ADDIS ABABA UNIVERSITY

Addis Ababa Institute of Technology (AAiT)

School of Electrical and Computer Engineering

Fault Location Estimator Design for Power Distribution System Using Artificial Neural Networks

By: Samuel Shawul Tessema

Thesis
Submitted to the School of Graduate Studies of Addis Ababa University
for Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Advisor:
Dr. Dereje Shiferaw

July, 2018

ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Fault Location Estimator Design for Power Distribution System Using Artificial Neural Networks

---

A Thesis in Electrical Engineering

By: Samuel Shawul Tessema
July 20,2018
Addis Ababa

A Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science

The undersigned have examined the thesis entitled **Fault Location Estimator Design for Power Distribution System Using Artificial Neural Networks** presented by **Samuel Shawul Tessema**, a candidate for the degree of **Master of Science** and hereby certify that it is worthy of acceptance.

Dereje Shiferaw (PhD)		
Advisor	Signature	Date
Dr. Eng. Getachew Biru		
Internal Examiner	Signature	Date
Mesfin Tilahun		
External Examiner	Signature	Date
Chair person	Signature	Date

Signed Declaration

This thesis is my original work and has not been presented for a degree in any other university, and that all sources of material used for the thesis have been duly acknowledged.

Name: Samuel Shawul Tessema

Signature:_____ Date _____

Place: Addis Ababa, Ethiopia

This is certifying that the above statement made by the student is correct and true to the best of my knowledge and belief. This thesis work has been submitted for examination with my approval as a university advisor.

Dr. Dereje Shiferaw

(Advisor)

Signature:_____ Date _____

Abstract

Fault location in distribution system is critical issue to increase the availability of power supply by reducing the time of interruption for maintenance in electric utility companies. In this thesis fault location estimator for power distribution system using artificial neural network is developed for line to ground, line to line, line to line to ground and three phase to ground faults in distribution system. To develop this estimator one of rural radial power distribution feeder in Ethiopia, Oromia, Assela substation Gumguma line feeder is used as a test feeder. This feeder is simulated using ETAP software to generate data for different fault condition, with different fault resistance and loading conditions, which is the fault phase voltage and current. The generated data is preprocessed and put as an input for neural network to be trained. MATLAB R2016a neural network toolbox to train ANN and programming toolbox is used to develop graphic user interface for fault estimator. The feed forward multi layer network topologies of neural network with improved back propagation, Levenberg Marquardt learning algorithm is used to train the network.

After the network (6-15-8-4) is trained the mean square error performance, regression plot and error histogram analysis was made and found to have an excellent performance with regression coefficient 0.99929 , validation performance of 0.000102 and error histogram range -0.015 to 0.019. In this thesis for practical implementation the fault records at the test feeder is handled by intelligent electronic device (IED) installed at the substation feeders. The fault record of IED can be read by PCM600 tool using laptop or manually using IEDs human machine interface, this fault recorded data feed to the graphic user interface to estimate the fault location as well as the fault type. Finally it is found that artificial neural networks are one of the alternate options in fault estimator design for distribution system where sufficient distribution network data are available with narrow fault location distance range from the substation. This has benefits in assisting for maintenance plan, saving efforts in fault location finding and economical benefits by reducing interruption time.

Key words: Artificial neural network, fault, power distribution system, intelligent electronic device (IED).

Acknowledgments

First of all I would like to thanks almighty God, for all things happened and not happened.

I would like to express my deep gratitude to my advisor **Dr. Dereje Shiferaw** for his willingness to be my advisor, his advice, suggestion and guidance throughout my thesis work. I would like to acknowledge Dr. Eng Getachew Biru for his suggestion in getting advisor to my thesis. I would like to acknowledge Mintesinot (EEU employee) for his help in getting data for distribution network and Jaswinder Sing(MSc.) for his help in providing thesis related documents for further reference. I would like to thanks Mr. Amare Assefa (MSc.) power engineering chair for being providing update information related to thesis schedule through my mail. I would like to thanks my family (father and mother) for their support and follow up in all my life.

Table of Contents

List of figures

List of tables

List of Abbreviations

W	bold capital letter matrix representation
p	bold small letter vector representation
FFNN	Feed-Forward Neural Network
FBNN	Feed-back Neural Network
ANN	Artificial neural network
EMTDC	Electro Magnetic Transient Design and Control
PSCAD	Professional's simulation tool for analyzing power systems transients
ETAP	Electrical transient analyzer program
MATLAB	Mathematics Laboratory
IEC	International Electronic Commission
DT	Distribution transformer
LV	Low voltage
MV	Medium voltage
ABB	Asian brown bivory
IED	Intelligent Electronic Device
PCM	Protection and Control Manger
DFT	Discrete Fourier Transform
RMS	Root Mean Square
IP	Internet protocol
LAN	Local area Network
WAN	Wide Area Network
LG	Line to ground
LLG	Line to Line to Ground
LL	Line to Line
3LG	Three phase to Ground
MSE	Mean square error
SSE	sum square error
IJEEE	International Journal of Electrical and Electronics Engineers
RBF	Radial basis function

CHAPTER 1

INTRODUCTION

1.1 Introduction

As the need for electricity is increasing in the world the expansion and installation in distribution networks to provide electric access to reach each customer in rural and urban is also increasing. With the installation of distribution lines most customers in developing countries are getting access for electricity and use in their daily activities. This access is not always reliable due to different reasons, mainly due to distribution system faults. In electric power system, a fault is any abnormal electric current flow in electrical component as defined in [1]. These faults cannot be completely avoided since a portion of these faults also occur due to natural reasons which are way beyond the control of mankind. Short circuit fault is a fault in which current bypasses the normal load. An open-circuit fault occurs if a circuit is interrupted by some failure. In three-phase systems, a fault may involve one or more phases and ground, or may occur only between phases. The prospective short circuit current of a fault can be calculated for power systems. In power systems, protective devices (relays, intelligent electronic devices) detect fault conditions and operate circuit breakers and other devices to limit the loss of service due to a failure. Power distribution lines and transmission lines are the main link in power system to address the generated power to the customers. Fault occurs due to failure of insulation of the distribution system, bridging of energized phase conductors by objects, accidents etc as mentioned in [2]. These events affect the value of the voltage and current on the distribution system and sometimes the entire power system. Electric power systems will always be exposed to the failure of their components. When a fault occurs on a line, it is crucial for the fault location to be identified as accurately as possible, allowing the damage caused by the fault to be repaired quickly before the line is put back into service. Fault location on power lines enables the technicians to pinpoint the location of a fault on power lines following a disturbance. If a fault location cannot be identified quickly and this causes prolonged line outage during a period of peak load, severe economic losses may occur and reliability of service may be questioned. The growth in size and complexity of power systems has increased the impact of failure to locate a

fault and therefore heightened the importance of fault location research studies, attracting widespread attention among researchers in recent years.

1.2 Problem statement

The average fault location time in city area is almost a half of the same time value in the rural areas where power lines are radial. Further, the average repairing time in city area is about one third of the same variable in the rural areas, because of several reasons. Operative maintenance team can do the fault location by systematic network topology changing in city area, which is impossible in radial power lines in rural areas. Usually, city is situated in the center of consumer and rural areas are in different directions around the center, so operative maintenance team is always in the space center during the normal power system function [1].

In electric utilities power distribution system frequently observed that the power interruptions due to different reasons, mainly due to different type of electrical faults. As reports indicate 70% of faults are single line to ground fault in [2]. Existing distribution system fault location is done by patrolling the distribution line from the substation or the switching station in urban. By opening sections and closing breakers step by step till circuit breaker trips and identified the faulty sections, which makes the electrical equipment stress and time taking to get faulty location, finally it may leads to failure to the distribution system equpments.Most of time during energization of new distribution line in rural it is too much exhaustive work to identify the fault location. From my experience in rural electrification faulty new distribution line energization will take more time to get the fault location as there is no means to know the fault location apart from visual inspection of insulators crack and conductors short. The maintenance time to clear the faulted line and restore the system is most of the time is too long, as records in the substations daily log sheet in EEU indicate on average 2-3 hours per day required to restore the rural feeder line. The maintenance team finds the fault location following the distribution line from the substation. It is too exhaustive work to find the fault location with branched and lengthy power distribution lines. Long time interruption has significant impact in the daily social and economical activities of the customers. The main problem is getting the actual fault location is the big task, apart from the customers call for their information for power interruption.

1.3 Overview of proposed solutions

The aim of this thesis is to design fault location estimator to estimate the location of the fault in overhead line of radial power distribution using the available fault record information at the intelligent electronic devices installed at the feeders of the substations using trained artificial neural network. To train the network the best way is to use data of the actual or practical test of the fault records, alternatively by modeling the distribution line using the distribution line parameter and simulate faults at different positions in the distribution network.Artificial neural networks are well known for solving many engineering problems related to classification and optimization. Its ability to recognize complex patterns has made it possible to use in locating a fault where the training input can be from measurement data such as voltage and current of feeder and the output is the location of fault.To train a neural network to perform some task, we must adjust the weights and bias of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights and bias. In this paper the fault classification and location estimation of medium voltage power distribution line in electric utility distribution network for the purpose of reducing power interruption durations by easily getting fault location and repairing to restore the distribution system for operation as much as possible in short time. Intelligent electronic devices (IEDs) or smart meters connected to the substation feeder are responsible to measure the voltage and current during fault. By simulating different type of the power distribution system faults such as line to ground, line to line and symmetrical faults at different location and training using artificial neutral networks to locate and detect the fault distance from the substation.

1.3.1 Main objective

Main objective of this thesis is to design fault location estimator for overhead line radial power distribution using the available fault record information at the intelligent electronic devices installed at the feeders of the substations by training the artificial neural network.

1.3.2 Specific objectives

- To analyze common type of faults by modeling test feeder in power distribution system using ETAP software to generate data for input to train the neural network.

- To discuss basic theories of artificial neural network.
- To train neural network to classify and locate fault using MATLAB software.
- To develop user interface for practical fault estimator implementation.

1.4 Literature review

1.4.1 Intoduction

A fault in the distribution system is located through conventional approaches, such as upon receiving a complaint from a customer, a technical staff is deployed to find the fault by patrolling the suspected faulted feeder. Meanwhile, for an underground cable system, switching operations were widely practiced to identify the faulted section. Thus, the locating process is time consuming and might expose additional stress to the equipment during the switching on/off of a section. This conventional fault detection rely on visual inspections of the faulted line parts resulting in long and tedious foot or aerial patrols. These methods were expensive and prone to more errors. Due to these problems, many automated fault location methods have been introduced by researches to expedite the process of locating faults. Some of the fault detection methods are selected from technical journal paper of previous similar works is reviewed in this section.

1.4.2 Impedance Based Method

In [3] in impedance based fault location methods are using impedance as seen from a monitored node to estimate the location of fault from measurement of voltage or current at the monitored substation. Based on Ohms Law, voltage and current from the monitoring node can be used to determine fault. The simple formulation of fault location solution with absolute values is:

$$d = \frac{V}{I * Z_l} \quad \text{..... (1.1)}$$

where: V ,voltage during the fault, Volt, I , current during the fault, Ampere, Z_l , line impedance in ohms per length unit, d ,distance to the fault, length unit such as miles.

In [4] some improvement to the impedance based fault location is made considering the cappacitive effect of the power distribution line. The obtained result shows the capacitive effect

should not be neglected in the distribution system fault location with impedance based.In [3] and [4] this method is good for less complex distribution network and the limitations with this method is as the complexity of distribution systems increase and various uncertainty factors such as length of conductors, type of conductor and cables and unknown fault resistance makes it difficult to address using impedance-based method fault location.

1.4.3 Technique based on travelling wave phenomenon

In [5] presents the travelling wave technique which is based on the reflection and transmission of the generated travelling waves along the faulty power networks. Traveling wave based fault location in distribution line or transmission line fault at any point on voltage wave propagates step wave toward both the source and load direction. One method of determining the fault location is precise time measurement of traveling wave arrival at both ends. Traveling wave uses naturally occurring surges and waves occurred by faults. The fault distance (d_f) can be calculated using the following expression:

$$d_f = \frac{V*(t_2 - t_1)}{2} \quad \text{..... (1.2)}$$

Where *V*, the velocity of the traveling wave, t_1 time when waveform started to travel, t_2 time when waveform arrived at the record node.The limitation with this method is single end line recorder and spreaded recorders along with the distribution line is used and it is costy for practical implementation. In [6] travelling wave based fault location technique is presented for radial distribution systems with distributed generation (DG). The proposed technique extracts the fault initiated high frequency components of the voltage signals, which are recorded only at the substation, by using wavelet transformation technique. The fault location procedure described here assumes that voltage measurements are available only at the sending end. The advantage of this approach is its insensitivity to naturally occurring in feed from the distributed generators during a fault. But sometimes in feed is typically unpredictable and makes the traveling wave based fault location methods vulnerable to errors.

1.4.4 Knowledge based fault location techniques

Uncertainty of line parameter affecting variables, such as length of cables and unknown fault resistance, coupled with the complex structure of distribution management systems tends to make fault location through impedance and travelling wave techniques inaccurate. As a result of this, knowledge-based technique for locating faults has receiving attention from researchers in the last few years. In general, the technique requires information such as substation and distribution switch status, line measurements, atmospheric conditions, and information provided by fault detection devices installed along the distribution feeders. This information is analyzed using artificial intelligence methods to locate a fault. The three major knowledge based techniques used in power systems are based on the following:

a) Artificial neural networks.
b) Fuzzy Logic systems (FL)
c) Hybrid methods

a) Artificial Neural Networks (ANN)

In paper [8] presents fault location in distribution system using neural network. In this case, the EMTDC simulation software is used in PSCAD environment. The root mean square fault voltage and currents are used as an input. The sigmoid transfer function is used between the input and hidden layers and the pure linear transfer function are used between the hidden and output layers the trained network initiates and determines the final output of the network, which is distance. The result obtained result is accurate, but the scope of the work is limited to short length low voltage distribution lines which has laterals and determined by the flags on branch's and no sub laterals.

In papers [9] and [10] present on detecting, classifying and locating faults on electric power transmission lines based on artificial neural networks. The methods employed make use of the phase voltages and phase currents as inputs to the neural networks. Feed forward networks have been employed along with back propagation algorithm for each of the three phases in the fault location process. Analysis on neural networks with varying number of hidden layers and neurons per hidden layer has been provided to validate the choice of the neural networks in each step. Simulation result demonstrates satisfactory performances of AAN in fault detection,

classification and location. In this paper the scope of work is limited to transmission lines.In [11] fault location in distribution network with distributed generation using radial basis neural network is presented, this scheme determines the fault by normalizing the fault current, using two radial basis function one of RBF that determine the distance between the distributed generator and the source feeder and the second RBF determines the exact fault line. This shows that ANNs are become best tool in fault location.

b) Fuzzy Logic systems (FL)

It is based on the theory of fuzzy sets that was developed for a domain in which definitions of activities and observations are fuzzy, or not well-described, without sharp boundaries. In [12] presents fault detection technique used fuzzy logic-based algorithm to identify types of faults in radial, unbalanced distribution system. The parameters used include fault resistance, fault inception angle, system topology and loading levels. A hybrid approach of neuro-fuzzy based learning and fault classification approach based on the online learning system was proposed. In this work, a method of fault location based on the conventional offline neuron controller approach is compared with the suggested hybrid approach for learning and convergent time evaluation for distributed systems.This method of fault detection applies the three phase feeder currents and phase voltages as the inputs to the fuzzy inference system (FIS). Different levels of the fault currents and voltages for different fault conditions on the distribution lines are classified into various degrees of membership functions low, normal and high. The limitation with this system is the fault location is classified in zone which is not clearly indicate the actual location.

c) Hybrid Methods

The hybrid approach for fault detection and location is adopted to formulate a new topology that checks the benefits offered by the combination of two or more methodologies. Toward this end, the main goal is to improve the accuracy and reliability of the resulting hybrid technique, while reducing or inhibiting the limitations of the individual methodologies. In [15] presented a new approach based on wavelet multi-resolution analysis and feed forward back propagation neural network. When a fault occurs in transmission line, it initiates a transition condition. These signals have a finite life i.e. they decay to zero in a finite time. Transients produce overvoltage

and over currents in the power system, which can damage it depending upon its severity; they also contain useful information which can be used for analyzing disturbances in transmission lines. The consequences of transients are presence of high frequency components in voltage and current fault signals. Fourier transform of a signal gives information about all the frequencies present in the signal but does not give any information about the time at which these frequencies were present. Wavelet transform is tool which helps the signal to analyze in time as well as frequency domain effectively. It provides non-uniform division of frequency domain i.e. it uses short window at high frequencies and long window at low frequencies. Using multi-resolution analysis a particular band of frequencies present in the fault signal can be analyzed.

In papers [13] and [14] developed a fault analysis system comprising of an expert system (ES), neural networks (NNs), and a fault analysis package. The inputs used were fault voltage and current waveforms, binary information from protection relays, and circuit breakers status. The ES performed the fault section estimation task, while the NNs took care of the fault detection, fault resistance, fault point, and multiple faults decisions. The fault analysis package is used to verify the decisions made by the NNs. The limitation with this method is much complex and expensive to practical implementation.

To summarize this section a number of literatutes on different fault location methods are reviewed. In [3] and [4] impedance based fault location is presented this method is good for less complex distribution and it is inaccurate for complex distribution system. In [5] traveling wave based is presented this method is costy as it requers fault recorders at the source and near the load, expensive for practical implementation. As the main concern of this thesis is on artificial neural network based fault location different papers [3], [6] are reviewed. In [8],[9] papers ANN based fault location is done in transmission lines and very short length of distribution lines.In this thesis the proposed fault location covers for radial medium voltage distribution systems, which has multiple laterals and sub laterals, longer lengths and different type of conductors sizes to fill the gaps in [8] and [9] which is the scope limitation to transmission line and short length distribution lines with some additional features which is ultimatly to show the fault distance and branch and sub branch of fault in the faulty feeder, by selecting one of the real operation test feeder in Ethiopia.

1.5 Thesis organization outline

This chapter gives an introduction to the power system faults, problem of statements and an overview of the solution to address the problem as well as the main objetive and the specific objectives of the reserch is described.As well as collection of different literetures review from prevous similar works. Fault location in distribution systems in different techniques.

Chapter 2: Describes about the power system faults mainly distribution systems faults, distribution system protections and different type of shunt faults and the fault conditions of the faults are described.

Chapter 3: Discusses the basic theory of the artificial neural network, overview of ANN, comparison of ANN and biological neuron, different types of ANN, benefits of ANN and different type of learning rules for ANN.

Chapter 4: Discusses distribution network simulation for data generation and artificial neural network training processes are described in detail. It also discusses the practical implementation of the fault location estimator design and the benefit of the estimator is described in detail.

Chapter 5: Describes the conclusion, recommendation and future works of the thesis.

CHAPTER 2

DISTRIBUTION SYSTEM FAULTS

2.1 Introduction

This chapter generally describes a typical distribution system and reviews the shunt faults in power distribution system. This will help us in the data collection from the simulation of distribution network in different conditions for different type of faults in distribution system. Two major components of fault resistance, arc and ground resistances are briefly described in this chapter.

2.2 Typical Distribution System

Distribution networks of an electric power system link bulk sources of energy to customers' facilities. If an outage occurs on a distribution circuit, supply to the customers is interrupted. It is estimated that 80% of all interruptions occur due to failures in distribution systems as mentioned in documents [16] and [17]. Figure 2-1 below shows a typical distribution system which includes sub-transmission circuits, substations, feeders, transformers. At the High Voltage/Medium Voltage (HV/MV) substations, voltages are stepped down to lower levels of, 33kV, 15kV and 0.400kV secondary circuits and services to customers' facilities. Distribution substation may include distribution transformers, buses, reactors, capacitors, circuit breakers, isolators and reclozers. The distribution transformers steps down the voltages from the medium voltage levels to lower levels for local distribution.

2.3 Distribution system fault detection and protection

The main objective of protection system is to minimize the duration of fault and to protect power equipments from damage. Distribution systems experience different type of series and shunt faults. The commonly used equipment for detecting and isolating the faulted circuits in a distribution system are fuses, relays, circuit breakers, and current and voltage transformers. Some of protection equipments are described below in [19].

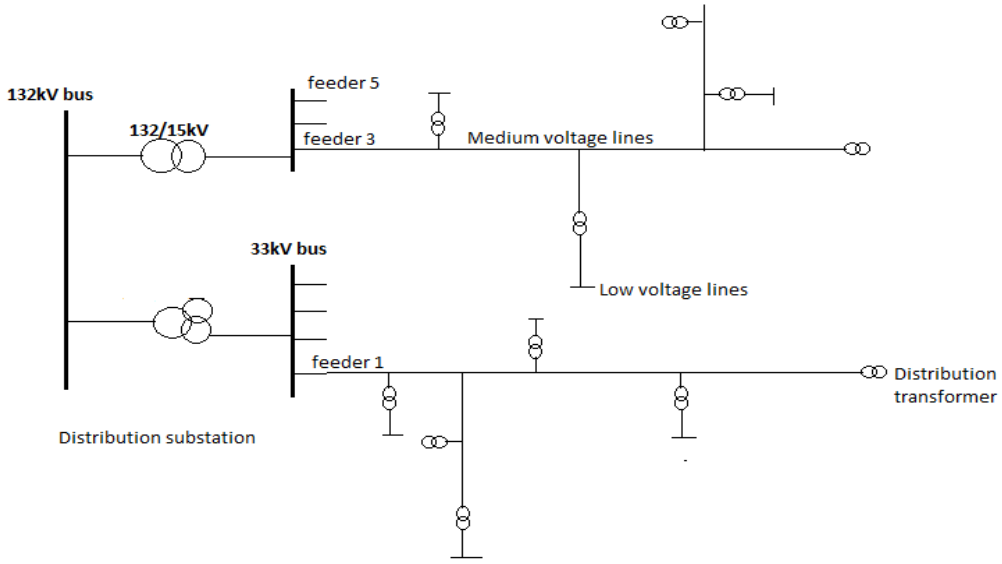


Figure 2-1 typical distribution system which includes sub-transmission circuits, substations, feeders, transformers

2.3.1 Fuses

A fuse is an over current protection device used in power system network. Under normal operating conditions, the heat built up in the fuse element is dissipated to the surrounding air and thus, the fuse remains at a temperature below its melting point. During fault conditions such as a short circuit, the heats become very great and cannot be dissipated fast enough. This causes the fuse element to heat up and melt, thereby breaking the circuit.

2.3.2 Instrument transformers

Instrument transformers are transducers used to transform high electric current and voltages to lower values proportional to the primary magnitudes thereby providing isolation between the electric power circuit and the measuring instruments. These transducers current transformers (CTs) and voltage transformers (VTs) measure the current and voltage in a network and provide low level signals to relays in order to detect abnormal conditions

2.3.3 Relays

A protective relay is a device capable of detecting changes in the received signal and if the magnitude of the received signal is outside a preset range, it operates to initiate appropriate

control action in order to protect the power system.To safeguard the investment in transmission and distribution lines, several types of protection techniques are used. Earth fault, over-current, differential, directional, etc are some of these techniques. A single technique or combinations of two or more techniques are employed to detect faults on transmission and distribution lines. The digital protective relay is a protective relay that uses a microprocessor to analyze power system voltages, currents or other process quantities for the purpose of detection of faults in an electric power system or industrial process system.

2.4 Power system faults cause and consequence

2.4.1 Fault

An electrical power system fault is the unintentional and undesirable creation of a conducting path or a blockage of electric current. It is any abnormal condition in a power system. The steady state operating mode of a power system is balanced 3-phase AC. However, due to sudden external or internal changes in the system, this condition is disrupted. When the insulation of the system fails at one or more points or a conducting object comes into contact with a live point, a short circuit or a fault occurs.

2.4.2 Causes of power system fault

The causes of faults in distribution system are numerous, for example:

- Lightning, heavy winds, trees falling across lines, vehicles colliding with towers or poles
- Birds shorting lines
- Aircraft colliding with lines
- Small animals entering switchgear
- Line breaks due to excessive loading

2.4.3 Consequences of faults

Fire is a serious result of major uncleared faults, may destroy the equipment of its origin, but also may spread in the system causing total failure. The short circuit, the most common type of fault that may have any of the following consequences as mentioned in [25]:

- A great reduction of the line voltage over a major part of the power system, leading to the breakdown of the electrical supply to the consumer and may produce wastage in production,
- An electrical arc – often accompanying a short circuit may damage the other apparatus in the system,
- Damage to the other apparatus in the system due to overheating and mechanical forces,
- Disturbances to the stability of the electrical system and this may even lead to a complete blackout of a given power system,
- Considerable reduction of voltage on healthy feeders connected to the system having fault, which can cause abnormal currents drawn by motors or the motors will be stopped (causing loss of industrial production) and then will have to be restarted.

2.4.4 Type of power system faults

Distribution system faults can be sub-divided into two major categories:

a) High impedance faults

b) Low impedance faults

2.4.4.1 High Impedance Faults (HIFs)

It can be defined as electrical contacts between a bare current carrying conductor and an insulated foreign object. This is usually as a result of a current carrying conductor touching a high impedance surface after breaking. Such surfaces can be the road, sand, grass, etc. and it is a threat to human life and the environment. Another variant is when the current carrying conductor does not break, but comes in contact with insulated grounded objects.

2.4.4.2 Low Impedance faults (LIFs)

It includes conventional shunt faults like:

- Single line-to-ground fault
- Line-to-line fault
- Double line-to-ground fault
- Balance 3 phase-to-ground fault

Experience has shown that between 70 and 80 percent of faults are single phase to ground fault. From statistics of faults 50% of fault is overhead line faults due to exposure to different

things and atmosphere as explained in [18]. All faults, except the three-phase faults causes power systems to operate in unbalanced modes.

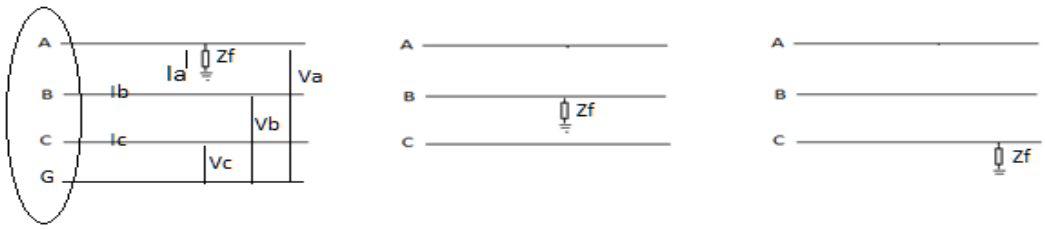
(a) Single-phase-to-ground faults

The following three types of single-phase-to-ground faults are experienced

(a) Phase A-to-ground faults.

(b) Phase B-to-ground faults.

(c) Phase C-to-ground faults.



(a) (b) (c)

Figure 2-2 single phase to ground fault with fault impedance

Consider a single line-to-ground fault from phase A to ground (G) at the general three-phase bus shown in Figure 2-2 (a). For generality, we include fault impedance Z_f. In the case of a bolted fault $Z_f = 0$. Fault conditions in phase domain Single line-to-ground fault of phase A:

$$I_b = I_c = 0 \; and \; V_a = Z_f I_a \quad \text{(2.1)}$$

Using the sequence transformation matrix fault conditions in sequence domain Single line-to-ground fault of phase A,

$$I_0 = I_1 = I_2 \; and \; V_0 + V_1 + V_2 = Z_f(I_0 + I_1 + I_2) = 3Z_f I_1 \quad \text{(2.2)}$$

Where $I_0, I_1, I_2 \; and \; V_0, V_1, V_2$ zero, positive and negative sequence current and voltage respectively.

Equation (2.2) can be satisfied if interconnection is made in series the sequence networks as follows:

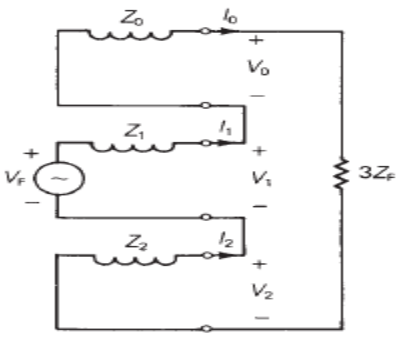


Figure 2-3 the sequence network connection for single phase to ground

From the figure the sequence fault current at phase a can be calculated:

$$I_a = 3I_1 = \frac{3V_a}{Z_0+Z_1+Z_2+3Z_f} \quad \text{..... (2.3)}$$

Similarly it can be calculated for other phases, detail derivation is found in [19].

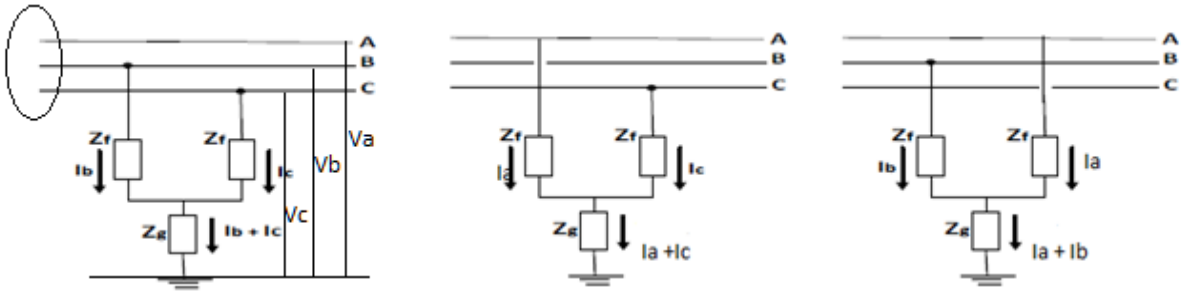
(b) Two-phase-to-ground faults

Two-phase-to-ground faults are of the following three types.

(a) Phase B and phase C-to-ground faults.

(b) Phase C and phase A-to-ground faults.

(c) Phase A and phase B-to-ground faults.



(a) (b) (c)

Figure 2-4 phase to phase to ground fault with fault resistance and ground resistance

Consider a line-to-line-to-ground fault from phase B and C to ground at the general three-phase bus shown in Figure 2-4 (a). For generality, we include a fault impedance Z_f. Fault conditions in phase domain line-to-line-to-ground fault of phase B and C:

$$I_a = 0 \; and \; V_b = V_c = (\frac{Z_f}{2} + Z_g)(I_b + I_c) \quad \text{(2.4)}$$

Using the sequence transforming matrix and transforming (2.4) to the fault conditions in sequence domain line-to-line-to-ground fault of phase B and C:

$$I_0 + I_1 + I_2 = 0 \; , \; V_0 - V_1 = (\frac{Z_f}{2} + 3Z_g)I_0 \; \; and \quad V_2 = V_1 \quad \text{(2.5)}$$

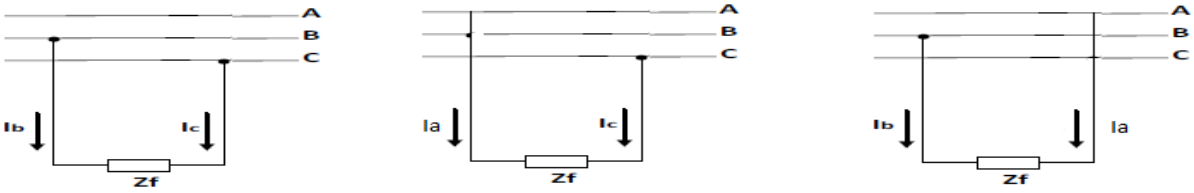
(c) Phase-to-phase faults

The three types of phase-to-phase faults that can be experienced on lines are as follows.

(a) Phase B-to-phase C faults.

(b) Phase C-to-phase A faults.

(c) Phase A-to-phase B faults.



(a) (b) (c)

Figure 2-5 phase to phase to ground fault with fault resistance and ground resistance

Consider a line-to-line fault from phase B and C to ground at the general three-phase bus shown in Figure 2-5 (a). For generality, we include a fault impedance Z_f. Fault conditions in phase domain line-to-line fault of phase B and C:

$$I_a = 0 \; , \; \; I_b = -I_c \; and \; \; V_b - V_c = Z_f I_b \quad \text{(2.6)}$$

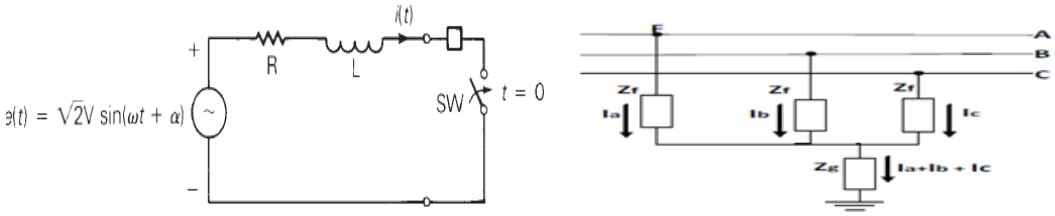
Using the sequence transforming matrix and transforming (2.6) to the fault conditions in sequence domain of line-to-line fault of phase B and C:

$$I_0 = 0 \; , \; I_1 = -I_2 \; \; and \; \; V_1 - V_2 = Z_f I_1 \quad \text{(2.7)}$$

(d) Balanced three-phase faults

Three phase faults that have equal fault resistances in the three phases are called balanced three-phase faults. These faults can be approximated at the terminals of unloaded synchronous machine. Let consider the series R-L circuit shown in figure 2-6 (a). For simplicity, assume zero fault impedance; that is, the short circuit is a solid or bolted fault. The current is assumed to be zero before SW closes, and the source angle $\boldsymbol{\alpha}$ determines the source voltage at t = 0. Writing the Kirchhoff's voltage law equation for circuit shown in figure 2-6 (a).

$$L\frac{di(t)}{dt} + Ri(t) = \sqrt{2}\,Vsin(wt+\alpha) \qquad t \geq 0 \text{……..(2.8)}$$



(a) (b)

Figure 2-6 phase to phase to ground fault with fault resistance and ground resistance

The solution to equation (2.8) which is the approximation of fault current is given by:

$$i(t) = i_{ac}(t) + i_{dc}(t) \text{……..(2.9)}$$

$$i(t) = \sqrt{2}\frac{V}{Z}(sin(wt+\alpha-\theta) + sin(\boldsymbol{\alpha}-\theta)e^{t/T})\text{A} \text{ ……..……(2.10)}$$

Where $Z = \sqrt{(wL)^2 + R^2} = \sqrt{X^2+R^2}\,Ohm$…….……(2.11)

$$\theta = \tan^{-1}\frac{X}{R} = \tan^{-1}\frac{wL}{R} \;\; and\; T = \frac{L}{R}\; second \text{ …..……..……(2.12)}$$

2.5 Fault Resistance

A fault resistance consists of two major components, arc resistance and ground resistance. It is either constant for the duration of a fault or it varies with time due to the elongation of the arc and its ultimate extinction. In phase to phase faults, fault resistance is entirely due to the arc. However, for faults involving the ground, fault resistances are composed of both the arc and

ground resistances. The ground resistance includes resistances of the contact between the conductor and the ground and the resistance of the ground path for the flow of current in the ground in situations where the snapped conductor touches the ground. In situations where a broken conductor touches the tower, the ground resistance includes resistance of the contact between the conductor and the tower, and the resistance of the ground path for the flow of current in the ground and tower footings. Usually, for inter-phase faults, fault resistances are small and in general values do not exceed 0.5 Ohm [18]. They may, however, become much higher during earth faults, because tower footing resistance may be as high as 10Ohm [18]. If there is a flashover of an insulator, the connection of towers with earth wires makes the resulting fault resistance smaller. In practice, it seldom exceeds 3ohm. For some earth faults the fault resistances may become much higher, which happens in cases of fallen trees, or if a broken conductor lays on the high-resistive soil. Different distribution system fault resistance values are compared in [7].

CHAPTER 3

THEORY OF ARTIFICIAL NEURAL NETWORK

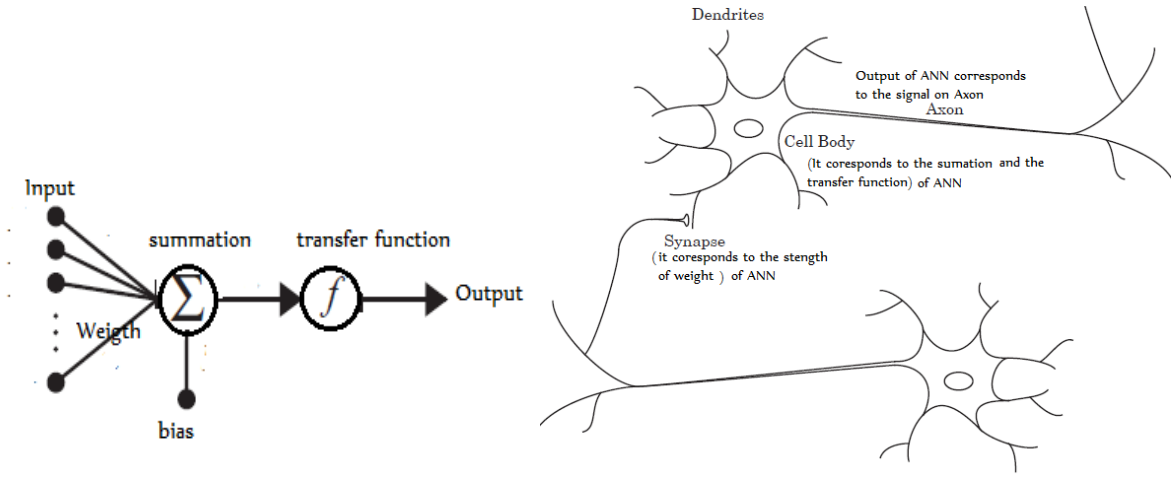
3.1 Literatures on artificial neural network

The modern view of neural networks began in the 1940s with the work that shows networks of artificial neurons could, in principle, compute any arithmetic or logical function as mentioned in [20]. This work is often acknowledged as the origin of the neural network field. The first practical application of artificial neural networks came in the late 1950s, with the invention of the perceptron network and associated learning rule mentioned in [21]. Rosenblatt and his colleagues built a perceptron network and demonstrated its ability to perform pattern recognition. The first successful neuro-computer was developed during 1957 and 1958 by Frank Rosenblatt, Charles Wightman, and others. Rosenblatt was the founder of neuro computing. Slightly later than Rosenblatt, Bernard Widrow developed a different type of neural network processing element called ADALINE (Adaptive linear neuron), which was equipped with a powerful new learning law which, unlike the perceptron learning law. The ADALINE network is very similar to the perceptron, except that its transfer function is linear, instead of hard-limiting, is still in widespread use as mentioned in [31]. A number of literatures are reviewed and compiled in [32] that give explanation on history and application of neural networks in control system. The applications of neural networks are expanding because neural networks are good at solving problems, not just in engineering, science and mathematics, but in medicine, business, finance and literature as well. Their application to a wide variety of problems in many fields makes them very attractive. Also, faster computers and faster algorithms have made it possible to use neural networks to solve complex industrial problems that formerly required too much computation as mentioned in [33].

3.2 Artificial neuron versus biological neuron

The idea of ANNs is based on the belief that working of human brain by making the right connections can be imitated using silicon and wires as living neurons and dendrites. The human brain is composed of 100 billion nerve cells called neurons. They are connected to other thousand cells by axons. Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the

neural network. A neuron can then send the message to other neuron to handle the issue. ANNs are composed of multiple nodes, which imitate biological neurons of human brain [9]. The neurons are connected by links (weights) and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value. Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values.The main similarities between biological and artificial neural networks are, the building blocks of both networks are simple eventhough artificial neurons are much simpler than biological neurons they are highly interconnected. and the connections between neurons determine the function of the network. The following figure 3-1 shows a simple ANN and biological neuron.



(a) (b)

Figure 3-1 simple artificial neural network and biological neuron

3.3 Advantages of neural network

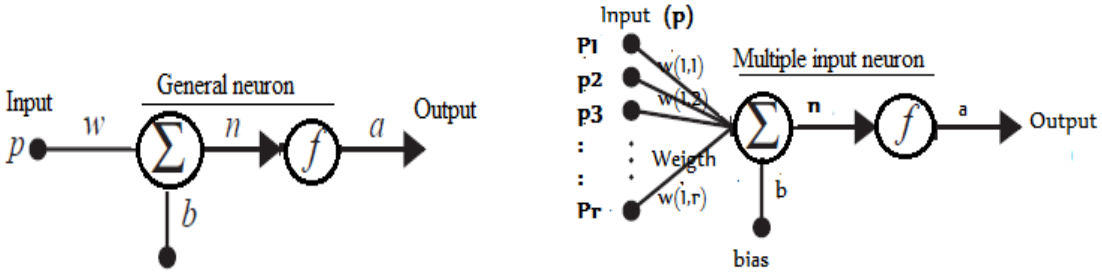
The following is a list of some of the advantages of ANNs:

1. Adaptivity: ANNs have the ability to adapt their synaptic weights to reflect changes. Also, a network can be easily retrained to adjust to some minor changes in the operating environment.

2. Robustness: Neural networks have demonstrated the ability to perform even when the inputs are degraded or noisy.
3. Flexibility: NN can be applied to several types of problems.
4. Generalization: ANNs when properly trained can provide the correct response to untrained input patterns that share similarity with the training patterns.

3.4 Artificial neural network model

Neuron can be either single input (a) or multiple input neurons (b) as figure shown below. For the single input neuron shown figure 3-2 (a) The scalar input, p, is multiplied by the scalar weight, w and the bias, b is added to form network out put, n that goes into a transfer function,f , which produces the scalar neuron output,a.



(a) (b)

Figure 3-2 shows single and multiple input neurons

The output of the single input neuron is calculated as: $n = wp + b$,

$$\boldsymbol{a} = f(n) = f(wp + b) \quad \text{.....(3.1)}$$

The actual output depends on the particular transfer function that is chosen.Multiple input neuron has more than one input. A neuron with inputs P_r is shown in Figure above (b). The individual inputs are each weighted by corresponding elements $w_{(1,1)}$, $w_{(1,2)}$,..... $w_{(1,r)}$ of the weight matrix **W.** The output of the multiple input neurons in [24] calculated as:

$$n = w_{(1,1)}p_1 + w_{(1,2)}p_2 + \ldots\ldots + w_{(1,r)}p_r + b \text{ ,}$$

$$\boldsymbol{a} = f(n) = f(w_{(1,1)}p_1 + w_{(1,2)}p_2 + \ldots + w_{(1,r)}p_r + b) \quad \ldots\ldots(3.2)$$

in matrix form, $\boldsymbol{a} = f(\boldsymbol{W}\boldsymbol{p} + b)$.

A particular transfer function in a neuron is chosen to satisfy some specification of the problem that the neuron is attempting to solve. There are variety of transfer functions mentioned in [22] some of the commonly used transfer functions are hardlimit,linear and log-sigmoid transfer functions which are discussed below.

(a) Hard-Limit Transfer Function

This transfer function operates by restricting the output to 0 if the net input n for the neuron is below 0 or gives a 1 if n is greater than or equal to 0.

$$Hardlim(0) = \begin{cases} 1, & n \geq 0 \\ 0 & \text{if otherwise} \end{cases}$$

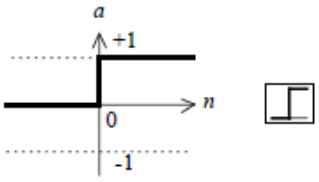
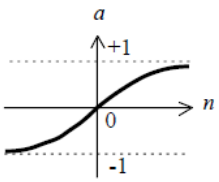


Figure 3-3 *hardlim(n)* function

(b) *Logsig* and *tansig* Transfer Function

A squashing function of the form shown below that maps the input to the interval (0,1) is:

$$tansig(n) = \frac{2}{(1+e^{-2n})-1} \qquad ,logsig(n) = \frac{1}{1+e^{-n}}$$



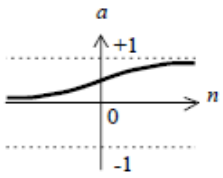


Figure 3-4 the *Logsig* and *tansig* transfer function

(c) Purelin Transfer Function

Purelin(n)= n

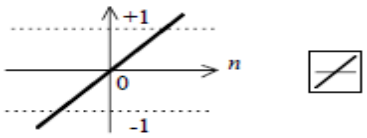


Figure 3-5 the purelin transfer function

3.5 Topologies of neural network

Neural networks can generally be classified according to their network topology. These are:

1. Feed forward neural network and
2. Feedback neural network

3.5.1 Feed forward neural networks

In feed forward neural networks information moves only in the forward direction from the input layers, via the hidden layers to the output layers. It is the most popular and commonly used network in practical application. Feed forward networks are static, that is, they produce only one set of output values rather than a sequence of values from a given input. Feed forward networks are memory-less in the sense that their response to an input is independent of the previous network state. Examples of feed-forward networks include the single-layer perceptron, multi-layer perceptron, radial basis function, learning vector quantization network, probabilistic neural network, generalized regression neural network, etc.

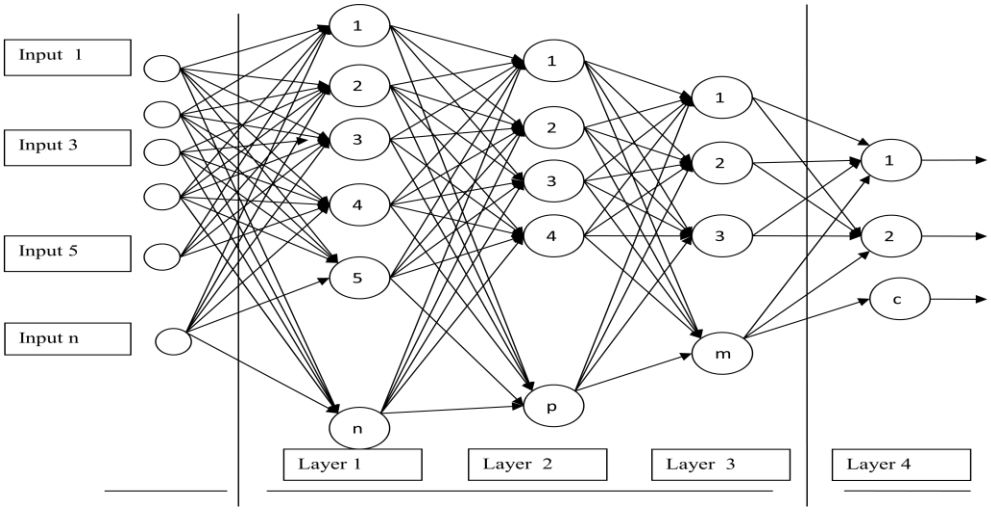


Figure 3-6 Feed-Forward neural network topology (multi layer neuron)

Figure above shows neurons with several layers.A layer whose output is the network output is called an output layer. The other layers are called hidden layers. The network shown above has an output layer (layer 4) and three hidden layers (layers 1, 2 and 3). Few examples of feed forward neural networks are described below.

3.5.1.1 Single Layer Perceptrons

Single line perceptron has only input and output neutrons. It has no hidden layer

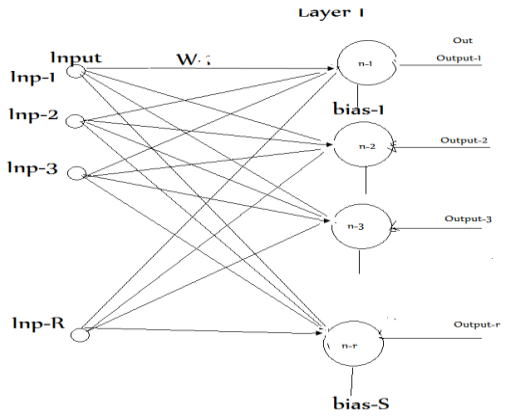


Figure 3-7 single layer perceptrom

3.5.1.2 Multilayer Perceptrons

The multilayer perceptrons consist of the input layer, hidden layer, and an output layer. The input layer and hidden layer are referred to as source nodes, while the output layers are regarded as computational nodes. The input layer propagates signals through the network in a forward direction from layer to layer. Figure 3-6 shows the one of the feed-forward neural network topology, multi layer neuron. Multilayer perceptron is used in this thesis.

3.5.1.3 Radial Basis Network

The radial basis network is a two-layer network. There are two major distinctions between the radial basis function (RBF) network and a two layer perceptron network. First, in layer 1 of the RBF network, instead of performing an inner product operation between the weights and the input (matrix multiplication), we calculate the distance between the input vector, **p** and the rows

of the weight matrix,**w**. Second, instead of adding the bias,b, we multiply by the bias. Therefore, the net input for neuron i in the first layer is calculated as follows:

$$n_i^1 = \|p - w_i^1\| b_i^1 \quad \text{..... (3.3)}$$

Each row of the weight matrix acts as a center point, a point where the net input value will be zero. The bias performs a scaling operation on the transfer (basis) function, causing it to stretch or compress. Radial basis function is local. This means that the output is close to zero if you move very far in either direction from the center point. The output layer of the RBF network is a standard linear layer.

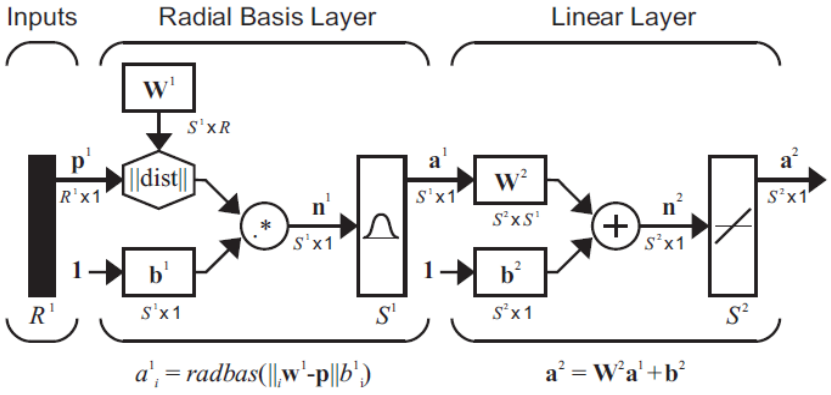


Figure 3-8 shows radial basis network taken from [22]

Multi layer perceptron and RBF networks are universal approximators, they perform their approximations in different ways. For the RBF network, each transfer function is only active over a small region of the input space the response is local. If the input moves far from a given center, the output of the corresponding neuron will be close to zero. This has consequences for the design of RBF networks. We must have centers adequately distributed throughout the range of the network inputs, and we must select biases in such a way that the entire basis functions overlap in a significant way.

3.5.2 Feedback neural networks

In the feed-back neural networks there are feedback connections from one layer to another. That is, there is a bi-directional data flow and data are also propagated from the outputs to the input layers. Feedback, networks are dynamic systems. When a new input pattern is presented,

the neuron outputs are computed. Because of the feedback paths, the inputs to each neuron are then modified, which leads the network to enter a new state. Examples of feed-back networks include Elman networks, recurrent network, etc.

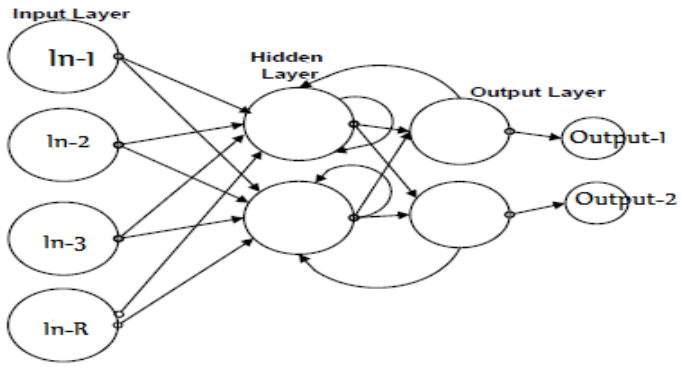


Figure 3-9 Feed-backward neural network topology (Recurrent network)

3.6 Training artificial neural network

The most significant property of a neural network is that it can train, and can improve its performance through training. Training is a process by which some parameters of a neural network i.e. synaptic weights and bias are adapted through a continuous process of stimulation in the environment in which the network is embedded. The network becomes more knowledgeable about the tasks after each iteration. The objective of the training process is to adjust the ANN weights and biases to obtain minimal deviations between the target and the calculated ANN outputs in relation to the average of all input samples. The finally found bias and weight matrices which applied to the network should hopefully map any input to a correct output.

3.6.1 Learning (training) Types

There are three types of learning paradigms namely, supervised learning, self-organized or unsupervised learning, and reinforced learning.

(a) Supervised Learning

The learning algorithm would fall under this category if the desired output for the network is also provided with the input while training the network. By providing the neural network with

both an input and output pair it is possible to calculate an error based on its target output and actual output. It can then use that error to make corrections to the network by updating its weights.

(b) Unsupervised Learning

In unsupervised learning, the weights and biases are modified in response to network inputs only. There are no target outputs available. This is especially useful in such applications as vector quantization.

(c) Reinforcement Learning

Reinforcement learning is similar to supervised learning, except that, instead of being provided with the correct output for each network input, the algorithm is only given a grade. The grade is a measure of the network performance over some sequence of inputs.

3.6.2 Training (Learning) Rules

Learning rules (training algorithms) refer to a procedure for adjusting the weights and biases of neural networks with the aim of minimizing the discrepancies between the network outputs and the training targets.The purpose of the learning rule is to train the network to perform some task. In incremental training, the network elements (weights) are updated after each training vector has been presented. Whereas, batch training, involves updating the weights and biases after all of the input and target vectors have been presented to the network. There are different kinds of learning algorithm. For example, the perceptron learning rule, back-propagation algorithm, conjugate gradient descent algorithm, quick propagation algorithm, Quasi-Newton algorithm, Levenberg-Marquardt algorithm, and Kohonen training, error correction learning, Boltzmann learning, Hebbian learning and competitive learning. Some of these learning rules that help this thesis are disscused below.

3.6.2.1 Perceptron Learning Rule

This learning rule is an example of supervised training, in which the learning rule is provided with a set of examples of proper network behavior. Each input is applied to the network, the network output is compared to the target. The learning rule then adjusts the weights and biases of

the network in order to move the network output closer to the target. From figure 3-2 and Equation (3.1) output of the network is given by:

$$\boldsymbol{a} = f(\boldsymbol{W}\boldsymbol{p} + b) \quad \text{.....(3.4)}$$

Where f is the transfer function, $\boldsymbol{W}$ weight matrix , $\boldsymbol{p}$ input vector and $\boldsymbol{b}$ bias.

The weight and bias adjestment for multilayer perceptron in this rule mathimatical expressed as:

$$\boldsymbol{W}^{new} = \boldsymbol{W}^{old} - (\boldsymbol{t} - \boldsymbol{a})\boldsymbol{p}^{\boldsymbol{T}} \quad \text{.....(3.5)}$$

$$\boldsymbol{b}^{new} = \boldsymbol{b}^{old} - (\boldsymbol{t} - \boldsymbol{a}) \quad \text{.....(3.6)}$$

where $\boldsymbol{t}$ target output, $\boldsymbol{a}$ output of the network, $\boldsymbol{W}^{new}$ new weight matrix and $\boldsymbol{W}^{old}$ the previous weight matrix ,$\boldsymbol{p}^{\boldsymbol{T}}$ transpose of input vector, $\boldsymbol{b}^{new}$ new bias vector and $\boldsymbol{b}^{old}$ previous bias vector values. The drawback of the perceptron rule is that it is only suitable for linearly separable samples and would fail to converge if the samples are not linearly separable.

3.6.2.2 Back propagation learning method

Back-propagation algorithm rule is the most frequently used algorithm for feed forward multilayer neural network. It is a supervised learning algorithm which requires a set of training data with known input and output vectors. It uses steepest gradient descent of error which propagates backwards for updating the synaptic weights and bias. In multilayer networks with nonlinear transfer function, the chain rule is used in the calculation of the mean square error (MSE) derivatives. The advantage of this algorithm is the simplicity of calculation during weight updates the back propagation (of error) or generalized delta rule algorithm summurized as follows:

Step 1: It propagates the input forward through the network:

$$\boldsymbol{a}^{\boldsymbol{m+1}} = \boldsymbol{f}(\boldsymbol{W}^{\boldsymbol{m+1}}\boldsymbol{a}^{\boldsymbol{m}} + \boldsymbol{b}^{\boldsymbol{m+1}}) \text{ for m= 1,2,3.....M, } \text{.....(3.7)}$$

where M number of layer

$a^0 = \mathbf{p}$, the neuron in the first layer is the external input

$a^M = \mathbf{a}$ the layer is considerd as the net output

Step 2: to propagate the sensitivities backward through the network, the sensitivity of F to changes in the i^{th} element of the net input at layer m, F is the function of weight,net output and bias.

$$s_i^M = \frac{\partial F}{\partial n_i^m} \ , where \ , n_i^m = \sum_{j=1}^{s^{m-1}} W_{i,j}^m \ a_j^{m-1} + b_i^m \quad \text{.....(3.8)}$$

We can now write out the recurrence relation for the sensitivity by using the chain rule in matrix form

$$s_i^M = \partial F/(\partial \boldsymbol{n}^m) = (\frac{\partial n^{m+1}}{\partial n^m})^T \frac{\partial F}{\partial n^{m+1}}) = \dot{\boldsymbol{F}}^m(n^m)\dot{(}\ W^{m-1}) \frac{\partial F}{\partial n^{m+1}}$$

$$= \dot{\boldsymbol{F}}^m(\ W^{m-1})\ s^{m+1} \quad \text{.....(3.9)}$$

Step 3: the weights and biases are updated using the approximate steepest descent rule,α learning rate

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T \quad \text{.....(3.10)}$$

$$b^m(k+1) = b^m(k) - \alpha s^m \quad \text{.....(3.11)}$$

Detail derivation of backpropagation algorithm is desribed in [23].The limitations of the back-propagation training algorithm are the training time, no guarantee for convergence to a global minimum, no exact rule for setting the number of hidden layers and neurons for the best performance, and instability if the learning rate is too large.

3.6.2.3 Improved Variations of the Back-Propagation Method

The practical implementation back propagation algorithm requires long training time for large network with large number of training patterns. Some methods have been developed to overcome the slow rate of learning which is discussed below.

a) Heuristic Modifications of Back propagation

(i) Batching

The parameters (weights and bias) are updated only after the entire training set has been presented. The gradients calculated for each training example are averaged together to produce a more accurate estimate of the gradient.

(ii) Momentum

The back-propagation algorithm minimizes the criterion by the method of steepest descent. The smaller we make the learning rate parameter α, the smaller will the changes to the synaptic weights in the network one iteration to the next and the smoother will be the trajectory in weight

space. This improvement, however, is attained at the cost of slower rate of learning. If on the hand, we make the learning parameter α too large so as to speed up the rate of learning the learning process may become unstable. A simple method of increasing the rate of learning and yet avoiding the danger of instability is to modify the delta rule by including a momentum term. Another feature of momentum is that it tends to accelerate convergence when the trajectory is moving in a consistent direction. When the momentum filter is added to the parameter (weight and bias) and the equations for the momentum modification to back propagation is as follows:

$$\Delta W^m(k) = \gamma \Delta W^m(k-1) - (1-\gamma)\alpha s^m (a^{m-1})^T \quad \text{.....(3.12)}$$

$$\Delta b^m(k) = \Delta b^m(k-1) - (1-\gamma)\alpha s^m \quad \text{.....(3.13)}$$

where γ is the momentum coefficient that must satisfy 0 <γ< 1.

(iii)Variable Learning Rate

There are many different approaches for varying the learning rate. We will describe a very straight forward batching procedure where the learning rate is varied according to the performance of the algorithm. The rules of the variable learning rate back propagation algorithm (VLBP) described in [24] are:

1. If the squared error (over the entire training set) increases by more than some set percentage P (typically one to five percent) after a weight update, then the weight update is discarded, the learning rate is multiplied by some factor 0 <q< 1 and the momentum coefficient(if it is used) is set to zero.
2. If the squared error decreases after a weight update, then the weight update is accepted and the learning rate is multiplied by some factor h>1. If γ has been previously set to zero, it is reset to its original value.
3. If the squared error increases by less than P, then the weight update is accepted but the learning rate is unchanged. If γ has been previously set to zero, it is reset to its original value.

b) Numerical Optimization Techniques

(i) Steepest decent

The steepest descent method is a general minimization method which updates parameter values in the downhill direction. Performance index is quantitative measure of network performance. Let the performance index that we want to minimize F(**x**) where **x** is a vector

$$\boldsymbol{F(x) = F(x_1, x_2, x_3 \ldots x_n)} \quad \text{.....(3.14)}$$

We begin from some initial guess,x_0 and then update our guess in stages according to an equation of the form

$$x_{k+1} = x_k + \alpha_k \boldsymbol{p}_k \quad \text{.....(3.15)}$$

Or $\Delta x_k = (x_{k+1} - x_k) = \alpha_k p_k$ ..... (3.16)

Where p_k is the search direction, and α_k is learning rate, we would like to have decreasing function, $F(x_{k+1}) < F(x_k)$. The direction of $\mathbf{p_k}$ is chosen, for sufficiently small learning rate by considering first order Taylor series expansion of F(x) , for old guess about $\mathbf{x}_k$:

$$F(x_{k+1}) = F(x_k + \Delta x_k) \approx F(x_k) + g_k^T \Delta x_k \quad \text{.....(3.17)}$$

Where g_k is the gradient at x_k

$$g_k = \nabla F(x)]_{x=x_k} \quad \text{.....(3.18)}$$

For $F(x_{k+1})$ to be less than $F(x_k)$ from (3.15) $g_k^T \Delta x_k$ must be negative $g_k^T \Delta x_k = \alpha_k g_k^T p_k < 0$, for α_k small and positive

$$g_k^T p_k < 0 \quad \text{.....(3.19)}$$

For any $p_k$vector that satisfy (3.19) is decent direction, for a function to be steepest when the $g_k^T p_k$ is most negative. It will be most negative when the direction vector is the negative of the gradient that is when $p_k = -g_k$

Using iteration the steepest decent method produced as follows:

$$x_{k+1} = x_k - \alpha_k p_k \quad \text{.....(3.20)}$$

(ii) Newton Method

The Newton method is a method for minimizing a sum-of-squares performance index function. It presumes that the universal performance index function is approximately quadratic in the parameters near the optimal solution [6].

$$F(x) = \frac{1}{2}x^T \boldsymbol{A}x + \boldsymbol{d}^T x + c \text{……..(3.21)}$$

Where the matrix A is symmetric, d is vector and c is constant

Gradient: $\nabla F(x) = Ax + d$ ……..……..…(3.22)

and hessian matrix, $\nabla^2 F(x) = A$, ……..……..…….. (3.23)

All higher derivatives of the quadratic function are zero. Newton's method is defined based on the second-order Taylor series:

$$F(x_{k+1}) = F(x_k + \Delta x_k) \approx F(x_k) + g_k^T \Delta x_k + \frac{1}{2}\Delta x_k^T A_k \Delta x_k \text{ ……... (3.24)}$$

The principle behind Newton's method is to locate the stationary point of this quadratic approximation to F(x). If we use Eq. (3.22) to take the gradient of this quadratic function with respect toΔx_k and set it equal to zero, we find

$$g_k + A_k \Delta x_k = 0 \text{ …… ..……. (3.25)}$$

solving for Δx_k, $\Delta x_k = -A_k^{-1} g_k$ then newtons method is defined :

$$x_{k+1} = x_k - A_k^{-1} g_k \text{…….……..…. (3.26)}$$

(iii) Conjugate Gradient

Steepest descent is the simplest algorithm, but is often slow in converging. Newton's method is much faster, but requires that the Hessian matrix and its inverse be calculated. The conjugate gradient algorithm is something of a compromise; it does not require the calculation of second derivatives, and yet it still has the quadratic convergence property. (It converges to the minimum of a quadratic function in a finite number of iterations.)

The congugate gradient backpropagation algorithm is summurized as follows:

1. Select the first search direction to be the negative of the gradient.

   Po = g_0 ,where $g_k = \nabla F(x)]_{x=xk}$

2. selecting the learning rate , $\propto_k$ tominimize the function along the search direction:

$$x_{k+1} = x_k + \alpha_k p_k \text{…….(3.27)}$$

3. Select the next search direction

$$p_k = -gx_k + \beta_k p_{k-1} \quad \text{(3.28)}$$

$$\text{With } \beta_k = \frac{\Delta g_{k-1}^T g_k}{\Delta g_{k-1}^T p_{k-1}} \; or \; = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} or \; = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}} \quad \text{(3.29)}$$

4. If the algorithm has not converged, continue from step 2.

(iv)Levenberg Marquardt Algorithm

The Levenberg-Marquardt algorithm is a variation of Newton's method that was designed for minimizing functions that are sums of squares of other nonlinear functions. This is very well suited to neural network training where the performance index is the mean squared error.

Basic algorithm described in [24]:

From Newton's method for optimizing a performance index F(x) is

$$x_{k+1} = x_k - A_k^{-1} g_k \quad \text{(3.30)}$$

$$g_k = \nabla F(x)]_{x=xk} and \; \nabla^2 F(x) = A_k$$

If we assume that *F(x)* is a sum of squares function, *v(x)*

$$F(x) = \sum_{i=1}^{N} v_i^2 (x) = v(x)^T v(x) \quad \text{(3.31)}$$

then the j[th] element of the gradient would be

$$[\nabla F(x)]_j = \frac{\partial F(x)}{\partial x_j} = 2 \sum_{i=1}^{N} v_i (x) \frac{\partial v_i(x)}{\partial x_j} \quad \text{(3.32)}$$

The gradient can therefore be written in matrix form:

$$\nabla F(x) = 2J(x)^T v(x) \quad \text{(3.33)}$$

Where ***J(x)*** is Jacobian matrix

$$\boldsymbol{J}(x) = \begin{bmatrix} \frac{\partial v_1(x)}{\partial x_1} & \frac{\partial v_1(x)}{\partial x_2} & \cdots & \frac{\partial v_1(x)}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial v_n(x)}{\partial x_1} & \frac{\partial v_n(x)}{\partial x_2} & \cdots & \frac{\partial v_n(x)}{\partial x_n} \end{bmatrix} \quad \text{(3.34)}$$

Next we want to find the Hessian matrix. The k,j element of the Hessian matrix would be

$$[\nabla^2 F(x)]_{k,j} = \frac{\partial^2 F(x)}{\partial x_k \partial x_j} = 2\sum_{i=1}^{N}\left\{\frac{\partial v_i(x)}{\partial x_k}\frac{\partial v_i(x)}{\partial x_j} + v(x)\frac{\partial^2 v_i(x)}{\partial x_k \partial x_j}\right\} \quad \text{(3.35)}$$

The Hessian matrix can then be expressed in matrix form:

$$\nabla^2 F(x) = 2J(x)^T J(x) + 2\sum_{i=1}^{N} v_i(x)\nabla^2 v_i(x) \quad \text{(3.36)}$$

If we assume that is $\sum_{i=1}^{N} v_i(x)\nabla^2 v_i(x)$ is small, we can approximate the Hessian matrix as

$$\nabla^2 F(x) \cong 2\boldsymbol{J}(x)^T \boldsymbol{J}(x) \quad \text{(3.37)}$$

If we then substitute Eq. (3.37) and Eq. (3.36) into Eq. (3.35), we obtain

$$x_{k+1} = x_k - [2J(x_k)^T J(x_k)]^{-1} 2J(x_k)^T v(x_k)$$

$$= x_k - [J(x_k)^T J(x_k)]^{-1} J(x_k)^T v(x_k) \quad \text{(3.38)}$$

One problem with the equation (3.38) is that the matrix H= $J^T J$ may not be invertible. This can be overcome by using the following modification to the approximate Hessian matrix:

G= H+ μI and therefore the matrix will be invertible. Detail derivation is found in[24]

Where **μ** is a damping factor that ensures the positiveness of the Hessian and **I** is the identity matrix. This leads to the Levenberg Marquardt algorithm

$$x_{k+1} = x_k - [\boldsymbol{J(x_k)^T J(x_k)} + \boldsymbol{\mu_k I}]^{-1} \boldsymbol{J(x_k)^T v(x_k)} \quad \text{(3.39)}$$

Or

$$\Delta x_k = -[\boldsymbol{J(x_k)^T J(x_k)} + \boldsymbol{\mu_k I}]^{-1} \boldsymbol{J(x_k)^T v(x_k)} \quad \text{(3.40)}$$

CHAPTER 4

DISTRBUTION NETWORK SIMULATION AND ARTIFICIAL NEURAL NETWORK TRAINING PROCESS

4.1 Introduction

A typical distribution feeder (figure 2-1 shown in chapter 2) is characterized by having only one path for power flow from the source to the load centers. Feeders may consist of the following: Main feeder, branches or tap-offs, loads and distribution transformers. A test feeder considers under this thesis is one of radial medium voltage (MV) rural power distribution feeder located in Ethiopia, Oromia region. It is taped from Assela substation (latitude 7º54'42''N, 39º04'59''E longitude), this distribution substation gives power supply to the industries around Assela and the town as well as the rural villages that located around Assela with in zone. It is lightly loaded with 1.3MW and has maximum length of 100.6km feeders with rated voltage of 33kV.The main reason for the selection of this as test feeder is that the length of the feeder suitability for the case studies and easily availability of the data for the distribution lines and the substation in detail. The distribution feeder data and substation data is obtained by letter communication, site visit and some of the data collected through phone communication. Test feeder single line diagram in ETAP is show in figure 4-1 below. This chapter covers the generation of the data required for the development of the fault location method and the artificial neural network training process.

4.2 Experiments for data generation

Extensive simulations were done in order to obtain data for fault conditions under different loading condition and different fault positions with different fault resistance values. The steady state simulation covered is to analyze the power flow in different loading condition of the feeder. Fault conditions involving different fault types namely single line-to-ground (LG), line to line (LL), double line to ground (LLG) and three line to ground (3LG.) faults were simulated on the test feeder respectively. The above mentioned fault types were simulated using various locations in the test feeder with different fault resistances values in distribution system as described in chapter two.

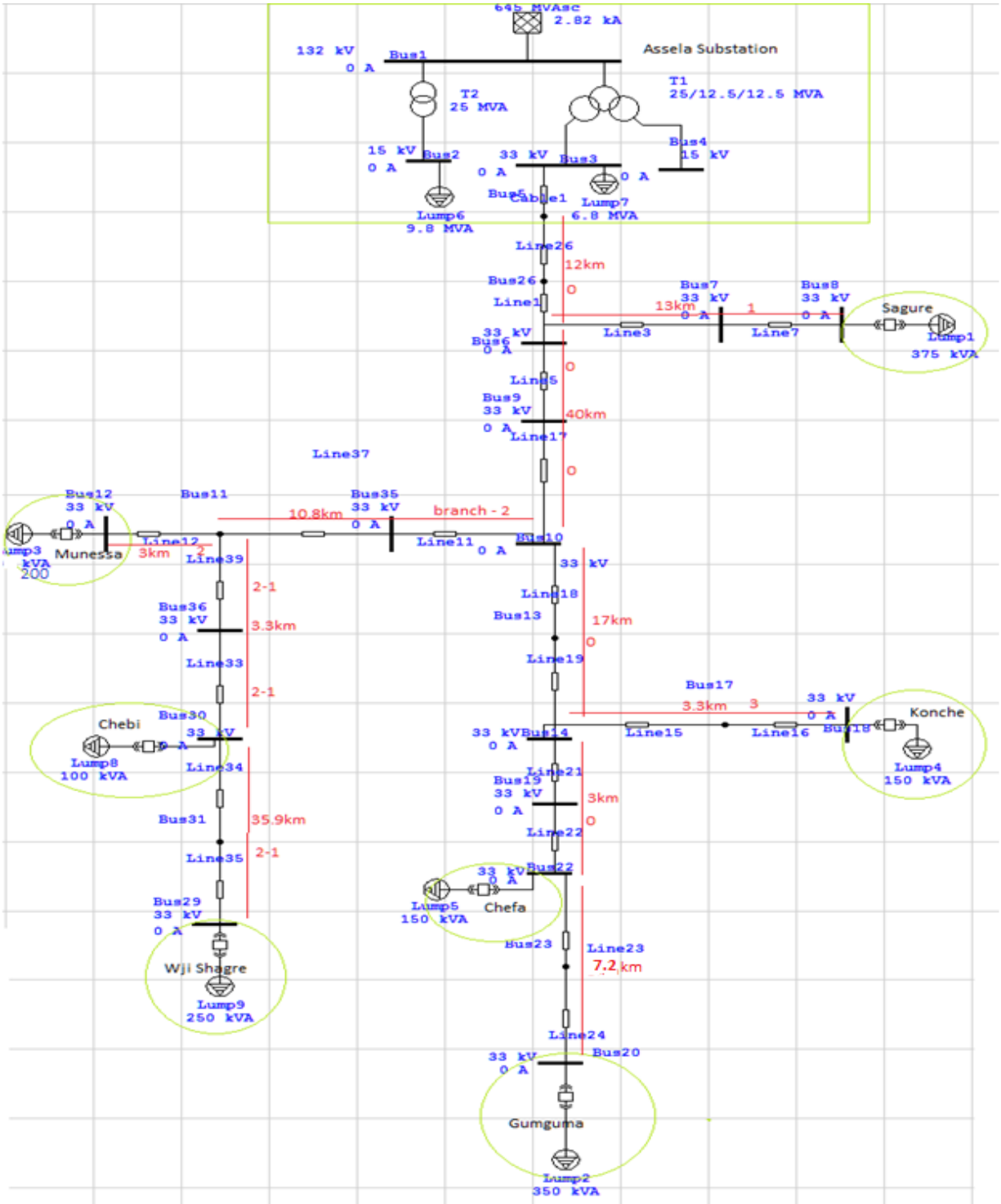


Figure 4-1 Single line diagram for test feeder simulated in ETAP software

4.2.1 Feeder Modeling

The distribution test feeder is modeled using ETAP (Electrical transient analyzer program) power station 4.0. It consists of distribution substations, radial medium voltage overhead line that distributes power to six rural villages at each taping point. There is section switches, three main laterals as well as sub laterals on branch two as shown in figure 4-1. Detail configuration of the line is mentioned in appendix B. There are 29 nodes in the feeder; these nodes are used to make faults at different length of the line from the substation. ETAP software version 11.4 with power station environment of 4.0 is used for simulation of distribution feeder for data generation. This software use IEEE/ANSI standards to do analysis in the simulation of electrical systems.

4.2.2 Transformer Models

The voltage source used for the test feeder was an external grid provided in the ETAP library. Two transformer models were made use of in this feeder. The transformer is the substation power transformer. The power transformers at the substation are 25/12.5/12.5MVA, 132/33/15kV transformer and 25MVA 132/15kV transformer. The parameters used are shown in appendix A. The parameters for the configuration of the transformers model are shown in figures 4-2(a) and 4-2(b)

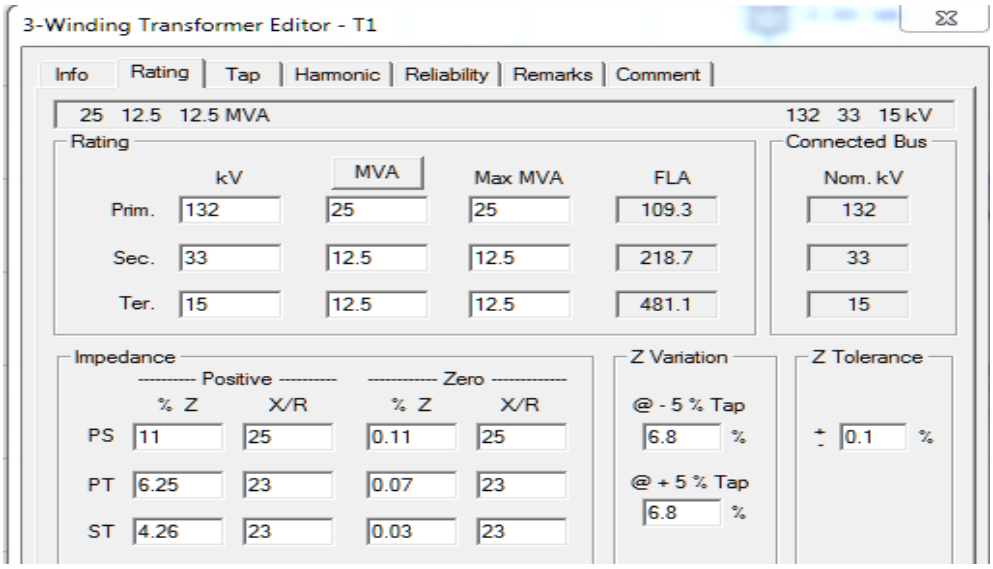


Figure 4-2 (a) three winding transformer

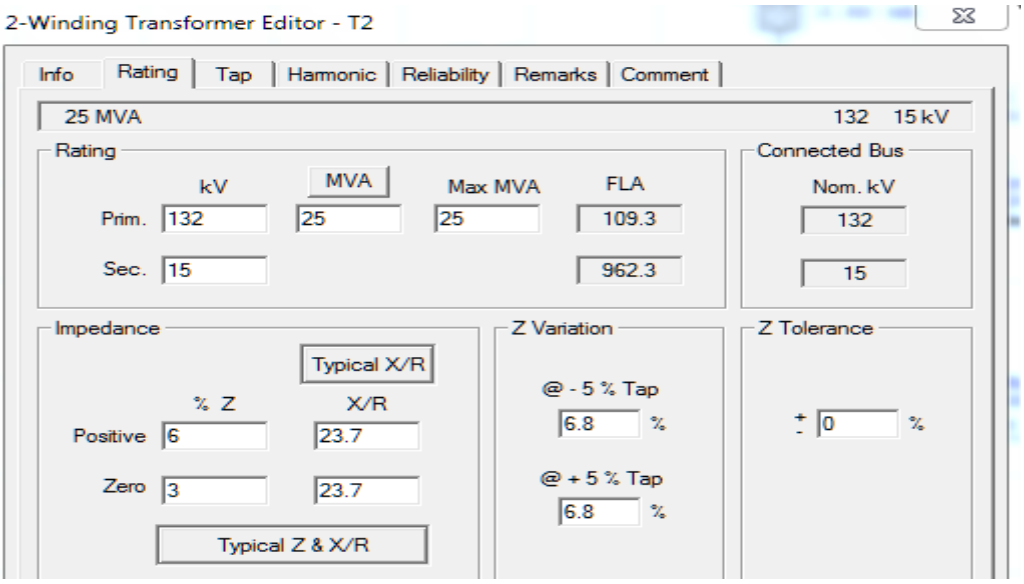


Figure 4.2 (b) two winding transformer parameter

4.2.3 Line Models

The geometric line modeling method is used in the ETAP power station software. The line configurations is three phase type horizontal with uniform spacing, the type of overhead conductor are used is aluminum with different sizes. The main feeder has a total length of 78.2km. The longest branch of feeder from the substation is 100.6km

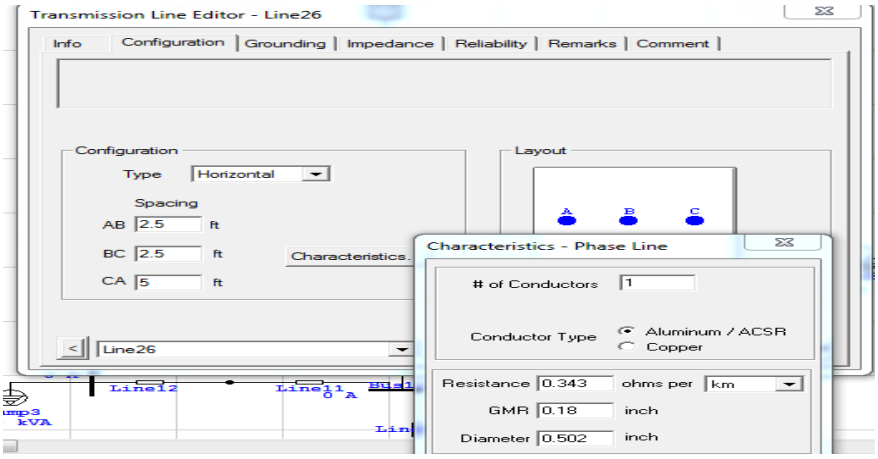


Figure 4-3(a) configurations and modeling of test feeder in ETAP

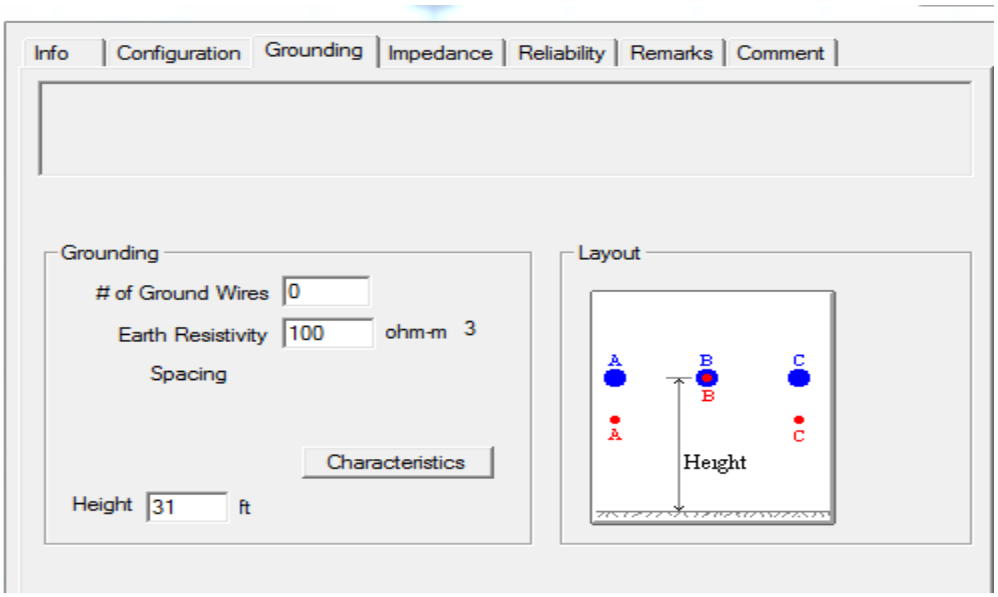


Figure 4-4(b) configurations and modeling of test feeder in ETAP

4.2.4 Load models

In the larger context of power systems, loads are usually modeled in an aggregated way rather than considering an individual appliance. Load may refer to an entire household, a city block, or all the customers within a certain region as mentioned in [26]. In this thesis the village level block loads are considered. Loads on a distribution feeder can modeled as wye-connected or delta connected. The loads can be three-phase, two-phase, or single-phase with any degree of unbalance, and can be modeled as: constant real and reactive power (constant PQ), constant current, constant impedance, or any combination of the above. All models are initially defined by a complex power per phase and an assumed line-to-neutral voltage (wye load) or an assumed line-to-line voltage (delta load). The notation for the specified complex powers and voltages are as follows as described in [27]

$$\begin{aligned} phase\ a{:}\ S_a &= |S_a|/\theta_a = P_a + jQ_a\ and\ V_{an} = |V_{an}|/\delta_a \\ phase\ b{:}\ S_b &= |S_b|/\theta_b = P_b + jQ_b\ and\ V_{bn} = |V_{bn}|/\delta_b \\ phase\ c{:}\ S_c &= |S_c|/\theta_c = P_c + jQ_c\ and\ V_{cn} = |V_{cn}|/\delta_c \end{aligned} \quad \text{..... (4.1)}$$

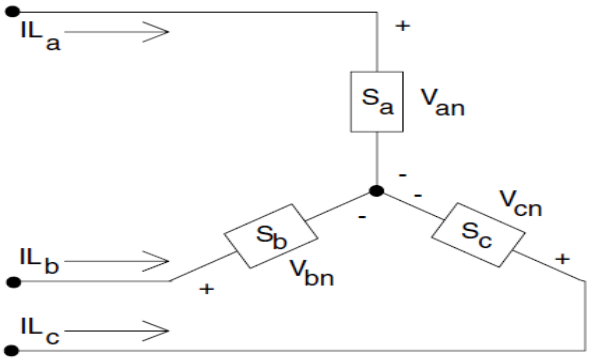


Figure 4-5 the model of a wye-connected load

The line currents for constant real and reactive power loads (PQ loads) are given by

$$I_a = \left(\frac{S_a}{V_{an}}\right)^* = \frac{|S_a|}{|V_{an}|} \ /(\delta_a - \theta_a) = |I_a| \ /\alpha_a$$
$$I_b = \left(\frac{S_b}{V_{bn}}\right)^* = \frac{|S_b|}{|V_{bn}|} \ /(\delta_b - \theta_b) = |I_b| \ /\alpha_b \qquad \text{..... (4.2)}$$
$$I_c = \left(\frac{S_c}{V_{cn}}\right)^* = \frac{|S_c|}{|V_{cn}|} \ /(\delta_c - \theta_c) = |I_c| \ /\alpha_c$$

The load in this thesis for simulation in ETAP software is modeled as the wye lump loads as constant power as shown in figure below.

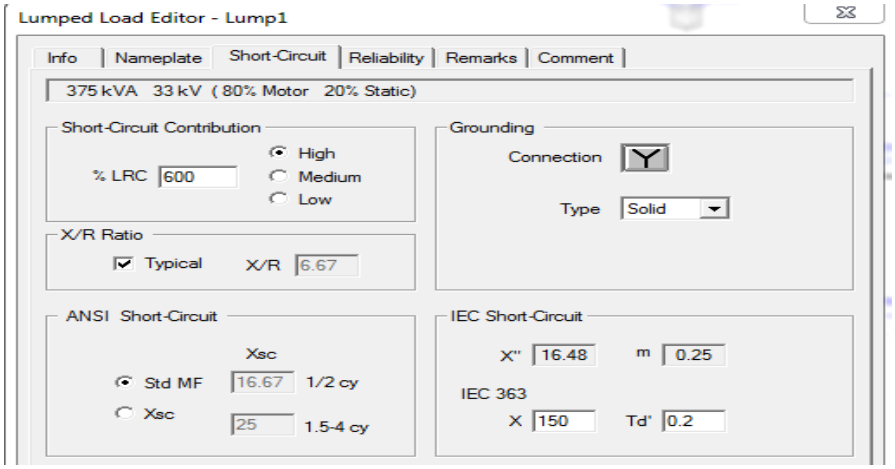


Figure 4-6 shows the configuration of loads in the test feeder

4.2.4 Test feeder simulation and data generation

The following block diagram representation shows methodology followed to generate data that used as input to the neural network training.

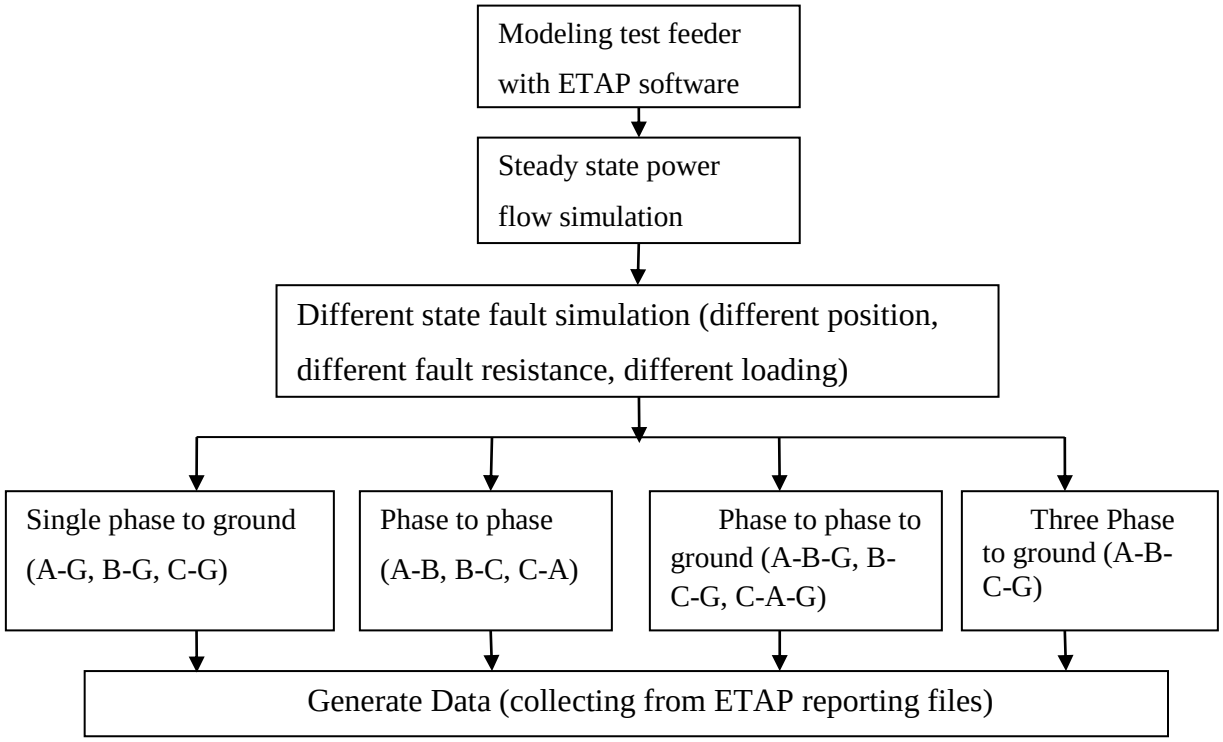


Figure 4-7 Summary of simulation for data generation process

The following considerations has been made to the test feeder during fault simulation

- For this case study thesis the sum loads of the total capacity of the village's distribution transformers are represented as maximum loads.
- Some branches and sub-branches have different aluminum conductor size than main feeder.
- The conductor's length count starts from the substations gantries.
- The spacing between conductors and height of poles remains the same in the feeder.

The following assumptions are usually made in fault analysis in three phase distribution line

- All sources are balanced and equal in magnitude & phase

- Sources represented by the Thevenin's voltage prior to fault at the fault point
- Large systems may be represented by an infinite bus-bars
- Transformers are on nominal tap position
- Loads currents are negligible compared to fault currents

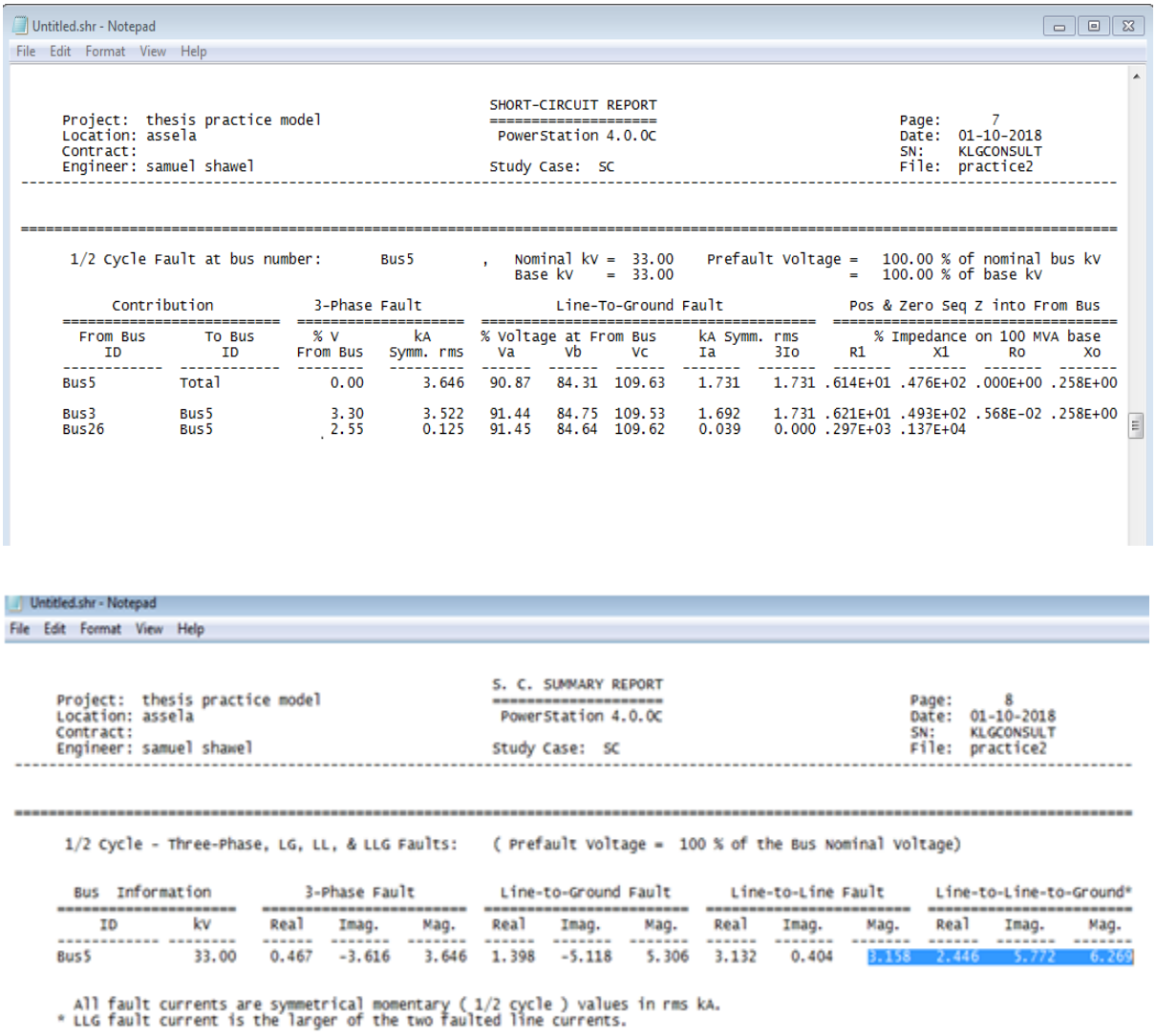
Table 4-1 Summary of taps length, conductor types and distribution transformer quantity in the test feeder

Rural villages	**MV line tap length**	**Conductor type and Area**	**Transformer rating used & Qty.**	**Circuit components**
Gumguma	78.2km (from Assela)	AAAC 95mm2	2x50kVA	DT and loads
			1x25kVA	
			2x100kVA	
Munesa	13.8km(from main feder)	AAAC 95mm2	2x100kVA	DT and loads
Chefa	online with gumguma	AAAC 95mm2	1x100kVA	DT and loads
			1x50kVA	
Chebi	3.3km(from munessa)	AAC50mm2	1x100kVA	DT and loads
Konche	2.9km(from main feeder)	AAC50mm2	1x50,1x100kVA	DT and loads
Sagure	13km(from main feeder)	AAC50mm2	1x50kVA	DT and loads
			1x25kVA	
			2x100kVA	
			1x200kVA	
Wajishagre	38.2km(from munessa)	AAC 50mm2	1x50,2x100kVA	DT and loads

Table 4-2 shows the plan for simulation of test feeder in fault condition

No.	**Fault Condition.**	**Fault and System parameter**
1.	Fault location (distance from substation(km))	3,6,9,12.. of the main feeder, and all the laterals and sub laterals (with three km increment, short length laterals by 1km increment)
2.	Fault types	single line-to-ground (LG), double line (LL), double line to ground (LLG), and three line to ground (3LG.)
3.	Fault resistance in ohm Ω	0, 0.5, 2.5, 5, 10, 25, and 50 for (LG)
4.	Loading	feeder loading of 0, 50%, and 100%

The figure below shows the sample short circuit analysis report generated by ETAP software for the single line to ground fault, line to line, three phases to ground fault simulated on the main feeder, at distance of 3km from the substation with fault resistance of 10ohm, with maximum feeder loading condition. The single line to ground the percentage voltage values and the short circuit current in kA (kilo Ampere) is reported in text form as shown below. Using the fault conditions of faults mentioned in chapter three, the fault voltage and current measurement during the ½ cycle of the occurrences of the fault is obtained from the report.



Untitled.shr - Notepad

File Edit Format View Help

SHORT-CIRCUIT REPORT

Project: thesis practice model
Location: assela
Contract:
Engineer: samuel shawel

PowerStation 4.0.0C

Study Case: SC

Page: 7
Date: 01-10-2018
SN: KLGCONSULT
File: practice2

1/2 Cycle Fault at bus number: Bus5 , Nominal kV = 33.00 Prefault Voltage = 100.00 % of nominal bus kV
Base kV = 33.00 = 100.00 % of base kV

Contribution		3-Phase Fault	Line-To-Ground Fault					Pos & Zero Seq Z into From Bus			
From Bus ID	To Bus ID	% V From Bus	kA Symm. rms	% Voltage at From Bus Va	Vb	Vc	kA Symm. rms Ia	3Io	% Impedance on 100 MVA base R1	X1	Ro	Xo
Bus5	Total	0.00	3.646	90.87	84.31	109.63	1.731	1.731	.614E+01	.476E+02	.000E+00	.258E+00
Bus3	Bus5	3.30	3.522	91.44	84.75	109.53	1.692	1.731	.621E+01	.493E+02	.568E-02	.258E+00
Bus26	Bus5	2.55	0.125	91.45	84.64	109.62	0.039	0.000	.297E+03	.137E+04		

Untitled.shr - Notepad

File Edit Format View Help

S. C. SUMMARY REPORT

Project: thesis practice model
Location: assela
Contract:
Engineer: samuel shawel

PowerStation 4.0.0C

Study Case: SC

Page: 8
Date: 01-10-2018
SN: KLGCONSULT
File: practice2

1/2 Cycle - Three-Phase, LG, LL, & LLG Faults: (Prefault Voltage = 100 % of the Bus Nominal Voltage)

Bus Information		3-Phase Fault			Line-to-Ground Fault			Line-to-Line Fault			Line-to-Line-to-Ground*		
ID	kV	Real	Imag.	Mag.	Real	Imag.	Mag.	Real	Imag.	Mag.	Real	Imag.	Mag.
Bus5	33.00	0.467	-3.616	3.646	1.398	-5.118	5.306	3.132	0.404	3.158	2.446	5.772	6.269

All fault currents are symmetrical momentary (1/2 cycle) values in rms kA.
* LLG fault current is the larger of the two faulted line currents.

Figure 4-8 Sample short circuit analysis report generated by ETAP software

The table 4-3 below shows the example data collected on template from the simulation short circuit reports to be pre processed for the input to the neural network

Table 4-3 template used to collect data from the ETAP generated report

	1 of phase to ground(case a-g)									3 phase(abc-g)	
Resistance value(ohm)	length of line(km)	branch of line	sub branch of line	voltage(%)			current(kA)			voltage(kv)	current(kA)
				Va	Vb	Vc	ia	ib	ic	3p	3p
0	0	0	0	0.01	86.66	86.55	5.455	0	0	0	3.646
0.5	0	0	0	13.93	80.03	93.24	5.306	0	0	0	3.646
2.5	0	0	0	54.98	8.29	108.84	4.19	0	0	0	3.646
10	0	0	0	90.87	84.31	109.63	1.731	0	0	0	3.646
25	0	0	0	97.35	93.49	104.87	0.742	0	0	0	3.646
50	0	0	0	98.88	96.76	102.6	0.377	0	0	0	3.646
0	3	0	0	0	91.42	95.87	3.597	0	0	0	3.144
.5	3	0	0	9.17	90.63	97.19	3.494	0	0	0	3.144
2.5	3	0	0	38.84	89.65	101.02	2.96	0	0	0	3.144
10	3	0	0	79.91	93.83	103.21	1.523	0	0	0	3.144
25	3	0	0	92.89	97.21	101.98	0.708	0	0	0	3.144
50	3	0	0	96.75	98.58	101.12	0.369	0	0	0	3.144

4.3 Artificial neural network training process

The neural network training process is an iterative procedure that begins by collecting data and preprocessing it to make training more efficient. At this stage, the data also needs to be divided into training, validation and testing sets. Once the data is ready, we need to choose the appropriate network type and architecture.

Then we select a training algorithm that is appropriate for the network and the problem we are trying to solve. After the network is trained, we have to analyze the performance of the network. This analysis may lead us to discover problems with the data, the network architecture, or the training algorithm. The entire process (as shown in the figure 4-8) is then iterated until the network performance is satisfactory. The training process mainly categorized into three steps: pre training, training and post training steps.

4.3.1 Pre training step

The steps that need to be performed before the network is trained can be grouped into three categories: selection of data, data preprocessing, and choice of network type and architecture.

(a) Selection of Data

Neural networks will only be as good as the data that is used to train it.

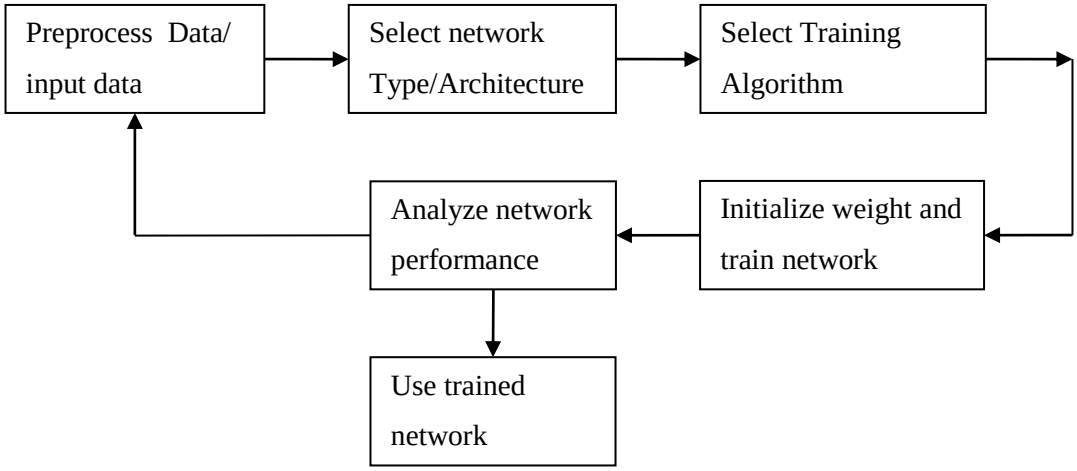


Figure 4-9 shows the block diagram for artificial neural network training process

The training data must span the full range of the input space for which the network will be used. It may not be possible to precisely define the active region of the input space. However, we can often collect data during standard operation of the system we model. After collecting the data, we generally divide it into three sets: training, validation and testing. The training set will generally make up approximately 70% of the full data set, with validation and testing making up approximately 15% each [28]. It is important that each of these sets be representative of the full data set. The simplest method for dividing the data is to select each set at random from the full data set.

(b) Data Preprocessing

The main purpose of the data preprocessing stage is to facilitate data for network training.It consists of such steps as normalization, nonlinear transformations, feature extraction, coding of discrete inputs or targets, handling of missing data, etc. In this thesis the collected data from ETAP short circuit report is change to the voltage and current transformers ratios multiple that set to the IED at the substation feeder. The percentage value of voltage is converted to the actual VT secondary voltage value and these values rewritten in multiple of VT secondary nominal voltage value, similarly for current values. After the input/output pairs organized it can be normalized by dividing the maximum value from the input output pair.

(c) Choice of Network Architecture

After exhaustive study the generally universal function approximators are selected. In this thesis the radial basis network and the multilayer feed forward neural network are used. One of the improved back propagation learning algorithms, Levenberg Marquardt is used to train the network because of the most commonly used algorithm for neural network training. The network have one input layer, one output layer and the number of neurons and hidden layers depend on the performance of trained network.

4.3.2 Training the network

After the data has been prepared, and the network architecture has been selected, we are ready to train the network. Before training the network, we need to initialize the weights and biases. The values to be initialized can be random values.

4.3.3 Post training analysis

Before using a trained neural network, we need to analyze it to determine if the training was successful. One useful tool for analyzing neural networks trained for approximation is a regression between the trained network outputs and the corresponding targets. We fit a linear function of the form

$$\boldsymbol{a}_q = mt_q + c + e \quad \text{.....(4.3)}$$

Where *m* and *c*, are the slope and offset, respectively, of the linear function, t_q is a target value, a_q is a trained network output, and *e* is the residual error of the regression

$$m = \frac{\sum_{q=1}^{Q}(t_q-\bar{t})(a_q-\bar{a}\ )}{\sum_{q=1}^{Q}(t_q-t)^2}, c = \bar{a} - m\bar{t}, \quad \text{.....(4.4)}$$

Where $\bar{a} = \frac{1}{Q}\sum_{q=1}^{Q} a_q \quad , \bar{t} = \frac{1}{Q}\sum_{q=1}^{Q} t_q$

In addition to computing the regression coefficients, we often also compute the correlation coefficient (R) between t_q and a_q Q is the number of input to target pair. R is a measure of how well the neural network's targets can track the variations in the outputs (0 shows no correlation at all and 1 shows complete correlation).

$$R = \frac{\sum_{q=1}^{Q}(t_q-\bar{t})(a_q-\bar{a}\ )}{(\ Q-1)s_t s_a} \quad \text{.....(4.5)}$$

Where

$$s_t = \sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q}\left(t_q - \bar{t}\right)}, and\ s_a = \sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q}(a_q - \bar{a}\ )} \quad \text{.....(4.6)}$$

Another tool that can identify outliers is a histogram of the errors, which shows the distribution of error.

4.4 Generalization

Generalization is the concept of getting the wisdoms in the data that used for training neural network. Based on the wisdom earned from data the trained network is capable to perform to the new data that not ever seen in the training set. One of the problems that occur during neural network training is over fitting and extrapolation. During over fitting the error on the training set is driven to a very small value, but when new data is presented to the network the error is large. Extrapolation is when the network extrapolates beyond the limit range of the input data. The network has memorized the training examples, but it has not learned to generalize to new situations. [22] and [9]

The key strategy we will use for obtaining good generalization is to find the simplest model that explains the data. There are at least five different methods that used to produce simple networks: growing, pruning, global searches, regularization, and early stopping. Growing methods start with no neurons in the network and then add neurons until the performance is adequate. Pruning methods start with large networks, which likely over fit, and then remove neurons (or weights) one at a time until the performance degrades significantly. Global searches, such as genetic algorithms, search the space of all possible network architectures to locate the simplest model that explains the data. Regularization and early stopping, keep the network small by constraining the magnitude of the network weights, rather than by constraining the number of network weights. This are commonly used generalization methods in [22] described below.

4.4.1 Early stopping

It is the simplest method used for improving the generalization in neural network training. The idea behind this method is that as training progresses the network uses more and more of its weights, until all weights are fully used when training reaches a minimum of the error surface. If

training is stopped before the minimum is reached, then the network will effectively be using fewer parameters and will be less likely to over fit. In order to use early stopping effectively, we need to know when to stop the training. The cross-validation method, that uses a validation set to decide when to stop. The input data to the network divided into test set training set and validation sets. The training set is used to compute gradients or Jacobeans' and to determine the weight update at each iteration. The validation set is an indicator of what is happening to the network function in between the training points, and its error is monitored during the training process. When the error on the validation set goes up for several iterations, the training is stopped, and the weights that produced the minimum error on the validation set are used as the final trained network weights as mention in [9]

4.4.2 Regularization

It is another method for improving generalization by modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set. It is the concept adding a penalty, or regularization, term that involved the derivatives of the approximating function (neural network in our case), which forced the resulting function to be smooth. The typical performance function that is used for training feed forward neural networks is the mean sum of squares of the network errors in [22].

$$F = mse = \frac{1}{n}\sum_{i=1}^{N} e_i^2 = \frac{1}{n}\sum_{i=1}^{N}(t_i - a_i)^2 \quad \text{.....(4.7)}$$

The generalization is improved if we modify the performance function by adding a term that consists of the mean of the sum of squares of the network weights and biases

$$msereg = \gamma mse + (1-\gamma)msw \quad \text{.....(4.8)}$$

Where γ is performance ratio and *msw* is mean square weight, $msw = \frac{1}{n}\sum_{j=1}^{N} w_j^2$.

4.5 Training neural network for fault location and fault type identification

4.5.1 Fault type identification

Under this thesis ten different types of faults are considered for study and coded as the following table below for purpose of training neural network. In this fault identification the faults, line to ground to each phase, line to line and line to line to ground and three phase faults

are coded as a serial numbers from 1 to 10 as shown in the table below. For training the neural network at the fourth neuron in the output layer is dedicated for fault type identifier.

Table 4-4 shows the different type of faults and the given code for training ANN

Fault type		The fault conditions during fault in phase						Code
		V_a	V_b	V_c	I_a	I_b	I_c	
LG	AG	Z_f*I_a	0	0	I_f	0	0	1
	BG	0	Z_f*I_b	0	0	I_f	0	2
	CG	0	0	Z_f*I_c	0	0	I_f	3
LLG	ABG	$Z_f(I_a+I_b)$	$Z_f(I_a+I_b)$	V_c	(I_a+I_b)	(I_a+I_b)	0	4
	BCG	V_a	$Z_f(I_c+I_b)$	$Zf(I_c+I_b)$	0	(I_c+I_b)	(I_c+I_b)	5
	CAG	$Z_f(I_a+I_c)$	V_b	$Z_f(I_a+I_c)$	(I_a+I_c)	0	(I_a+I_c)	6
LL	AB	V_a	V_b	V_c	I_a	$-I_a$	0	7
	BC	V_a	V_b	V_c	0	I_b	$-I_b$	8
	CA	V_a	V_b	V_c	$-I_c$	0	I_c	9
3L	ABCG	0	0	0	I_f	I_f	I_f	10

In fault location the distance of fault from the substation, the branch and sub branch under the main branch is coded as the sequence of numbers count from the substation. The input for the fault location and type identifier is the three phase fault voltage and currents (6 input neuron) and the output of the network is the fault distance from the substation in (km), the branch and sub branch number and fault type, which has 4 output neurons.

4.5.2 ANN training for fault location

This section presents the fault location training process using the data of 2694 different type of fault simulation with different loading condition and different fault resistance values. This training process starts by preprocessing the data generated by the ETAP software. To make uniform the generated data with the actual intelligent electronic device (IED) fault reading values by dividing the simulated data to the corresponding voltage and current transformer ratios of the feeder line. Additionally to increase the efficiency of ANN in training both the input and output pair is normalized to the numbers between 1 and -1 by dividing both inputs and out puts to the maximum value. MATLAB R2016a neural network toolbox is used for training the neural network. In MATLAB, the implementation of neural networks is performed by means of matrix

manipulation of the inputs, outputs, weights and biases vectors. Vectors from a training set are presented to the networks sequentially. The weights and bias of the network would be updated if the output of the network is different from the target dataset. The MATLAB neural network has the possibility to train the network either command line or graphic user methods. In this thesis graphic neural network training method is used for training the network for its simplicity in training process. In MATLAB software before training the data it randomly divided with 70% of data as training set, 15% validation set and 15% testing set. Training set used to compute gradients to determine the weight updates, validation sets used to check the trained network to be not to over fitting and testing sets used to test the trained network. The weights and biases are initialized by the MATLAB as default weights and the bias to random values between the input ranges; we can also change the bias and weight initial values as we like.

The fault location estimator can be considered as function approximation problem. This problem can be solved by universal function approximation networks, like the radial basis and the multilayer feed forward network topology. The multilayer feed forward network topology can be trained using neural network fitting tool or customized neural network tool or using command line. The performance analysis of the trained neural network is based on their regression result, MSE, error histogram, and independent testing using a hold-out test dataset. The well trained network can be used for further processing in fault location estimator design.

Network1 (6-21-4 ANN configuration)

This network is trained using the neural fitting tool with the *tansig* activation function between the input and the hidden layer and *purelin* between the hidden and output layer, with the 6 input neurons, 21 neurons in hidden layer and 4 neurons in output layer. Figure 4-9 shows the overview of ANN training. The feed forward back propagation topology, two layer neural network is used. The Levenberg Marquardt variation back propagation learning rule is used. As there is no defined rule for the selection of number of neurons and layers in the hidden layer for back propagation algorithm, trial and error is made till well trained network is obtained.

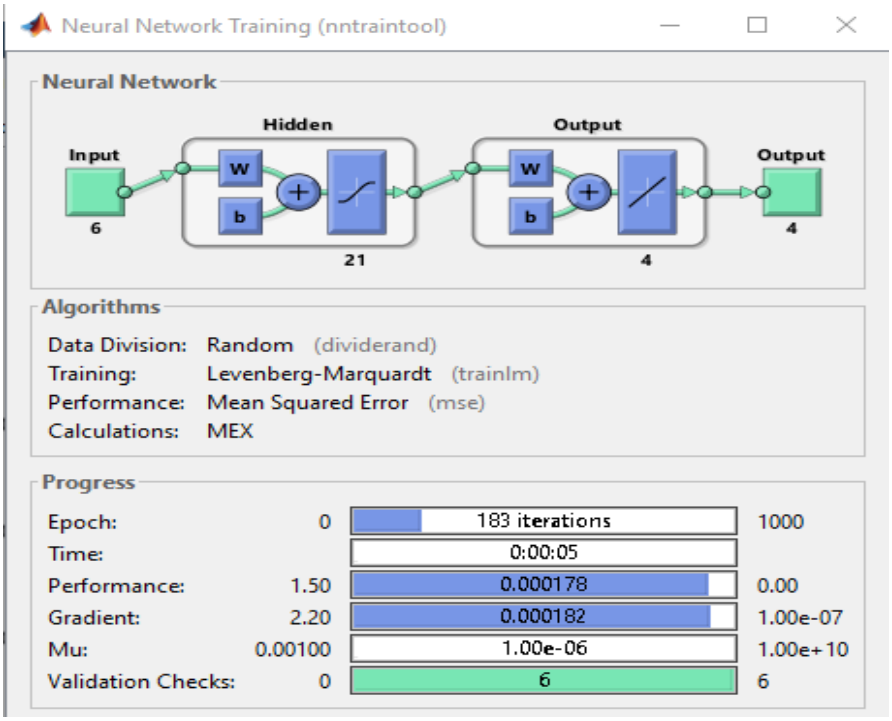


Figure 4-10 overview of 6-21-4 neural network training process

During the training state of the network the gradient, the Marquardt adjustment parameter and the validation check is shown in figure below. As shown in the figure 4-10 the gradient value varies during training and finally 0.000182 at epoch 183. The validation check is made at different epochs and for continuous 6 validation checks made at epoch 177 to 183 and training stops. The training parameters during training are as follows: minimum gradient is 10^{-7}, performance goal set to 0 in order to achieve maximum possible training performance.

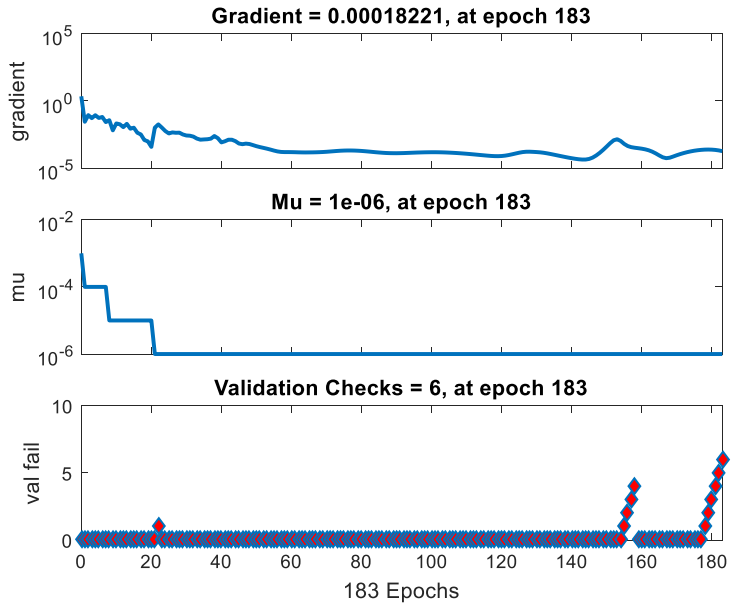


Figure 4-11 shows the training state of the 6-21-4 ANN configuration

The performance of the trained neural network was based on their regression result, MSE and error histogram values are shown in figure 4-11 below. As shown in this figure the mean square error (MSE) of the validation data set at epoch 177 is 0.000222 and the error which is the difference between the target and network output distribution is ranges from -0.022 to 0.024 with most of the data concentrated near to zero.

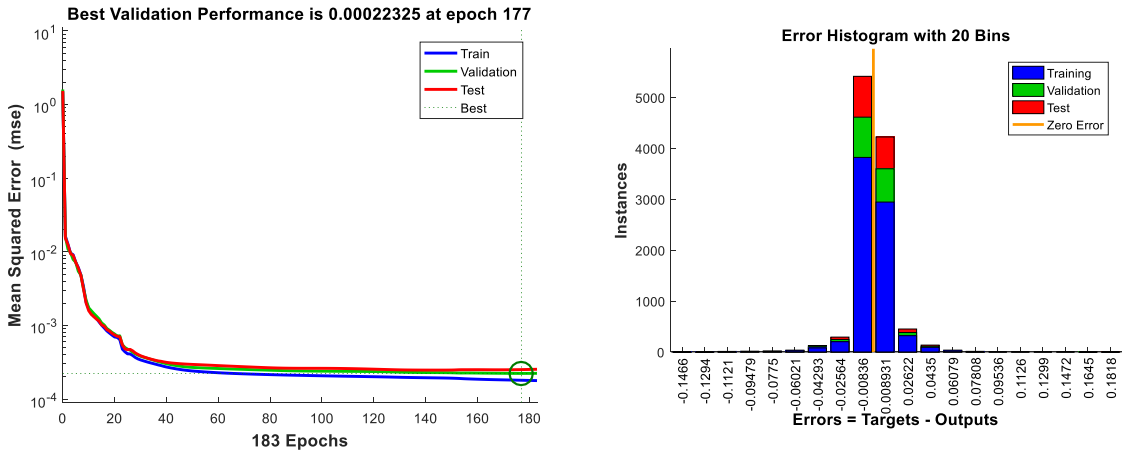


Figure 4-12 MSE performance and error histogram of 6-21-4 ANN configuration

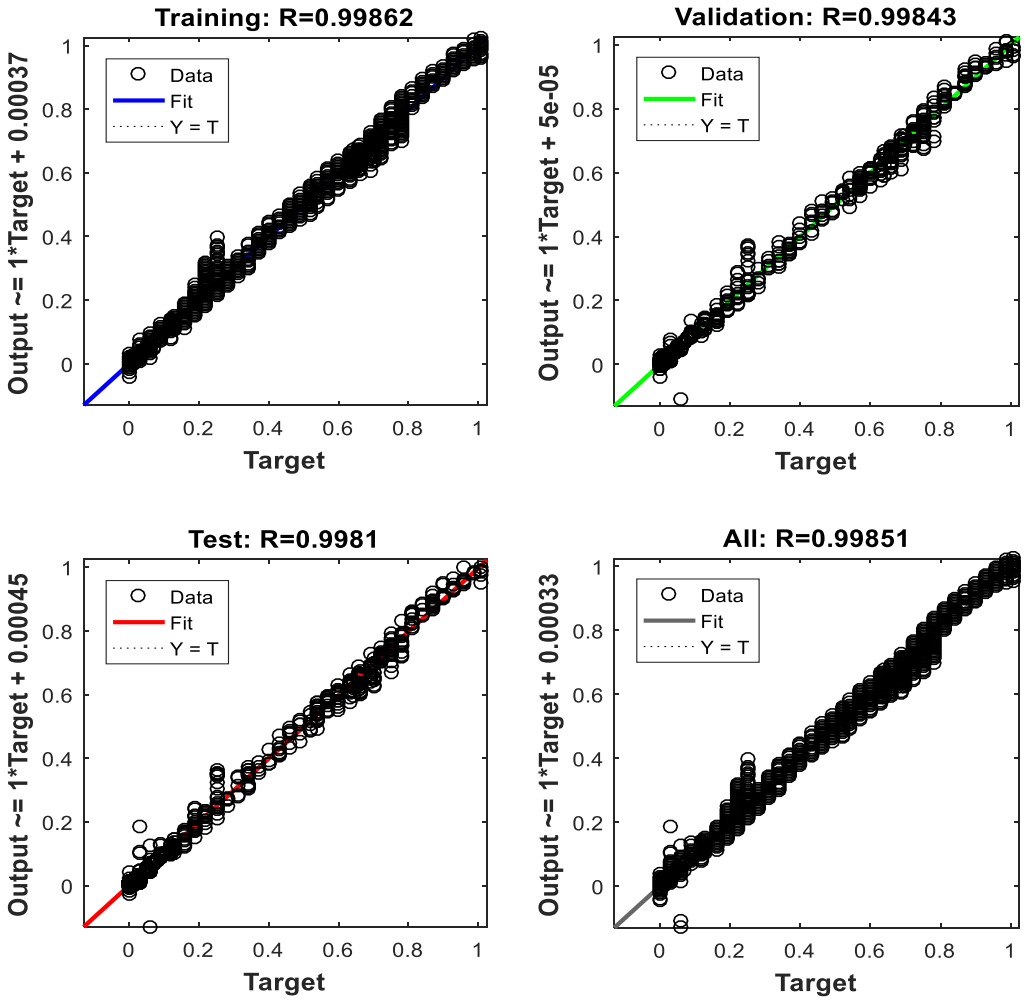


Figure 4-13 the regression plot of 6-45-3 ANN configuration

Another performance measure of trained neural network is the regression plot of the trained network training set, validation set and testing set. The figure above shows the correlation of the trained network output and the target values of the train, test and validation as well as overall. The correlation coefficient (R) is a measure of how well the neural network's targets can track the variations in the outputs (0 being no correlation at all and 1 being complete correlation). The correlation coefficient in this case has been found to be 0.99851 which indicates good correlation.

Network2 (6-41-4 ANN configuration)

This network is trained using the neural fitting tool by increasing the neuron number in hidden layer, the *tansig* activation function between the input and the hidden layer and *purelin* between the hidden and output layer, with the 6 input neurons, 41 neurons in hidden layer and 4 neurons in output layer. Figure 4-13 shows the overview of ANN training. the feed forward back propagation topology, two layer neural network is used. The Levenberg Marquardt variation back propagation learning rule is used. As there is no defined rule for the selection of number of neurons and layers in the hidden layer for back propagation algorithm, trial and error is made till well trained network is obtained.

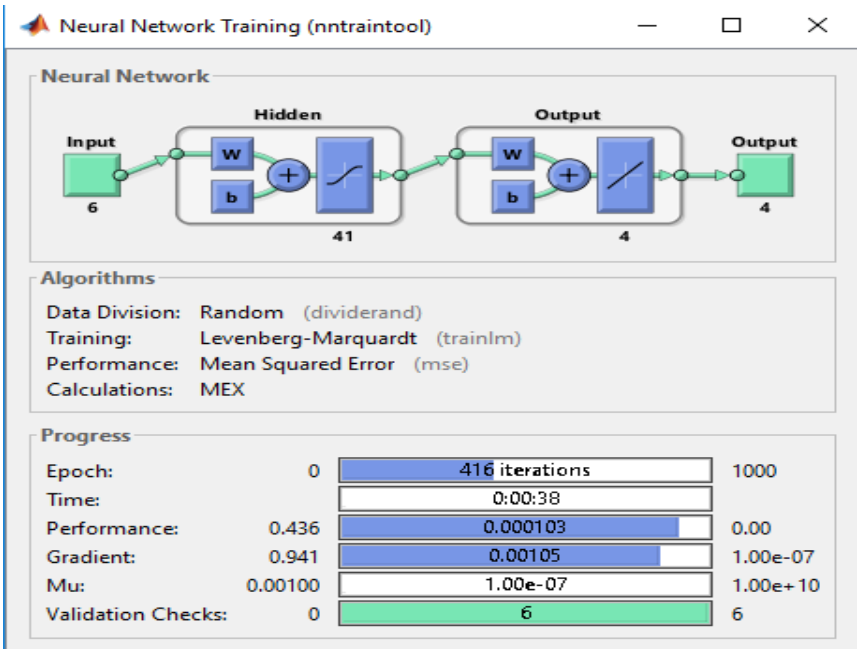


Figure 4-14 overview of 6-41-4 neural network training process

During the training state of the network the gradient, the Marquardt adjustment parameter and the validation check is shown in figure below. As shown in the figure 4-14 the gradient value varies during training and finally 0.000104 at 416 epoch. The validation check is made at different epochs and for continuous 6 validation checks made at epoch 410 to 416 and training stops. The training parameters during training are as follows: minimum gradient is 10^{-7}, performance goal set to 0 in order to achieve maximum possible training performance.

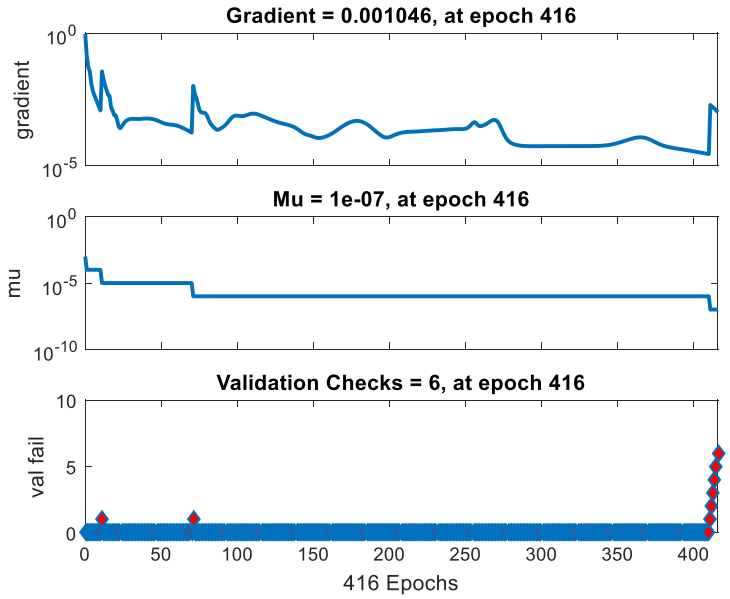


Figure 4-15 shows the training state of the 6-41-4 ANN configuration

The performance of the trained neural network was based on their regression result, MSE and error histogram values are shown in figure 4-15 below. As shown in this figure the mean square error (MSE) of the validation data set at epoch is 0.0001438 and the error which is the difference between the target and network output distribution is ranges from -0.017 to 0.019 with most of the data concentrated near to zero.

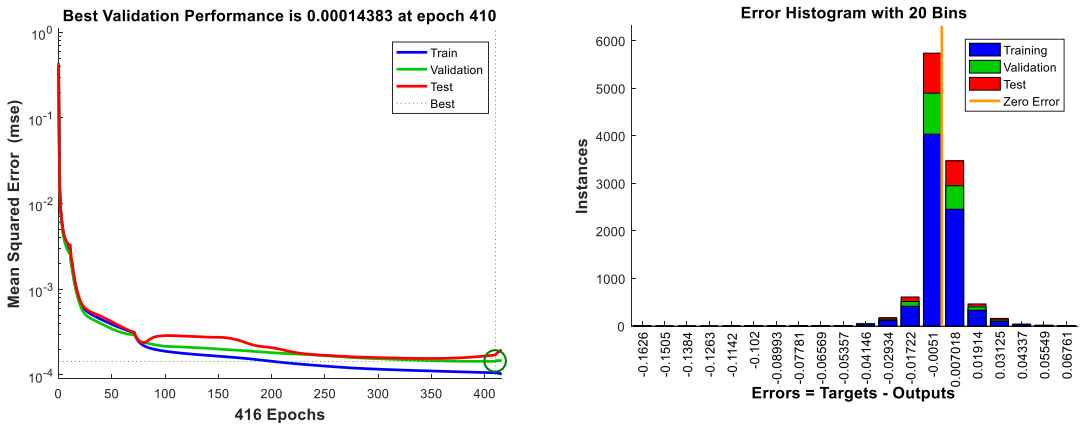


Figure 4-16 MSE performance and error histogram of 6-41-4 ANN configuration

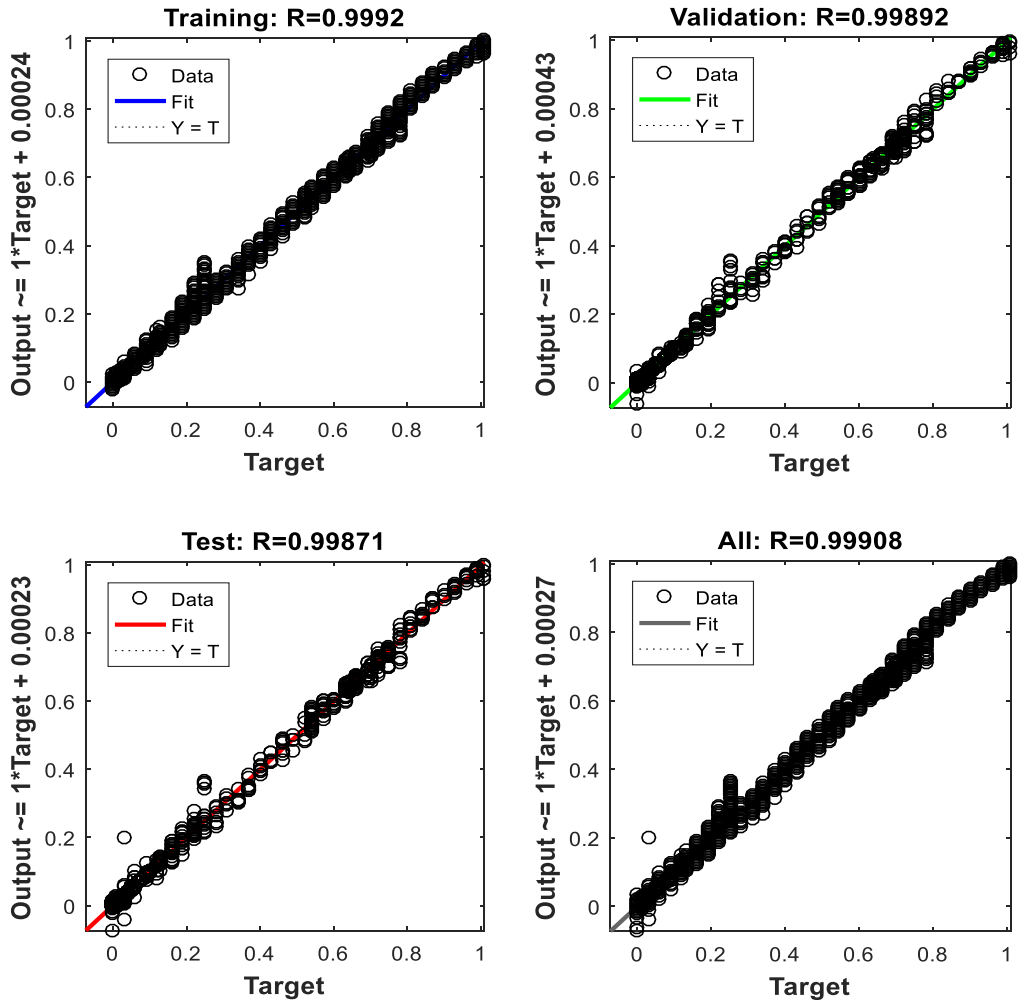


Figure 4-17 the regression plot of 6-41-4 ANN configuration

Another performance measure of trained neural network is the regression plot of the trained network training set, validation set and testing set. The figure above shows the correlation of the trained network output and the target values of the train, test and validation as well as overall. The correlation coefficient (R) is a measure of how well the neural network's targets can track the variations in the outputs (0 being no correlation at all and 1 being complete correlation). The correlation coefficient in this case has been found to be 0.99908 which indicates an excellent correlation.

Network3 (6-15-8-4 ANN configuration)

This network is trained using the neural network training tool with the *tansig* activation function between the input and the hidden layer , hidden layer and hidden layer and *purelin* between the hidden and output layer, with the 6 input neurons, 15 and 8 neurons in hidden layer 1 and 2 respectively and 4 neurons in output layer. Figure 4-17 shows the overview of ANN training. The feed forward back propagation topology, three layer neural network is used. The Levenberg Marquardt variation back propagation learning rule is used

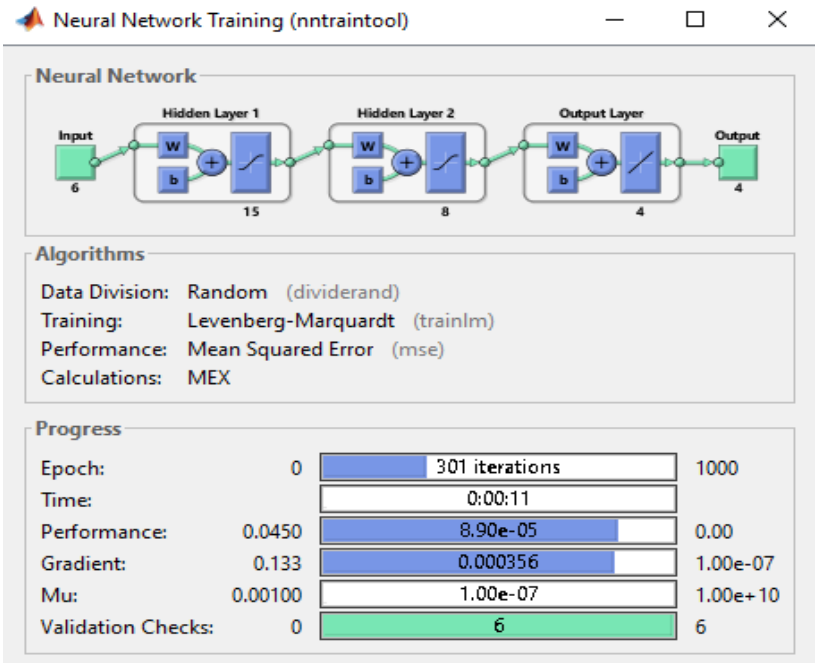


Figure 4-18 the overview of 6-15-8-4 neural network training process

The gradient, the Marquardt adjustment parameter and the validation check during training process is shown in figure 4-18 below. As shown in the figure the gradient value varies during training and finally 0.0003555 at 301 epochs. The validation check is made at different epochs and for continuous 6 validation checks made at epoch 295 to 301 and training stops. The training parameters during training are as follows: minimum gradient is 10^{-7}, performance goal set to 0 in order to achieve maximum possible training performance.

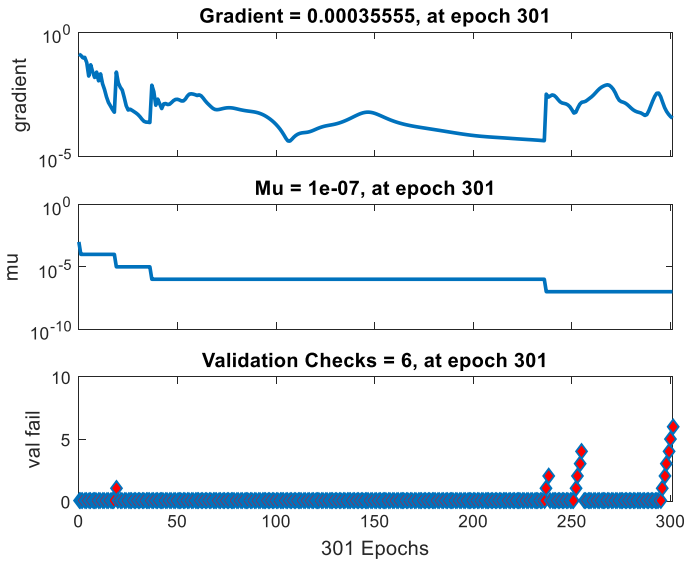


Figure 4-19 shows the training states of the 6-15-8-4 ANN configuration

The performance of the trained neural network was based on their regression result, MSE and error histogram values are shown in figure 4-19 below. As shown in the figure the mean square error of the validation data set at epoch 295 is 0.000102 and the error which is the difference between the target and network output distribution is ranges from -0.015 to 0.019 with most of the data concentrated near to zero.

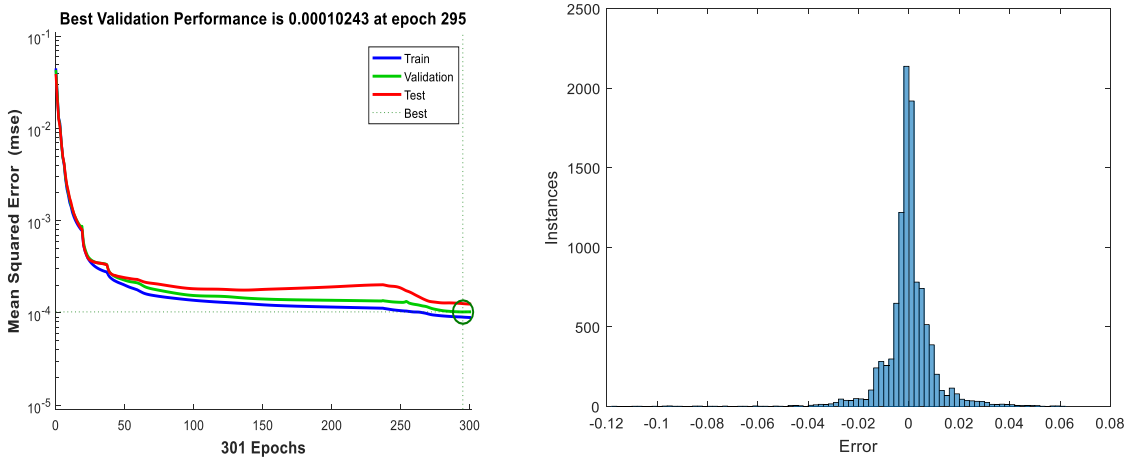


Figure 4-20 MSE performance error and error histogram of 6-15-8-4 ANN configuration

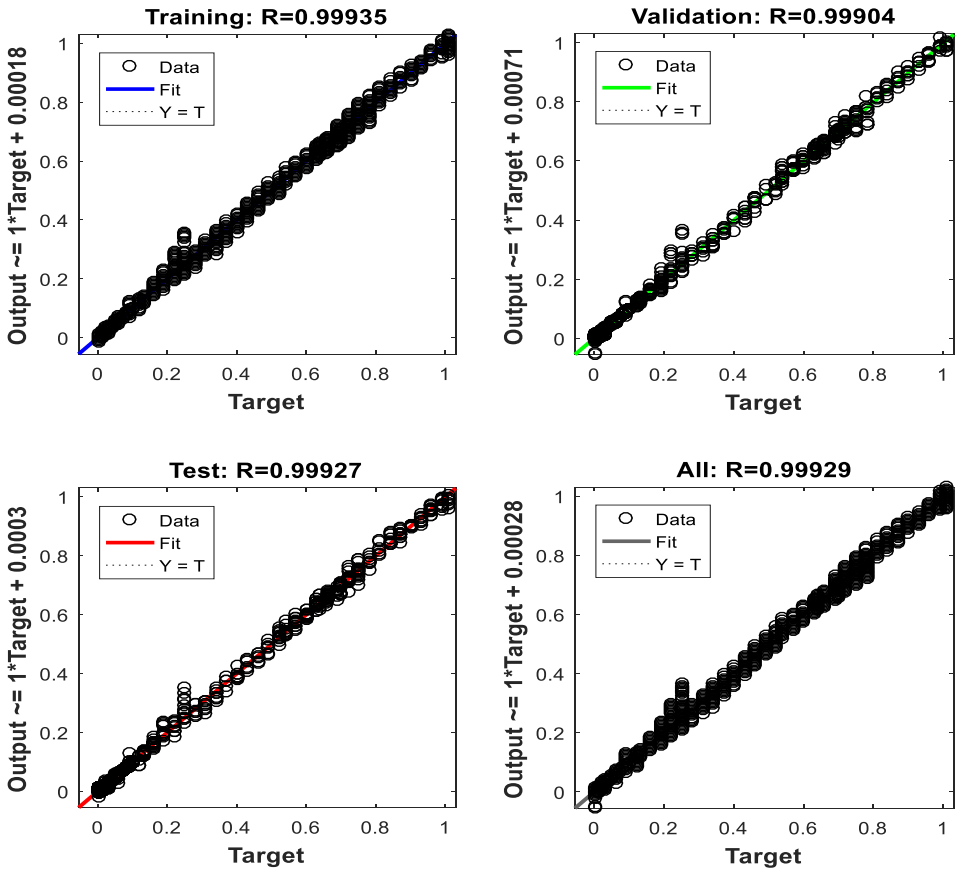


Figure 4-21 the regression plot of 6-15-8-4 ANN configuration

Another performance measure of trained neural network is the regression plot of the trained network training set, validation set and testing set. The figure 4-20 shows the correlation of the trained network output and the target values of the train, test and validation as well as overall of 6-15-8-4 ANN configuration. As shown in the figure the correlation of the overall trained network is 0.99929 which shows an excellent correlation.

Network4 (Radial basis network)

This network has two layers the *radbas* activation function and the *purelin* as shown in the figure below.

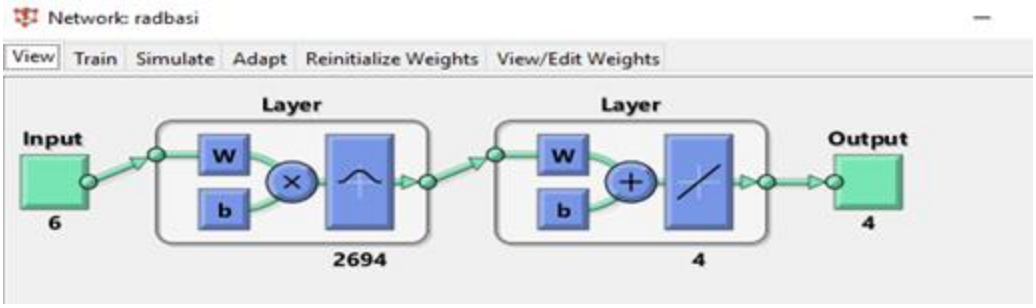


Figure 4-22 Overview of radial basis neural network configuration

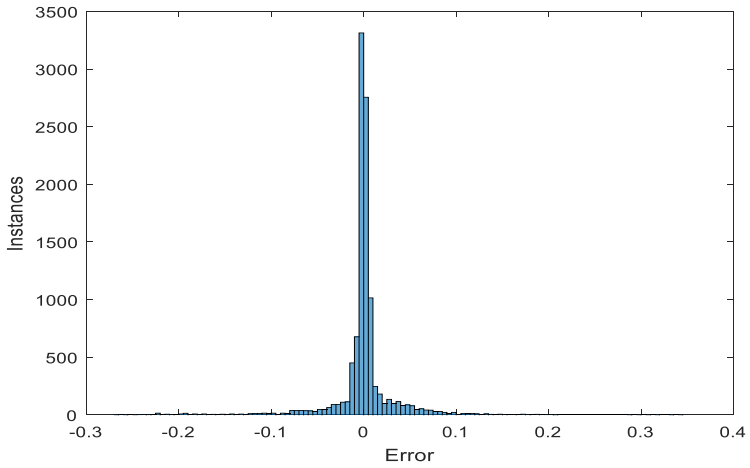


Figure 4-23 the error distribution plot of the exact fit radial basis network

The radial basis function calculates the distance between the input and the initialized weight and scaled by the initialized bias. The number of neurons used is the same as the number of total input used to train the network. One of the performance measures to see the exact fit radial basis is the error histogram as shown in figure 4-22 above.

4.5.3 Discussion of results

To obtain well trained neural network a number of different ANN configurations that are not presented here are made and four of ANN configuration are presented to show different configuration of ANN. Network1 (6-21-4 ANN configuration) two layer network, the MSE performance is 0.000222, the error ranges -0.025 to 0.024 and the overall correlation between the target and network output is 0.99851. Network2(6-41-4 ANN configuration) the two layer

network, with increased number of hidden layer neuron by 20 additional neurons as compared to network1, the obtained the MSE performance is 0.0001438, the error ranges -0.017 to 0.02 and the overall correlation between the target and network output is 0.99908. Network 3(6-15-8-4 ANN configuration) the three layer network, with increased number of hidden layer by 1 additional layer and reduced number of neurons in hidden layer as compared to network2, the obtained the MSE performance is 0.000102, the error ranges -0.015 to 0.019 and the overall correlation between the target and network output is 0.99929. Network4 different type network used for training (exact fit radial basis network) this networks output is mainly affected by the initial weight initialization values. The error values ranges from -0.05 to 0.05 as shown from error histogram. As shown above the MSE, regression plot, validation performance of the three networks becomes improved as the number of neurons and layers in the network increase. As there is no defined rule for optimized number of neurons, layers in the neural network training based on the performance results of the above network the network configuration 6-15-8-4 is selected as final well trained network that used for the fault location estimator. The table 4-5 below shows the comparison between trained neural network responses to the sample simulated faults location with the actual fault location. As shown in the table random sample new different type of faulty condition untrained 30 data given to the trained network and response of network to the new data is shown in table. As show in the table the maximum deviation in distance from the actual is 4.69km. The average distance, branches and sub branch location deviation from target is 0.269km, 0.099, and 0.038 respectively, as well as for fault type 0.092.

Table 4-5 trained neural network response and actual fault location comparison

Actual fault location for sample simulation				**Trained ANN network response to new data for different faults**				**Error (Difference between Actual target and network output)**			
Distance (km)	Branch	sub branch	Fault type	Distance (km)	Branch	sub branch	Fault type	Distance (km)	Branch	sub branch	Fault type
9	0	0	9	8.82	0.28	-0.21	9.33	0.18	-0.28	0.21	-0.33
75	2	1	8	79.69	1.43	0.52	8.01	-4.69	0.57	0.48	-0.01
66	2	1	7	67.07	1.37	0.4	7	-1.07	0.63	0.6	0
22	1	0	8	26.28	0.76	0.16	8.01	-4.28	0.24	-0.16	-0.01
49	0	0	8	47.11	0.29	0.33	8.07	1.89	-0.29	-0.33	-0.07
34	0	0	7	33.37	0.66	0.35	7.03	0.63	-0.66	-0.35	-0.03
69	0	0	7	68.87	1.4	0.4	6.99	0.13	-1.4	-0.4	0.01
57	0	0	8	55.4	1.03	0.38	8.07	1.6	-1.03	-0.38	-0.07
6	0	0	7	5.96	-0.59	0.08	6.74	0.04	0.59	-0.08	0.26
96	2	1	7	94.15	1.9	0.44	6.94	1.85	0.1	0.56	0.06

65.8	2	0	2	65.82	1.05	0.33	2.16	-0.02	0.95	-0.33	-0.16
57	0	0	2	54.71	0.77	0.14	2.13	2.29	-0.77	-0.14	-0.13
60	0	0	2	60.67	0.96	0.3	2.22	-0.67	-0.96	-0.3	-0.22
78.2	0	0	2	74.06	1.37	0.51	2.21	4.14	-1.37	-0.51	-0.21
16	0	0	1	16.42	0.76	0.22	1.28	-0.42	-0.76	-0.22	-0.28
19	0	0	1	20.03	0.25	-0.06	1.1	-1.03	-0.25	0.06	-0.1
64	2	1	1	64.66	1.09	0.44	1.16	-0.66	0.91	0.56	-0.16
66	2	1	1	65.22	1.17	0.32	0.95	0.78	0.83	0.68	0.05
66	2	1	1	65.08	1.17	0.33	0.94	0.92	0.83	0.67	0.06
100.6	2	1	1	99.96	1.91	0.62	0.93	0.94	0.09	0.38	0.07
57	0	0	1	57.35	1.01	0.25	0.97	-0.35	-1.01	-0.25	0.03
3	0	0	3	4.52	0.04	-0.21	2.57	-1.52	-0.04	0.21	0.43
9	0	0	3	13.39	0.24	0.14	2.62	-4.39	-0.24	-0.14	0.38
12	0	0	3	11.41	0.04	-0.21	2.93	0.59	-0.04	0.21	0.07
54	0	0	1	52.9	0.93	0.2	1.11	1.1	-0.93	-0.2	-0.11
6	0	0	10	5.65	-0.04	-0.36	10.09	0.35	0.04	0.36	-0.09
70	3	0	10	70.63	1.15	0.44	10.15	-0.63	1.85	-0.44	-0.15
78	2	1	10	80.44	1.63	0.58	9.97	-2.44	0.37	0.42	0.03
49	0	0	10	50.88	0.73	0.17	10.05	-1.88	-0.73	-0.17	-0.05
19	0	0	3	20.42	0.11	-0.15	3.02	-1.42	-0.11	0.15	-0.02

As seen in the table most neural network response is similar to the actual target values for distance, branch and sub branch, but some of the values have deviation it might be due to lack of sufficient data representation from the active region of the data during training set selection.

4.6 Fault location estimator development and implementation

As detail discussion in the above sections on the distribution system simulation for data generation and neural network training process for test feeder to the desired fault location estimator design is covered. In the continuation of training process of ANN using the well trained NN to the fault estimator development. This development has the hardware and software implementations the overall hard ware and software combination layout is shown in figure 4-23. The major component in practical implementation is intelligent electronic device (IED), which is installed at the feeder in the substation. In this feeder ABB REF615 series IED is used for fault recording and protection of the distribution line and PCM600 IED software tool that used for reading the fault record using computer. The graphic user interface software is developed using MATLAB for interaction of the users with the fault estimator software developed.

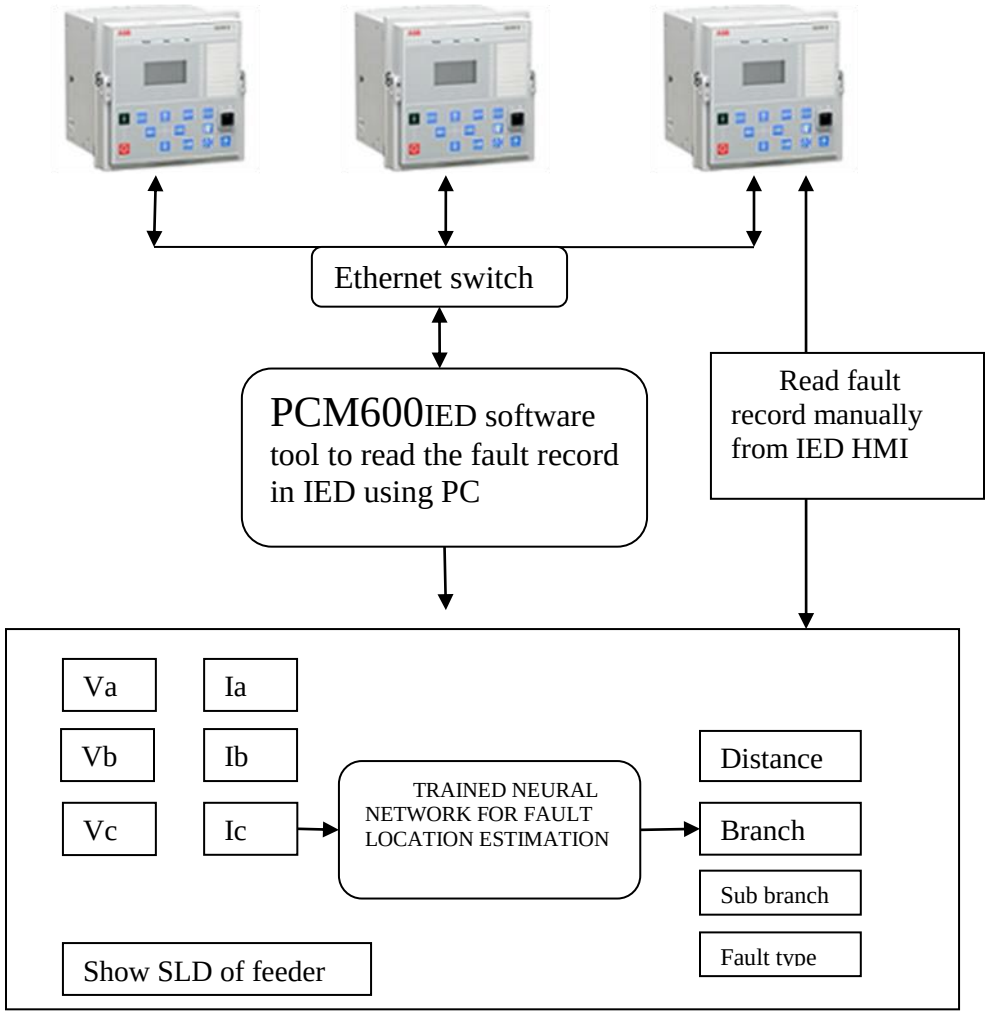


Figure 4-24 general block diagram representation of overall fault location

Most of the feeders in Ethiopia have IEDs, made in ABB, AREVA, Micom which are most known electrical equipment manufacturing companies. So we can utilize this IEDs without additional cost for fault location estimator design.

4.6.1 Intelligent electronic device (IED)

Intelligent electronic device (IED) is digital microprocessor based electrical devices that use in different applications. One of application of this device is in substation automation. Under this thesis one of ABB REF615 series is a product family of IEDs designed for protection, control, measurement and supervision of utility substations and industrial switchgear and equipment is used in the selected test feeder. The design of the IEDs has been guided by the IEC 61850

standard for communication and interoperability of substation automation devices. It is the most powerful, advanced and simplest feeder protection relay in its class, perfectly offering time and instantaneous over current, negative sequence over current, breaker failure, thermal overload, and voltage metering and protection. The relay continuously measures the phase currents, the sequence components of the currents and the residual current. With the VT option, phase, ground and sequence voltage measurements plus power, energy and power factor measurements are included. The values measured can be accessed locally via the user interface on the relay front panel or remotely via the communication interface of the relay. The relay has the capacity to store records of 100fault events. The records enable the user to analyze the four most recent power system events. Each record includes the current values, the pickup times of the protection blocks, time stamp, etc. The fault recording can be triggered by the pickup signal or the trip signal of a protection block, or by both. The available measurement modes include DFT, RMS and peak-to-peak. In addition, the maximum demand phase currents with date and time stamp are separately stored as recorded data. This IED use protection and control IED Manager PCM600 Ver. 2.3 or later to communicate through computers locally using RJ45 port. The local human machine interface (HMI) of IED is shown below which is taken from [29].

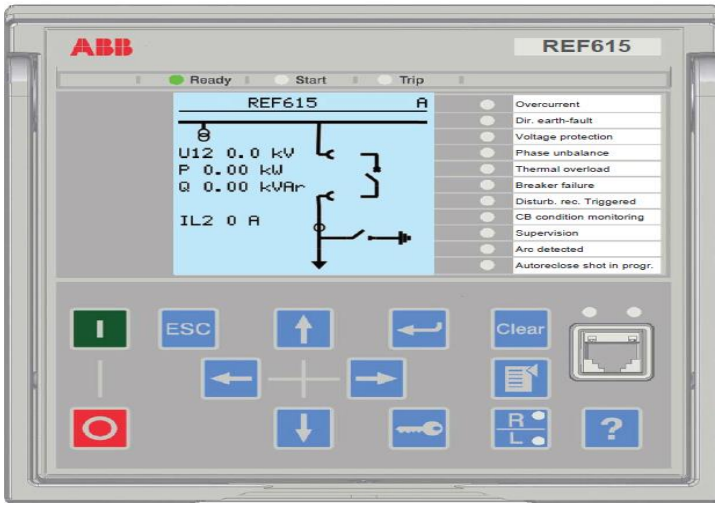


Figure 4-25 front view of ABB REF615 IED

The Protection and Control IED ManagerPCM600 tool provides versatile functionalities for the entire life-cycle of protection and control IED applications, at all voltage levels. PCM600

interacts with IEDs over the fast and reliable TCP/IP via corporate LAN or WAN, or alternatively directly through the communication port at the front of the IED. PCM600 tool is able to read and write all configuration and setting data of an IED with a single command. The user interface, workflow and the IEC61850 based data model in PCM600 are designed according to the philosophy of protection and control of IEDs, ensuring smooth and seamless integration between the tool and the IEDs.

The PCM600 tool fault record is shown below, the highlighted the maximum current values and the line to ground voltage records are used for further processing in fault location estimator design in this thesis.

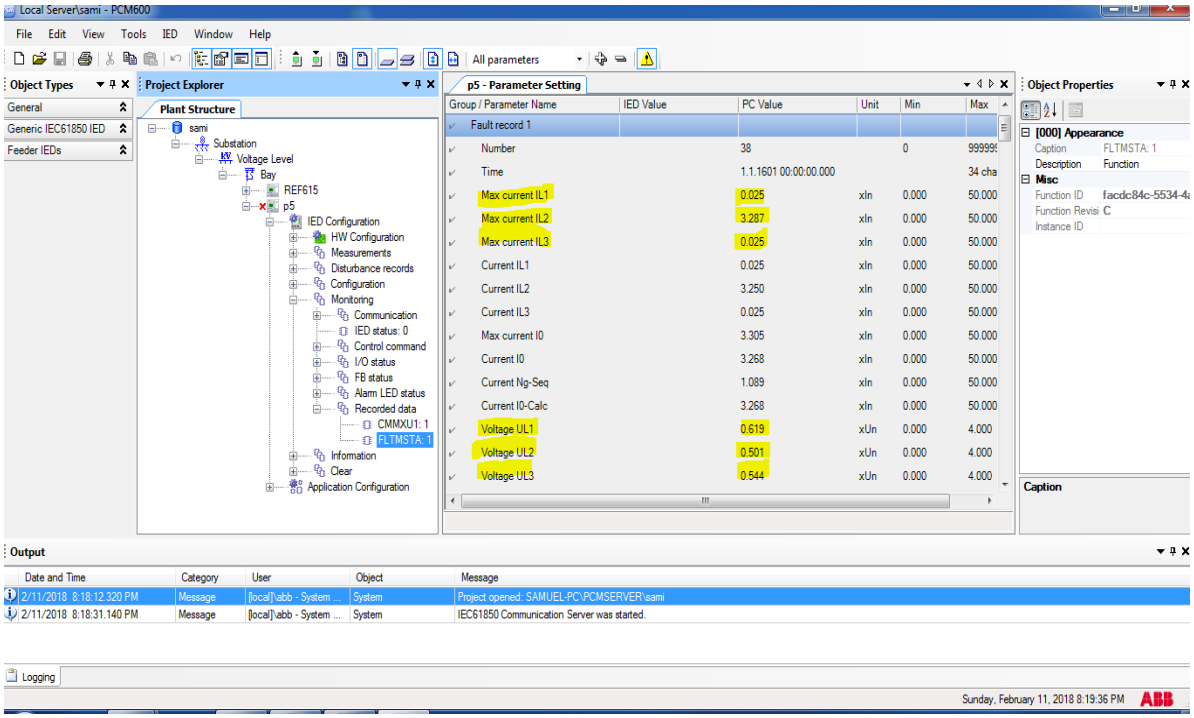


Figure 4-26 the fault record 1 of the last event on the feeder read by PCM600

4.6.2 Software development

The programming for the fault location estimator and fault type identifier is developed by MATLAB programming tool box. As shown in the figure 4-23 the fault record reading from the IEDs through PCM600 or manually using IEDs human machine interface (HMI),which is the maximum line current and the line to ground voltage is given as input to the graphic user

interface input. The program behind checks all necessary data preprocessing and load the trained artificial neural network is ordered to simulate the given input and finally the output displays the estimate fault location by distance, branch and sub branch level and type of fault. The user interface is developed by the MATLAB graphic user interface tool box. The MATLAB generated code for the program is found in Appendix C.

When the new fault record is given as input to the user interface the trained neural network will loaded and the new data introduced with that trained artificial neural network and finally the fault location in distance, faulty branch and sub branch and fault type will be displayed. The developed user interface is shown in figure below.

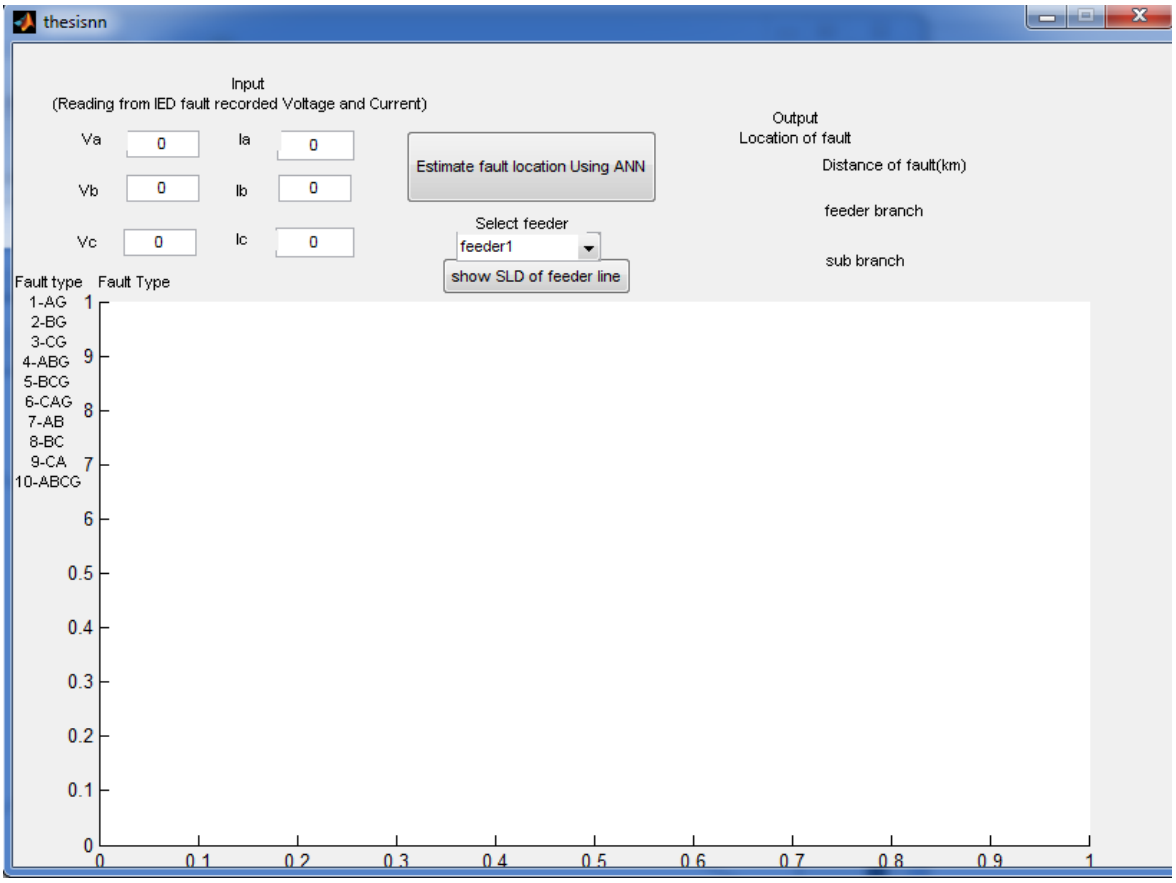


Figure 4-27 the graphic user interface developed for fault estimator design

The fault record from IED the maximum current records and the phase voltage fault records will be given to the User interface and the output will be displayed on the right side of the

window. The below window shows the input of fault record is given to the user interface and the output with the selected single line diagram of the feeder is shown.

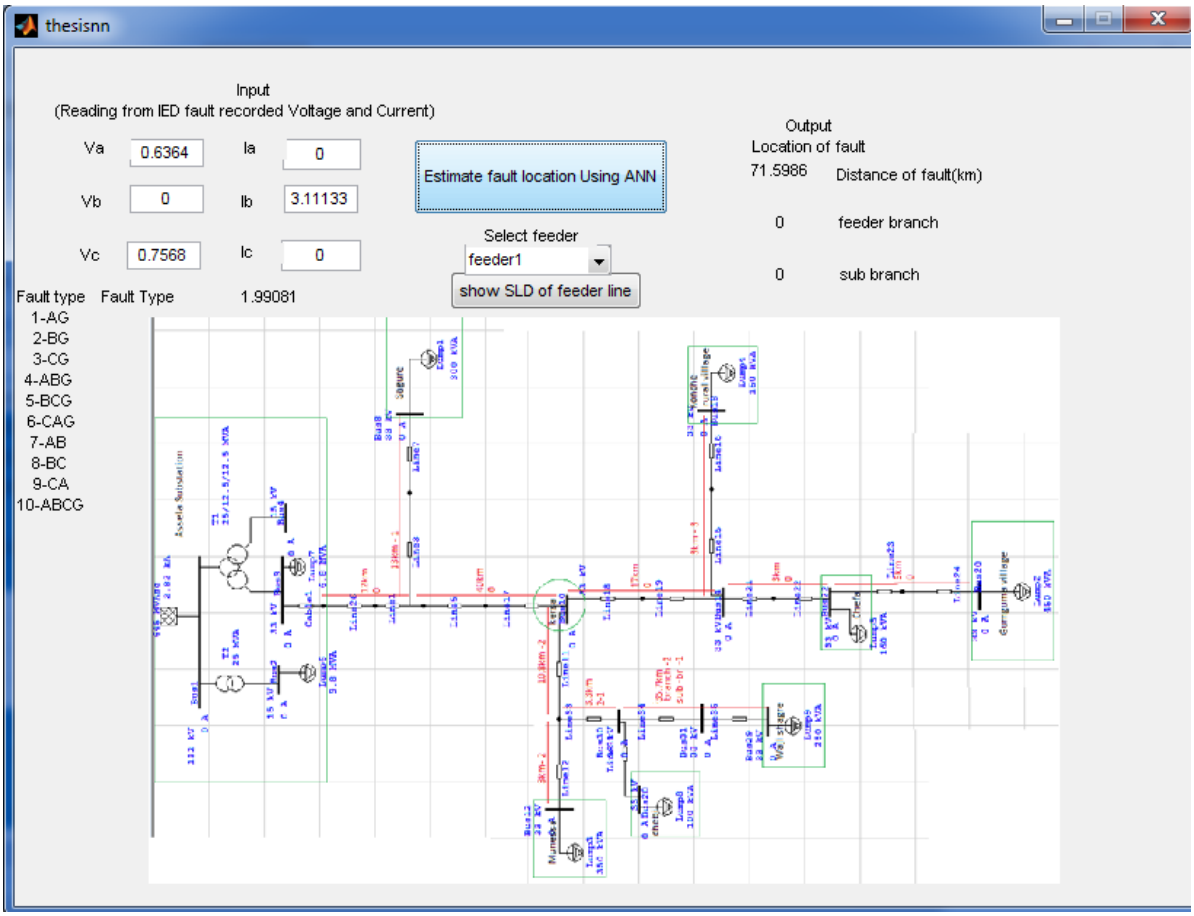


Figure 4-28 the graphic user interface developed for fault estimator design, input fault data given and fault location shown as output

4.6.3 Benefits of fault location estimator

4.6.3.1 Time and effort saving

After the fault, the related relaying equipment enables the associated circuit breakers to energize the faulted sections. Once the fault is cleared and the participated faulted phase(s) are declared, the adopted fault locator is enabled to detect the fault position. Then, the maintenance crews can be informed of that location in order to fix the resultant damage. Later, the line can be energized again after finishing the maintenance task.

4.6.3.2 Improving the system availability

There is no doubt that fast and effective maintenance processes directly lead to improve the power availability to the consumers. This consequently enhances the overall efficiency of the power nets. These concepts of (availability, efficiency, quality etc) have an increasingly importance nowadays due to the new marketing policies resulting from deregulation and liberalization of power and energy markets.

4.6.3.3 Assisting future maintenance plans

Analyzing the location of these faults can help to pinpoint the wake spots on the overall distribution nets effectively. This hopefully assists the future plans of maintenance schedules and consequently leads to avoid further problems in the future. These strategies of preventive maintenance enable to avoid those large problems such as blackouts and help to increase the efficiency of the overall power system.

4.6.3.4 Economic factor

All the mentioned benefits can be reviewed from the economical perspective. As the time and effort saving, increasing the power availability and avoiding future accidents can be directly interpreted as a cost reduction or a profit increasing. This is an essential concept for competitive marketing. Thus the importance of proper fault location schemes for power system utilities is obvious.

CHAPTER 5

CONCLUSION, RECOMMENDATION AND FUTURE WORKS

5.1 Conclusion

In this thesis fault location estimator for power distribution system using artificial neural network is developed for line to ground, line to line, line to line to ground and three phases to ground faults. To develop this estimator one of rural radial power distribution feeder in Ethiopia, Oromia, Assela substation Gumguma line feeder is used for design as the test feeder. This feeder is simulated using ETAP software to generate different fault condition, with different fault resistance and loading condition. MATLAB R2016a neural network toolbox to train ANN and programming tool box is used to develop graphic user interface for fault estimator. The fault condition generated data from ETAP software is normalized and this normalized data used as input for neural network to train neural network and to estimate the fault location with trained network efficiently. The feed forward multi layer network topologies of neural network with improved back propagation; Levenberg Marquardt learning algorithm is used to train the network. The performance of the trained network is analyzed by mean square error, regression plot and error histogram. Different artificial neural network configurations are trained with variable hidden layer neuron and variable number of layers as well as varying different values of weight and bias initialization. Network configuration of ANN 6-15-8-4 is selected as final trained network for fault location estimator which was found as excellent performance with regression coefficient 0.99929, validation performance of 0.000102 and error histogram range -0.15 to 0.17 of network. The fault records at the test feeder is handled by intelligent electronic device (IED) installed at the substations feeder. The fault record of IED can be read by PCM600 tool using laptop or manually using IEDs human machine interface, the read data feed to the graphic user interface to estimate the fault. We can conclude that artificial neural networks are one of the alternate options for fault location estimator design for distribution system where sufficient distribution network data are available with average error of fault location distance 0.296km, branch 0.09 and sub branch 0.038 from the actual location as well as fault type identification 0.092. Training network performance depend on the amount of data given as input to the neural network, the more data to be trained the more accurate will be the trained network.

5.2 Recommendation

- This fault estimator design methodology can be used for any radial distribution system which is long to reduce the fault restoration time in a condition where the substation operators and the maintenance crew are well organized.
- With the similar methodology for sub transmission lines fault location estimator design can be applied.
- It is also possible to use the fault estimator for ring type distribution systems, on express feeders and line feeders if sufficient real time operation data of distribution networks are available.

5.3 Future works

- If there is sufficient data like GPS points of feeder line poles or maps of feeder line, the fault location estimators can be integrated with geographical location of the fault instead of using distance, branch and sub branch.
- If the substations are automated the stored fault record of feeder used for the fault location estimator design. It is recommended to use the data of fault records in IED and reports of maintenance crew for fault restoration area to store in data base and use this data for neural network training to increase the efficiency of the fault location estimation.
- In future it can be done to estimate the fault location automatically using the fault record data from IEDs remotely to estimate the fault locations away from the substations by maintenance team.

5.4 Limitation

- The big challenge in neural network training is how much data is sufficient for successful neural network training. Unless the neural network is trained and watched for the performance of the trained network.

References

[1] Marinko Stojkov,"*Power System Fault Data and Time Series*",HEP group, Distribution, Slavonski Brod,2009, Croatia pp 8,11.

[2] I.J. Nagrath and D.P. Kothari," *Modern power system fault analysis*" third edition,Tata McGraw-Hill private limited, 2009, pp 369,357.

[3] G. M. Morales-Espana, J. Vargas-Torres, H., *Elimination of Multiple Estimation for Fault Location in Radial Power Systems by Using Fundamental Single-End Measurements,* IEEE Transactions on, vol.24 No 3, july 2009, pp. 1382-1389.

[4] R.H. Salim, K.C.O. Salim,A.S. Bretas, *Further improvements on impedance-based fault location for power distribution systems*, April, 2010, pp 1-12.

[5] L. J. Awalin, H. Mokhilis, A. Abubakar, *Recent Developments in Fault Location Methods for Distribution Networks*, ISSN 0033-2097, R. 88 NR 12a/2012, pp 3.

[6] Cansin Y. Evrenoso˜glu, Ali Abur, *Fault Location in Distribution Systems with Distributed Generation,* 15th PSCC, Liege, 22-26 August 2005 Session 10, Paper 5, Page 3

[7] Daqing Hou, *Comparing Fault Resistance Coverage of Different Distribution System Grounding Methods*, Schweitzer Engineering Laboratories, Inc.Revised edition, Oct. 2010 pp 9.

[8] M. Zangiabadi, M. Reza, H. Abolhasan, K. Mahmood Attari, *Fault location in distribution system based on artificial neuralnetwork and application of GIS*, May 2003, 17th International Conference on Electricity Distribution,pp 1-7.

[9] Suhaas Bhargava Ayyagari, *Artificial neural network based fault location for transmission lines* , University of Kentucky , 2011,pp 1-97

[10] Eisa Bashier M. Tayeb,*Faults Detection in Power Systems Using Artificial Neural Network,* Volume-02, Issue-06 2013 , pp-69-75

[11] H. Zayandehroodi,A Mohamed , *Automated fault location in power system with distribution generation using radial basis neural network*, journal of applied science:10(20):3032-3041,2010 pp 2-11

[12] N. Kumar, M. Sharma, A. Sinha ,I. Bhushan,"*Fault detection on radial power distribution system using fuzzy logic*",IJEEE:ISN2321-2055, vol 07,issue 02,december 2015. pp 399-406.

[13] Adeyemi Charles Adewole,*''Investigation of methodologies for fault detection and diagnosis in electric power system protection*" October, 2012. pp 250- 311

[14] Fukuyama,Y., Ueki , *Fault analysis system using neural networks and artificial intelligence*, Proceedings of the Second International Forum, 1993,pp. 20-25.

[15] Mamta Patel, PhD ,*Fault Detection and Classification on a Transmission Line using Wavelet Multi Resolution Analysis and Neural Network* ,vol-47 no 22, June 2012 pp 27-31.

[16] Ratan Das, *Determining the location of faults in distribution systems*, University of Saskatchewan , 1998. pp 21-23

[17] Mm.S. Sharma, J.D.Glover, T.J. Overbye" *power system analysis*", 5th edition, Cengage learning, 2012, page 479.

[18] Jan Iżykowski ,"*Renewable Energy Systems power system faults*", Printpap lodz ,Wrocław University of Technology,2011, pp 9-15

[19] A. Charles Adewole, *Investigation of methodologies for fault detection and diagnosis in electric power system protection*, October 2012, pp 17-21.

[20] W. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics, 1943, Vol. 5, pp. 115–133.

[21] F. Rosenblatt, *The perceptron A probabilistic model for information storage and organization in the brain*, Psychological Review, Vol. 65, pp. 386–408, 1958.

[22] MATLAB R2016a neural nerwork tool box.

[23] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press.1996.

[24] Martin T. Hagan , Howard B. Demuth , Mark Hudson Beale, Orlando De Jesús"*Neural network design* " second edition,ebook website:hagan.okstate.edu/nnd.html, page (8-12)-(8-13),(9-10)-(9-11)

[25] Jan Iżykowski ,"*Renewable Energy Systems power system faults*", Printpap lodz ,Wrocław University of Technology,2011, pp 7-8.

[26] Alexander von maer, *Electric power system concepts introduction*, IEEE press wiley inter science, 2002, pp 127.

[27] Wiliam .H kersting "*Distribution system modeling and analysis*" CRC press, 2002, page 252

[28] M. T. Hagan , H. B. Demuth , M. H. Beale, O. De Jesús"*Neural network design* " second edition,ebook , page volume 1012, pp 22-3,22-5

[29] ABB group, *Relion protection and control ABB 615 series technical manual*, revision E, 2010, pp 361.

[30] *ETAP version 11.0.0 user guide*, operation technology Inc.

[31] N. Yadav, An *Introduction to neural Network Methods for Differential Equations*, springer, year 2015, pp 13-14.

[32] G.Lalithamma and S. Puttaswam, *Literature Review of Applications of Neural Network in control systems*, International Journal of Scientific and Research Publications, Volume 3, Issue 9, September 2013,pp 1-6.

[33] Martin T. Hagan , Howard B. Demuth , Mark Hudson Beale, Orlando De Jesús"*Neural network design* " second edition,ebook website:hagan.okstate.edu/nnd.html, page (1-5)

Appendix A: Data used for feeder simulation

3 winding TRANSFORMER DATA				
Rated Power, Prated (HV)	=	25		MVA
Rated Power, Prated (MV)	=	12.5		MVA
Rated Power, Prated (LV)	=	12.5		MVA
Rated Voltage				
HV, Vnom[1]	=	132		kV
MV, Vnom[2]	=	33		kV
LV, Vnom[3]	=	15		kV
% Impedance (HV/MV)	=	11		
% Impedance (HV/LV)	=	6.25		
% Impedance (MV/LV)	=	4.26		
Vector Group	=	YNyn0d11		
HV SIDE Primary-winding 1, CT Ratio (Inom,a)	200	1	A	(200/1A)
MV SIDE Primary-winding 2, CT Ratio (Inom,b)	300	1	A	(300/1A)
LV SIDE Primary-winding 3, CT Ratio (Inom,c)	600	1	A	(600/1A)
OLTC Range	+	12.5		%
OLTC Range	-	12.5		%
Highest voltage tolerence, Vmax	=	148.5		kV
Lowest voltage tolerence,Vmin	=	115.5		kV

Source: transformer name plate and records in EEP

2 winding TRANSFORMER DATA			
Rated Power, Prated (HV)	=	25	MVA
Rated Power, Prated (LV)	=	25	MVA
Rated Voltage			
HV, Vnom[1]	=	132	kV
LV, Vnom[2]	=	15	kV
% Impedance (HV/LV)	=	11	
Vector Group	=	YNyn0d11	
OLTC Range	+	12.5	%
OLTC Range	-	12.5	%
Highest voltage tolerence, Vmax	=	148.5	kV
Lowest voltage tolerence,Vmin	=	115.5	kV

Details of MV line length and loads

vilages	**MV line length of tap**	**Conductor type and Area**	**Transformer rating used & Qty.**
Gumguma	78.2km (from Assela)	AAAC 95mm2	2x50kVA
			1x25kVA
			2x100kVA
Munesa	13.8km(from kersa)	AAAC 95mm2	2x100kVA
Chefa	online with gumguma	AAAC 95mm2	1x100kVA
			1x50kVA
Chebi	3.3km(from munessa)	AAC50mm2	1x100kVA
Konche	2.9km(from main feeder)	AAC50mm2	1x50,1x100kVA
Sagure	13km(from main feeder)	AAC50mm2	1x50kVA
			1x25kVA
			2x100kVA
			1x200kVA
Wajishagre	38.2km(from munessa)	AAC 50mm2	1x50,2x100kVA

Conductors detail						
	unit	**area**		**unit**		
		$50mm^2$	$95mm^2$		$0.06in^2$	$0.15in^2$
wire strand	mm	7	7	inch	7	7
diameter of wire	mm	3.1	4.19	inch	0.1052	0.1672
Resistance(25oC)	ohm/km	0.542	0.343	ohm/kft	0.2646	0.1049
over all conductor diameter	mm	9.3	12.57	inch	0.316	0.502
actual area	mm2	52.83	96.52	in2	0.0608	0.1537
GMR	mm	2.8956	0.1824	ft	0.0095	0.0152
Inductance	ohm/km	0.35049	0.31574	ohm/kft	0.1069	0.0963
Capacitance	Mohm/km	2.22426	2.97311	Mohm/kft	0.6784	0.9068

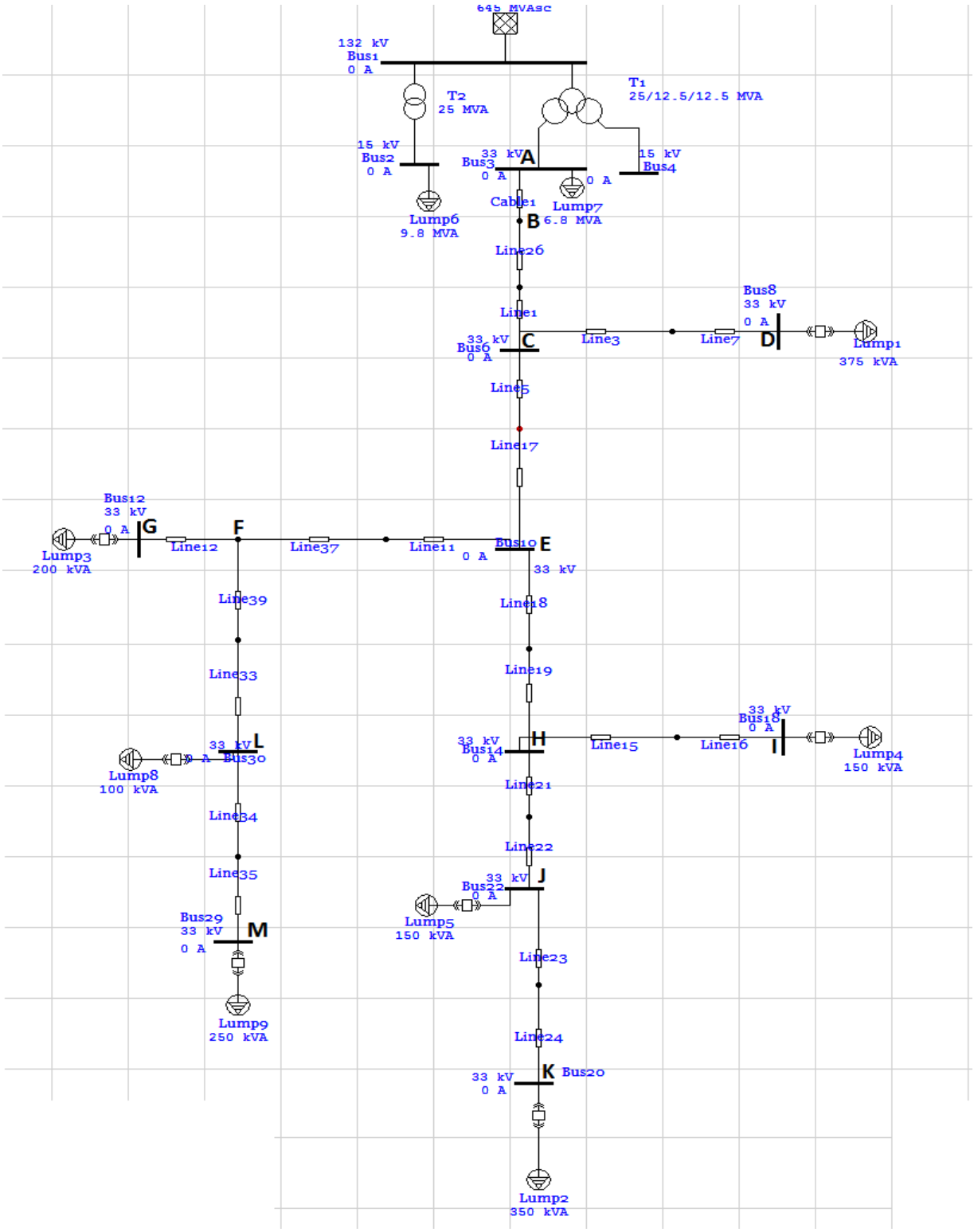
source :Aberdare cables owned by power tech and EEU/UEAP technical spec of conductors

Subatation relay(REF 615) setting		
overcurent	0.8In	In= 150 CT ratio=75-150/1/1A
earth fault	0.2In	In= 150 CT ratio=75-150/1/1A
overvoltage	1.15Un	Un=100 VT ratio 33kV/110V
overvoltage	0.90Un	Un=100 VT ratio 33kV/110V

Cable data power cable single core XLPE	
resistance	0.099
reactance	0.02

Grid representation	
Assela 132kv Substation	
MVASc(1P)	190MVA
MVASc(3P)	644.96MVA
Isc(3P)	2.82kA
Isc(1P)	2.494kA
x/R,	2.725

Appendix B: Configuration of test feeder



1)Line configuration

Config description	length	Type/size of conductor	Total distance fom point A	Branch	Sub branch
A-B	60m	Power cable cupper		0	0
B-C	12km	AAAC 95mm^2	12km	0	0
C-D	13km	AAC 50mm^2	25km	1	0
C-**E**	40km	AAAC 95mm^2	52km	0	0
E-F	10.8km	AAAC 95mm^2	62.8km	2	0
F-G	3km	AAAC 95mm^2	65.8km	2	0
F-L	3.3km	AAC 50mm^2	65.1km	2	1
L-M	35.5km	AAC 50mm^2	100.6km	2	1
E-H	17km	AAAC 95mm^2	69km	0	0
H-I	3.3km	AAC 50mm^2	72.3km	3	0
H-J	3km	AAAC 95mm^2	71	0	0
J-K	7.2km	AAAC 95mm^2	78.2km	0	0

2) Loading conditions considerations made to simulate the faulty condition

Nodes	Total installed DT capacity (100%) loaded	No load	50% load	100% load	150% load
D	375kVA	0	187.5	375	562
G	200kVA	0	100	200	300
L	100kVA	0	50	100	150
M	250kVA	0	125	250	375
I	150kVA	0	75	150	225
J	150kVA	0	75	150	225
K	350kVA	0	125	350	375

3) Applicable IEEE/ANSI standards

- IEEE 80-1986, IEEE 80-2000, IEEE 665-1995 for grid ground system
- IEEE CFS.116 standard for transformer tap optimization
- IEEE C37 /ANSI series short circuit analysis, complete compliance with IEC 60056,60282,61363,60781.
- IEEE 1584-2002 standard for arc flash analysis
- IEEE 519A standard for harmonic analysis

Appendix C: MATLAB code for GUI implementation of fault estimator

```
functionvarargout = Samithesisgui(varargin)
% SAMITHESISGUI MATLAB code for Samithesisgui.fig
%SAMITHESISGUI, by itself, creates a new SAMITHESISGUI or raises the existing
%      singleton*.
%
% Edit the above text to modify the response to help Samithesisgui

% Last Modified by GUIDE v2.5 02-Jan-2018 10:26:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @Samithesisgui_OpeningFcn, ...
'gui_OutputFcn',  @Samithesisgui_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

ifnargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before Samithesisgui is made visible.
functionSamithesisgui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no samuelargs, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Samithesisgui (see VARARGIN)

% Choose default command line samuel for Samithesisgui
handles.samuel = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Samithesisgui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
functionvarargout = Samithesisgui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning samuelargs (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% Get default command line samuel from handles structure
varargout{1} = handles.samuel;

function input1_Callback(hObject, eventdata, handles)
% hObject    handle to input1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
input1 = str2double(get(hObject,'String'));
if (isempty(input1))
set(hObject,'String','0')
end
guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of input1 as text
% str2double(get(hObject,'String')) returns contents of input1 as a double
% --- Executes during object creation, after setting all properties.
function input1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function input2_Callback(hObject, eventdata, handles)
% hObject    handle to input2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
input2 = str2double(get(hObject,'String'));
if (isempty(input2))
set(hObject,'String','0')
end
guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of input2 as text
% str2double(get(hObject,'String')) returns contents of input2 as a double
% --- Executes during object creation, after setting all properties.
function input2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
 % Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function input3_Callback(hObject, eventdata, handles)
% hObject    handle to input3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
input3 = str2double(get(hObject,'String'));
if (isempty(input3))
set(hObject,'String','0')
end
guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of input3 as text
% str2double(get(hObject,'String')) returns contents of input3 as a double

% --- Executes during object creation, after setting all properties.
function input3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function input4_Callback(hObject, eventdata, handles)
% hObject    handle to input4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
input4 = str2double(get(hObject,'String'));
if (isempty(input4))
set(hObject,'String','0')
end
guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of input4 as text
%str2double(get(hObject,'String')) returns contents of input4 as a double
% --- Executes during object creation, after setting all properties.
function input4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
 % Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function input5_Callback(hObject, eventdata, handles)
% hObject    handle to input5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
input5 = str2double(get(hObject,'String'));
if (isempty(input5))
set(hObject,'String','0')
end
guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of input5 as text
% str2double(get(hObject,'String')) returns contents of input5 as a double
```

```
% --- Executes during object creation, after setting all properties.
function input5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



function input6_Callback(hObject, eventdata, handles)
% hObject    handle to input6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
input6 = str2double(get(hObject,'String'));
if (isempty(input6))
set(hObject,'String','0')
end
guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of input6 as text
%        str2double(get(hObject,'String')) returns contents of input6 as a
double


% --- Executes during object creation, after setting all properties.
function input6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to input6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


Va = get(handles.input1,'String');
%v1 = srt2double(Va);
Vb = get(handles.input2,'String');
%v2 = str2double(Vb);
```

```
Vc = get(handles.input3,'String');
%v3 = str2double(Vc);
Ia = get(handles.input4,'String');
%i1 = str2double(Ia);
Ib = get(handles.input5,'String');
%i2 = str2double(Ib);
Ic = get(handles.input6,'String');

inpt ={Va;Vb;Vc;Ia;Ib;Ic};
s = str2double(inpt)./100;

load('finalcombinednet.mat');

y = sim(ffbp,s);
y1= num2str(y);

set(handles.text9,'string',y1);
 t = y(2);
if (-0.005<=t)&&(t<0.005)
y(2)= 0;
elseif (0.005<=t)&& (t<0.013)
y(2)= 1;
elseif (0.013<=t)&&(t<0.025)
y(2)= 2;
elseif (2.5<=t)&&(t<3.5)
y(2)= 3;
elseif (t>0.035)
%printf('Error')
end
 m = y(3);
if (-0.005<=m)&&(m<0.004)
y(3)= 0;
elseif (0.004<=m)&&(m<0.018)
y(3)= 1;
elseif (0.015<=m)&&(m<0.025)
y(3)= 2;
elseif (m>2.5)
% printf('Error');
end

set(handles.text17,'string',abs(y(1))*100);
set(handles.text18,'string',y(2));
set(handles.text19,'string',y(3));
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
sami = imread('feder.png');
%sami = imread('munessa.png');
%set(handles.axes1,imshow(sami));
%set(handles.showline,'string',imshow(sami));
imshow(sami);

% --- Executes on selection change in popupmenu2.
```

```
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
str = get(hObject, 'String');
val = get(hObject,'Value');
% Set current data to the selected data set.
switchstr{val};
case'feder1'% User selects feder1.
handles.current_data = handles.feder1;
case'feder2'% User selects feder2.
handles.current_data = handles.feder2;

end
guidata(hObject,handles)
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2


% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```