Square) values are crucial for determining the effective power that a system can deliver or handle. These values are utilized in calculating power, current, voltage, and other parameters in AC systems.

For instance, when dealing with AC power, the formula to calculate power P using RMS values of voltage V_{RMS} and I_{RMS} is:

$$\mathrm{P} = \mathrm{V_{RMS}} \times \mathrm{I_{RMS}} \times \text{Power Factor} \qquad (3.9.1)$$

Where:

- V_{RMS} is the Root Mean Square voltage.

- I_{RMS} is the Root Mean Square current.
- Power Factor is the cos of the phase angle between current and voltage.

RMS values are crucial in power system analysis, helping to determine the effective power transferred, minimizing losses, and ensuring the safe and efficient operation of electrical equipment within the system. The Root Mean Square (RMS) value of a set of values, such as voltage or current in an AC system, is calculated by taking the square root of the mean (average) of the squares of the values. For a discrete set of values x1, x2, x3, …., xn, the RMS value is calculated using the following formula :

1- Square each individual value: $x_1^2, x_2^2, x_3^2, ..., x_n^2$.
2- Calculate the mean (average) of these squared values: $\frac{x_1^2+ x_2^2 +\cdots+ x_n^2}{n}$
3- Take the square root of the result obtained in step 2 to get the RMS value: $\sqrt{\frac{x_1^2+ x_2^2 +\cdots+ x_n^2}{n}}$

For continuous functions, such as a sine wave for voltage or current in an AC system:

1- Square the function: $f^2(t)$.
2- Calculate the average over one full period (cycle) using integration: $\frac{1}{T}\int_0^T f^2(t)\,dt$ where T is a period of one cycle.
3- Take the square root of the result obtained in step 2 to get the RMS value: $\sqrt{\frac{1}{T}\int_0^T f^2(t)\,dt}$

For a simple sine wave with amplitude A, the RMS value is calculated as $\frac{A}{\sqrt{2}}$. This calculation helps determine the equivalent DC value that conveys the same amount of power as the original AC signal.

As shown in the symmetrical components can be used to calculate transmission lines sequence impedance. Where sequence impedance has a vital role in calculating the value of fault current in different types of faults and the types of faults are symmetrical and unsymmetrical faults.

3.10 FAULT CURRENT CALCULATION

Fault current is calculated depending on different types of faults mentioned in the current chapter using equation (3.4.6) to equation (3.5.9). Two different scenarios were considered during fault calculation when a line is available in system or when two parallel lines are in system. Table 4 shows the sequence impedance of the generator used. However, Table 5 shows the sequence impedance of the system while one line is available, and Table 6 shows two parallel are available in system.

Table 4. Generator Data [1]

Type of generator	Zero sequence (P.U)	Positive sequence (P.U)	Negative sequence (P.U)
Turbo – Generator (solid rotor)	0.05	0.15	0.13

Table 5. Sequence Impedance Of The System When Single Line

Sequence Impedance (P.U)	**Z_0**	**Z_1**	**Z_2**
	$j0.0505$	$j0.15011$	$j0.13011$

Table 6. Sequence Impedance Of The System When Two Parallel Line

Sequence Impedance (P.U)	**Z_0**	**Z_1**	**Z_2**
	$j0.0502$	$j0.15$	$j0.13$

Table 7 and Table 9 present the fault current at the end of the line in a system when a fault occurs with single and both generation in service, respectively. Table 8 and Table 10 show the fault current at the end of the line when two parallel lines are connected, with single and both generation in service, respectively.

Table 7. Fault Current For Single Line

Type of Fault	**1LG**	**LL**	**2LG**	**3LG**
Current (A)	4007.58	4946.96	5318.18	5,681.82
Current (pu)	5.29	6.53	7.02	7.50

Table 8. Fault Current For Parallel Lines

Type of Fault	**1LG**	**LL**	**2LG**	**3LG**
Current (A)	2765.15	3022.73	3303.03	3469.70
Current (pu)	3.65	3.99	4.36	4.58

Table 9. Fault Current For Signle Line With Both Generations

Type of Fault	**1LG**	**LL**	**2LG**	**3LG**
Current (A)	4446.97	4924.24	5075.76	5689.39
Current (pu)	5.87	6.50	6.7	7.51

Table 10. Fault Current For Parallel Lines With Both Generations

Type of Fault	**1LG**	**LL**	**2LG**	**3LG**

Current (A)	2765.15	3022.73	3303.03	3469.70
Current (pu)	2886.36	3007.58	3303.03	3462.12

CHAPTER 4: ARTIFICIAL NEURAL NETWORK

The following chapter will explain artificial neural networks with configuration with different kinds of methods along with comparison between and how it can be implemented.

4.1 MACHINE LEARNING

Machine learning refers to the process by which a computer program gains knowledge from experience (E) to execute a particular task (T), thereby enhancing its performance measure (P) over time. In essence, it seeks to create mathematical models using historical data to forecast future results.

The core concept of machine learning is to leverage historical data to train models that can generate accurate predictions or make informed decisions when presented with new, unfamiliar data. Figure 16 illustrates the essence of this process, historical data is utilized to train a model, and once the model is trained, it is employed to predict or infer future outcomes based on new input data [36].

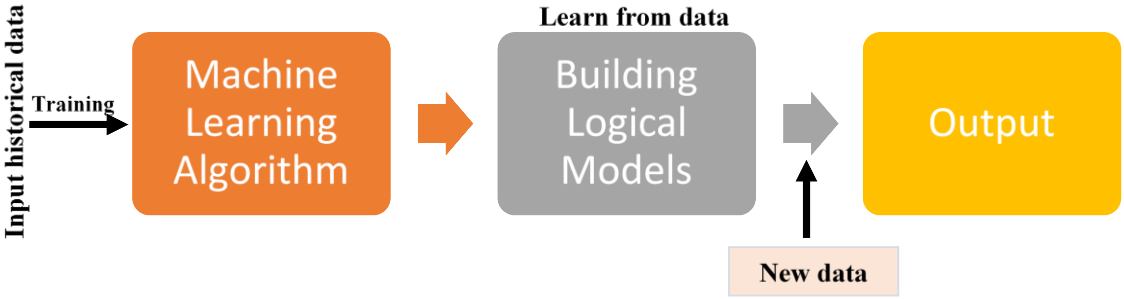


Figure 16. Machine learning procedures

In Figure 16, the left side represents the historical data used for training the machine learning model. Through algorithms and learning processes, this data is analyzed, and patterns are extracted to create a mathematical model capable of making predictions. After the model has been trained, it can be utilized on new or unseen data to forecast future outcomes or produce valuable insights.

Machine learning algorithms come in various forms, each suited for different types of tasks such as classification, regression, clustering, and more. By leveraging historical data, machine learning aims to uncover relationships, patterns, and trends that enable the prediction of future outcomes, thereby assisting in decision-making processes across diverse domains [36].

4.2 CLASSIFICATION OF MACHINE LEARNING

ML can be classified into various types based on different criteria, such as the learning style, the type of the problem being addressed, and the approach used for learning and making predictions. Here is some common classification machine learning based on types:

4.2.1 Supervised Learning

In supervised learning, the algorithm is provided with a labeled dataset, where each input is associated with a corresponding output label. The model uses this data to learn patterns and relationships, enabling it to make accurate predictions or decisions when confronted with new, unseen data. This learning approach is typically applied to tasks such as classification, where the goal is to predict categorical outcomes, and regression, which involves predicting continuous values. Figure 17 explains the process of supervised learning. Supervised learning can be divided into two subcategories[37]:

1- Regression

2- Classification

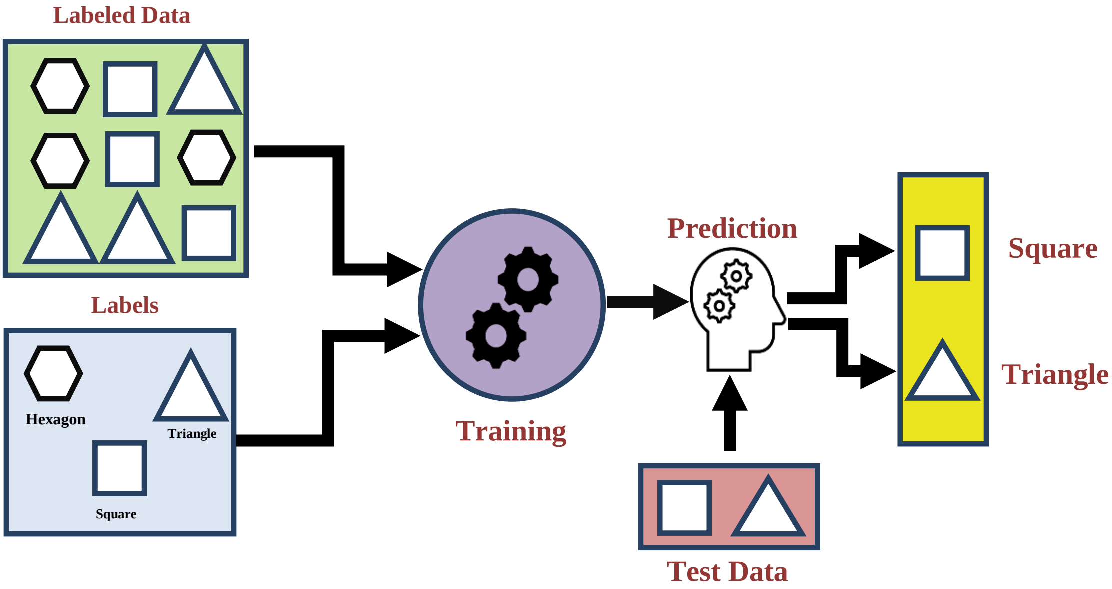


Figure 17. Process of supervised learning

4.2.2 Unsupervised Learning

Unsupervised learning works with unlabeled data, where the algorithm seeks to identify patterns, structures, or relationships within the data without predefined labels. It focuses on discovering inherent groupings or insights from the data. Common tasks in unsupervised learning include clustering, which involves grouping similar data points, and dimensionality reduction, where the number of features is reduced while preserving the most relevant information. Figure 18 describes the unsupervised learning. Unsupervised learning can be divided in two subcategories[37]:

1- Clustering

2- Association

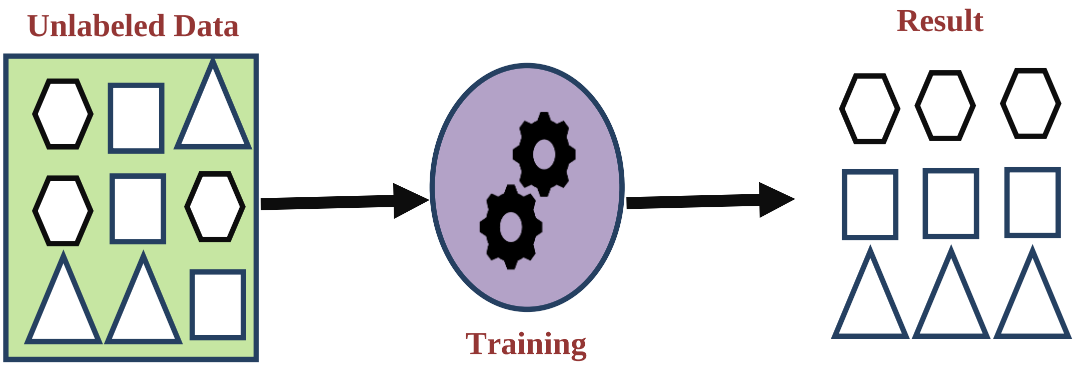


Figure 18. Process of unsupervised learning

4.2.3 Reinforcement Learning

Reinforcement learning involves an agent that learns to make decisions by interacting with its environment. Through trial and error, the agent receives feedback in the form of rewards or penalties based on its actions. This feedback guides the agent in adjusting its strategies to maximize cumulative rewards over time, allowing it to improve performance as it explores different actions and outcomes. Figure 19 give an explanation of the process of reinforcement learning[37].

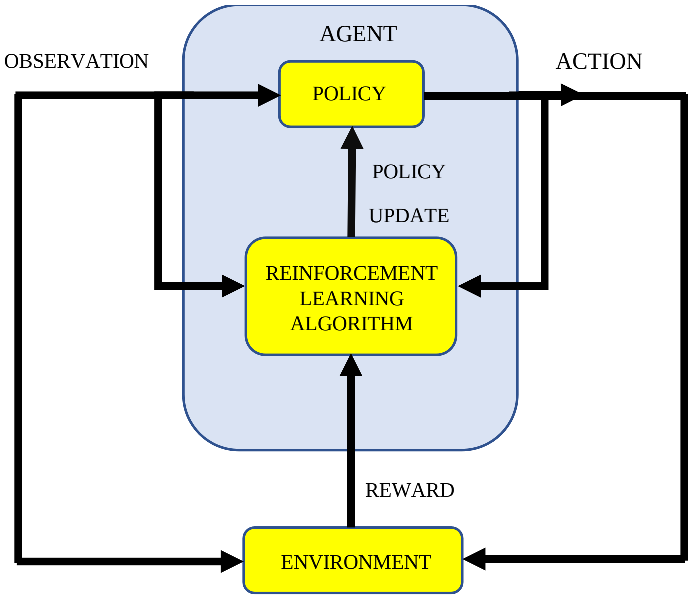


Figure 19. Process of reinforcement learning

These classifications are not mutually exclusive, and some machine learning approaches can combine elements from multiple types. The selection of the appropriate learning type—whether supervised, unsupervised, or reinforcement learning—depends on several factors, including the nature of the problem, the availability of labeled data, the complexity of the task, and the specific outcomes desired. Each approach offers unique advantages, making it crucial to align the learning type with the problem's requirements to achieve optimal results.

Before exploring machine learning in depth, it is essential to understand the foundational processes that drive it. These processes form the backbone of how machine learning operates, and a solid grasp of them is necessary before diving into the various techniques used within the field. The following sections offer an introductory overview, outlining the basic principles and functionalities of these processes.

4.3 DATA PREPROCESSING

Data preprocessing is an essential step prior to training a model with historical data to assist algorithms in achieving rapid convergence. Numerous methods exist for data preprocessing, which vary depending on the type of problem and available dataset. One prevalent technique involves normalization, which transforms the data into a scale typically between 0 to 1. This normalization process aims to standardize the data. Below are some common methods used for normalization[38]:

4.3.1 Min-max Normalization:

In this method, the below equation (4.3.1.1) is used to perform normalization for each vector (X_1, X_2, X_3, …., X_n) of inputs.

$$X_{1Norm} = \frac{X_1 - X_{1min}}{X_{1max} - X_{1min}} \qquad (4.3.1.1)$$

Where:

- X_{1min} is the minimum value to X_1 dataset.
- X_{1max} is the maximum value to X_1 dataset.

4.3.2 Standardization

In this approach, our aim is to adjust the variables by both shifting and scaling them to conform to a normal distribution characterized by a mean of zero and a standard deviation of one. Accordingly, equation (4.3.1.2) can be applied in this context.

$$X_{1Norm} = \frac{X_1 - \mu}{\sigma} \qquad (4.3.1.3)$$

Where:

- μ is the mean of $x_{1.}$
- σ is the standard deviation of $x_{1.}$

Following that, the developer needs to engage in data wrangling, addressing issues such as missing data and outliers, either by eliminating them or modifying them through estimations or by using mean values. Occasionally, it becomes essential to modify the dataset by employing variable transformations to yield improved outcomes. Table 11 provides an overview of the prevalent variable transformation techniques commonly utilized in machine learning, along with illustrative examples[38].

Table 11. Variable Transformation Types

Transformation Name	Example
Functional modification of variable	X➞X_2 or Sin(X)
Combination of variables	Width and Length ➞Area

Categorical to integer	red→1 yellow→2 Green→3
Complex physical models	X→F(X) or X→F(X)'

4.4 COST OR LOSS OF FUNCTION

A cost function, often referred to as a loss function or objective function, evaluates how far off the model's predictions are from the actual target values. It serves as a measure of the model's performance on the dataset. During training, the goal is to minimize the cost function by optimizing the model's parameters, leading to more accurate predictions.

For instance, in Table 12 the linear regression, a common cost function is the Mean Squared Error (MSE) and Mean Absolute Error (MAE), given by[38]:

Table 12. Types Of Regression Cost Function

Cost Function Name	Formula
Mean Squared Error (MSE)	$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i\hat{y}_i)^2$ Where: • y_i is actual output. • $\hat{y}_i$ is estimated output.
Mean Absolute Error (MAE)	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$ Where: • y_i is actual output.

	• $\hat{y}_i$ is estimated output.

In classification problems, a common cost function for binary classification using logistic regression is the Binary Cross-Entropy (log loss), given in Table 13.

Table 13. Binary Classificaion And Multi Class Cost Function

Cost Function Name	Formula
Cross-Entropy For Binary Classification	$h = \frac{1}{N}\sum_{i=1}^{N}[-y_i \log \sigma_i - (1 - y_i) \log(1 - \sigma_i)]$ Where: • y_i is actual output. • σ_i is Softmax probability distribution of class w_{ith}.
Cross-Entropy For Multiclass Classification	$h = -\sum_{j=0}^{M}\sum_{i=0}^{N}[y_{ij} \log a_i]$ Where: • y_i is actual output. • σ_i is Softmax probability distribution of class w_{ith}.

4.5 GRADIENT DESCENT

Gradient Descent is a widely used optimization technique in machine learning aimed at minimizing the cost or loss function by iteratively updating the model's parameters. It works by moving in the direction of the steepest negative gradient of the cost function to gradually approach the optimal parameter values. For this algorithm to function effectively, the cost function must be differentiable and ideally convex, ensuring that the gradient points toward the global minimum. Below Figure 20 illustrates the convex concept [38].

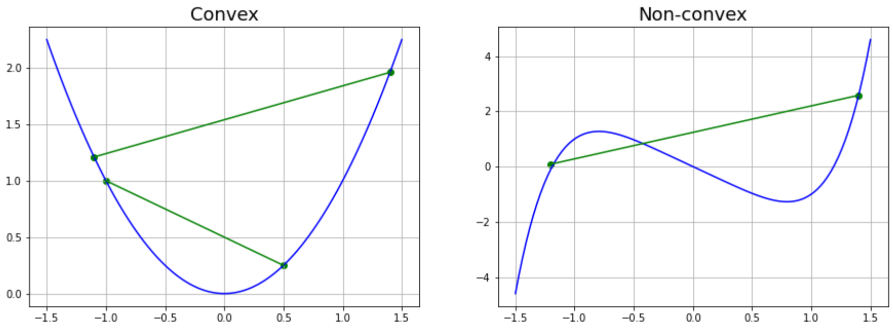


Figure 20. Convex and non-convex function [38]

Here's a simplified explanation of the Gradient Descent algorithm:

1- Initialization:

Start by initializing the model's parameters (weights and biases) with some random values.

2- Compute Gradient:

Determine the gradient of the cost function for each parameter. The gradient reveals both the direction and the size of the steepest rise in the cost function.

The gradient is computed using techniques such as partial derivatives or automatic differentiation, as shown in equation (4.5.1)

$$\nabla f(p) = \begin{bmatrix} \frac{df}{\partial x_1}(P) \\ \vdots \\ \frac{df}{\partial x_n}(P) \end{bmatrix} \qquad (4.5.1)$$

3- Update Parameters:

   - Adjust the parameters in the direction opposite to the gradient to minimize the cost function.
   - The update rule for each parameter P in the iteration n is often represented as in equation (4.5.2):

$$P_{n+1}=P_n-\eta \nabla f(P_n) \qquad (4.5.2)$$

Where:

P_{n+1} is future value.

P_n is the current value.

η is the learning rate.

4- Iteration:

   - Repeat steps 2 and 3 until convergence or a specified number of iterations.
   - Convergence occurs when the change in the cost function becomes negligible, indicating that the parameters have reached an optimal or near-optimal solution.

4.6 ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks (ANNs) are composed of layers of interconnected neurons that process information for tasks like classification, regression, and pattern recognition. Key elements in an ANN's configuration include the number of layers, neurons per layer, activation functions, optimizer, and loss function. Figure 21 illustrates an ANN showing how its connection [38].

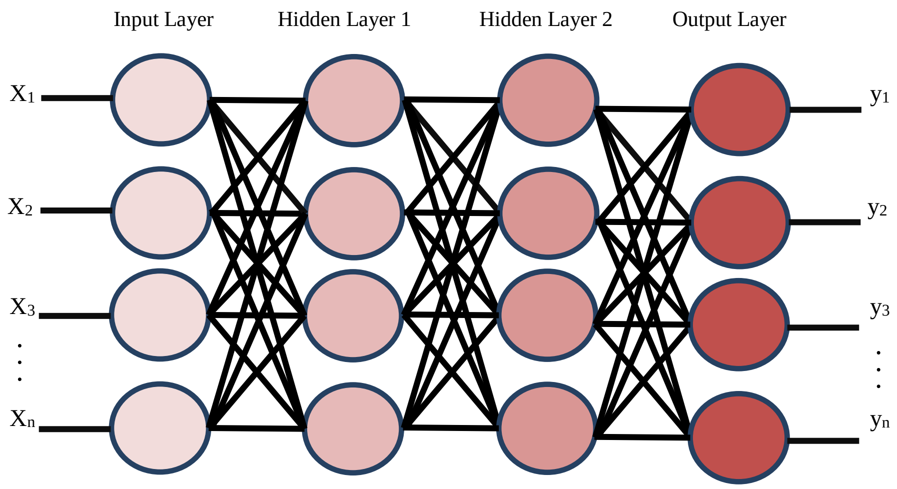


Figure 21. Artificial neural network (ANN)

Here's a basic outline of configuring an ANN:

1. Input Layer:

   The neurons in this layer correspond to the input features, with the quantity of neurons determined by the dimensionality of the input data.

2. Hidden Layers:

   - These layers are situated between the input and output layers. The number of hidden layers and the number of neurons within each layer are hyperparameters that require fine-tuning.

3. You can try different configurations, including the number of neurons in each layer, the total number of layers, and the types of layers used (such as dense, convolutional, or recurrent).

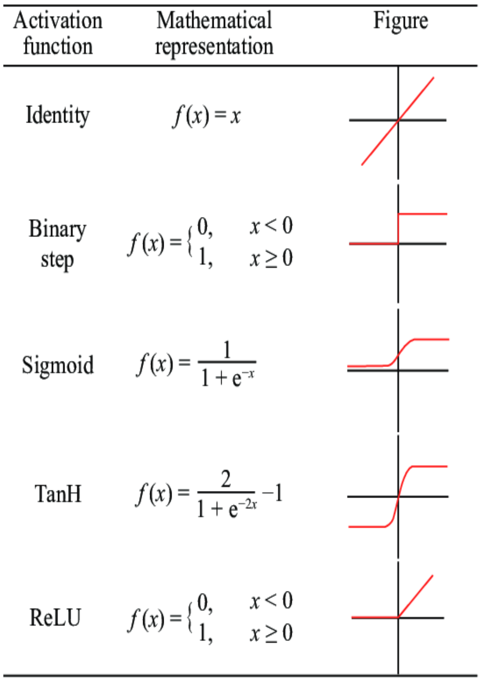
4. Output Layer:

   - The configuration of the output layer is determined by the specific task at hand. For example, in binary classification, a single neuron with a sigmoid activation function is typically employed. In contrast, for multi-class classification, the output layer consists of a number of neurons equal to the number of classes, usually utilizing a SoftMax activation function.

- For regression tasks, a single neuron without an activation function or using a linear activation function could be used.

5. Activation Functions:

Activation functions play a crucial role in introducing non-linearities to neural networks, which enables them to model complex relationships between inputs and outputs. By allowing the network to learn and approximate non-linear functions, these functions help capture intricate patterns within the data, making them essential for the effectiveness of deep learning models. The selection of an appropriate activation function is influenced by factors such as the specific problem domain, the architecture of the network, and the characteristics of the data being processed. Figure 22 shows different activation functions.



Activation function	Mathematical representation	Figure
Identity	$f(x) = x$	
Binary step	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	
TanH	$f(x) = \frac{2}{1 + e^{-2x}} - 1$	
ReLU	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	

Figure 22. Types of activation functions used in anns [38]

6. Loss Function:

   These loss functions help to quantify the performance of the neural network during training by providing a measure of how well the predictions match the actual targets. The objective during training is to minimize the value of the chosen loss function, achieved by adjusting the model's parameters through techniques like gradient descent, thus improving the model's predictive capabilities. The choice of the loss function depends on the specific task and the nature of the problem being addressed.

7. Regularization and Dropout:

   Regularization and dropout are two techniques commonly used in Artificial Neural Networks (ANNs) to prevent overfitting and improve the generalization of the model to unseen data. Will be discussed in more detail in the following sections.

4.7 REGULARIZATION

Regularization methods such as L1 and L2 regularization aim to prevent overfitting by adding a penalty term to the loss function that depends on the magnitude of the weights in the neural network.

L1 Regularization: Adds a penalty proportional to the absolute value of the weights: $\lambda \sum |W|$

.

L2 Regularization (Weight Decay): Adds a penalty proportional to the squared magnitude of the weights: $\lambda \sum W^2$

.

Here, λ is the regularization parameter that controls the strength of regularization. Increasing λ increases the penalty on large weights, encouraging the model to use smaller weights and prevent overfitting[38].

4.8 DROPOUT

Dropout is a regularization method that randomly disables a portion of neurons in a neural network during each training iteration. By setting a specific percentage of neurons to zero at random (based on the dropout rate), this technique helps prevent the network from becoming overly dependent on certain neurons or from learning noise within the training data. As a result, dropout encourages the network to learn more robust and generalized features. During inference or testing, all neurons remain active, but their outputs are adjusted according to the dropout rate to compensate for the effect of dropout during training.

Both regularization techniques and dropout aim to prevent the network from memorizing the training data, enhancing its ability to generalize to new, unseen data. By limiting the model's capacity to fit noise and unnecessary details in the training dataset, these strategies contribute to creating models with improved performance and reliability on unseen data. While regularization methods like L1 and L2 act directly on the weights, dropout introduces an element of randomness during training, making the network more resilient and less susceptible to overfitting. Thus, it is clear that various configurations significantly impact the performance of an Artificial Neural Network (ANN) and should be carefully selected before beginning the training process. Figure 23 presented a typical neuron cell in an ANN [38].

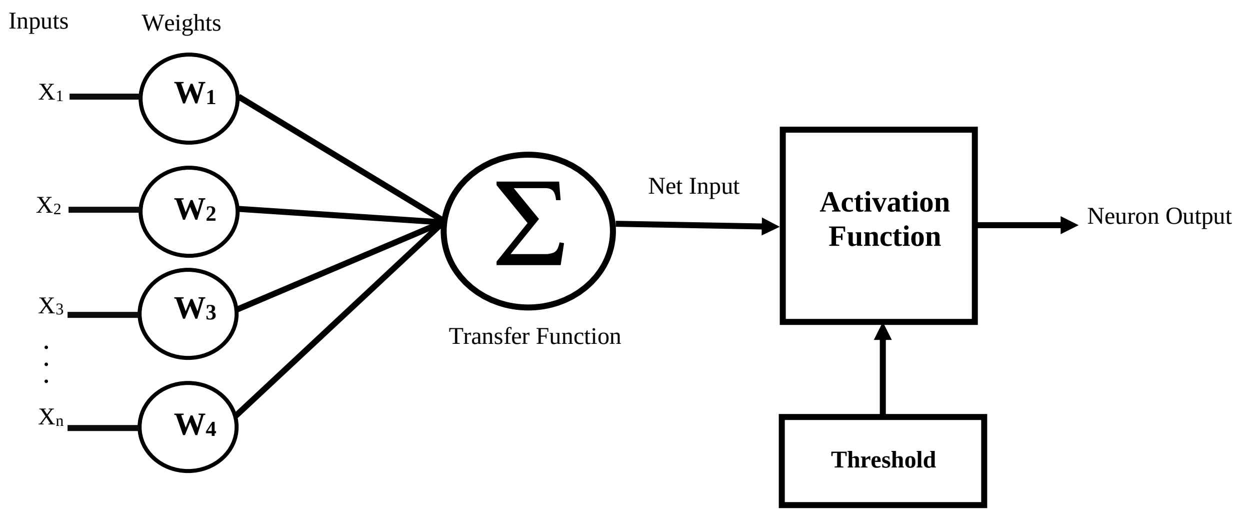


Figure 23. Typical neuron in ANN

4.9 TRAINING ALGORITHM

Multiple techniques are employed to train an Artificial Neural Network (ANN) to achieve varying output performance levels. This study will utilize two specific techniques, with their outputs being thoroughly examined and discussed in the final chapter. The chosen training algorithms are:

1- The Levenberg-Marquardt algorithm.

2- The Scaled Conjugate Gradient algorithm.

4.9.1 Levenberg Marquardt Algorithm

The Levenberg-Marquardt algorithm is an optimization method commonly used for training artificial neural networks. It combines elements of both gradient descent and Gauss-Newton methods to efficiently minimize the error between the network's predictions and the actual target values. The algorithm is particularly effective for nonlinear least squares problems, such as those encountered in training neural networks. The basic idea behind the Levenberg-Marquardt algorithm is to adjust the weights of the neural network in a way that reduces the error between the predicted outputs and the actual targets. This adjustment is done iteratively, with each iteration aiming to improve the network's performance[39].

The algorithm updates the weights of the network using the following equation (4.9.1.1).

$$\Delta w=(J^T J+\lambda I)^{-1} J^T e \qquad (4.9.1.1)$$

where:

- Δw is the update vector for the weights of the network.
- J is the Jacobian matrix, which contains partial derivatives of the network's outputs with respect to its weights.
- e is the error vector, representing the difference between the network's predicted outputs and the actual target values.
- λ is a regularization parameter that controls the step size of the update. It is adjusted dynamically during the training process to balance between the gradient descent and Gauss-Newton steps. This parameter ensures that the algorithm converges smoothly and avoids overshooting the minimum.
- I is the identity matrix.

The Levenberg-Marquardt algorithm starts with an initial guess for the weights of the network. It then iteratively updates the weights using the above equation until the error converges to a minimum or reaches a predefined threshold.

By incorporating both gradient descent and Gauss-Newton methods, the Levenberg-Marquardt algorithm can effectively handle various types of nonlinearities and converge to a good solution more quickly than traditional gradient descent methods alone[39].

4.9.2 Scaled Conjugate Gradient Algorithm

The Scaled Conjugate Gradient (SCG) algorithm is another optimization technique commonly used for training artificial neural networks. It belongs to the family of conjugate gradient methods and is particularly efficient for problems with large parameter spaces, such as neural network training. The SCG algorithm aims to iteratively adjust the weights of the neural network to minimize the error between the predicted outputs and the actual target values. It combines the concepts of conjugate gradients with a dynamically scaled step size to achieve faster convergence. The main update equation (4.9.2.1) in the SCG algorithm is[40]:

$$\Delta W = \frac{S}{||g||} \qquad (4.9.1.2)$$

Where:

- Δ_w is the update vector for the weights of the network.
- S is the step size, which is dynamically adjusted to ensure efficient progress towards the minimum error. It is computed based on the previous and current gradients.
- g is the gradient vector, representing the gradient of the error function with respect to the weights of the network.

The SCG algorithm also utilizes information from previous iterations to determine the direction of the update. It ensures that the search direction is conjugated to the previous search directions, which helps avoid oscillations and ensures smooth convergence. Additionally, step size S is adaptively scaled to balance between the magnitude of the update and the curvature of the error surface. This scaling ensures that larger steps are taken in regions where the error surface is relatively flat, while smaller steps are taken in regions with steeper gradients. The SCG algorithm continues to update the weights iteratively until the error converges to a minimum or reaches a predefined threshold. Overall, the Scaled Conjugate Gradient algorithm offers a computationally efficient and effective approach to training neural networks by dynamically adjusting step sizes and utilizing conjugate search directions to achieve faster convergence[40].

4.10 PERFORMANCE

Performance parameters of Artificial Neural Networks (ANNs) are essential metrics utilized to assess the effectiveness of these models in various tasks. These parameters offer valuable insights into the network's accuracy, efficiency, and generalization capabilities, aiding in the evaluation and optimization of neural network architectures and training procedures. Here's a more detailed exploration of some common performance parameters of ANNs:

4.10.1 Accuracy

Accuracy is a key performance metric for assessing the effectiveness of classification models, including Artificial Neural Networks (ANNs). It quantifies the ratio of correctly classified instances to the total number of instances within the dataset. Here is a more detailed explanation of how accuracy is calculated in the context of

ANNs as in equation (4.10.1.1).

$$Accuracy(\%)=\frac{Total\ Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}\times 100 \qquad (4.10.1.1)$$

Accuracy serves as a clear indicator of a model's overall correctness in its predictions. Nonetheless, it's crucial to take into account the class distribution and dataset characteristics when evaluating accuracy, particularly in cases of imbalanced datasets where one class may overshadow others. In such instances, relying solely on accuracy can lead to an incomplete assessment of the model's performance; hence, it is important to also consider additional metrics like precision, recall, and F1 Score.

In conclusion, accuracy is an important metric for assessing the performance of classification models, including ANNs, as it reflects the proportion of correctly classified instances and offers a straightforward measure of effectiveness. However, it should be evaluated alongside other metrics, especially in situations involving imbalanced class distributions.

4.10.2 Precision and Recall

Precision quantifies the fraction of true positive predictions relative to the total number of positive predictions made by the model, whereas recall assesses the proportion of true positive predictions out of all actual positive instances present in the dataset. These metrics are especially helpful for imbalanced datasets. They are calculated as in equation (4.10.2.1) and (4.10.2.2) [39].

$$Precision=\frac{True\ Positives}{True\ Positives+False\ Positives} \qquad (4.10.2.1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad (4.10.2.1)$$

Where:

- True Positives (TP): These are the cases that are correctly predicted as positive by the model.
- False Positives (FP): These are the cases that are incorrectly predicted as positive by the model but are negative.
- False Negatives (FN): These are the cases that are incorrectly predicted as negative by the model but are positive.
- True Negatives (TN): These are the cases that are correctly predicted as negative by the model.

4.10.3 F1 Score

The F1 Score is a widely utilized metric in classification tasks, including those involving Artificial Neural Networks (ANNs). It balances precision and recall, offering a single value that reflects the model's overall performance. The F1 Score is calculated as the harmonic mean of precision and recall. It balances both metrics and is especially useful when you want to avoid overemphasizing either Precision or Recall: Here's how to calculate the F1 Score in more detail as in equation (4.10.3.1).

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (4.10.3.1)$$

The F1 Score starts from 0 to 1, where 1 indicates perfect precision and recall, while 0 indicates poor performance. It is a single metric that provides a comprehensive evaluation of the model's ability to classify instances correctly, considering both false positives and false negatives. In summary, the F1 Score is calculated based on the precision and recall of the model's predictions, providing a balanced measure of its performance in classification tasks[39].

4.10.4 Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) serves as a crucial evaluation metric in the context of Artificial Neural Networks (ANNs), particularly in forecasting and predictive modeling tasks. MAPE offers a measurable assessment of the accurateness and consistency of predictions generated by ANNs, aiding analysts in gauging the effectiveness of these models in capturing underlying patterns and making accurate forecasts. The calculation of MAPE within the framework of ANNs involves comparing the predicted values generated by the neural network with the corresponding actual values from the dataset.

The formula for calculating MAPE within the context of ANNs is as in equation (4.10.4.1).

$$MAPE(\%)=\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i-\hat{y}_i|}{|y_i|}x100 \qquad (4.10.4.1)$$

Where:

- Actual Target Values (y_i): These are the true values of the target variable in your dataset. Where i represents each individual sample in the dataset.
- Predicted Values ($\hat{y}_i$): These are the values predicted by your ANN for each corresponding input sample.

- Number of Samples (n): The total number of observations in the dataset.

The application of MAPE within ANNs offers several advantages. Firstly, it provides a standardized measure of forecasting accuracy, enabling analysts to objectively evaluate the performance of neural network models across different datasets and forecasting horizons. Additionally, MAPE offers a clear and intuitive interpretation, as it represents the average percentage deviation between predicted and actual values. This simplicity facilitates communication and decision-making, allowing stakeholders to easily assess the reliability of forecasts generated by ANNs.

However, it is important to note that MAPE also has limitations, particularly in cases where actual values are close to or equal to zero. In such scenarios, the percentage errors can become extremely large or undefined, leading to inflated MAPE values. Analysts should be mindful of these limitations and consider other error metrics, such as Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE), when dealing with datasets containing zero or near-zero values. Overall, MAPE serves as a valuable tool in assessing the accuracy of predictions generated by ANNs, providing insights into their effectiveness in capturing underlying patterns and making reliable forecasts[39].

4.10.5 Mean Squared Error (MSE)

Mean Squared Error (MSE) is a commonly used metric for evaluating the performance of regression models, including Artificial Neural Networks (ANNs). It measures the average squared difference between the actual target values and the predicted values generated by the model. Here's a more detailed explanation of how to calculate MSE in the context of ANNs as in equation (4.10.5.1).

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i\text{-}\widehat{y}_i)^2 \qquad (4.10.5.1)$$

Where:

- Actual Target Values(y_i)**:** These are the true values of the target variable in your dataset. Where **a** represents each individual sample in the dataset.
- Predicted Values $(\widehat{y}_i)$: These are the values predicted by your ANN for each corresponding input sample.
- Number of Samples (n): The total number of observations in the dataset.

The MSE quantifies the average squared difference between the actual and predicted values. A lower MSE specifies that the model's predictions are closer to the actual values, while a higher MSE specifies larger deviations between predictions and actuals. MSE values closer to zero indicate better model performance. Larger MSE values indicate poorer model performance, with larger prediction errors. It is important to compare MSE values across different models or iterations to assess improvements or regressions in performance.

In summary, the Mean Squared Error is a fundamental metric used in assessing the performance of regression models, providing a quantitative measure of how well the model's predictions align with the actual target values.

RMSE is the square root of the MSE and provides a measure of the standard deviation of the errors using equation (4.10.5.2)[39].

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i\text{-}\widehat{y}_i)^2} \qquad (4.10.5.2)$$

4.10.6 Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) is a key evaluation metric used in Artificial Neural Networks (ANNs) to measure the accuracy of model predictions. It provides a clear and interpretable assessment of the average magnitude of errors between predicted and actual values, making it a useful tool for evaluating ANN performance in various tasks, such as regression and forecasting. To calculate MAE, one computes the absolute difference between the predicted and actual values for each observation and then averages these absolute errors across all observations. The formula for calculating MAE within ANNs is as follows shown in (4.10.6.1).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}| \qquad (4.10.6.1)$$

Where:

- Actual Target Values (y_i)**:** These are the true values of the target variable in your dataset. Where i represents each individual sample in the dataset.
- Predicted Values $(\widehat{y_i})$: These are the values predicted by your ANN for each corresponding input sample.
- Number of Samples (n): The total number of observations in the dataset.

The Mean Absolute Error (MAE) calculation provides the average absolute difference between predicted and actual values, serving as a clear and intuitive measure of prediction accuracy. Unlike other metrics like Mean Squared Error (MSE), which disproportionately penalize larger errors, MAE is more robust to outliers, making it particularly useful in situations where the magnitude of errors is a primary concern. One of the main benefits of MAE is its simplicity and ease of interpretation; the resulting value reflects the average absolute deviation, allowing analysts and stakeholders to easily evaluate the accuracy and reliability of predictions made by ANNs. Moreover, MAE is scale-independent, meaning it is unaffected by the units of the data, which allows for the comparison of ANN performance across different datasets and applications.

However, while MAE is advantageous, it may not fully capture the variability in prediction errors, especially when larger errors are infrequent. In such cases, other metrics like Mean Squared Error (MSE) might offer a more comprehensive assessment of prediction accuracy. Despite this limitation, MAE remains a widely used and valuable metric for evaluating ANN models, providing a straightforward measure of prediction accuracy that is easily interpretable and applicable across a variety of domains and tasks[39].

4.10.7 R-Squared (R^2)

In the context of Artificial Neural Networks (ANNs), R^2 R-squared) is a statistical measure commonly used to evaluate the goodness of fit of a regression model. It assesses how well the model explains the variance in the dependent variable.

$$R^2=1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\tilde{y}_i)^2} \qquad (4.10.7.1)$$

Where:

- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.
- $\tilde{y}_i$ is the mean of the actual values.

It ranges from 0 to 1. A value of 1 specifies that the model perfectly explains the variance in the dependent variable, while a value of 0 indicates that the model does not explain any of the variance. A higher R^2 value indicates that the model's predictions closely match the actual target values, suggesting a better fit. A lower R^2 value suggests that the model does not explain much of the variance in the dependent variable.

In summary, R^2 is a valuable metric for assessing the predictive power of regression models, including ANNs. It quantifies how well the model explains the variance in the dependent variable and provides insights into the model's overall performance. These performance parameters collectively provide a comprehensive evaluation of the ANN's performance across various tasks, including classification and regression. By considering these metrics, practitioners can gain valuable insights into the strengths and weaknesses of their neural network models, facilitating informed decision-making and further optimization[39].

CHAPTER 5: RESULTS

In this chapter, The outcomes derived from the simulation of trained feed-forward Artificial Neural Networks (ANNs) are analyzed, setting the stage for a comprehensive discussion of their performance in the subsequent chapter. Throughout the testing phase of these ANNs, a variety of testing models are employed, each meticulously configured to examine fault locations ranging from 0.2 Kilometers to 9.8 Kilometers. Across this spectrum, all 10 fault types are systematically applied to each model, ensuring a robust assessment of their efficacy, with the results meticulously recorded for further analysis.

It is worth noting that the testing process involves two distinct types of ANNs: one utilizing the Levenberg-Marquardt (LM) algorithm, while the other harnesses the Scaled-Conjugate-Gradient (SCG) algorithm. This deliberate differentiation allows for a comparative evaluation of their respective performance characteristics. The testing phase represents a crucial stage in the evaluation of the trained ANNs, as it serves to validate their predictive capabilities under diverse fault conditions. By systematically varying the fault locations and types, we aim to assess the ANNs' adaptability and generalization capabilities across a broad spectrum of real-world scenarios. This comprehensive approach not only enhances the reliability of our findings but also offers valuable insights into the ANNs' potential utility in practical applications. Each testing models depicted in Figure 3 to Figure 6 are meticulously designed to replicate real-world conditions, with fault locations spanning a wide range to capture the full spectrum of potential scenarios encountered in practical settings. By subjecting the ANNs to these varied conditions, we aim to gauge their performance across different fault severities and locations, thereby enabling a more assessment of their effectiveness. Furthermore, the utilization of both LM and SCG algorithms enables us to explore the impact of algorithmic choice on the ANNs' performance metrics. While LM is renowned for its robustness and efficiency in optimizing network parameters, SCG offers distinct advantages in terms of computational efficiency and scalability. By comparing the performance of ANNs trained with these two algorithms, valuable insights can be gained into their relative strengths and weaknesses, thus informing future algorithm selection and optimization strategies.

In summary, this chapter lays the groundwork for a comprehensive analysis of the trained ANNs' performance, providing insights into their predictive capabilities under varying fault conditions. Through meticulous testing and evaluation, aim to elucidate the factors influencing their performance and identify avenues for further improvement.

5.1 GENERATING DATA SET

To effectively train Artificial Neural Networks (ANNs) for the identification, classification, and estimation of faults within power systems, it is crucial to generate appropriate input and target data using models that accurately represent real-world power system scenarios. Figure 24 shows how the ANN will be connecting to the real systems. As outlined in the preceding chapter, four distinct systems will be leverage to construct our training dataset, each serving as a distinct case study:

1- System 1: A single line configuration with a generation source located at the sending end.

2- System 2: Parallel lines configuration with a single generation source at the sending end.

3- System 3: A single line configuration with generation sources located at both the sending and receiving ends.

4- System 4: Parallel lines configuration with generation sources situated at both the sending and receiving ends.

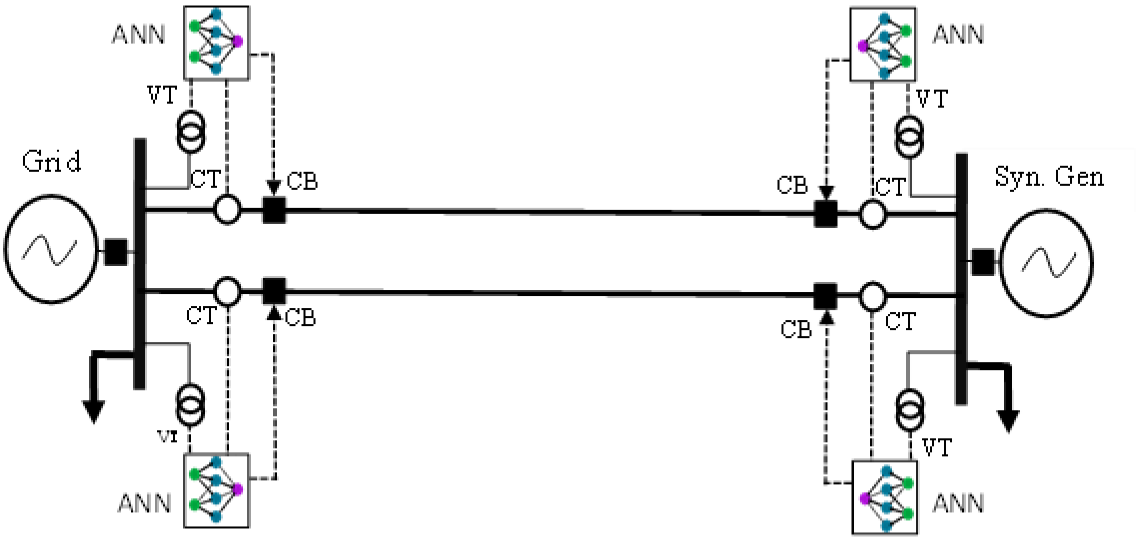


Figure 24. ANNs replacing protection relays in real system

In each scenario, two different systems will be used for data collection. The first system collects data solely from the sending end, while the second system gathers data from both the sending and receiving ends.

Prior to commencing the generation of the training dataset, it is essential to encode the target data into binary bits, adhering to the following format in Figure 25:

Identification	Classification				Location
Fault / No Fault	Red Phase Association	Yellow Phase Association	Blue Phase Association	Earth Association	Fault Distance
Binary Number	Binary Number	Binary Number	Binary Number	Binary Number	Decimal Number
0	0	0	0	0	0.0

Figure 25. Encode the target data into binary bits

The target dataset comprises 6 bits, each serving a distinct purpose in fault identification, classification, and location. The initial bit functions as a binary indicator, signaling the presence of a fault within the system or confirming its healthy status. Following this, the subsequent four bits, also binary, serve to classify the faulty phase and determine the involvement of the earth in the fault occurrence. Lastly, the final bit assumes a decimal form, representing the distance from the measuring point to the location of the fault. This comprehensive representation facilitates precise fault analysis and localization within the power system.

Now, the selection of inputs for the neural network must be meticulous to ensure that the network can effectively discern all the various purposes encoded in the target data. These inputs have been carefully chosen to encapsulate diverse aspects of the power system's behavior and fault characteristics, enabling the neural network to differentiate between different fault scenarios with precision and reliability. Furthermore, by leveraging a rich set of inputs, the neural network can learn to generalize well across different scenarios and adapt to varying system conditions, enhancing its robustness and efficacy in fault detection and localization tasks. Across all cases, a set of 8 inputs will be utilized to provide comprehensive information for accurate fault identification, classification, and localization as follows:

1- I_S is the RMS current of three-phase measured from the sending end.

2- V_S is the RMS voltage of three-phase measured at the sending end.

3- I_{ZS} is the zero-sequence current measured from the sending end.

4- V_{zS} is the zero-sequence voltage measured at the sending end.

5- I_r is the RMS current of three-phase measured from the receiving end.

6- V_r is the RMS voltage of three-phases measured at the receiving end.

7- I_{zr} is the zero-sequence current measured from the receiving end.

8- V_{zr} is the zero-sequence voltage measured at the receiving end.

Now that the system has been constructed and both inputs and target data have been finalized, the next step is to execute the simulation process, which involves simulating all 12 faults occurring at various locations within the system. MATLAB Simulink serves as the primary tool for conducting these simulations, facilitating the generation of data representing fault scenarios.

Once the simulation is initiated, the generated data is transmitted to the workplace where further processing occurs. Here, the data undergoes various preparatory steps to ensure its suitability for training the Artificial Neural Networks (ANNs). While the simulation process may require significant computational resources and time investment, its importance cannot be overstated. By meticulously examining fault scenarios, the system gains insights into diverse fault patterns and behaviors, thereby enhancing its capability to handle unforeseen circumstances and improve overall performance.

Once the data is processed and refined, it is ready to be utilized to train the ANN. This marks a critical milestone in the development of the fault detection and classification system, as it empowers the ANNs to learn from a diverse range of fault scenarios and effectively distinguish between different fault types and locations.

In the training models, a cable length of 10 kilometers is employed. Consequently, faults are initiated at a starting point of 0.1 kilometers along the cable, gradually increasing the fault distance by increments of 0.1 kilometers until reaching the cable's end. This systematic approach ensures comprehensive coverage of fault scenarios across the entire length of the cable, enabling the training models to effectively learn and adapt to various fault conditions and distances. This results in a total of 100 scenarios for each case. Within each scenario, 100 samples are collected for every fault, leading to a substantial dataset. Consequently, the total training set from each case comprises 120,000 samples, reflecting the extensive scope and granularity of the data collected for training the models.

For training the ANNs, the Neural Fitting Tool (nftool) in Matlab is used where inputs and target data are selected as following percentage stated in Table 14:

Table 14. The Percentage Of Training, Validation And Testing Sets

Set Name	Percentage	Number of Samples
Training Set	60%	72,000 Samples
Validation Set	20%	24,000 Samples
Testing Set	20%	24,000 Samples

The ANNs used in the research consist of three layers (input, hidden, and output). The input layer has 8 neurons for system has input from sending end only and 16 neurons for system has input from sending and receiving ends, the hidden layer has 50 neurons, and the output layer has 6 neurons only. Figure 26 and Figure 27 show the ANNs configuration used in this research.

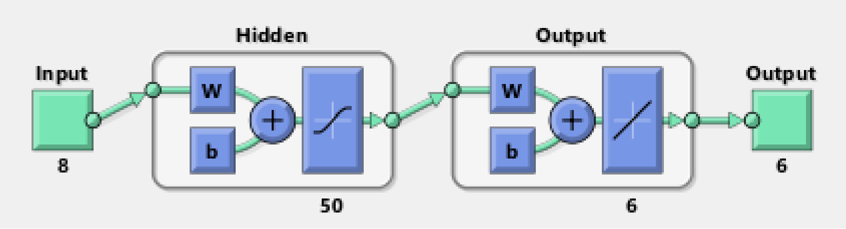


Figure 26. ANN configuration of 8 inputs system

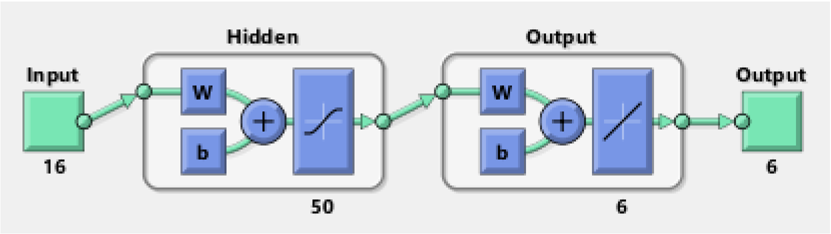


Figure 27. ANN configuration of 16 inputs system

To train the algorithm, two carefully selected training algorithms are employed to train the Artificial Neural Networks (ANNs) for each scenario, resulting in a total of 16 sub-cases shown in

Table 16 with their configurations. The LM and SCG backpropagation algorithms are employed for network training, with their corresponding configuration parameters shown below in Table 15.

Table 15. Configuration Parameters Of Training Algorithms

Configuration Parameters	LM	SCG
Maximum number of epochs to train	1000	1000
Performance goal	0	0
Maximum validation failures	2	2
Initial μ	0.001	N/A
μ decrease factor	0.1	N/A
μ increase factor	10	N/A
Maximum μ	$1x10^{10}$	N/A
σ_0	N/A	$5x10^{-5}$

Table 16. Cases And Their Corresponding Configurations

Case Number	System Configuration	Inputs	Training Algorithm
Case 1	A single line configuration with a generation source located at the sending end.	I_s,V_s, I_{z_s} and V_{zs}	LM
Case 2	A single line configuration with a generation source located at the sending end.	I_s,V_s, I_{z_s} and V_{zs}	SCG
Case 3	Parallel lines configuration with a single generation source at the sending end.	I_s,V_s, I_{z_s} and V_{zs}	LM
Case 4	Parallel lines configuration with a single generation source at the sending end.	I_s,V_s, I_{z_s} and V_{zs}	SCG
Case 5	A single line configuration with generation sources located at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}	LM
Case 6	A single line configuration with generation sources located at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}	SCG
Case 7	Parallel lines configuration with generation sources situated at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}	LM
Case 8	Parallel lines configuration with generation sources situated at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}	SCG
Case 9	A single line configuration with a generation source located at the sending end	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	LM
Case 10	A single line configuration with a generation source located at the sending end	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	SCG

Case 11	Parallel lines configuration with a single generation source at the sending end.	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	LM
Case 12	Parallel lines configuration with a single generation source at the sending end.	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	SCG
Case 13	A single line configuration with generation sources located at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	LM
Case 14	A single line configuration with generation sources located at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	SCG
Case 15	Parallel lines configuration with generation sources situated at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	LM
Case 16	Parallel lines configuration with generation sources situated at both the sending and receiving ends.	I_s,V_s, I_{z_s} and V_{zs}, I_r, V_r, I_{zr} and V_{zr}	SCG

Now, after training all ANNs are completed below are the results of the training in Table 17.

Table 17. Results Of The Training

	Training		Validation		Testing	
Case#	MSE	R^2	MSE	R^2	MSE	R^2
Case 1	0.1199	0.9855	0.1218	0.9854	0.1202	0.9854
Case 2	0.1202	0.9855	0.1249	0.9849	0.1220	0.9854
Case 3	0.1219	0.9854	0.1233	0.9850	0.1131	0.9863
Case 4	0.1220	0.9853	0.1190	0.9857	0.1193	0.9855
Case 5	0.1199	0.9855	0.1218	0.9854	0.1204	0.9854
Case 6	0.1216	0.9853	0.1213	0.9854	0.1222	0.9825
Case 7	0.1199	0.9855	0.1218	0.9854	0.1204	0.9854
Case 8	0.1207	0.9855	0.1254	0.9848	0.1225	0.9853
Case 9	0.1203	0.9855	0.1218	0.9854	0.1194	0.9855
Case 10	0.1177	0.9858	0.1230	0.9851	0.1268	0.9848
Case 11	0.1175	0.9858	0.1228	0.9852	0.1267	0.9848
Case 12	0.1851	0.9857	0.1194	0.9856	0.1278	0.9846
Case 13	0.1203	0.9855	0.1218	0.9854	0.1194	0.9855
Case 14	0.1181	0.9857	0.1233	0.9851	0.1272	0.9848
Case 15	0.1203	0.9855	0.1218	0.9854	0.1193	0.9855
Case 16	0.1180	0.9858	0.1233	0.9851	0.1272	0.9848

The Mean Squared Error (MSE) represents the average squared difference between outputs and targets. Lower values indicate better performance, with zero indicating no error. Regression R values quantify the correlation between outputs and targets. An R value of 1 indicates a strong positive relationship, while 0 signifies a random relationship.

In Table 17, the training performance indicates that the lowest result is observed in case 11, characterized using parallel lines, a single generation, and the LM learning algorithm, with data obtained from the receiving end. Conversely, the best validation performance is evident in case 4, which also utilizes parallel lines and a single generation at the sending end but employs the SCG learning algorithm. As for testing performance, the optimal outcome is demonstrated in case 3, which shares similarities with case 4 by employing parallel lines and a single generation at the sending end, while employing the LM learning algorithm.

Comparing the performance values, LM learning outperforms SCG learning, albeit at a slower convergence rate in finalizing the training process. Additionally, utilizing receiving end data yields similar training performance values, albeit with increased processing time due to the higher number of inputs and corresponding input neurons.

5.2 TESTING PROCESS

Figure 28 illustrates the testing process of ANNs within the testing system, showcasing various simulations conducted to predict fault type, fault location, and identify faulty systems. The subsequent section will delve into the performance metrics associated with these predictions, offering a comprehensive analysis of the ANNs' efficacy in real-world fault detection scenarios.

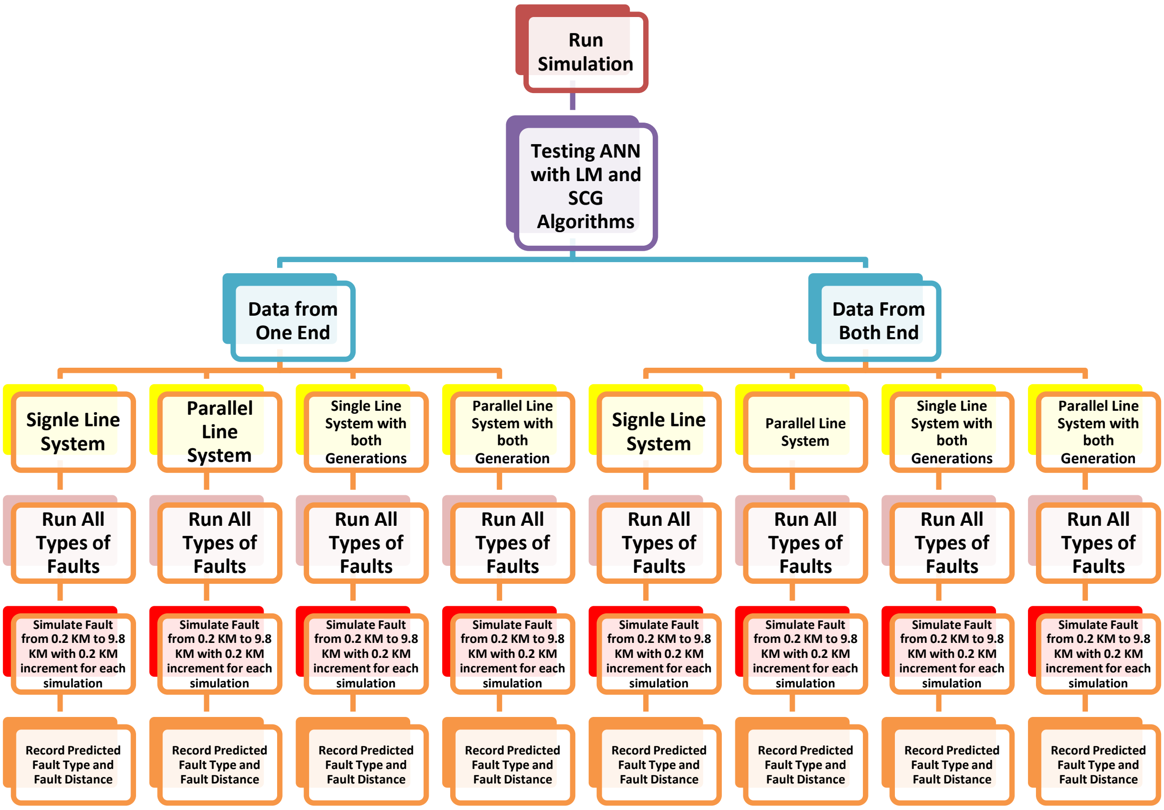

Run Simulation
Testing ANN with LM and SCG Algorithms
Data from One End
Data From Both End
Signle Line System
Parallel Line System
Single Line System with both Generations
Parallel Line System with both Generation
Signle Line System
Parallel Line System
Single Line System with both Generations
Parallel Line System with both Generation
Run All Types of Faults
Run All Types of Faults
Run All Types of Faults
Run All Types of Faults
Run All Types of Faults
Run All Types of Faults
Run All Types of Faults
Run All Types of Faults
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Simulate Fault from 0.2 KM to 9.8 KM with 0.2 KM increment for each simulation
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance
Record Predicted Fault Type and Fault Distance

Figure 28. The testing procedure of trained ANNs

5.3 TESTING RESULTS

5.3.1 Classification Results

The classification results center on the identification and categorization of outcomes derived from the testing phase. A total of 539 simulations were conducted to meticulously assess the accuracy and reliability of the ANNs across four distinct systems. These simulations were designed to scrutinize the ANNs' ability to accurately detect and classify fault types, fault locations, and faulty systems.

Upon completion of the testing phase, the performance results were meticulously compiled and summarized to provide a comprehensive overview of the ANNs' efficacy. These performance metrics offer valuable insights into the ANNs' predictive capabilities, shedding light on their strengths and areas for improvement.

The summary of performance results serves as a crucial reference point for evaluating the effectiveness of the ANNs in real-world fault detection scenarios. By quantifying metrics such as accuracy, precision, and recall, a deeper understanding will be gained of the ANNs' performance across different systems and fault conditions. This comprehensive analysis lays the foundation for further refinement and optimization of the ANNs, with the ultimate goal of enhancing their reliability and applicability in practical settings.

Table 18. The Confusion Matrix Of Case1

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 19. Performance Parameter For Classification Results Of Case 1

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 20. The Confusion Matrix Of Case 2

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 21. Performance Parameter For Classification Results Of Case 2

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 22. The Confusion Matrix Of Case 3

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 23. Performance Parameter For Classification Results Of Case 3

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 24. The Confusion Matrix Of Case 4

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
Predicted	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 25. Performance Parameter For Classification Results Of Case 4

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 26. The Confusion Matrix Of Case 5

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 27. Performance Parameter For Classification Results Of Case 5

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 28. The Confusion Matrix Of Case 6

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 29. Performance Parameter For Classification Results Of Case 6

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 30. The Confusion Matrix Of Case 7

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 31. Performance Parameter For Classification Results Of Case 7

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 32. The Confusion Matrix Of Case 8

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 33. Performance Parameter For Classification Results Of Case 8

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 34. The Confusion Matrix Of Case 9

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 35. Performance Parameter For Classification Results Of Case 9

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 36. The Confusion Matrix Of Case 10

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 37. Performance Parameter For Classification Results Of Case 10

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 38. The Confusion Matrix Of Case 11

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 39. Performance Parameter For Classification Results Of Case 11

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 40. The Confusion Matrix Of Case 12

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 41. Performance Parameter For Classification Results Of Case 12

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 42. The Confusion Matrix Of Case 13

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 43. Performance Parameter For Classification Results Of Case 13

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 44. The Confusion Matrix Of Case 14

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 45. Performance Parameter For Classification Results Of Case 14

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 46. The Confusion Matrix Of Case 15

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 47. Performance Parameter For Classification Results Of Case 15

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Table 48. The Confusion Matrix Of Case 16

		Actual										
		RN	YN	BN	RY	YB	RB	RYN	YBN	RBN	RYB	RYBN
Predicted	RN	49	0	0	0	0	0	0	0	0	0	0
	YN	0	49	0	0	0	0	0	0	0	0	0
	BN	0	0	49	0	0	0	0	0	0	0	0
	RY	0	0	0	49	0	0	0	0	0	0	0
	YB	0	0	0	0	49	0	0	0	0	0	0
	RB	0	0	0	0	0	49	0	0	0	0	0
	RYN	0	0	0	0	0	0	49	0	0	0	0
	YBN	0	0	0	0	0	0	0	49	0	0	0
	RBN	0	0	0	0	0	0	0	0	49	0	0
	RYB	0	0	0	0	0	0	0	0	0	0	49
	RYBN	0	0	0	0	0	0	0	0	0	0	49

Table 49. Performance Parameter For Classification Results Of Case 16

	Accuracy	Precision	Recall	F-1 Score
RN	100%	100%	100%	1
YN	100%	100%	100%	1
BN	100%	100%	100%	1
RY	100%	100%	100%	1
YB	100%	100%	100%	1
RB	100%	100%	100%	1
RYN	100%	100%	100%	1
YBN	100%	100%	100%	1
RBN	100%	100%	100%	1
RYB	100%	100%	100%	1
RYBN	100%	100%	100%	1

Upon analyzing the classification results presented above from Table 18 to Table 49, it is evident that the ANNs demonstrate exceptional proficiency in identifying all types of faults, achieving a remarkable 100% accuracy rate. Notably, both types of learning algorithms employed—namely, the Levenberg-Marquardt (LM) and Scaled-Conjugate-Gradient (SCG)—yield identical outputs in fault identification and classification tasks. This uniformity underscores the consistency and reliability of the ANNs across different algorithmic frameworks. Furthermore, the inclusion of receiving end data in the training process does not yield discernible differences in the ANNs' performance regarding fault identification and classification. This suggests that the ANNs possess robust generalization capabilities, enabling them to effectively leverage available data sources irrespective of their origin. However, it is crucial to note that the RYB and RYBN faults are considered identical for classification purposes. Consequently, the ANNs' outputs are deemed accurate for both fault types, reflecting their ability to navigate subtle distinctions in fault classifications with precision. Despite the nuanced challenges posed by fault classification, the ANNs exhibit consistently high precision in both fault identification and classification tasks. This underscores their capacity to deliver accurate and reliable predictions, minimizing the likelihood of misclassification errors and enhancing overall fault diagnosis efficacy.

In summary, the results highlight the ANNs' remarkable performance across various fault detection and classification tasks. Their ability to maintain high accuracy and precision, regardless of the learning algorithm employed or the data utilized, underscores their potential as valuable tools for fault diagnosis in practical applications. These findings lay a solid foundation for further exploration and optimization of ANNs in fault detection systems, with the aim of enhancing system reliability and operational efficiency.

5.3.2 Regression Results

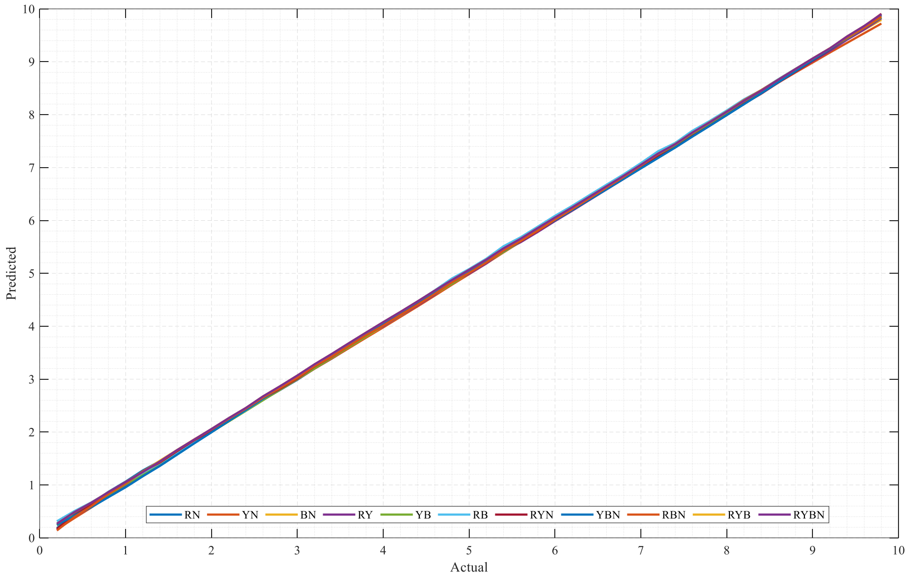


Figure 29. Case 1 regression results of testing system

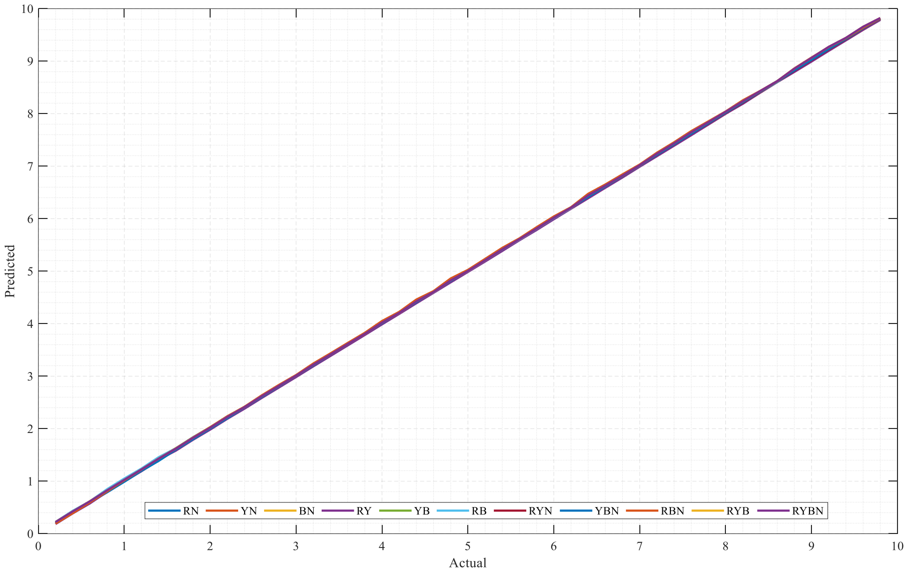


Figure 30. Case 2 regression results of testing system

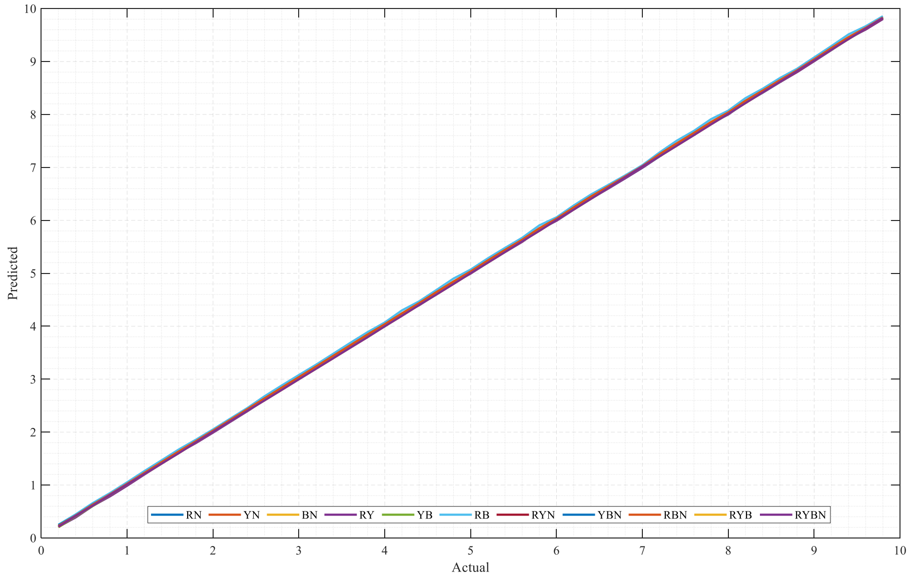


Figure 31. Case 3 regression results of testing system

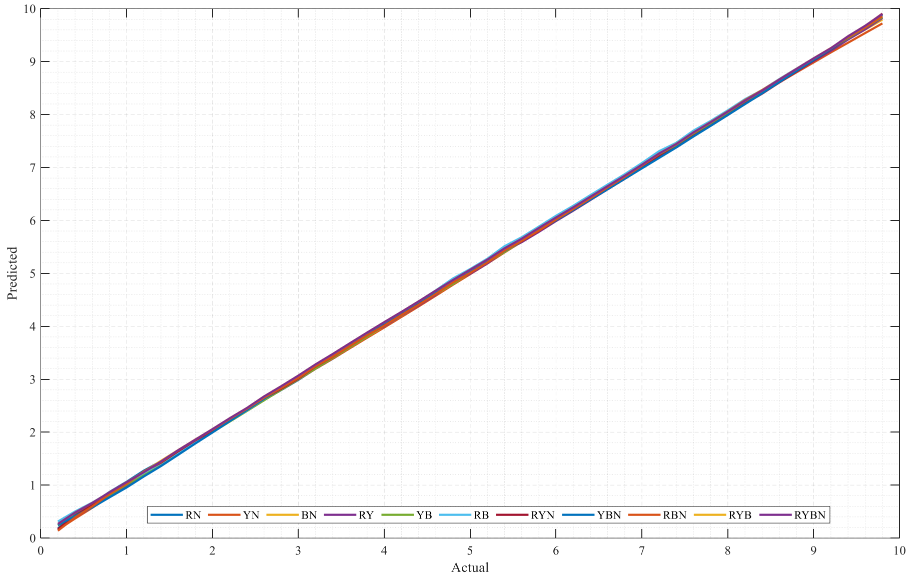


Figure 32. Case 4 regression results of testing system

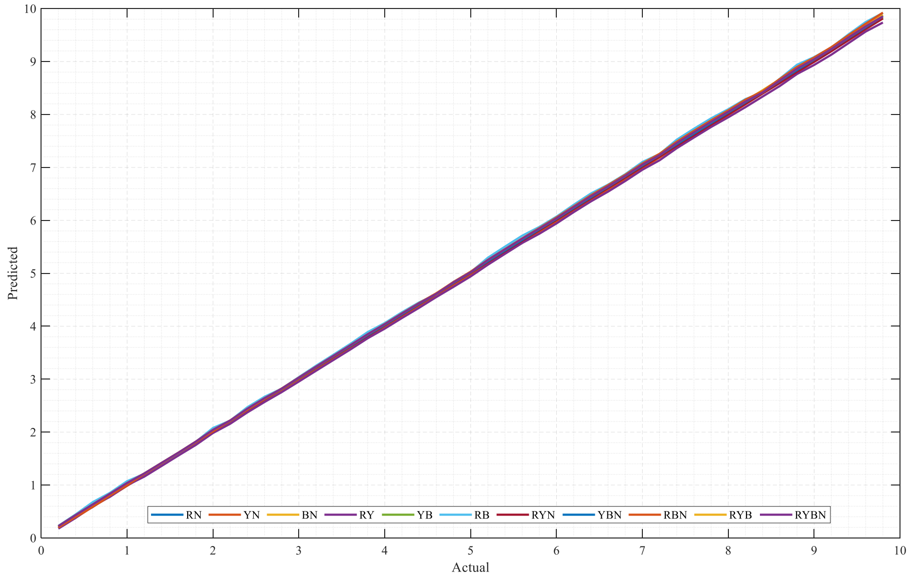


Figure 33. Case 5 regression results of testing system

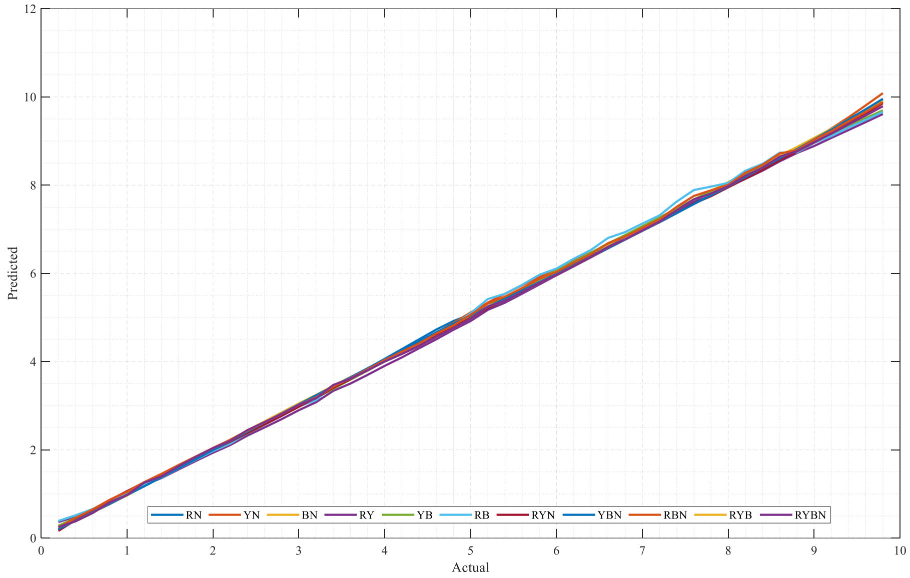


Figure 34. Case 6 regression results of testing system

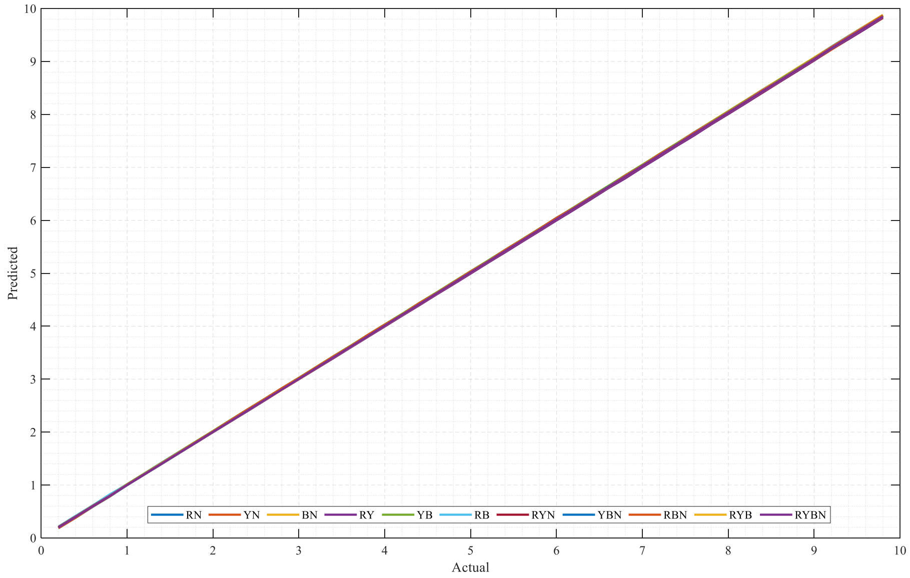


Figure 35. Case 7 regression results of testing system

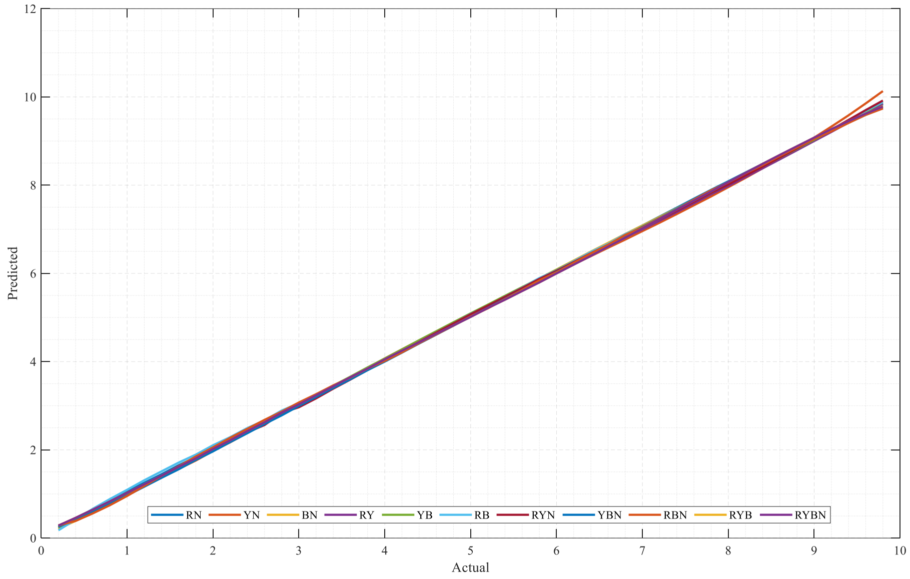


Figure 36. Case 8 regression results of testing system

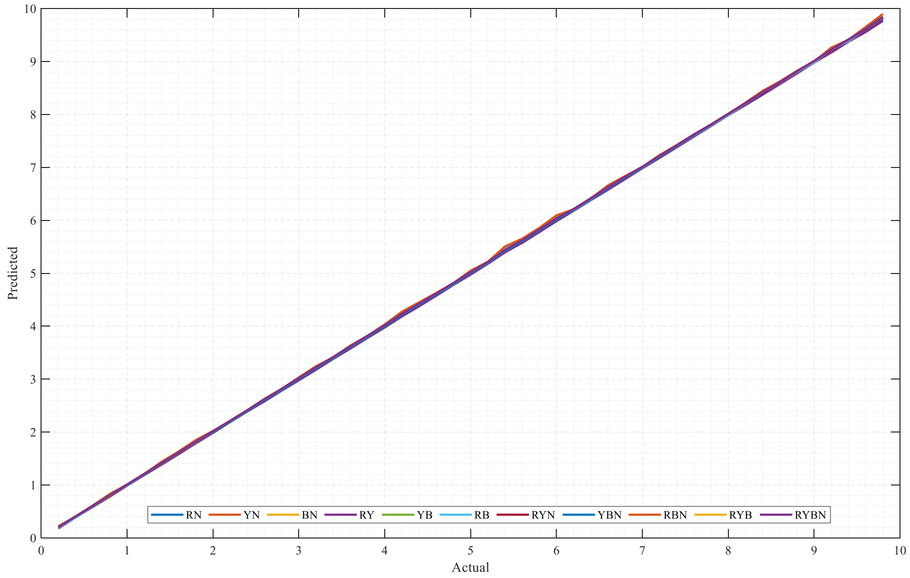


Figure 37. Case 9 regression results of testing system

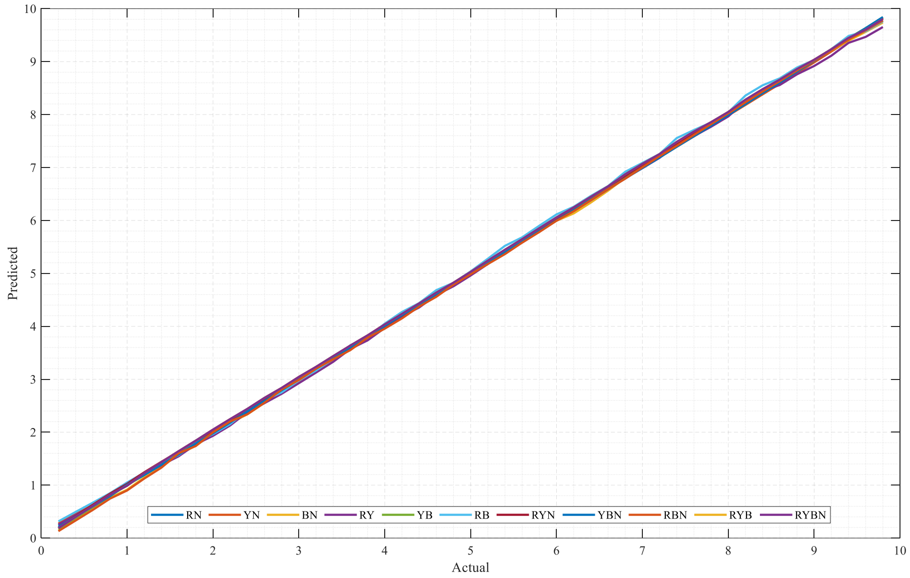


Figure 38. Case 10 regression results of testing system

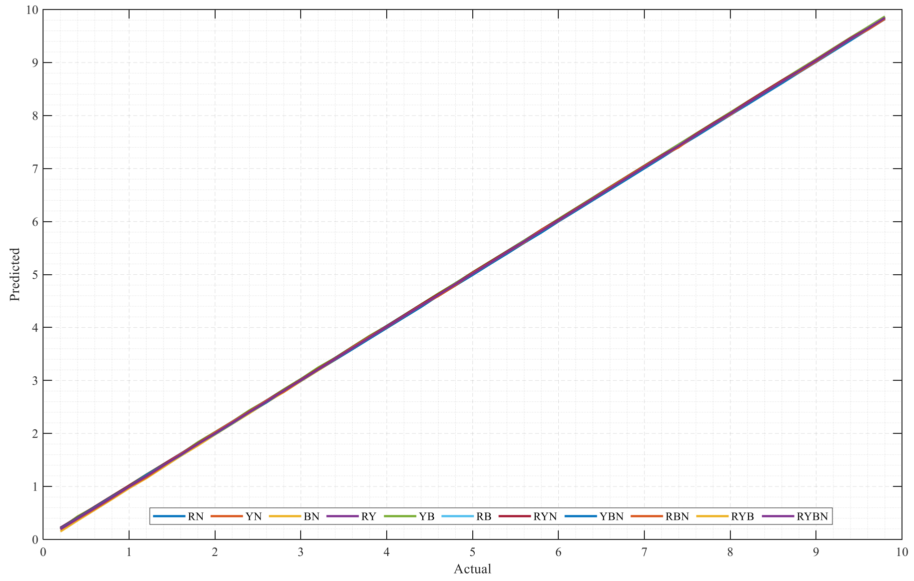


Figure 39. Case 11 regression results of testing system

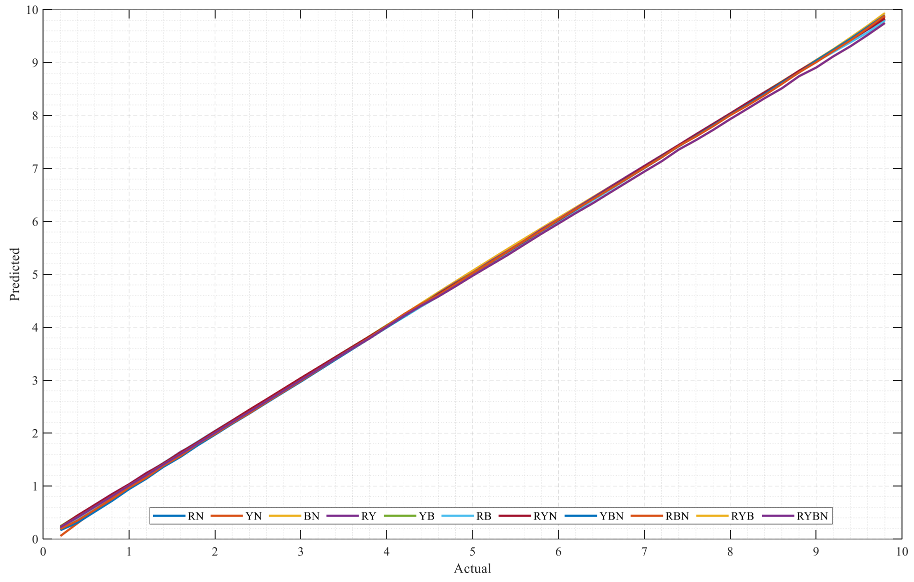


Figure 40. Case 12 regression results of testing system

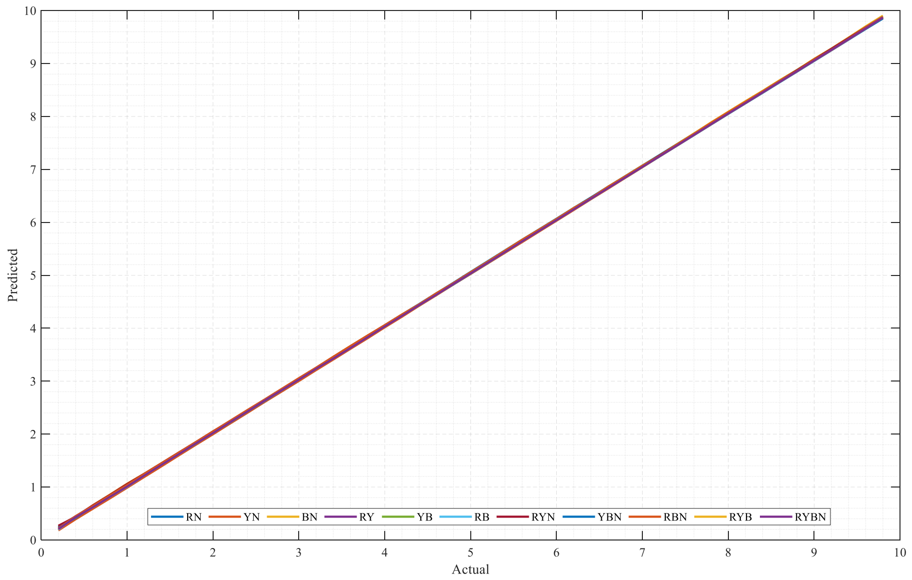


Figure 41. Case 13 regression results of testing system

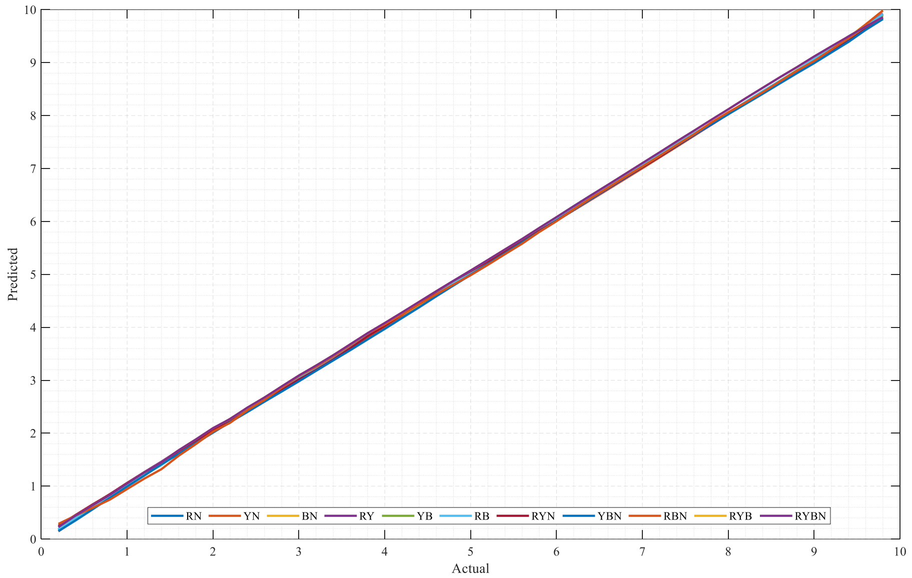


Figure 42. Case 14 regression results of testing system

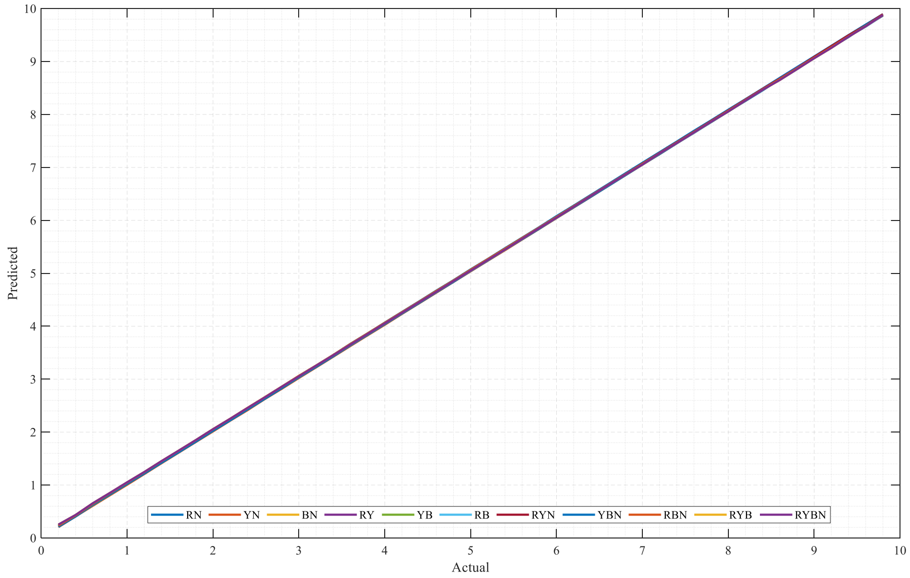


Figure 43. Case 15 regression results of testing system

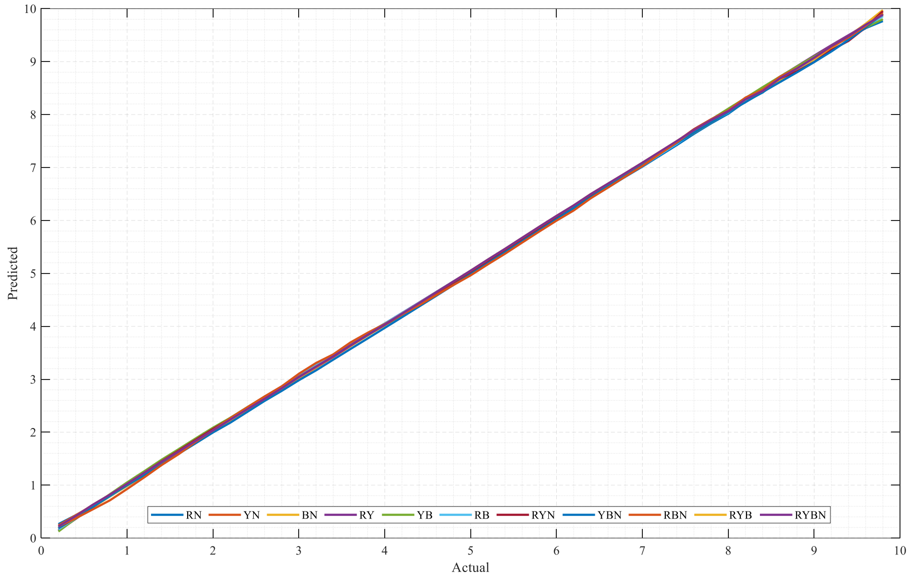


Figure 44. Case 16 regression results of testing system

Table 50. Performance Of Regression Model

Case Number	MAPE	MSE	MAE
Case 1	7.758	0.005	0.189
Case 2	24.342	0.038	0.493
Case 3	9.596	0.010	0.244
Case 4	19.793	0.022	0.400
Case 5	12.394	0.020	0.385
Case 6	21.790	0.041	0.501
Case 7	6.320	0.010	0.259
Case 8	20.055	0.031	0.468
Case 9	6.293	0.005	0.171
Case 10	16.566	0.018	0.341
Case 11	9.192	0.011	0.294
Case 12	14.525	0.014	0.320
Case 13	18.722	0.029	0.532
Case 14	19.994	0.035	0.516
Case 15	19.630	0.034	0.579
Case 16	18.064	0.035	0.531

The regression models' outcomes are visually represented from Figure 29 to Figure 44, revealing nearly accurate or closely aligned predicted fault values with the actual data. Notably, the similarity between the RYB and RYBN faults results in identical predicted values being plotted for both. Among the various cases examined, Case 1 and Case 9 exhibit the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE) values, underscoring their superior predictive accuracy. Additionally, Case 7 and Case 9 emerge with the lowest Mean Absolute Percentage Error (MAPE) values, signifying their robust performance in minimizing prediction errors relative to the actual fault values. These findings highlight the effectiveness of the regression models in capturing fault trends and underscore the importance of meticulous analysis in identifying optimal model configurations for enhanced predictive capabilities.

5.4 Results with Different Systems

The results presented below provide a comprehensive analysis of the performance of a predictive model under various conditions, including different fault resistances (1 Ω, 5 Ω, 10 Ω), load percentages (110%, 150%, 200%), and varying lengths. Each case (Case 1 to Case 16) is meticulously documented with actual and predicted identification and classification outcomes, as well as corresponding distance measurements. The data reveals a consistent pattern of high accuracy, precision, and recall across all conditions, indicating the model's robustness and reliability. This detailed comparison between actual and predicted values under diverse scenarios is crucial for validating the model's effectiveness in real-world applications, ensuring that it can accurately identify and classify conditions with minimal error. The insights gained from this analysis are instrumental in refining the model and enhancing its predictive capabilities, ultimately contributing to the advancement of fault detection and classification systems in electrical engineering.

5.4.1 Classification Results

Table 51: Classification Results of Case 1

	Accuracy	Precision	Recall	F-1 Score
RN	100.0	100.0	100.0	1.0
YN	100.0	100.0	100.0	1.0
BN	100.0	100.0	100.0	1.0
RY	100.0	100.0	100.0	1.0
YB	71.4	100.0	71.4	0.8
RB	100.0	100.0	100.0	1.0
RYN	100.0	100.0	100.0	1.0
YBN	100.0	100.0	100.0	1.0
RBN	100.0	100.0	100.0	1.0
RYB	0.0	0.0	0.0	0.0
RYBN	100.0	100.0	100.0	1.0

The accuracy of the YB fault is notably affected when it comes to fault resistance due to the increased difficulty in correctly identifying and classifying faults as resistance values rise. It observes that for fault resistances of 5 **Ω** and 10 **Ω**, the predicted identification and classification often fail to match the actual conditions.

Table 52: Classification Results of Case 2

	Accuracy	Precision	Recall	F-1 Score
RN	100.0	100.0	100.0	1.0
YN	95.2	100.0	95.2	1.0
BN	95.2	100.0	95.2	1.0
RY	90.5	100.0	90.5	1.0
YB	71.4	100.0	71.4	0.8
RB	71.4	100.0	71.4	0.8
RYN	90.5	100.0	90.5	1.0
YBN	71.4	100.0	71.4	0.8
RBN	71.4	100.0	71.4	0.8
RYB	38.1	100.0	38.1	0.6
RYBN	57.1	100.0	57.1	0.7

The results above indicate the accuracy and performance of predictions for different fault types. The model achieved perfect prediction accuracy (100%) for RN faults, while for YN, BN, and RYN, the accuracy was slightly lower at 95.2% and 90.5%, respectively. For other fault types such as YB, RB, YBN, and RBN, the accuracy dropped to 71.4%, showing more difficulty in correctly predicting these faults. The lowest accuracy was observed for RYB and RYBN faults, with accuracies of 38.1% and 57.1%, respectively. The confidence levels (predicted) are consistently high across all fault types (100%), but the lower precision for some faults indicates areas where the model may need improvement.

Table 53: Classification Results of Case 3

	Accuracy	Precision	Recall	F-1 Score
RN	100.0	100.0	100.0	1.0
YN	95.2	100.0	95.2	1.0
BN	100.0	100.0	100.0	1.0
RY	100.0	100.0	100.0	1.0
YB	81.0	100.0	81.0	0.9
RB	71.4	100.0	71.4	0.8
RYN	100.0	100.0	100.0	1.0
YBN	95.2	100.0	95.2	1.0
RBN	100.0	100.0	100.0	1.0
RYB	28.6	100.0	28.6	0.4
RYBN	71.4	100.0	71.4	0.8

The model performs exceptionally well for faults like RN, BN, RY, and RYN, achieving 100% accuracy, precision, and recall, with an F1-score of 1.0. However, the model's performance drops for certain fault types, such as YB and RB, where accuracy is 81.0% and 71.4%, respectively, though precision remains high at 100%.

Table 54: Classification Results of Case 4

	Accuracy	Precision	Recall	F-1 Score
RN	100.0	100.0	100.0	1.0
YN	100.0	100.0	100.0	1.0
BN	100.0	100.0	100.0	1.0
RY	71.4	100.0	71.4	0.8
YB	71.4	100.0	71.4	0.8
RB	71.4	100.0	71.4	0.8
RYN	95.2	100.0	95.2	1.0
YBN	100.0	100.0	100.0	1.0
RBN	71.4	100.0	71.4	0.8
RYB	0.0	0.0	0.0	0.0
RYBN	95.2	100.0	95.2	1.0

The model demonstrates perfect accuracy, precision, and recall, achieving an F1-score of 1.0. Faults like RY, YB, RB, and RBN show moderate performance with 71.4% accuracy, high precision, and 71.4% recall, leading to an F1-score of 0.8. Despite some mixed results, the model is generally reliable for most fault types.

Table 55:Classification Results of Case 5

	Accuracy	Precision	Recall	F-1 Score
RN	81.0	100.0	81.0	0.9
YN	100.0	100.0	100.0	1.0
BN	81.0	100.0	81.0	0.9
RY	100.0	100.0	100.0	1.0
YB	71.4	100.0	71.4	0.8
RB	61.9	100.0	61.9	0.8
RYN	71.4	100.0	71.4	0.8
YBN	81.0	100.0	81.0	0.9
RBN	81.0	100.0	81.0	0.9
RYB	33.3	100.0	33.3	0.5
RYBN	66.7	100.0	66.7	0.8

The performance analysis of the model across different fault types highlights varying levels of success. The model performs exceptionally well for fault types of YN and RY, achieving 100% accuracy, precision, recall, and an F1-score of 1.0, indicating perfect predictions. For RN, BN, YBN, and RBN fault types, the model shows strong but slightly lower performance with an accuracy of 81.0% and an F1-score of 0.9. Faults like YB, RB, RYN, and RYBN show moderate performance, with accuracy ranging from 61.9% to 71.4% and F1-scores of 0.8. Overall, the model shows solid results but needs improvement in handling more complex fault types.

Table 56: Classification Results of Case 6

	Accuracy	Precision	Recall	F-1 Score
RN	66.7	100.0	66.7	0.8
YN	81.0	100.0	81.0	0.9
BN	81.0	100.0	81.0	0.9
RY	71.4	100.0	71.4	0.8
YB	100.0	100.0	100.0	1.0
RB	71.4	100.0	71.4	0.8
RYN	81.0	100.0	81.0	0.9
YBN	81.0	100.0	81.0	0.9
RBN	100.0	100.0	100.0	1.0
RYB	23.8	100.0	23.8	0.4
RYBN	71.4	100.0	71.4	0.8

The model's performance across different fault types shows varied results. Fault types YB and RBN exhibit perfect performance, achieving 100% accuracy, precision, recall, and an F1-score of 1.0, indicating flawless predictions. Faults such as YN, BN, RYN, and YBN demonstrate strong performance, with an accuracy of 81.0% and F1-scores of 0.9, indicating high but not perfect prediction accuracy. Moderate performance is seen for fault types RN, RY, RB, and RYBN, where accuracy ranges from 66.7% to 71.4% with F1-scores of 0.8.

Table 57: Classification Results of Case 7

	Accuracy	Precision	Recall	F-1 Score
RN	100.0	100.0	100.0	1.0
YN	100.0	100.0	100.0	1.0
BN	81.0	100.0	81.0	0.9
RY	71.4	100.0	71.4	0.8
YB	76.2	100.0	76.2	0.9
RB	71.4	100.0	71.4	0.8
RYN	71.4	100.0	71.4	0.8
YBN	95.2	100.0	95.2	1.0
RBN	81.0	100.0	81.0	0.9
RYB	57.1	100.0	57.1	0.7
RYBN	42.9	100.0	42.9	0.6

The model demonstrates excellent performance with perfect accuracy, precision, recall, and F1-scores of 1.0 in several cases, showing flawless predictions. Some fault types, like the ones with 81.0% and 76.2% accuracy, still achieve high precision but slightly lower recall and F1-scores of 0.9, indicating minor discrepancies in predictions. Moderate performance is observed for fault types with 71.4% accuracy, maintaining strong precision but slightly lower recall and F1-scores of 0.8.

Table 58: Classification Results of Case 8

	Accuracy	Precision	Recall	F-1 Score
RN	71.4	100.0	71.4	0.8
YN	81.0	100.0	81.0	0.9
BN	71.4	100.0	71.4	0.8
RY	71.4	100.0	71.4	0.8
YB	71.4	100.0	71.4	0.8
RB	71.4	100.0	71.4	0.8
RYN	71.4	100.0	71.4	0.8
YBN	71.4	100.0	71.4	0.8
RBN	71.4	100.0	71.4	0.8
RYB	28.6	100.0	28.6	0.4
RYBN	42.9	100.0	42.9	0.6

The model shows consistent performance across most fault types, achieving an accuracy of 71.4% with perfect precision (100%) and a recall of 71.4%, leading to an F1-score of 0.8. This indicates reliable predictions for the majority of fault types, balancing accuracy and precision. However, the model struggles with two specific fault types: RYB and RYBN.

Table 59: Classification Results of Case 9

	Accuracy	Precision	Recall	F-1 Score
RN	100.0	100.0	100.0	1.0
YN	85.7	100.0	85.7	0.9
BN	85.7	100.0	85.7	0.9
RY	85.7	100.0	85.7	0.9
YB	81.0	100.0	81.0	0.9
RB	100.0	100.0	100.0	1.0
RYN	100.0	100.0	100.0	1.0
YBN	71.4	100.0	71.4	0.8
RBN	76.2	100.0	76.2	0.9
RYB	42.9	100.0	42.9	0.6
RYBN	57.1	100.0	57.1	0.7

The model demonstrates excellent performance for several fault types, achieving 100% accuracy, precision, recall, and F1-scores of 1.0 for RN and RB, and RYN, indicating perfect classification. For other fault types like YN, BN, and RY, the accuracy is slightly lower at 85.7%, but the model still maintains strong precision and recall, resulting in an F1-score of 0.9. YB, YBN and RBN show declining performance, with YBN and RYBN fault types having lower accuracy (71.4% and 57.1%) and F1-scores of 0.8 and 0.7, respectively.

Table 60: Classification Results of Case 10

	Accuracy	Precision	Recall	F-1 Score
RN	61.9	100.0	61.9	0.8
YN	76.2	100.0	76.2	0.9
BN	66.7	100.0	66.7	0.8
RY	57.1	100.0	57.1	0.7
YB	57.1	100.0	57.1	0.7
RB	71.4	100.0	71.4	0.8
RYN	61.9	100.0	61.9	0.8
YBN	47.6	100.0	47.6	0.6
RBN	95.2	100.0	95.2	1.0
RYB	38.1	100.0	38.1	0.6
RYBN	57.1	100.0	57.1	0.7

The model's performance varies across different fault types. It performs well for RN, YN, RB, and RBN with F1-scores ranging from 0.8 to 1.0, showing strong precision and recall. However, for fault types like RY, YB and YBN, the accuracy drops below 60%, with lower F1-scores (0.6 to 0.7), indicating that the model struggles to classify these faults accurately.

Table 61: Classification Results of Case 11

	Accuracy	Precision	Recall	F-1 Score
RN	71.4	100.0	71.4	0.8
YN	100.0	100.0	100.0	1.0
BN	95.2	100.0	95.2	1.0
RY	100.0	100.0	100.0	1.0
YB	95.2	100.0	95.2	1.0
RB	71.4	100.0	71.4	0.8
RYN	76.2	100.0	76.2	0.9
YBN	100.0	100.0	100.0	1.0
RBN	71.4	100.0	71.4	0.8
RYB	42.9	100.0	42.9	0.6
RYBN	57.1	100.0	57.1	0.7

Fault types of YN, RY, and YBN achieved perfect scores across all metrics, indicating exceptional performance. BN and YB also performed well, with high accuracy and F-1 scores of 1.0. However, fault types of RN, RB, RYN, RBN, and RYBN showed moderate performance, with accuracy and recall around 71.4% to 76.2%, and F-1 scores ranging from 0.7 to 0.9.

Table 62: Classification Results of Case 12

	Accuracy	Precision	Recall	F-1 Score
RN	90.5	100.0	90.5	1.0
YN	100.0	100.0	100.0	1.0
BN	95.2	100.0	95.2	1.0
RY	71.4	100.0	71.4	0.8
YB	100.0	100.0	100.0	1.0
RB	71.4	100.0	71.4	0.8
RYN	90.5	100.0	90.5	1.0
YBN	100.0	100.0	100.0	1.0
RBN	71.4	100.0	71.4	0.8
RYB	4.8	100.0	4.8	0.1
RYBN	71.4	100.0	71.4	0.8

The results show high accuracy, precision, and recall for most categories, with YN, YB, YBN, and RN achieving perfect precision and recall scores of 100%, resulting in a perfect F-1 score of 1.0. Categories like RY, RB, RBN and RYBN show strong precision (100%) but lower recall (71.4%), resulting in a slightly reduced F-1 score of 0.8. Overall, most categories perform exceptionally well.

Table 63: Classification Results of Case 13

	Accuracy	Precision	Recall	F-1 Score
RN	81.0	100.0	81.0	0.9
YN	100.0	100.0	100.0	1.0
BN	71.4	100.0	71.4	0.8
RY	71.4	100.0	71.4	0.8
YB	71.4	100.0	71.4	0.8
RB	100.0	100.0	100.0	1.0
RYN	95.2	100.0	95.2	1.0
YBN	71.4	100.0	71.4	0.8
RBN	71.4	100.0	71.4	0.8
RYB	19.0	100.0	19.0	0.3
RYBN	71.4	100.0	71.4	0.8

The results demonstrate strong performance for several categories, with YN, RB, and RYN achieving perfect precision, recall, and F1 scores, indicating excellent model performance in these areas. Categories such as RN, BN, RY, YB, YBN, RBN, and RYBN show perfect precision (100%) but lower recall (71.4% to 81.0%), leading to F1 scores ranging from 0.8 to 0.9.

Table 64: Classification Results of Case 14

	Accuracy	Precision	Recall	F-1 Score
RN	61.9	100.0	61.9	0.8
YN	71.4	100.0	71.4	0.8
BN	71.4	100.0	71.4	0.8
RY	71.4	100.0	71.4	0.8
YB	90.5	100.0	90.5	1.0
RB	66.7	100.0	66.7	0.8
RYN	71.4	100.0	71.4	0.8
YBN	71.4	100.0	71.4	0.8
RBN	71.4	100.0	71.4	0.8
RYB	23.8	100.0	23.8	0.4
RYBN	47.6	100.0	47.6	0.6

The results show mixed performance across different categories. Categories like YB achieve high accuracy (90.5%) and perfect precision, recall, and an F1 score of 1.0, indicating strong performance. Other categories, such as RN, YN, BN, RY, RB, RYN, YBN, and RBN, demonstrate perfect precision but lower recall (61.9% to 71.4%), resulting in F1 scores of 0.8. Additionally, RYBN has moderate accuracy (47.6%) and an F1 score of 0.6, suggesting it also needs improvement.

Table 65: Classification Results of Case 15

	Accuracy	Precision	Recall	F-1 Score
RN	81.0	100.0	81.0	0.9
YN	100.0	100.0	100.0	1.0
BN	71.4	100.0	71.4	0.8
RY	76.2	100.0	76.2	0.9
YB	81.0	100.0	81.0	0.9
RB	100.0	100.0	100.0	1.0
RYN	81.0	100.0	81.0	0.9
YBN	71.4	100.0	71.4	0.8
RBN	76.2	100.0	76.2	0.9
RYB	28.6	100.0	28.6	0.4
RYBN	71.4	100.0	71.4	0.8

The results indicate strong overall performance across most categories. YN, RB, and RN achieve perfect precision and high recall, resulting in F1 scores of 1.0 and 0.9, respectively. Other categories such as BN, RY, YB, RYN, and RBN also exhibit high precision and good recall, with F1 scores ranging from 0.8 to 0.9.

Table 66: Classification Results of Case 16

	Accuracy	Precision	Recall	F-1 Score
RN	71.4	100.0	71.4	0.8
YN	71.4	100.0	71.4	0.8
BN	95.2	100.0	95.2	1.0
RY	66.7	100.0	66.7	0.8
YB	71.4	100.0	71.4	0.8
RB	66.7	100.0	66.7	0.8
RYN	71.4	100.0	71.4	0.8
YBN	71.4	100.0	71.4	0.8
RBN	76.2	100.0	76.2	0.9
RYB	23.8	100.0	23.8	0.4
RYBN	47.6	100.0	47.6	0.6

The results reveal varied performance across different categories. BN stands out with high accuracy (95.2%), perfect precision, recall, and an F1 score of 1.0, indicating optimal performance. Categories such as RN, YN, RY, YB, RB, RYN, and YBN show perfect precision but lower recall, leading to F1 scores of 0.8. RBN performs slightly better with an F1 score of 0.9. In contrast, RYBN shows lower accuracy and recall, with F1 scores of 0.6. These lower scores suggest that RYBN needs significant improvement.

Comparing the results from using sending end inputs and both end inputs reveals some notable differences in performance metrics. In the first set (sending end inputs) of cases, the accuracy, precision, recall, and F1 scores are consistently high, often reaching 100% across various fault types such as RN, YN, and BN. However, in the second set (both ends inputs) of cases, there is a noticeable drop in performance for several fault types. For instance, RN’s accuracy drops to as low as 61.9% in Case 10, and RY’s recall decreases significantly in later cases. Despite these fluctuations, some fault types like YB maintain high precision across all cases. The best performance recorded is in Case 1, where multiple fault types (RN, YN, BN, RY, RB, RYN, YBN, RBN, and RYBN) achieved perfect scores of 100% in accuracy, precision, recall, and F1, showcasing the highest reliability and consistency.

Comparing the cases where LM leaning algorithm is used with cases with SCG learning algorithm) reveals some interesting trends in terms of accuracy, precision, recall, and F1 scores. The LM cases generally exhibit higher consistency and performance across most fault types, with several instances of perfect scores (100%) in accuracy, precision, recall, and F1, particularly in Cases 1 and 3. In contrast, SCG cases show more variability and lower performance metrics, with notable drops in accuracy and recall, such as in Case 10 where RN’s accuracy is only 61.9%. Despite this, some SCG cases like Case 2 and Case 6 still maintain high precision and recall for certain fault types. The best performance recorded is in Case 1, where multiple fault types achieved perfect scores, highlighting the highest level of reliability and consistency.

5.4.2 Regression Results

Table 67: Regression Results of Case 16

Case#	MAPE	MSE	MAE
Case 1	1.88	46.32	3.65
Case 2	1.22	18.73	2.40
Case 3	0.97	10.61	1.91
Case 4	0.73	6.97	1.44
Case 5	1.41	25.66	2.78
Case 6	1.00	12.55	2.05
Case 7	1.22	17.18	2.48
Case 8	0.85	8.89	1.75
Case 9	1.47	22.76	2.99
Case 10	1.25	17.61	2.67
Case 11	0.77	8.11	1.59
Case 12	0.95	12.25	2.05
Case 13	0.88	9.76	1.78
Case 14	0.98	10.68	1.89
Case 15	0.93	11.72	1.90
Case 16	1.09	14.01	2.17

Comparing the LM cases (1, 3, 5, 7, 9, 11, 13, 15) with the SCG cases (2, 4, 6, 8, 10, 12, 14, 16) in terms of MAPE, MSE, and MAE, the SCG cases generally show better performance. For instance, Case 4 has the lowest MAPE (0.73), MSE (6.97), and MAE (1.44), indicating the highest accuracy and lowest error rates. In contrast, the odd-numbered cases have higher error metrics, with Case 1 having the highest MAPE (1.88) and MSE (46.32). Overall, the SCG cases demonstrate more consistent and lower error values.

When comparing one end inputs with both end inputs, the first group tends to have higher error metrics. Case 4 stands out with the best performance, having the lowest MAPE, MSE, and MAE among all cases. In the second group, Cases 11 and 13 show strong performance with low error values, but they do not surpass the metrics of Case 4. This indicates that the first group has more variability and higher errors compared to the second group.

Table 68 highlights the impact of fault resistance (ranging from 1 Ω to 20 Ω) on the regression model's performance. The data reveals that as fault resistance increases, the predicted fault distance exhibits a rapid rise, it affects the model's accuracy and leads to a significant rise in fault distance estimation.

Table 68: Effect Of Fault Resistanc on The Regression Model of Case 1

Fault	**Actual**	**Predicted**			
		1 Ω	**5 Ω**	**10 Ω**	**20 Ω**
RN	1	3.51	14.14	14.44	21.34
	5	5.82	12.91	14.44	20.43
	9	9.59	14.44	14.44	19.54
YN	1	1.39	7.73	13.07	16.74
	5	5.49	10.25	13.07	16.60
	9	9.47	13.07	13.07	16.39
BN	1	1.49	4.32	12.32	17.68
	5	5.32	8.32	12.32	18.03

	9	9.39	12.32	12.32	18.18
RY	1	1.09	14.87	17.72	22.28
	5	5.36	16.23	17.72	22.16
	9	9.41	17.72	17.72	21.99
YB	1	3.98	19.57	20.70	23.29
	5	7.08	19.74	20.70	23.21
	9	10.67	20.70	20.70	23.06
RB	1	0.44	15.55	20.12	17.73
	5	5.65	17.96	20.12	17.79
	9	10.42	20.12	20.12	17.81
RYN	1	4.85	21.33	22.09	28.28
	5	8.02	21.59	22.09	27.66
	9	11.16	22.09	22.09	27.04
YBN	1	3.37	11.31	15.41	20.21
	5	5.64	12.35	15.41	21.20
	9	9.25	15.41	15.41	22.00
RBN	1	0.58	13.32	16.00	27.09
	5	3.54	14.14	16.00	27.02
	9	8.27	16.00	16.00	26.57
RYB	1	4.76	19.98	20.26	30.32
	5	6.37	19.47	20.26	30.37
	9	9.71	20.26	20.26	30.34
RYBN	1	4.76	19.98	20.26	30.32
	5	6.37	19.47	20.26	30.37
	9	9.71	20.26	20.26	30.34

CHAPTER 6: DISCUSSION

The study delved into examining 16 cases, aiming to assess the efficacy of Artificial Neural Networks (ANNs) in utilizing remote end data, employing various backpropagation learning algorithms. While prior research primarily concentrated on utilizing extracted features from fault signals as inputs for training ANNs to localize fault distance, none have explored the impact of incorporating remote end data. This study sought to bridge this gap, investigating whether local data alone is sufficient for fault identification and localization or if remote end data enhances these processes.

Furthermore, the study aimed to evaluate different system types using two distinct algorithms, namely LM (Levenberg-Marquardt) and SCG (Scaled Conjugate Gradient). It scrutinized how these learning algorithms influence the performance of ANNs in detecting, classifying, and locating various fault types. By comparing the outcomes obtained from employing LM and SCG algorithms, the study aimed to discern their respective contributions to the ANN's effectiveness in fault detection, classification, and localization across different fault scenarios.

The findings of the study reveal a remarkable success rate in identifying and classifying faults across all cases, with an accuracy of 100%. This indicates a high level of precision and recall, as evidenced by the F1 score of 1 for all cases, signifying perfect performance in both metrics. Upon scrutinizing the data presented from Table 18 to Table 49, it becomes apparent that both learning algorithms, LM and SCG, exhibit efficacy in fault identification and classification when employed by ANNs.

Moreover, an intriguing observation emerges from the study: the incorporation of remote data into the ANNs does not appear to influence the accuracy of fault identification and classification in transmission lines. Across all cases examined, the inclusion of remote data did not yield any discernible impact, with a consistent 100% accuracy rate recorded for these tasks. This suggests that the ANNs' ability to identify and classify faults remains robust regardless of whether local or remote data is utilized.

Overall, this study contributes valuable insights into the effectiveness of ANNs in fault detection and classification, shedding light on the role of different learning algorithms and the impact of incorporating remote data in transmission line analysis.

In Table 50, a significant pattern emerges: the prediction accuracy on the validation set consistently exceeds that of the training set across all algorithms. This observation suggests the absence of overt overfitting on the training data, implying a commendable level of generalization achieved by the models.

Figure 29 through Figure 44 present a discernible pattern where the accuracy of fault distance prediction experiences a slight decline for certain cases located at the terminals of the transmission lines. Despite this fluctuation, it's noteworthy that both algorithms consistently produce comparable results across various fault types. This underscores the robustness and reliability of the models in predicting fault distances, despite minor deviations in accuracy under specific conditions. Further investigation is merited to elucidate the underlying factors influencing the observed variations in prediction accuracy at the extremities of the transmission lines. Nonetheless, the consistent similarity in outputs between both algorithms underscores their efficacy in addressing diverse fault scenarios and emphasizes their potential practical applicability in real-world settings for accurate fault distance prediction.

upon analyzing the provided data in Table 50, it is evident that there are variations in the performance metrics (MAPE, MSE, and MAE) across different cases.

Here is a comparison based on the performance metrics:

- The MAPE values range from as low as 6.293% (Case 9) to as high as 24.342% (Case 2).
- Cases 1, 7, and 9 exhibit relatively low MAPE values, indicating a high level of accuracy in fault localization.
- Cases 2, 6, and 15 have notably higher MAPE values, suggesting less accurate fault localization in these scenarios.
- The MSE values vary from 0.005 (Case 9) to 0.041 (Case 6).
- Cases 1, 3, 7, and 9 demonstrate relatively lower MSE values, indicating better precision in fault localization.
- Cases 6, 8, and 14 exhibit higher MSE values, implying less precise fault localization in these instances.
- The MAE values range from 0.171 (Case 9) to 0.579 (Case 15).
- Cases 1, 7, 9, and 11 showcase lower MAE values, indicating more accurate fault localization.
- Cases 13, 14, 15, and 16 display higher MAE values, suggesting less accurate fault localization in these cases.
- Overall, Cases 1, 7, and 9 consistently demonstrate relatively better performance across all three metrics, while Cases 2, 6, 8, 13, 14, 15, and 16 exhibit higher error rates, indicating potential areas for improvement in fault localization.

Table 50 highlights a significant enhancement in results when employing the LM algorithm for fault localization, as evidenced by the MAPE scores, with notable exceptions observed in cases 15 and 16 where SCG demonstrated superior performance. Additionally, LM consistently outperforms SCG across all cases in terms of MSE. However, when considering MAE, SCG exhibits superior performance in the final six cases. Overall, the LM algorithm demonstrates superior performance when utilizing local data, whereas SCG exhibits robust performance when incorporating data from both ends of the system.

Both techniques employed for training the ANNs contribute to enhancing the outcomes, effectively meeting the objective of identifying, classifying, and localizing faults in transmission lines. While the degree of improvement varies from case to case, the overarching goal is consistently achieved, with overall accuracy notably enhanced. Additionally, the utilization of the SCG learning algorithm results in reduced training time, a notable advantage that enhances efficiency in fault detection and localization tasks.

CHAPTER 7: CONCLUSION

The study delved into analyzing 16 cases to evaluate how effectively Artificial Neural Networks (ANNs) could utilize data from remote locations, using different backpropagation learning algorithms. Previous research mostly focused on using specific features extracted from fault signals to train ANNs to pinpoint fault distance. However, no previous studies have explored the effect of including remote end data. This study aimed to fill this gap by investigating whether relying solely on local data is enough for fault identification and localization, or if incorporating remote end data improves these processes.

Additionally, the study aimed to assess various system types by employing two distinct algorithms: LM (Levenberg-Marquardt) and SCG (Scaled Conjugate Gradient). It scrutinized the impact of these learning algorithms on the performance of ANNs in detecting, classifying, and locating different fault types. By comparing the results obtained from using LM and SCG algorithms, the study sought to discern their respective contributions to the effectiveness of ANNs in fault detection, classification, and localization across diverse fault scenarios.

The study demonstrates enhanced results through the utilization of both terminal data and the LM learning algorithm across most cases. While the SCG learning algorithm exhibits faster convergence speeds compared to the LM algorithm in fault location applications, both algorithms yield comparable results in fault identification and classification. Leveraging data from both ends is not novel to transmission line protection systems, as cable differential protection operates on the same principle.

However, there remains a gap for analysis concerning varying loads and scenarios involving high resistance faults occurring on transmission cables. Due to time and equipment constraints (specifically personal computer limitations), these aspects were not investigated in this study. Nevertheless, the research demonstrates that ANNs can serve as efficient protection relays, capable of detecting faults and tripping faulty lines more rapidly than traditional techniques, owing to shorter processing times. Additionally, the study's findings indicate superior fault localization results compared to impedance-based techniques, although it acknowledges that the traveling wave method remains the most accurate. However, this method is also acknowledged to be more expensive and complex.

For future work, the analysis can be extended by incorporating different feature extraction techniques, such as Fast Fourier Transform (FFT) and Wavelet Transform (WT), to evaluate their impact on fault detection, classification, and localization accuracy. Additionally, experimenting with advanced ANN architectures, such as Recurrent Neural Networks (RNNs) for capturing temporal patterns and Graph Neural Networks (GNNs) for modeling spatial relationships, could provide deeper insights and potentially enhance the model's performance in handling complex fault scenarios.

REFERENCES

[1] E. Csanyi, “The structure of Electric Power Systems (generation, distribution and transmission of energy): EEP,” *Electrical Engineering Portal*, 2019. https://electrical-engineering-portal.com/electric-power-systems (accessed Mar. 15, 2024).

[2] R. Rengaraj, G. R. Venkatakrishnan, S. Shalini, R. Subitsha, S. Suganthi, and S. Sushmita Carolyn, “Identification and classification of faults in underground cables – A review,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1166, no. 1, p. 012018, 2021, doi: 10.1088/1757-899x/1166/1/012018.

[3] U. Cables, “CHAPTER-7 Underground Cables.”

[4] M. Mirzaei, M. Z. A. Ab Kadir, E. Moazami, and H. Hizam, “Review of fault location methods for distribution power system,” *Aust. J. Basic Appl. Sci.*, vol. 3, no. 3, pp. 2670–2676, 2009.

[5] P. System, R. Committee, I. Power, and E. Society, *IEEE Guide for Determining Fault Location on AC Transmission and Distribution Lines*, vol. 2014, no. June. 2014.

[6] J. Xie, “A Review of Machine Learning Applications in Power System Resilience Jian,” pp. 1–5, 2019, [Online]. Available: https://doi.org/10.1109/PESGM41954.2020.9282137.

[7] A. Elnozahy, K. Sayed, and M. Bahyeldin, “Artificial Neural Network Based Fault Classification and Location for Transmission Lines,” *IEEE Conf. Power Electron. Renew. Energy, CPERE 2019*, no. October, pp. 140–144, 2019, doi:

10.1109/CPERE45374.2019.8980173.

[8] G. Shingade and S. Shah, “Fault detection and location based SVM for three phase transmission lines utilizing positive sequence fault,” vol. 11, no. 3, pp. 61–70, 2023, doi: 10.15649/2346030X.3302.

[9] A. A. Fayyad *et al.*, “Electric Power Components and Systems Transmission : A Hardware Co-simulation Approach Transmission Line Protection Using High-Speed Decision Tree and Artificial Neural Network,” *Electr. Power Components Syst.*, vol. 49, no. 13–14, pp. 1181–1200, 2022, doi: 10.1080/15325008.2022.2050446.

[10] C. Asbery and Y. Liao, “Fault Identification on Electrical Transmission Lines Using Artificial Neural Networks,” *Electr. Power Components Syst.*, vol. 49, no. 13–14, pp. 1118–1129, 2022, doi: 10.1080/15325008.2022.2049659.

[11] N. N. Bon and L. Van Dai, “Fault Identification, Classification, and Location on Transmission Lines Using Combined Machine Learning Methods,” *Int. J. Eng. Technol. Innov.*, vol. 12, no. 2, pp. 91–109, 2022, doi: 10.46604/IJETI.2022.7571.

[12] L. Fiaschetti and P. A. Lotito, “Pr ep rin t n ot pe er r ie we d Pr ep rin t n er.”

[13] X. Wang, P. Zhou, X. Peng, Z. Wu, and H. Yuan, “Fault location of transmission line based on CNN-LSTM double-ended combined model,” *Energy Reports*, vol. 8, pp. 781–791, 2022, doi: 10.1016/j.egyr.2022.02.275.

[14] B. Roy *et al.*, “Deep Learning Based Relay for Online Fault Detection, Classification, and Fault Location in a Grid-Connected Microgrid,” *IEEE Access*, vol. 11, no. June, pp. 62674–62696, 2023, doi: 10.1109/ACCESS.2023.3285768.

[15] M. Najafzadeh, J. Pouladi, A. Daghigh, J. Beiza, S. Shahmohamadi, and M. De

Santis, “A New Method for Fault Detection and Location in a Low-Resistance Grounded Power Distribution Network Using Voltage Phasor of D-PMUs Data,” *Int. Trans. Electr. Energy Syst.*, vol. 2023, 2023, doi: 10.1155/2023/1754305.

[16] S. S. Shingare, P. Khampariya, and S. Bakre, “Application of ANN-Based Approach for Fault Location in Extra High Voltage Networks,” *Int. J. Eng. Trends Technol.*, vol. 71, no. 2, pp. 440–449, 2023, doi: 10.14445/22315381/IJETT-V71I2P245.

[17] V. Rizeakos, A. Bachoumis, N. Andriopoulos, M. Birbas, and A. Birbas, “Deep learning-based application for fault location identification and type classification in active distribution grids,” *Appl. Energy*, vol. 338, no. March, p. 120932, 2023, doi: 10.1016/j.apenergy.2023.120932.

[18] Y. Liang, A. He, J. Yuan, T. Wu, and Z. Jiao, “An accurate fault location method for distribution lines based on data fusion of outcomes from multiple algorithms,” *Int. J. Electr. Power Energy Syst.*, vol. 153, no. January, p. 109290, 2023, doi: 10.1016/j.ijepes.2023.109290.

[19] V. Veerasamy, N. I. Abdul Wahab, R. Ramachandran, M. Mansoor, M. Thirumeni, and M. L. Othman, “High impedance fault detection in medium voltage distribution network using discrete wavelet transform and adaptive neuro-fuzzy inference system,” *Energies*, vol. 11, no. 12, 2018, doi: 10.3390/en11123330.

[20] M. Dashtdar, M. Esmaeilbeig, M. Najafi, and M. E. N. Bushehri, “Fault Location in the Transmission Network Using Artificial Neural Network,” *Autom. Control Comput. Sci.*, vol. 54, no. 1, pp. 39–51, 2020, doi: 10.3103/S0146411620010022.

[21] Y. D. Mamuya, Y. Der Lee, J. W. Shen, M. Shafiullah, and C. C. Kuo, “Application of machine learning for fault classification and location in a radial

distribution grid," *Appl. Sci.*, vol. 10, no. 14, 2020, doi: 10.3390/app10144965.

[22] A. T. Radhi, W. H. Zayer, and A. M. Dakhil, "Classification and direction discrimination of faults in transmission lines using 1d convolutional neural networks," *Int. J. Power Electron. Drive Syst.*, vol. 12, no. 3, pp. 1928–1939, 2021, doi: 10.11591/ijpeds.v12.i3.pp1928-1939.

[23] D. Okojie, L. Idoko, D. Herbert, and A. Nnachi, "Study of transmission line boundary protection using a multilayer perceptron neural network with back propagation and wavelet transform," *Appl. Syst. Innov.*, vol. 4, no. 4, 2021, doi: 10.3390/asi4040095.

[24] O. O. Kalu and T. C. Madueme, "Application of artificial neural network (ANN) to enhance power systems protection: a case of the Nigerian 330 kV transmission line," *Electr. Eng.*, vol. 100, no. 3, pp. 1467–1479, 2018, doi: 10.1007/s00202-017-0599-y.

[25] Z. Tong *et al.*, "Fault Diagnosis and Location Method for Active Distribution Network Based on Artificial Neural Network," *Electr. Power Components Syst.*, vol. 46, no. 9, pp. 985–996, 2018, doi: 10.1080/15325008.2018.1460884.

[26] R. Alilouch and F. Slaoui-Hasnaoui, "Intelligent Relay Based on Artificial Neural Networks ANN for Transmission Line," *2022 IEEE 9th Int. Conf. Sci. Electron. Technol. Inf. Telecommun. SETIT 2022*, pp. 468–473, 2022, doi: 10.1109/SETIT54465.2022.9875449.

[27] M. M. Devi, A. Ganguly, and M. Sharma, "Soft Computing Technique for Automatic Detection, Classification and Location of Transmission Line Faults," *5th Int. Conf. Energy, Power, Environ. Towar. Flex. Green Energy Technol. ICEPE 2023*, no. Ll, pp. 1–6, 2023, doi: 10.1109/ICEPE57949.2023.10201547.

[28] M. Ben Hessine and S. Ben Saber, "18. Jiale, S.; Guobing, S.; Xu, Q.; Chao, Q.

Time-domain fault location algorithm for parallel.pdf," vol. 2014, 2014.

[29] K. Hosseini, "Short circuit fault classification and location in transmission lines using a combination of wavelet transform and support vector machines," *Int. J. Electr. Eng. Informatics*, vol. 7, no. 2, pp. 353–365, 2015, doi: 10.15676/ijeei.2015.7.2.14.

[30] I. M. Karmacharya and R. Gokaraju, "Fault Location in Ungrounded Photovoltaic System Using Wavelets and ANN," *IEEE Trans. Power Deliv.*, vol. 33, no. 2, pp. 549–559, 2018, doi: 10.1109/TPWRD.2017.2721903.

[31] K. Y. Chowdary and S. Kumar, "Detection, Location, and Classification of Fault Applying Artificial Neural Networks in Power System Transmission Line," *Proc. 2022 IEEE Int. Conf. Curr. Dev. Eng. Technol. CCET 2022*, pp. 1–6, 2022, doi: 10.1109/CCET56606.2022.10080158.

[32] H. Nourmohamadi, G. Gohil, and P. T. Balsara, "Fault Location and Classification for MVDC Networks," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 10, no. 1, pp. 589–603, 2022, doi: 10.1109/JESTPE.2021.3111825.

[33] H. A. Naji, R. A. Fayadh, and A. H. Mutlag, "ANN-based Fault Location in 11 kV Power Distribution Line using MATLAB," *2023 IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. JEEIT 2023*, pp. 134–139, 2023, doi: 10.1109/JEEIT58638.2023.10185849.

[34] A. A. M. Alabbawi, I. I. Alnaib, O. S. A. D. Y. Al-Yozbaky, and K. K. Mohammed, "Faults detection, location, and classification of the elements in the power system using intelligent algorithm," *Bull. Electr. Eng. Informatics*, vol. 12, no. 2, pp. 597–607, 2023, doi: 10.11591/eei.v12i2.4456.

[35] A. M. Moldovan and M. I. Buzdugan, "Prediction of Faults Location and Type in Electrical Cables Using Artificial Neural Network," *Sustain.*, vol. 15, no. 7,

2023, doi: 10.3390/su15076162.

[36] V. Duarte, S. Zuniga-Jara, and S. Contreras, “Machine Learning and Marketing: A Systematic Literature Review,” *IEEE Access*, vol. 10, no. July, pp. 93273–93288, 2022, doi: 10.1109/ACCESS.2022.3202896.

[37] M. I. H. Pathan, M. S. Shahriar, M. M. Rahman, M. S. Hossain, N. Awatif, and M. Shafiullah, “Comparative Analysis of Machine Learning Approaches in Enhancing Power System Stability,” in *Artificial Intelligence-based Smart Power Systems*, 2023, pp. 157–177.

[38] T. S. Ng, “Machine learning,” *Stud. Syst. Decis. Control*, vol. 65, pp. 121–151, 2016, doi: 10.1007/978-981-10-1509-0_9.

[39] G. A. Watson, *Numerical Analytics*. 1998.

[40] M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993, doi: 10.1016/S0893-6080(05)80056-5.