Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
Telecommunication Engineering Graduate Program

Neural Network based 3G Mobile Sites Fault Prediction: A Case Study in Addis Ababa, Ethiopia

Submitted by: Asmelash Tesfay

Advisor: Dr. Yihenew Wondie

A Thesis submitted to the School of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Award of Degree of Master of Science in Telecommunication Engineering (Telecommunication Network Engineering Track)

November, 2018

Addis Ababa, Ethiopia

Addis Ababa Institute of Technology
School of Electrical and Computer Engineering
Telecommunication Engineering Graduate Program

Neural Network based 3G Mobile Sites Fault Prediction: A Case Study in Addis Ababa, Ethiopia

Submitted by: Asmelash Tesfay

Signature

Advisor Signature

Evaluator Signature

Evaluator Signature

Abstract

As cellular mobile networks are evolving in technology and service type, the number of mobile network site infrastructure is increasing. Nowadays, faulty cellular mobile network sites per day in ethio telecom are significant in number and have big impact to customers and operator in QoS, revenue and maintenance cost. Fault maintenance techniques commonly applied by ethio telecom is corrective maintenance approach. This only helps to recover services after interruption. However, it is important to implement the proactive maintenance approach to make mobile sites reliable and available. This helps to provide services according to standards and improve the quality of service delivery to customers. To mitigate mobile network site faults before happening, fault occurrence time prediction is an important technique for the implementation of proactive maintenance strategy. This Neural Network based 3G Mobile Fault Occurrence Prediction research work is conducted based on the Nonlinear Auto regressive (NAR) Neural Network time series prediction method using Addis Ababa 3G mobile sites in a case study. To train the neural network 15,950 actual fault occurrence time data are used. The algorithm used to train the neural network is Levenberg-Marquardt which is fast and efficient, and an iterative approach of hidden layer neuron number selection is applied. Finally, the best model is selected with minimum value of mean square error of prediction. Also, the model is tested with actual fault occurrence time which was not used in the training and achieved 90.71% in prediction. Therefore, it is efficient in prediction accuracy, fast and adaptive with future data.

Key Words: Mobile Site, Fault, Corrective Maintenance, Proactive Maintenance, Fault Occurrence Time, Prediction, Neural Network, Time Series

Acknowledgment

I would like to express my deep gratitude to my advisor, Dr. Yihenew Wondie for all his great guidance and support throughout this research. As my advisor, his observations, guidance and comments help me to find the right direction of the work to move forward and to complete the thesis. I am very much thankful to him. Also, I want to thank to Dr. Yalemzewd Negash, Dean of Electrical and Computer Engineering School in AAiT, for his friendly advice during the program and all staffs who have role in the program.

I am thankful to ethio telecom for giving me full scholarship and allowing me to use data for my research. The data sets were very helpful for my experiments and evaluation. I want to thank to all ethio telecom staffs and friends who motivate, support and provide me special facilities during the program. Also special thanks to Helen Temesgen from National Network Operation Center (NNOC), ethio telecom, for her expertise support in data understanding.

To my family, thank you for your support & being exemplary through my study. I would like to dedicate the work to my son, Ruhus Asmelash for his unconditional love and performance. He is so much to me.

Table of Content

List of Figures

List of Tables

List of Acronyms

2G	Second Generation
3G	Third Generation
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BPNN	Back Propagation Neural Network
CDMA	Code Division Multiple Access
CTMC	Continuous Time Markov Chain
DSLAM	Digital Subscriber Line Access Multiplexer
EMS	Element Management System
GSM	Global System for Mobile Communication
KDD	Knowledge Discovery from Data
KPI	Key Performance Indicator
LM	Levenberg Marquardt
LTE	Long Term Evolution
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MPE	Mean Percentage Error
MSAG	Multi Service Access Gateway
MSE	Mean Squared Error
NAR	Nonlinear Autoregressive
NMSE	Normalized Mean Squared Error
NN	Neural Network
NRMSE	Normalized Root Mean Square Error

PSTN Public Switched Telephone Network

QoS Quality of Service

RAM Reliability, Availability and Maintainability

RMSE Root Mean Squared Error

SLA Service Level Agreement

SSE Sum of Squared Error

SVM Support Vector Machine

UNE Unavailable Network Elements

1. Introduction

The telecommunication industry is evolving to high capacity networks and multi-services delivery due to business requirements and technological innovations [1]. The growth in wireless technologies, particularly cellular mobile networks is attracting customers to ease their life. It is largely due to the seamless mobility support and in addition, it is continuously growing in capacity. The dependency of businesses on mobile communication services is growing significantly. The mobile site infrastructure number is increasing due to coverage and capacity demands of customers. If the number of users and network elements are increasing, maintenance frequency and maintenance cost are also increasing [2]. Besides, the management of network infrastructures are getting complex to provide reliable services. Thus, fault should be prevented, or restoration time should be low so that impacts on customers and operators will be as low as possible. Since mobile services such as voice, data and internet are becoming very sensitive to faults, operators are required to meet standards and Key Performance Indicators (KPIs) defined by regulatory bodies and Service Level Agreement (SLA) contracts signed with customers.

Furthermore, the demand of customers is changing to services with high reliability in addition to high bandwidth requirements to ensure fault free operation of critical businesses which is a new challenge for operators [3]. Regardless of the time interval, it is important for the customer to have a reliable network when they need to make calls. Therefore, predictive analysis, a proactive approach, is necessary due to its effectiveness. Intending to deal with the problem before it happens, predictive analysis utilizes mathematical algorithms, such as various modeling techniques, to illustrate the relationship among functioning variables of a system. These models can learn the behavior of the system under normal conditions based on these variables, and they can also monitor the patterns that forecast a troublesome scenario. Specifically, in telecommunications, operators would be able to estimate that a fault is going to occur in a certain amount of time, and thus, could take steps to prevent this fault. This would eliminate any possible difficulties that a customer would experience and is much better than the reactive approach. The prediction techniques are important since they are the precondition to proactive fault management [4].

Although network faults are unavoidable for telecommunication operators, there is a demand of proactive approaches to reduce faults. One promising direction potentially being less expensive than traditional fault tolerance methodologies is to predict the occurrence of faults to prevent them or to prepare repair mechanisms for an upcoming fault. This approach is called proactive fault handling [5]. Thus, network faults can be predicted and prevented to improve the overall network performance in terms of availability and reliability [6].

This research work is to conduct mobile sites fault time occurrence prediction based on real fault time series data. The motivation of this research is due to high unavailability of telecom networks especially in mobile network sites, faults are frequently occurring, and the fault sustains for long duration which affects businesses significantly.

Existing fault prediction methods consist of model-based approach, probabilistic method and knowledge-based approach, and lately increasing attentions have been focused on neural network-based methodology. Neural networks are effective in forecasting complex systems [7]. Moreover, there are other prediction techniques such as linear regression, exponential regression, and support vector machine (SVM). Also, ARMA model is a widely used linear time series model to trend and predict future behavior, however, because of the highly complicated and non-stationary nature of some processes, the observation of a set of past values is possibly nonlinear [2]. Therefore, there is a demand of prediction models to solve the nonlinear problems.

Neural network, which are commonly named Artificial Neural Network (ANN) are defined as a machine learning technique which takes its inspiration from biological neural networks, with both consisting of the same basic components. NNs are a very popular technique for prediction, and many variations were utilized to determine which variation had the greatest success. NNs have individual neurons that act as a processing unit by taking in one or more input to provide an output. At all neurons, an associated weight is assigned to every input, and this can adjust the strength. The neuron then proceeds to sum all the inputs and calculate an output to be passed on [5].

Currently the maintenance is becoming predictive maintenance, with the meaning of predicting fault and carrying out the maintenance only when necessary. This maintenance benefits to minimize downtime of equipment and service interruptions in providing telecom services. As a result, the quality of service experience by customers is going to be better than the corrective maintenance approach .Telecom operators, like ethio telecom can get benefits in revenue and customer satisfactions by improving the networks reliability and availability performances.

1.1. Statement of the Problem

Mobile site faults occur by base station hardware and software malfunctions, power outages, faulty links, multi-vendor incompatibility, and misconfiguration of parameters during network operation. Moreover, the rate of faults is essentially proportional to complexity of hardware and software that constitute the mobile network. This compromises network reliability, which is defined as the fraction of time the network, including its components, is operational. Hence, the results of these faults are a drop in a quality of service provided to customer and increase cost for the service provider or operator as the maintenance times remains high. The problem lies in the current corrective (reactive) approach of maintenance applied by most carriers and operators. This only responds to the problem and can only limit the magnitude of the problems caused, not prevent it from happening [4]. Additionally, cellular networks are inherently subject to cell outages caused by base stations hardware & software malfunctions, misconfiguration of cell parameters during routine network operation [8].

Similar network fault scenarios happen in ethio telecom, which is Ethiopia's sole telecom operator. The company owns a huge infrastructure and its customer base is growing. The operator is also making continuous network expansions with large amount of investments. Table 1.1 shows the current capacity of ethio telecom networks.

Table 1.1 Ethio telecom's network capacity [9, 10]

Description	Amount
Mobile network capacity	62 Million
Mobile network sites	7500
Fixed Line network capacity	3 Million
Fiber Route Length	21, 178Km
International Gateway Capacity	42 Gbps

Thus, ethio telecom became the largest mobile operator in Africa in terms of subscriptions, with 57.34 million mobile subscribers as of Nov 2017 [9, 10]. It has reached 85% mobile coverage. However, as an operator, ethio telecom is facing challenges like network faults in various networks including Multi Service Access Gateway (MSAG), Digital Subscriber Line Access Multiplexer (DSLAM), Public Switched Telephone Network (PSTN), Code Division Multiple Access (CDMA) networks, and Mobile networks including Global System for Mobile Communication (GSM), Universal Mobile Telecommunications System (UMTS) and Long Term Evolution (LTE). Table 1.2 indicates the average daily critical network faults share by technology. From this, it can be observed that faulty mobile site number share is the largest, which is 68.71% of the total network fault number per day which includes MSAG, PSTN, DSLAM and CDMA networks.

Table 1.2 Average daily faulty network share by technology

Month	Technology				
	Mobile (%)	CDMA (%)	DSLAM (%)	MSAG (%)	PSTN (%)
Aug-2017	64.74	23.78	3.74	6.58	1.17
Sep-2017	70.86	16.19	3.68	7.19	2.89
Oct-2017	68.75	15.64	4.65	8.78	2.20
Nov-2017	69.10	12.59	4.56	10.67	3.08
Dec-2017	65.91	11.44	5.73	12.84	4.08
Jan-2018	69.70	11.25	5.31	12.82	0.93
Feb -2018	71.89	9.40	4.94	13.39	0.38
Average	68.71	14.33	4.66	10.32	2.11

Source: Performance report of ethio telecom, Aug 2017 to Feb 2018[11]

Hence, daily critical network faults which are named as daily unavailable network elements (UNE) are significant in number and the services which are affected, the revenue loss and the maintenance cost are significant for the company. Table 1.3 below shows sample report of daily average and monthly total unavailable networks and duration of downtime ranges in days. Most faults sustain unsolved for a longer time in which their associated cost is high in terms of revenue loss during service interruption, increasing maintenance cost and QoS impacts. The sample report here does not include mobile access networks deployed by Ericson because it was in a project status during the time of network availability analysis of the data.

Table 1.3 Number of unavailable network elements for a duration of Aug-17 to Feb- 2018

Month	Daily Average UNE	Daily Average UNEs Per Month Based on Down Time Duration (in days)			UNEs >=3 days Down Time Duration (%)
		< 1 Day	1 <= Days < 3	>= 3 Days	
Aug-2017	853	412	159	282	33.06
Sep-2017	820	382	138	300	36.59
Oct-2017	601	292	107	202	33.61
Nov-2017	373	206	71	96	25.74
Dec-2017	311	189	54	69	22.19
Jan -2018	296	191	49	56	18.92
Feb-2018	423	276	82	65	15.37
Average	525	278	94	153	29.1

Source: Network performance report of ethio telecom, 2017-2018

The design target of network availability of mobile network sites (base stations) is 99.999% [12]. Table 1.4 indicates the operational performance of mobile sites in terms of availability. There is performance gap from the design target due to faults and results in high revenue loss roughly estimation up to 40 million Birr per month [11].

Table 1.4 Network Availability performances 2017

Technology/Region/vendor	Network availability(%)
GSM-Addis Ababa (Huawei)	99.004
3G-Addis Ababa(Huawei)	98.774
LTE-Addis Ababa(Huawei)	99.84
GSM-Region-Huawei	89.01
3G-Region-Huawei	90.14
2G & 3G-Region-ZTE	93.42
Average	95.03

Source: Network performance report of ethio telecom, 2017

Considering the huge mobile sites infrastructure in ethio telecom which is around 7500 (2G,3G and LTE) and the significant number of daily mobile site fault, quality of service is negatively affected. This implies, there is a need of network availability improvement of mobile sites. Thus, it is essential to do research on mobile sites fault occurrence time prediction which helps to mitigate the faults by implementing proactive maintenance techniques. In this, as a case study 3G Addis Ababa mobile sites are selected since the availability performance is less than 99.999% [12], have 99.7% site number coverage, and customers are very sensitive to faults in Addis Ababa.

1.2. Objective

1.2.1 General Objective

The main objective of this research is to predict fault occurrence time of 3G mobile sites based on Neural Network model approach by using the fault history data of the sites.

1.2.2 Specific Objectives

The specific aims of the research are:

- To undertake literature survey on fault, maintenance, time series, prediction, neural network concepts and fault prediction related works;
- To perform fault history data collection of 3G mobile sites in Addis Ababa;
- To select prediction technique, to model the system;
- To do neural network training simulation using MATLAB;
- To select the best linear mathematical model relationship between the predicted and actual data by using performance parameters mean square error (MSE) and Correlation factor (R value);
- To do performance analysis and evaluation of the predicted and actual fault occurrence time;

- To construct a prediction mathematical model representing the relationship between the predicted and actual fault occurrence time

1.3. Methodology

After the problem is identified and objectives are set, the methodology followed in this research work is represented by using flow chart diagram in Figure 1.1. It shows the main activities accomplished to complete the thesis work.

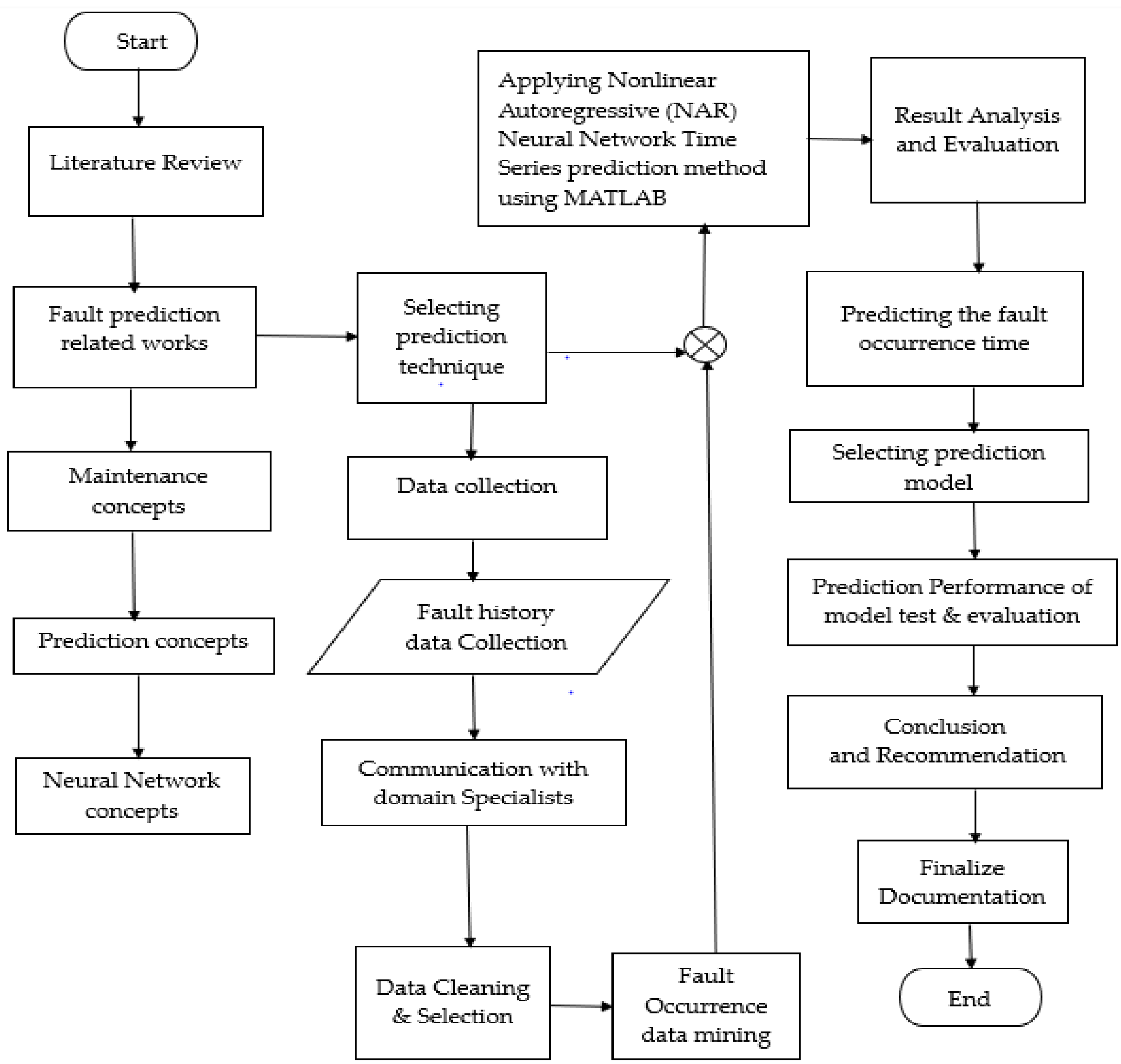

Figure 1.1 Research methodology flow chart diagram

1.4. Scope and Limitation

1.4.1 Scope

The scope of this research is to predict fault occurrence time of 3G mobile sites in Addis Ababa by using nonlinear autoregressive (NAR) Neural Network time series prediction technique. The data collected covers from February 2018 to June 2018 which is a five month period. Fault severity considered here is critical ones.

1.4.2 Limitation of the research

There were limitations in the research in getting the fault history data for more than five month period at one time. It was not available in the server due to data storage related configurations. There was also limitation in filtering only 3G mobile sites from the fault history data of mobile sites because the site identification number assigned to a site with 2G,3G and LTE networks is one number.

1.5. Contribution of the research

Predicting the fault occurrence time of mobile site is important for telecom operator to implement proactive maintenance strategy which helps to ensure network reliability and availability up on providing services. Moreover, this enables operators to deliver acceptable QoSs expected by regulatory bodies and SLA agreements with customers, reduce maintenance cost and increase revenue. It also helps to schedule resources based on priority.

1.6. Related Works Review

The author in [4] has aimed to introduce a proactive approach for failure prediction of time series data by surveying a wide range of techniques. The prediction techniques, like Deep Neural Network with Auto-encoders, Nonlinear Autoregressive Neural Network, Gaussian Kernel Support Vector Machine Regression, Exponential Regression, Linear

Regression, Neural Network and Linear Support Vector Machine Regression, were used along with Continuous Time Markov Chain (CTMC) analytical model. Moreover, to differentiate their performance among those techniques, Normalized Root Mean Square Error (NRMSE) was used for the error calculation. The results indicate that most of the models had relatively the same success in predicting the fault's inter-arrival time on the test data, besides the deep neural network with auto-encoders has a huge improvement in NRMSE demonstrates significantly better effective fault prediction than all other techniques. Moreover, Nonlinear Autoregressive Neural Network perform well. However, the linear models performed worse than their nonlinear counterparts. This shows that the data does not follow a linear trend and has a nonlinear relation.

To overcome outage damage caused by temporary failure and ensure excellent operation of the equipment, Wei et al. [7] presents an effective prediction model which combined the back propagation neural network (BPNN) with multi-agent cooperation grouping algorithm. The values of weights and thresholds of BPNN were obtained through optimization results of the multi-agent cooperation grouping algorithm and improves the forecasting precision of equipment failure.

To forecast the telecommunication network outage duration [13] used statistical analysis with Autoregressive Integrated Moving Average (ARIMA) with five statistical indicators, namely, Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Mean Percentage Error (MPE) and Mean Absolute Error (MAE) are used for comparison of performance. ARIMA (2, 0, 2) model is suitable in predicting the network outage of telecommunication networks. Also, Jie Zhao [14] uses auto-regressive moving average (ARMA) model for device down time forecasting based on transformed historical data. The 8 orders moving average method was adopted to obtain mean stationary time series with a defined historical data calculated by an algorithm. The proposed method is helpful to reflect the equipment condition and thereby can aid predictive maintenance in manufacturing process and reduce the downtime costs.

In [15], it is explained that neural networks have been applied very successfully in the identification and control of dynamic systems. The universal approximation capabilities of the multilayer perceptron have made it a popular choice for modeling nonlinear systems and for implementing general-purpose nonlinear controllers. Neural networks are powerful when applied to problems whose solutions require knowledge. Multilayer feed-forward neural networks with back propagation training are now commonly used for prediction and time scries forecasting.

Mathematical and statistical modeling are the prominent approaches used for failure predictions. A.Abu-Samah et al. [16] presented a methodology for failure prediction using Bayesian Network approach which is probabilistic, and is complemented with the extraction of rules for failure prediction with computation of lead time and predictability index. Its advantage is the use of predictors coming from multiple data sources as predictors in a single prediction model. It uses event driven predictors as temporal characteristics successfully to predict the potential failures. The proposed methodology is tested and found promising result.

D.Hunter et al. [17], proposed a solution to the problem of selecting proper size and topology of neural networks, this is to use the least possible number of neurons along with many training patterns. The problem is even more complex because often when the neural network is trained to very small errors, it may not respond properly for patterns not used in the training process.

From [4] Neural network has performed better for nonlinear data prediction, to differentiate their performance among those techniques, Normalized Root Mean Square Error (NRMSE) was used for the error calculation. But it has a fault data size limitation and covers only one-month data. For this research work the opportunity is to use actual mobile fault data up to 5-month period in applying the prediction method.

1.7. Research Organization

This research work contains six chapters. Chapter one deals with the introduction part which contains statement of the problem, objectives, methodology of the research approach, scope and limitations, and contribution of the research and related works review on fault prediction techniques. Chapter two contains theoretical discussion on fault, maintenance, Quality of Service (QoS) and SLA, time series, prediction and neural network. Chapter three explains about fault prediction methodology of the research. Chapter four includes network training and result evaluation part of the research. Chapter five contains the finding and discussion part and chapter 6 covers the conclusion and future work.

2. Theoretical Discussion

The theoretical concept of fault, maintenance, QoS and SLA, time series, prediction, neural networks, learning algorithms, benefit of neural networks, and data mining are discussed in the following sections. This helps to understand the basic concepts, technical definitions and their significances.

2.1. Fault in Telecom Networks

Fault is defined as the inability of an item to perform a required function, excluding that inability due to preventive maintenance, lack of external resources or planned actions. It is often the result of a failure of the item itself, but it may exist without the prior failure of the item [18].

Maintenance is defined as all actions performed on the item to retain it in or to restore it to a specified state. There are two common maintenance approaches to resolve faults. These approaches are corrective (reactive) and preventive (proactive) maintenance [6].
A preventive maintenance retains the item functionality and its aims are to detect and repair hidden faults, to use prescribed procedures, test all relevant function, carry out at scheduled intervals to reduce the probability of failures or the degradation of the functionality of the item. A corrective maintenance is the reestablishment of the item functionality. It is initiated after fault (defect or failure) detection. Corrective maintenance includes fault detection(recognition), localization (isolation and diagnosis), correction, function checkout [6].

The priority of the maintenance techniques depends on cost of dealing with failure or cost of preventing a failure under the given parameters like KPI, QoS, customer satisfaction, and revenue to generate. Figure 2.1 represents the summary of the questions to be considered when it is dealt with the maintenance techniques. It has a center point as a reference to show the weight of the maintenance cost direction under some given targets to be achieved. This helps to do analysis before decision is achieved to select the type of maintenance methods.

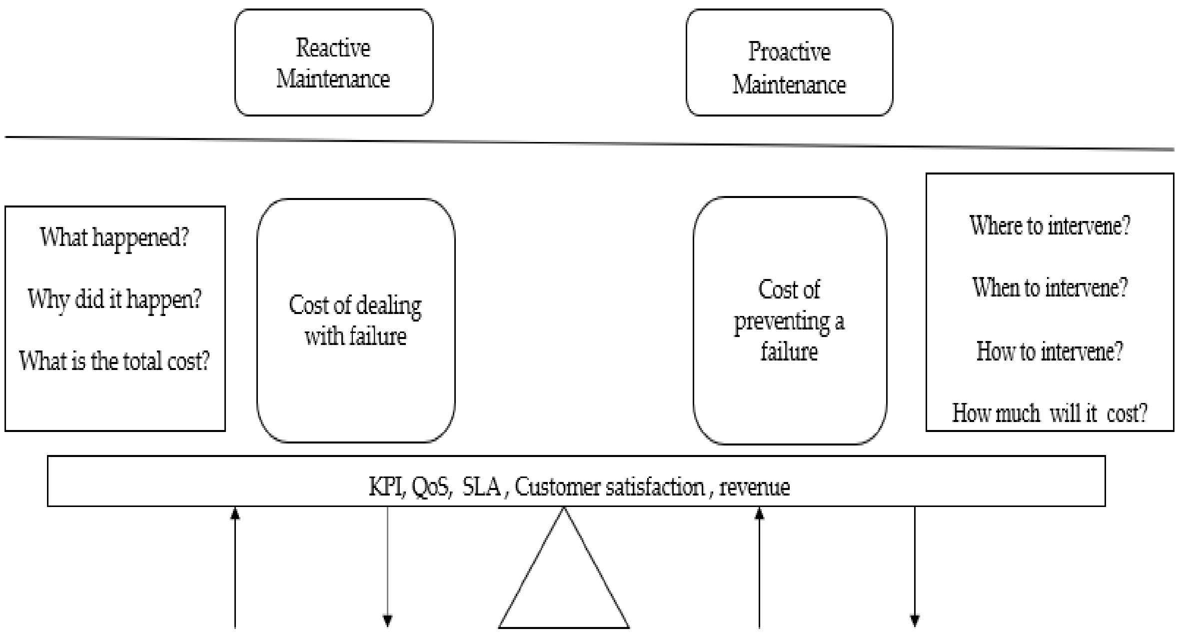

Figure 2.1 Summary of maintenance analysis techniques [6]

Telecom network faults are categorized depending their impact severity as warning, minor, major and critical faults in network management systems. On daily fault maintenance activity handling, priority is given for critical, major, minor and minor respectively. Because service impact and reason for the faults are different.

QoS is defined as the totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service. QoS comprises both network performance and non-network related performance. Examples of network performance are bit error rate, latency, etc., and for non-network performance provision time, repair time, range of tariffs and complaints resolution time, etc. The list of QoS criteria for a service would be dependent upon the service and their relevance could vary among the segments of the customer population [18].

SLA is defined as a formal document listing a set of performance characteristics and target values (or range) to be delivered for a service or portfolio of services by the service provider. It may include statements about performance, tariffing and billing, service delivery compensations and escalation procedures in cases of disagreements between service provider and SLA user customers [18].

2.2. Time Series

Time series is a sequential set of data points, measured typically over successive times. The measurements taken during an event in a time series are arranged in a proper chronological order. Neural networks have attracted increasing attentions in the domain of time series prediction. Time series is mathematically defined as set of vectors x(t), t=0,1, 2,…. where t represents the time elapsed. The variable is treated as random variable [19].

The term "time series" itself, denotes a data storing format, which consists of the two mandatory components - time units and the corresponding value assigned for the given time unit. Values of the series need to denote the same meaning and correlate among the nearby values [20]. A time series containing records of a single variable is termed as univariate. But if records of more than one variable are considered, it is called multivariate. A time series can be continuous or discrete. In a continuous time series observations are measured at every instant of time, whereas a discrete time series contains observations measured at discrete points of time. Usually in a discrete time series the consecutive observations are recorded at equally spaced time intervals such as hourly, daily, weekly, monthly or yearly time separations [19].

In practice a suitable model is fitted to a given time series and the corresponding parameters are estimated using the known data values. In time series prediction, past observations are collected and analyzed to develop a model which captures the under lying data generating process for the series. The future events are then predicted using the model.

2.3. Prediction

It is a technique, in which past values of one or more time series are used to predict future values [15]. To predict a time series, for example, it requires a neural network that maps the previous series values to future developments of the time series, i.e. having longer sections of the time series helps to have enough training samples. Before beginning to predict a time series, some questions must be answered about the time series we are dealing with and ensure that it fulfills some requirements. The questions to answer are:

i. Is there any evidence which suggests that future values depend in any way on the past values of the time series? or does the past of a time series include information about its future?
ii. Are there enough past values of the time series that can be used as training patterns?

Once the time series is predictable, it is possible to predict in different "step ahead" predictions. There are different types of step ahead predictions. Among these, one step ahead prediction and two steps ahead predictions are mentioned below.

One step ahead prediction: It is the first attempt to predict the next future values of a time series out of past value is called on-step-ahead prediction. The network receives the last n observed state parts of the system as input and output the prediction for the next state (or state part). If we want to include many past values, a multilayer perceptron will require considerably less computational effort [22].

Two steps ahead prediction: Its attempt is to predict two step ahead future values. This helps to see farther into the future. The prediction becomes more easier the more past values of the time series are available [22].

Moreover, to express the accuracy of prediction there are different prediction performance measures, such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Sum of Squared Error (SSE), Root Mean Squared Error (RMSE), Normalized Mean Squared Error (NMSE) [19].

Equation (2.1) shows mathematical formula for MSE.

$$\mathrm{MSE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{t}=1}^{\mathrm{n}} [e(t)]^{2} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots. \quad (2.1)$$

where e(t) is error between the actual and predicted values at time t and n is the number of data set.

To add one area of prediction application, that is in network management systems. They collect real-time information from network nodes and can take reactive and proactive action to fix problems of different kinds. That means modern network management systems can perform automatic recovery and prediction [15]. Self-Organizing Network can also be an example to such applications.

2.4. Neural Network

A Neural Network is a machine that is designed to model the way in which the brain performs a specific task or function of interest. It is usually implemented by using electronic components or is simulated in software on a digital computer. To achieve good performance, neural networks employ a massive interconnection of simple computing cells referred to as “neurons” or “processing units” [22].

The definition of a Neural Network when viewed as an adaptive machine: It is a massively parallel distributed processor made up of simple processing units that has a natural tendency for storing experiential knowledge and making it available for use. Neural Network resembles the brain in two characteristics: [22]

i. Knowledge is acquired by the network from its environment through a learning process
ii. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge

Neural networks are often classified as single layer networks and multilayer networks. The number of layers in a network can be defined as the number of layers of weighted interconnection links between various layers. While determining the number of layers, the input layer is not counted as a layer, because it does not perform any computation. A single layer network consists of one layer of connection weights. The network consists of a layer of units called input layer, which receive signals from the outside and a layer of units called output layer from which the response of the network can be obtained [21]. The architecture of a single layer neural network is shown in Figure 2.2 below.

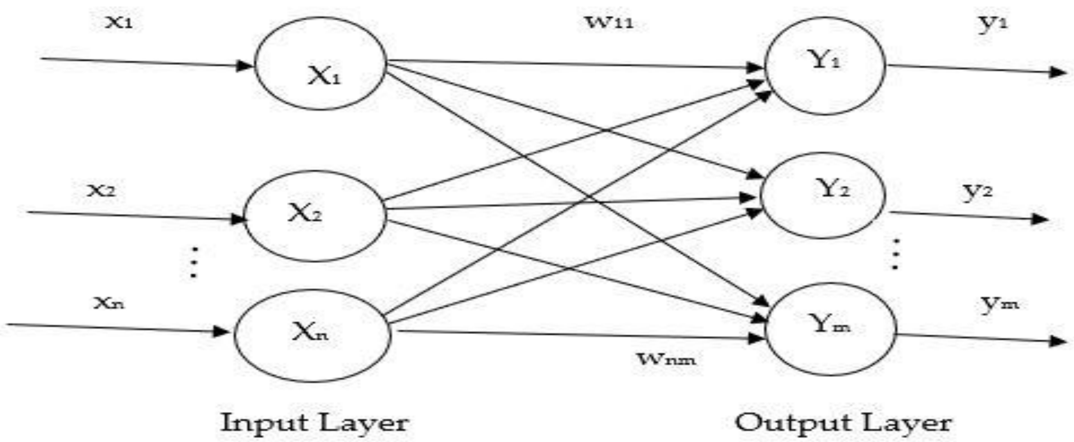

Figure 2.2 Single layer Neural Network [21]

In the single layer neural network, the variables are defined as follows:

x_1 ,……….,x_n are the inputs to the network

X_1, ………, X_n are the input neurons

Y_1 , ……….,Y_m are the output neurons

y_1 ,………,y_m are the outputs of the network

w_{11} ,…....,w_{nm} are the connection weights between the input and output neuron

A multilayer neural network consists of one or more layers of units (called hidden layers) between the input and output layers. Multilayer networks may be formed by simply cascading a group of layers; the output of one layer provides the input to the subsequent layer. A multilayer network with nonlinear activation function can solve any type of problem. However, training a multilayer neural network is very difficult [21]. The architecture of a multilayer neural network is shown in Figure 2.3 below.

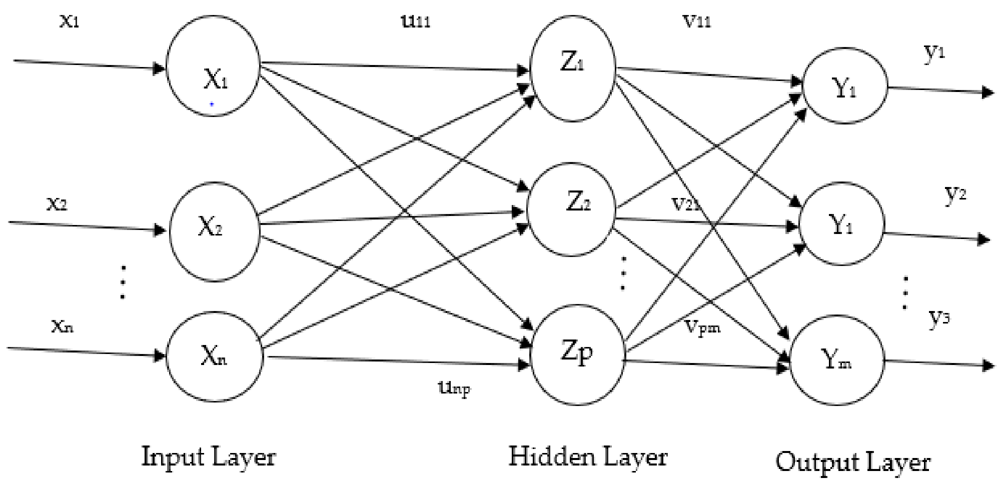

Figure 2.3 Multilayer Neural Network [21]

The variables in multilayer neural network are defined as follows:

x_1 ,……….,x_n , the inputs to the network

X_1, ………, X_n , the input neurons

Z_1,………..,Z_n , the hidden neurons

u_{11} ,………,u_{np} , the connection weight between the input and hidden neurons

v_{11},…………...,v_{pm} ,the connection weights between the hidden and output neurons

Y_1 , ……….,Y_m , the output neurons

y_1 ,………,y_m ,the outputs of the network

w_{11} ,..........,w_{nm} , the connection weights between the input and output neuron

Thus, the input to a neuron is the sum of the products of the inputs with their connection weights between the neuron and the inputs. Also, the output value of a neuron is obtained by evaluating the activation function at the input value to the neuron.

Moreover, given a specific application, optimizing the number of hidden layer neurons is important when establishing feedforward neural network to solve problems. Setting too few hidden units causes high training errors and high generalization errors due to under-fitting, while too many hidden units results in low training errors but still high generalization errors due to over-fitting. Several researchers have proposed some rules of thumb for determining an optimal number of hidden units for any application. It is argued that the best number of hidden units depends in a complex way on: the numbers of input and output units, the number of training cases, the amount of noise in the targets, the complexity of the function or classification to be learned, the architecture, the type of hidden unit activation function, the training algorithm, etc. [23].Therefore, it important to take care in selecting neuron numbers since random selection of number of hidden neurons might cause either overfitting or underfitting problems [24].

Neural network is also defined as computational models that are inspired by an animal's central nervous systems, these models are used to calculate or approximate unknown functions. Computational model depends on many inputs, example human brain. A neuron is also called as node, and Neural network is named as Artificial neural network [25].

Neural network uses three types of parameters:

i. The interconnection pattern between the different layers of neurons
ii. The learning process for updating the weights of the interconnections
iii. The activation function that converts a neuron's weighted input to its output activation

Neural networks are most effective and appropriate for pattern recognition and many other real-world problems like signal processing, classification problems. Superior results in pattern recognition can be directly provided in the forecasting, classification and data analysis. To get appropriate results, neural network requires correct data preprocessing, architecture selection and network training but still the performance of a neural network depends on the size of network.

Selection of hidden neurons in neural network is one of the major problems in the field.

The random selections of hidden neurons may cause the problem of either underfitting or overfitting. Overfitting arises because the network matches the data so closely as to lose its generalization ability over the test data. It is an adaptive system that changes its structure or internal information that flows through the network during the training time. If there are so many neurons in the hidden layers it might cause overfitting. This occurs when unnecessary more neurons are present in the network. If the number of neurons is less as compared to the complexity of the problem data, it takes towards the underfitting. It occurs when there are few neurons in the hidden layers to detect the signal in complicated data set [26].

Similarly, in [27] explained that deciding the number of neurons in the hidden layers is a very important part of deciding the overall neural network architecture. Though hidden layers do not directly interact with the external environment, they have a tremendous influence on the final output. Both the number of hidden layers and the number of neurons in each hidden layer must be carefully considered. An extremely large number of neurons in the hidden layers can increase the time it takes to train the network. The amount of training time can increase to the point that it is impossible to adequately train the neural network. Thus, some compromise must be reached between too many and too few neurons in the hidden layers.

There are many rule-of-thumb methods for determining the correct number of neurons to use in the hidden layers, such as the following:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer

2.4.1. Learning Algorithm

A learning algorithm is the procedure used to perform learning process. Its function is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective. Moreover, it is also possible for a neural network to modify its own topology, which is motivated by the fact that neurons in the human brain can die and new synaptic connections can grow [22]. A learning procedure is always an algorithm that can easily be implemented by means of a programing language. Also, a training set is defined as a set of training patterns, which uses to train the neural network.

There are three standards of learning neural network by presenting the difference between the regarding sets.

Unsupervised learning: It provides input patterns to the network, but no learning aides, biologically most plausible method, but is not suitable for all problems. Only the input patterns are given; the network tries to identify similar patterns and to classify them into similar categories.

Reinforcement learning: This learning method provides feedback to the network, whether it behaves well or bad. In reinforcement learning the network receives a logical or a real value after completion of a sequence, which defines whether the result is right or wrong, possibly, how right or wrong it was.

Supervised learning: This method provides training patterns together with appropriate desired outputs. Supervised learning is defined as the training set consists of input patterns with correct results so that the network can receive a precise error vector can be returned. This learning procedure is not always biologically reasonable, but it is extremely effective and therefore very practicable.

Additionally, learning a network can be offline or online. In offline learning a set of training samples is presented, then the weights are changed, the total error is calculated by means of an error function operation or simply accumulated. However, in online learning after every sample presented the weights are changed. Both training procedures have advantages and disadvantages. Offline training procedures are also called batch training procedures since a batch of results is corrected all at once. Such a training section of a whole batch of training samples including the related change in weight values is called epoch. The definition of the offline training is several training patters are entered the network at once, the errors are accumulated, and it learns for all patterns at the same time. In online learning the network learns directly from the error of each training sample [22].

2.4.2. Neural Network Architecture

An optimal neural network architecture may be considered as the one yielding the best performance in terms of error minimization, while retaining a simple and compact structure. There are two important issues concerning the implementation of neural networks, that is, specifying the network architecture (the number of nodes and layers in the network) and finding the optimal values for the connection weights which is done by training algorithm selected [28]. The neural network architecture consists mainly input layer, hidden layer, output layer, neurons and connection weights (lines).The Figure 2.4 below shows NAR Neural Network architecture .The variables are indicated in the figure and helps to create the mathematical relationship between input values and output values once the connection weights and the activation function in the hidden neuron and output neuron is determined .

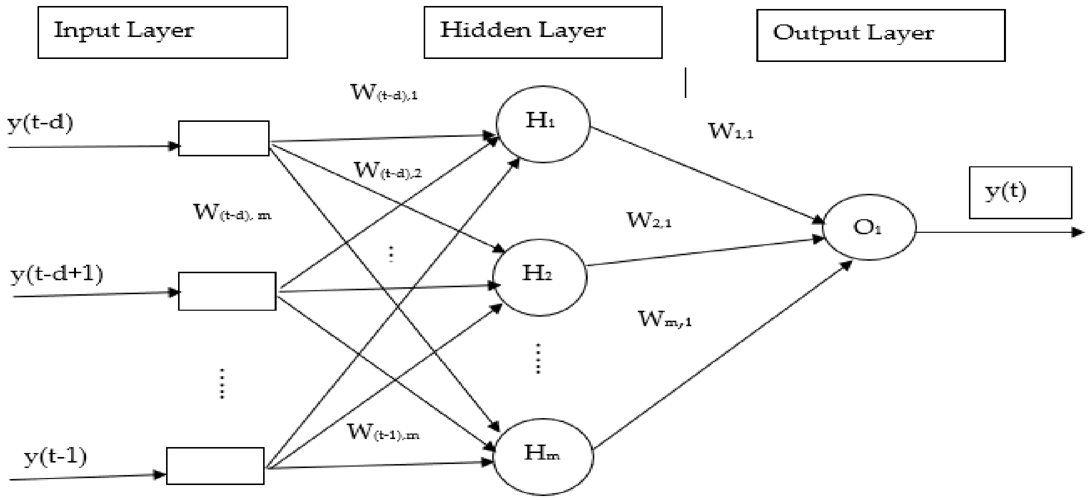

Figure 2.4 NAR Neural Network Architecture Model [22]

The variables in the model are defined as follows:

y(t) is the tth value of the series, which is the predicted value

d is the number of past values of series used as input

y(t-1) ,…..,y(t-d) are input data to the network i.e. sequence of faults occurrence time

$W_{(t-d),1}$,……, $W_{(t-1),m}$ are the connection weights between the inputs and the hidden layer neurons

H_m represents hidden neurons for all positive integers from 1 up to m number of neurons used in hidden layer

O_1 is the output neuron

The basic definition of the terms in the neural network architecture are explained below [20, 22, 25].

Input layer: The neurons in this layer receive the external input signals and perform no computation, but simply transfer the input signals to the neurons in another layer. The input data to the system can be binary, numerical. The input layer is fixed length vector containing user defined data

Hidden layer: The layer of neurons that are connected in between the input layer and the output layer is known as hidden layer. It is used to calculate the weighted sum of the inputs.

Output layer: The neurons in this layer receive signals from neurons in the hidden layer It represents the output containing fixed length vector of data

Neurons: The processing units of the neural network, which processes weighted inputs and produces outputs which might be used as inputs to other nodes

Connection weights: The amount of information about the input that is required to solve a problem is stored in the form of weights

Activation function: Are used to bring about a response that is bounded. There are various activation functions namely the linear function, sigmoid function, step function, bipolar sigmoidal and Gaussian function. In a neural network, each neuron has an activation function which specifies the output of a neuron to a given input.

Mathematically the linear and sigmoid activation functions can be expressed using Equations (2.1) and (2.2) below respectively [21].

(a). Linear function

$$F(x)=x \quad \text{for all } x \ldots\ldots (2.1)$$

(b). Sigmoid function

$$F(x)=\frac{1}{(1+e^{-x})} \quad \text{for all } x \ldots\ldots (2.2)$$

The graphical representation of the sigmoidal activation function is plotted and shown below in Figure 2.5. One of the activation functions commonly used for neurons is the sigmoid function. This function looks like an S hence called as sigmoid function [25].

The function has a value of rage between 0 and 1.

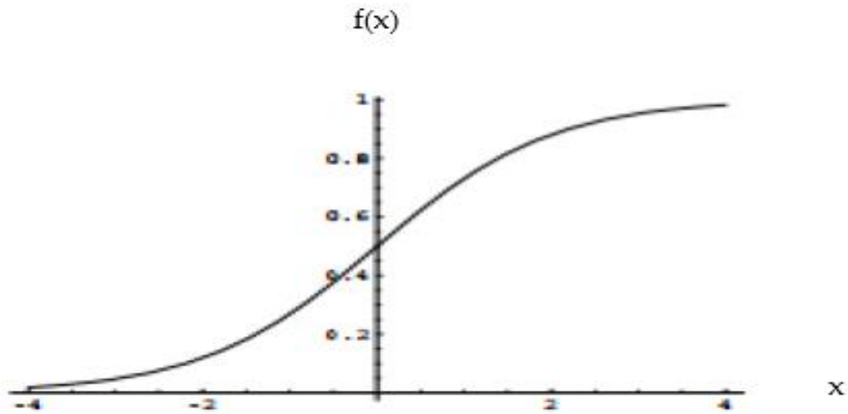

Figure 2.5 Sigmoidal activation function

Each input is weighed before it reaches the main body of the processing element by the connection strength or the weight factor (or simply weight) analogous to the synaptic strength. The amount of information about the input that is required to solve a problem is stored in the form of weights. Each signal is multiplied with an associated weight before it is applied to the summing block. The output of the adder is input value to the activation function and the neuron gives an output after the activation function is evaluated at the output of the adder value.

Moreover, the number of nodes in the hidden layer is of great importance. If the number is too less, the network can't establish complicated judgment boundary; on the contrary, if the number is overabundant, the network will lose the summarizing and judging ability [7]. The output result of the network is dependent on hidden layer result. In a network with a hidden layer and an output layer, the hidden layer is computed first and then the result of the hidden layer are used to compute the output layer. When one uses the neural network to learn the updating of the contributions depends on the steepness slope of the activation function. For adjusting connection weights learning rate is use. It determines how quickly the backpropagation algorithm converges toward a solution [25].

2.4.3. Neural Network Types

There are feedforward and feed backward neural network structures. In feedforward neural network structure, the input signal moves from input to output direction only but in the feed backward neural network or Recurrent neural networks, there is at least one signal input to the network from the output one [26].

2.4.4. Benefits of Neural Networks

Neural networks are most effective and appropriate for pattern recognition, predictions, and many other real-world problems. A few areas where neural networks are currently applied are mentioned below [26].

1. Classification: It is the assignment of each object to specific class, which is an important aspect in image classification for example recognition of printed or handwritten characters.
2. Signal processing: In digital communication systems, distorted signals cause inter-signal interference, one of the first commercial applications of neural networks was to suppress noise cancellation and it was implemented to remove the noise from the telephone line signal.
3. Speech recognition: In recent years, speech recognition has received enormous attention and involves three modules namely the front end which samples the speech signals and extracts the data, the word processor which is used for finding the probability of words in the vocabulary that match the features of the spoken words and the sentence processor which determines if the recognized word makes sense in the sentence.
4. Medicine: An auto associative network is developed to store large amount of medical records, each of which includes information on the symptoms, diagnosis

and treatment of a specific case. When trained network is presented with a set of symptoms, the network finds the best diagnosis and treatment.

5. Intelligent control: A problem of concern in industrial motor control is the ability to predict system failure. Neural networks have been used in many vehicular applications including trains and automatic gear transmission in cars.
6. Financial forecasting: One of the most desirable applications of neural networks have been in financial applications. The idea if one dealer can predict market values or assess risk better than his competitors even by a very small amount, leads to an expectation of huge financial benefit. Much of the work in this area is based on the time series prediction.
7. Function approximation: Many computational models can be described as functions mapping the input vectors to numerical outputs. Neural networks are employed to construct approximately the same output for the given input vector based on the available training data.
8. Condition monitoring: Neural networks can also be used in predicting the occurrence of malfunction of item
9. Process monitoring and control: The techniques applied in processes such as chemical and biochemical, pharmaceutical, water purification, food and beverage production, power distribution
10. Pattern analysis: Realtime audit of bandwidth systems, images and data management can also be tackled with the help of neural networks.

2.5. Data Mining

It is common to extract data from databases where the huge data is available. For example, fault management database systems contain fault data records. To make use of the relevant information from data some techniques are required. Data mining is defined as knowledge discovery from data (KDD). Knowledge discovery from data process contains the following activities and defined shortly to understand the terms [29].

Data cleaning: Noise and irrelevant data are removed

Data integration: Multiple data sources are combined in a common source

Data selection: Where data relevant to the analysis task are retrieved from the database

Data transformation: Where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations

Data mining: An essential process where intelligent methods are applied to extract data patterns

Pattern evaluation: To identify interesting patterns representing knowledge based on interestingness measures

Knowledge representation: Where visualization and knowledge representation techniques are used to present mined knowledge to users

Most of data mining activities are performed when the fault data is extracted from system.

3. Fault Prediction Methodology

The prediction method selected in this research work is Nonlinear Autoregressive (NAR) neural network time series. The problem is assumed to be a nonlinear because nonlinear models are appropriate for predicting variable changes in fault time series. The feature values of the time series are predicted from past values of that series. This form of prediction is known as Nonlinear Autoregressive and mathematically written as follows in Equation (3.1) below [30].

$$y(t)=f(y(t-1),y(t-1),\ldots\ldots y(t-d),W, B) \qquad (3.1)$$

where y(t) is a series and d is number of past values of y(t), f is a function which is relating future value nonlinearly with past values, connection weights(w) & bias(B) of neurons.

3.1 System Model

The NAR neural network time series prediction method system model is shown in block diagram in Figure 3.1 below. It consists the input, the hidden and the output layer of the network with main parameters required in each function such as time series fault input data, neurons, algorithms, activation functions and target output.

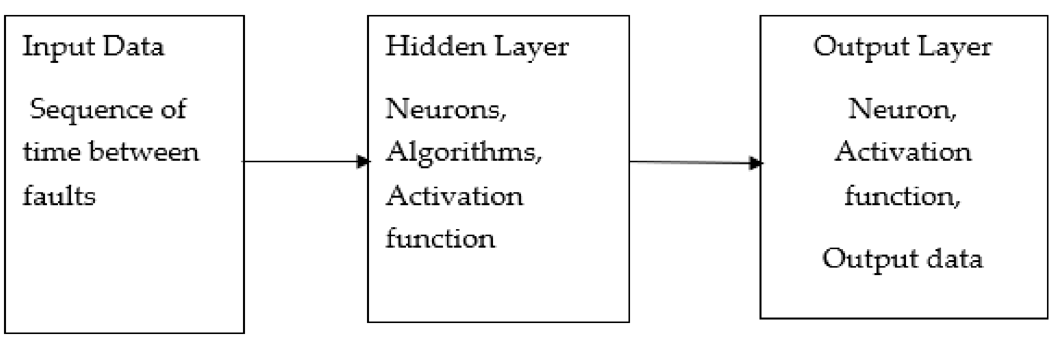

Figure 3.1 System model block diagram [22]

To develop the neural network model the parameters to set are number of hidden neurons, number of training pairs, the training algorithm and the neural network architecture to be used [22,24].

Thus, model selected is a feedforward neural network type because the input signal moves forward to output of the network and NAR Neural Network architecture is same to Figure 2.4.

3.2 Neural Network Tool

The Neural Net Time Series Tool utilizes sequential inter-arrival times to construct a network. It utilizes a two-layer feed-forward network. It has "hidden" layers between input and output layers and requires an activation function, which is usually non-linear, and in this case, is a sigmoid function. In the output layer the activation function is a linear function. Furthermore, Neural Net Time Series Tool is utilized to create a NAR Network. It is more suitable to use NAR for this data because the desired predictions depend on the past values. NAR predicts fault inter-arrival times, using past values of the fault inter-arrival times. In addition to varying neurons and algorithms, feedback delays can also vary to achieve best results. The network is a two-layer feed-forward network. The learning algorithm is Levenberg-Marquardt (LM). This algorithm typically requires memory but less time and efficient in prediction [30].

3.3 Data Collection

Fault history data of Addis Ababa 3G mobile sites are extracted from the database of Element Management System (EMS), where the fault data from communication networks is collected and aggregated at a network operations center. The amount of data collected is for five months length and 15,950 in number. It is categorized in weekly, monthly and 5-months basis, Table 3.1 shows the number of fault data collected to train network. The data has the fault occurrence date and time format.

Table 3.1 Number of Fault data collected

Data Category	Date coverage in month	Feb-18	Mar-18	Apr-18	May-18	Jun-18
Data 1(Week 1)	1-7	234	807	450	696	1060
Data 2(Week 2)	8-14	1630	778	438	921	634
Data 3(Week 3)	15-21	629	862	469	699	790
Data 4(Week 4)	22-28	822	797	779	506	857
Data 5(Week 5)	29-31	0	378	299	226	188
Monthly total data		3315	3622	2435	3048	3528
Five-month total data	15,950					

3.4. Data Preprocessing

Before the raw time series data is applied to the prediction methods, usually it undergoes several transformations. Proper data processing significantly affects the prediction quality. Neural network methods have strict requirements for the format of input data. After fault history of mobile sites data is extracted from the server of EMS, the fault occurrence time of critical 3G mobile sites faults are extracted and arranged in chronological order. The fault inter-arrival time duration which means the time difference between consecutive faults, is calculated in a sequence. This becomes the input data to the network. The input data is prepared in numerical form in hours unit for this purpose, but it can be converted to any time unit like days.

3.5. Hidden Layer Neuron Number Selection

Deciding the number of neurons in the hidden layer is a very important part of deciding the overall network architecture. There are different approaches to select the number of neurons in the hidden layer. These are trial, rule of thumb and the third one is the recommended formulas by different researchers. In the process of specifying the network size, an insufficient number of hidden nodes causes difficulties in learning data whereas an excessive number of hidden neurons might lead to unnecessary training time with marginal improvement in training outcome as well as make the estimation for a suitable set of interconnection weights more difficult [26].

A higher number of neurons in hidden layer tend the network to memorize, instead of learning and generalization, and it might lead to the problem of local minima. On the other hand, increasing the hidden nodes will help to adjust to larger fluctuation of target function and allow the model to consider the presence of volatilities in the data.

In this research an iterative approach of hidden neuron selection, from the smallest to highest iteration is followed by observing the MSE performance when the neuron number increases.

3.6. Training the network

The dataset is divided among the training, validation and testing with different percentages. The training data are presented to the network during training, and the network is adjusted according to its error. The validation data are used to measure network generalization, and to halt training when generalization stops improving. The testing data has no effect on training and so provide an independent measure of network performance during and after training. The target timesteps dataset defining the desired output y(t) is loaded to the network. The dataset is randomly divided among the three in the ratio of 70% for training ,15% for validation and 15% for testing the model.

To create the architecture of a nonlinear autoregressive neural network, the number of neurons and feedback delay parameters are configured. Figure 3.2 shows an illustration of the neural network architecture diagram created as a sample for 10 neurons in the hidden layer. It can be shown that the input, the weight (W), bias(B), the hidden and output layers, the activation functions in the hidden and the output layers, the neurons used in the hidden layer and the feedback delay number, which is 1 in this case. This network is a two-layer neural network architecture.

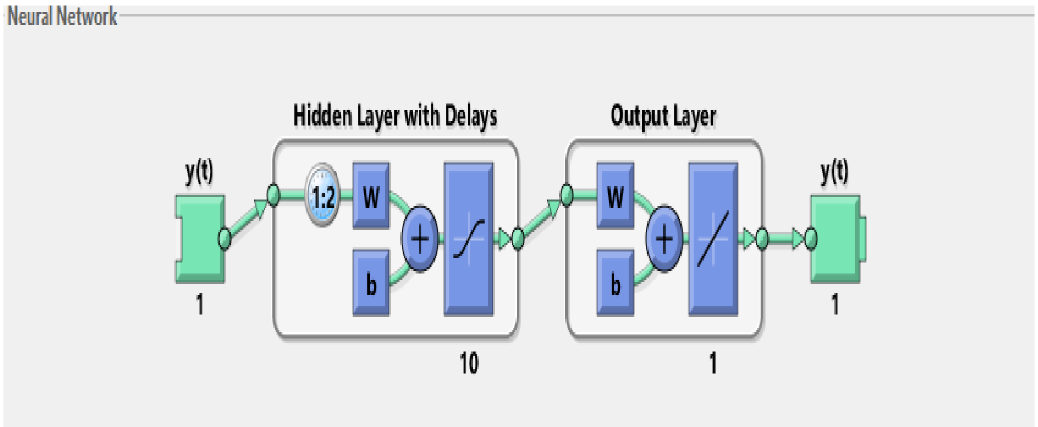

Figure 3.2 Neural Network Architecture

To train the network Levenberg- Marquardt (LM) learning algorithm is selected. This is algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples [28]. Figure 3.3 shows an illustration of neural network training performance plot diagram. It is MSE vs epoch number. The MSE indicates the minimum error value and the epoch indicates the number of iterations to train the whole complete data of the training. The training performance of the models obtained for training, validation and test data are available with MSE, correlation factor (R value) and the approximate linear regression function for all models are presented in graphics. The error performance tells how the predicted(output) value is approaches the target (actual) value.

From the graph below, it is possible to observe the best model validation performance is achieved at MSE value of 1.5253 at epoch 1 at the point marked by circle and hidden line.

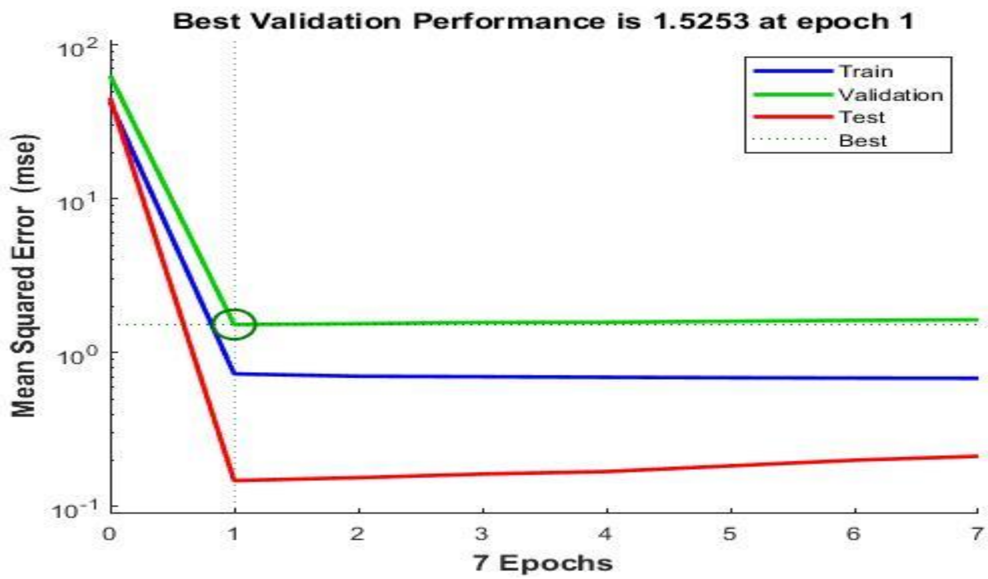

Figure 3.3 Neural Network Training performance graph sample

The training stops after the validation performance MSE starts to increase. Thus, the test model is selected for future data or new fault data predictions by taking best results in MSE and R values. Besides, the performance report summary for all models is indicated in Figure 3.4.

Results

	Target Values	MSE	R
Training:	354	7.29123e-1	2.61625e-1
Validation:	76	1.52527e-0	3.06994e-2
Testing:	76	1.47795e-1	3.46031e-1

Figure 3.4 Neural network model training performance result

For example, this result shows training data number is 354, training model performance is MSE of 0.729123 & R value of 0.261625, where e-2=0.01 and e-1=0.1. In similar way, for validation and testing performance can be observed.

4. Network Training and Result Evaluation

4.1 Neural Network Training

The neural network is trained using training, validation and testing data. The models obtained using testing data on weekly, monthly and Five-month data is summarized and explained in detail in next parts of this section.

4.1.1 Training 1- Using weekly data

The neural network is trained using weekly data category and the results with better mean square error (MSE) and correlation factor known as regression(R) values are summarized in Table 4.1 below. During the network training the proportion of dataset among training, validation and test are 70%,15% and 15% respectively. The best values of test MSE and R values are selected after doing iterative tests with different number of neurons for each week data category. In last column of Table 4.1, all the best approximate linear regressive mathematical models which relate the Output (Predicted) and the Target (Actual) values are presented.

From the linear regression approximation, the mathematical models are approximated in leaner function. Thus, the predicted and the actual values are related with a linear equation. Equation (4.1) below shows the leaner relationship between the two variables. Thus, the formula uses to find the output (predicted) value, given the new target (actual) value of fault occurrence time.

$$\text{Output}\sim = \propto \times \text{Target} + \beta \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (4.1)$$

where "∝" and "β" are constant value parameters related to the developed model. Which are obtained from the leaner approximation of the output and the target values.

Table 4.1 Weekly data training result

Month Name	Data category	Number of data	Training data (70%)	Validation data (15%)	Test data (15%)	Number of Neuron	Training MSE	Validation MSE	Test MSE	Training R	Validation R	Test R	Mathematical Model for test data
Feb-2018	Week 1	235	165	35	35	5	0.66926	0.783246	0.638591	0.461432	0.514865	0.559377	Output~= 0.32*Target+0.48
	Week 2	1630	1140	245	245	10	0.15215	0.150258	0.075022	0.509981	0.269135	0.202721	Output~= 0.14*Target+0.1
	Week 3	629	441	94	94	10	0.22216	0.317097	0.137658	0.419783	0.481949	0.385475	Output~= 0.47*Target+0.21
	Week 4	822	576	123	123	5	0.12707	0.143067	0.0694	0.399995	0.302053	0.500202	Output~= 0.19*Target+0.13
Mar-2018	Week 1	807	565	121	121	5	0.24919	0.237808	0.12341	0.176719	0.131328	0.003469	Output~= 0.0022*Target+0.19
	Week 2	778	544	117	117	20	0.2099	0.282448	0.116138	0.359397	0.0802181	0.246492	Output~= 0.077*Target+0.27
	Week 3	863	605	129	129	25	0.18517	0.365318	0.114401	0.234595	0.147588	0.054607	Output~= 0.042*Target+0.13
	Week 4	797	557	120	120	10	0.2378	0.287618	0.107107	0.361576	-0.0812256	0.443991	Output~= 0.49*Target+0.12
	Week 5	378	264	57	57	15	0.18983	0.193299	0.041157	0.147649	0.428163	0.405983	Output~= 0.18*Target+0.17
Apr-2018	Week 1	450	314	68	68	10	0.5686	0.192217	0.274158	0.36991	0.728001	0.482412	Output~= 0.37*Target+0.24
	Week 2	438	306	66	66	15	0.98291	0.123184	0.41199	0.236897	0.294042	0.570455	Output~= 0.34*Target+0.079
	Week 3	469	329	70	70	10	0.68555	0.262889	0.147436	0.423566	0.386526	0.01699	Output~= 0.0077*Target+0.36
	Week 4	779	545	117	117	10	0.16633	0.141426	0.154157	0.623033	0.43386	0.671524	Output~= 0.46*Target+0.11
	Week 5	299	209	45	45	5	0.12471	0.0714916	0.109196	0.442138	0.196863	0.275266	Output~= 0.38*Target+0.065
May-2018	Week 1	696	488	104	104	5	0.29985	0.118812	0.093706	0.210777	0.477075	0.324873	Output~= 0.12*Target+0.23
	Week 2	921	645	138	138	5	0.12193	0.1262	0.058383	0.441567	0.2331	0.308878	Output~= 0.21*Target+0.17
	Week 3	699	489	105	105	10	0.16477	0.0571531	0.082371	0.351558	0.528771	0.308843	Output~= 0.16*Target+0.2
	Week 4	506	354	76	76	10	0.72912	1.52527	0.147795	0.261625	0.0306994	0.346031	Output~= 0.14*Target+0.3
	Week 5	226	158	34	34	10	0.19723	1.07424	0.142108	0.528012	0.149835	0.710497	Output~= 0.25*Target+0.33
Jun-2018	Week 1	1060	742	159	159	5	0.71826	0.175681	0.056409	0.422489	-0.0098099	0.626773	Output~= 0.66*Target+0.083
	Week 2	634	444	95	95	20	0.23865	0.113384	0.251064	0.445099	0.534676	0.367435	Output~= 0.098*Target+0.23
	Week 3	790	552	119	119	5	0.28374	0.136444	0.090275	0.310903	0.179759	0.375791	Output~= 0.2*Target+0.14
	Week 4	857	599	129	129	20	0.29716	0.684626	0.089869	0.386358	0.010878	0.094703	Output~= 0.022*Target+0.34
	Week 5	188	132	28	28	5	0.53421	0.106537	0.964423	-0.477048	0.00869565	0.111796	Output~= 0.0028*Target+0.25

In last column of Table 4.1, all the best approximate linear regressive mathematical models which relate the Output (Predicted) and the Target (Actual) values are presented.

4.1.2 Training 2 - Using monthly data

The neural network is trained using monthly base data category and the mathematical model obtained with better mean square error (MSE) and correlation factor known as regression(R) value are summarized in Table 4.2 below.

Table 4.2 Monthly data training result

Month Name	Number of data	Training data (70%)	Validation data (15%)	Test data (15%)	Number of Neuron	Training MSE	Validation MSE	Test MSE	Training R	Validation R	Test R	Mathematical Model for test data
Feb-2018	3316	2322	497	497	5	0.19985	0.264523	0.191846	0.42531	0.285166	0.38146	Output~= 0.15*Target+0.16
Mar-2018	3622	2536	543	543	5	0.22952	0.201477	0.077955	0.26551	0.153604	0.31816	Output~= 0.13*Target+0.19
Apr-2018	2435	1705	365	365	5	0.39899	0.718494	0.37269	0.37195	0.200532	0.2875	Output~= 0.096*Target+0.25
May-2018	3048	2134	457	457	5	0.2753	0.190021	0.233043	0.29733	0.302676	0.32724	Output~= 0.095*Target+0.22
June-2020	3528	2470	529	529	10	0.20065	0.202537	0.147583	0.30027	0.356137	0.2998	Output~= 0.097*Target+0.17

In last column of Table 4.2, all the best approximate linear regressive mathematical models which relate the Output (Predicted) and the Target (Actual) values are presented.

4.1.3 Training 3 - Using Five-month data

The neural network is trained using all Five-month data and the results with better mean square error (MSE) and correlation factor known as regression value(R) for different neuron number are summarized in Table 4.3 below. In last column of Table 4.3, all the approximate linear regressive mathematical models which relate the Output (Predicted) and the Target (Actual) values are presented for neurons from 1 to 30 iteratively is shown. All parameters used, and performance result are indicated in all result tables.

Table 4.3 Five-month data training result

Data Category	Test No.	Data Number	Training data (70%)	Validation data (15%)	Test data (15%)	Number of Neuron	Training MSE	Validation MSE	Test MSE	Training R	Validation R	Test R	Test Relationship
	1	15,948	11164	2392	2392	1	0.27935	0.199174	0.190973	0.316314	0.307535	0.31556	Output~= 0.11*Target+0.17
	2	15,948	11164	2392	2392	2	0.24602	0.240163	0.300636	0.310644	0.335423	0.31643	Output~= 0.085*Target+0.19
	3	15,948	11164	2392	2392	3	0.24576	0.23848	0.303703	0.327241	0.301556	0.271	Output~= 0.083*Target+0.20
	4	15,948	11164	2392	2392	4	0.25074	0.243975	0.289554	0.302053	0.336637	0.30604	Output~= 0.1*Target+0.17
	5	15,948	11164	2392	2392	5	0.23252	0.228773	0.375369	0.321814	0.362282	0.25413	Output~= 0.064*Target+0.21
	6	15,948	11164	2392	2392	6	0.25302	0.25907	0.246561	0.301152	0.337982	0.36811	Output~= 0.11*Target+0.2
	7	15,948	11164	2392	2392	7	0.25112	0.257403	0.253716	0.328177	0.308078	0.28857	Output~= 0.095*Target+0.2
	8	15,948	11164	2392	2392	8	0.25717	0.216164	0.275998	0.324657	0.302575	0.27808	Output~= 0.11*Target+0.2
	9	15,948	11164	2392	2392	9	0.25153	0.238464	0.273366	0.317953	0.281159	0.34546	Output~= 0.1*Target+0.2
	10	15,948	11164	2392	2392	10	0.25307	0.236449	0.265755	0.320589	0.359334	0.27575	Output~= 0.092*Target+0.21
	11	15,948	11164	2392	2392	11	0.24646	0.31526	0.221625	0.329897	0.263823	0.31097	Output~= 0.11*Target+0.2
	12	15,948	11164	2392	2392	12	0.24889	0.323413	0.209566	0.306878	0.312789	0.34103	Output~= 0.1*Target+0.2
	13	15,948	11164	2392	2392	13	0.25261	0.304338	0.210048	0.314896	0.285464	0.32661	Output~= 0.11*Target+0.21
	14	15,948	11164	2392	2392	14	0.25188	0.281954	0.232606	0.309499	0.332887	0.31238	Output~= 0.099*Target+0.21
5- Month	15	15,948	11164	2392	2392	15	0.26422	0.181952	0.274515	0.300613	0.335743	0.35702	Output~= 0.11*Target+0.2
	16	15,948	11164	2392	2392	16	0.23329	0.290847	0.308928	0.349464	0.26751	0.23141	Output~= 0.082*Target+0.21
	17	15,948	11164	2392	2392	17	0.24919	0.233015	0.2826	0.32819	0.381633	0.24297	Output~= 0.08*Target+0.2
	18	15,948	11164	2392	2392	18	0.26154	0.315349	0.251984	0.329268	0.189256	0.24046	Output~= 0.15*Target+0.2
	19	15,948	11164	2392	2392	19	0.25396	0.285762	0.202445	0.334443	0.350464	0.27322	Output~= 0.11*Target+0.2
	20	15,948	11164	2392	2392	20	0.25115	0.250377	0.268537	0.326209	0.270963	0.30613	Output~= 0.1*Target+0.22
	21	15,948	11164	2392	2392	21	0.26396	0.275648	0.212402	0.296504	0.318831	0.25331	Output~= 0.099*Target+0.22
	22	15,948	11164	2392	2392	22	0.22015	0.383585	0.325347	0.318586	0.299773	0.19699	Output~= 0.097*Target+0.21
	23	15,948	11164	2392	2392	23	0.25226	0.283667	0.223809	0.328959	0.303058	0.28095	Output~= 0.093*Target+0.21
	24	15,948	11164	2392	2392	24	0.24585	0.210018	0.326509	0.332557	0.277251	0.29718	Output~= 0.086*Target+0.2
	25	15,948	11164	2392	2392	25	0.235	0.289204	0.297475	0.321116	0.325206	0.31306	Output~= 0.09*Target+0.22
	26	15,948	11164	2392	2392	26	0.23837	0.310136	0.24768	0.345506	0.260866	0.33273	Output~= 0.11*Target+0.2
	27	15,948	11164	2392	2392	27	0.25727	0.270078	0.240974	0.322388	0.214243	0.28567	Output~= 0.1*Target+0.2
	28	15,948	11164	2392	2392	28	0.24706	0.188428	0.344381	0.322367	0.293256	0.32706	Output~= 0.091*Target+0.22
	29	15,948	11164	2392	2392	29	0.22683	0.370656	0.262421	0.329688	0.255052	0.32531	Output~= 0.091*Target+0.23
	30	15,948	11164	2392	2392	30	0.24962	0.271419	0.274035	0.293584	0.326659	0.30769	Output~= 0.1*Target+0.2

4.2. Result Evaluation

The comparison between predicted and actual values of a network fault is performed for weekly, monthly and 5-month data using the selected models. The analysis is done based on the training data category. Using the linear mathematical model in Equation (4.1), the predicted value is calculated for the different models by using actual values from the existing fault data. Also, the formula applied to compare the output (predicted) value and the target (actual) value is defined in the Equation (4.2) below.

$$\text{Performance}(\%)=\frac{\text{Predicted value}}{\text{Actual value}}*100\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.2)$$

By applying the above equation weekly prediction performance is indicated in Table 4.4 below.

Table 4.4 Weekly prediction performance

Mathematical Model	Neuron Number	Target (Actual) [hr]	Output(Predicted) [hr]	Performance (%)	Error (Actual-Predicted)[hr]
Output~= 0.19*Target+0.13	5	0.1258	0.1539	122.3386	-0.028102
Output~=0.0022*Target+0.19	1	0.1589	0.1903	119.7921	-0.03144958
Output~= 0.49*Target+0.12	10	0.2256	0.2305	102.1915	-0.004944
Output~= 0.18*Target+0.17	15	0.2	0.2060	103.0000	-0.006
Output~= 0.12*Target+0.23	5	0.2922	0.2651	90.7132	0.027136
Output~= 0.66*Target+0.083	5	1.6558	1.1758	71.0127	0.479972
Output~= 0.098*Target+0.23	20	0.4744	0.2765	58.2823	0.1979088
Output~= 0.022*Target+0.34	20	0.7086	0.3556	50.1819	0.3530108

The result obtained when the best mathematical models are evaluated with an actual data which was not used during the process of training the model are shown in performance column of Table 4.4. The best achievement is 90.71% with 5 hidden neuron number. The graph in Figure 4.1 below shows the neural network model training performance. It can be observed that the best model validation performance is achieved at MSE value of 0.11881 at epoch 30 at the point marked by circle and hidden line.

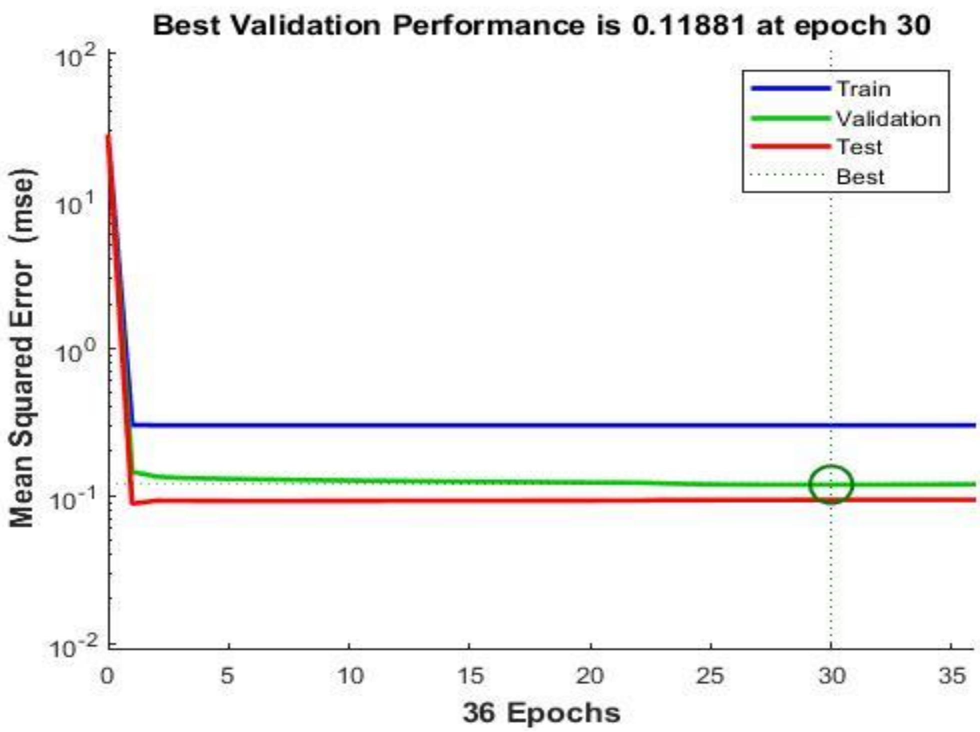

Figure 4.1 Neural network training performance achieved

The training stops after the validation performance MSE starts to increase or be constant. Thus, the test model is selected for future data or new fault data predictions by taking best results in MSE and R values. Besides, the performance report of the network for all models is indicated in Figure 4.1.

Results

	Target Values	MSE	R
Training:	488	2.99854e-1	2.10777e-1
Validation:	104	1.18812e-1	4.77075e-1
Testing:	104	9.37063e-2	3.24873e-1

Figure 4.2 Neural network model training performance result

The performance results obtained are training with MSE of 0.299854 & R value of 0.210777, validation with MSE of 0.118812 and R value of .477075, and Testing with MSE 0.0937063 and R value of 0.324873.

The monthly prediction performance detail is indicated in the Table 4.5 below. It can be observed the performance is a little higher than 100% which means the predicted value is bigger than the actual value. However, the performance value with 108 % is in the error of less than 10%.

Table 4.5 Monthly prediction Performance

Mathematical Model	Neuron Number	Target (Actual) [hr]	Output (Predicted)[hr]	Performance (%)	Error (Actual-Predicted)[hr]
Output~= 0.13*Target+0.19	5	0.2000	0.2160	108.00	-0.016
Output~= 0.097*Target+0.17	10	0.1478	0.1843	124.72	-0.0365366

The Five-month prediction performance detail is indicated in the Table 4.6 below. It is observed that performance is a 125% which means the predicted value is bigger than the actual value with a 25% error.

Table 4.6 Five-month prediction performance

Mathematical Model	Neuron Number	Target (Actual) [hr]	Output (Predicted)[hr]	Performance (%)	Error (Actual-Predicted)[hr]
Output~= 0.11*Target+0.17	1	0.1478	0.186258	126.02	-0.038458
Output~= 0.085*Target+0.19	2	0.1478	0.202563	137.05	-0.054763
Output~= 0.1*Target+0.17	4	0.1478	0.18478	125.02	-0.03698
Output~= 0.086*Target+0.2	24	0.1478	0.186258	126.02	-0.038458

Table 4.7 below indicates the summary of the linear mathematical models obtained with error of less than 50%. After models are evaluated by actual data, the predicted value obtained and the relative percentage of the predicted over the actual value are indicated.

Table 4.7 Summary of all models and prediction performance

Mathematical Model	Neuron Number	Target (Actual) [hr]	Output(Predicted) [hr]	Performance (%)	Error (Actual-Predicted)[hr]
Output~= 0.022*Target+0.34	20	0.7086	0.3556	50.18	0.3530108
Output~= 0.098*Target+0.23	20	0.4744	0.2765	58.28	0.1979088
Output~= 0.66*Target+0.083	5	1.6558	1.1758	71.01	0.479972
Output~= 0.12*Target+0.23	5	0.2922	0.2651	90.71	0.027136
Output~= 0.49*Target+0.12	10	0.2256	0.2305	102.19	-0.004944
Output~= 0.18*Target+0.17	15	0.2	0.2060	103.00	-0.006
Output~= 0.13*Target+0.19	5	0.2000	0.2160	108.00	-0.016
Output~=0.0022*Target+0.19	1	0.1589	0.1903	119.79	-0.03144958
Output~= 0.19*Target+0.13	5	0.1258	0.1539	122.34	-0.028102
Output~= 0.097*Target+0.17	10	0.1478	0.1843	124.72	-0.0365366
Output~= 0.1*Target+0.17	4	0.1478	0.18478	125.02	-0.03698
Output~= 0.11*Target+0.17	1	0.1478	0.186258	126.02	-0.038458
Output~= 0.086*Target+0.2	24	0.1478	0.186258	126.02	-0.038458
Output~= 0.085*Target+0.19	2	0.1478	0.202563	137.05	-0.054763

It can be concluded that the best achievement is 90.71% success. The mathematical model obtained using 5 neurons with MSE of 0.093706 and R value 0.325873 is formulated in Equation (3.6) below.

$$\text{Output} = 0.12\times\text{Target} + 0.23 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4.3)}$$

Moreover, Table 4.8 below shows the selected models with 10 % range error between the predicted and the actual value.

Table 4.8 List of best four models

Mathematical Model	Neuron Number	Target (Actual) [hr]	Output(Predicted) [hr]	Performance (%)	Error (Actual-Predicted)[hr]
Output~= 0.12*Target+0.23	5	0.2922	0.2651	90.71	0.027136
Output~= 0.49*Target+0.12	10	0.2256	0.2305	102.19	-0.004944
Output~= 0.18*Target+0.17	15	0.2	0.2060	103.00	-0.006
Output~= 0.13*Target+0.19	5	0.2000	0.2160	108.00	-0.016

Generally, Equation (4.3) can be taken as the best achieved linear mathematical model approximation to predict fault occurrence time of a 3G mobile sites. The mathematical model has a unit of hours, so it is better to convert the actual fault time in hours unit. The prediction performance result achieved when evaluated using Equation (4.2) is 90.71%.

5. Findings and Discussion

5.1. Findings

After the neural network model is trained using the weekly, monthly and 5-month dataset with different neuron numbers, the mathematical models which are relating the predictive(output) and the actual (target) are selected with lower MSE and higher correlation value from the testing data models. The prediction performance with new actual data is calculated using the mathematical models obtained. The following are the main findings from the training, testing the model and evaluation the result

- The performance of predicted over the target obtained by taking -29 to -50 % range error from the actual are 50.18%,58.28%,71.01%.
- The performance percentage of predicted over the target obtained by taking + or - 10% range error from the actual are 90.71%,102.19%,103.00% and 108.00%.
- The best prediction value obtained is 90.71% of the actual value
- The five-month prediction performance is compared using all mathematical models obtained by different neuron size
- The weekly and monthly performance are more related than the 5-month performance
- Generally, the predicted time must be smaller than the actual fault time
- By varying neuron size, the model performance is different and compared all models
- The most success was achieved with 5 neuron numbers, and training with Levenberg-Marquardt, the dataset is divided randomly 70% for training,15% for

validation and 15% for testing the model as independent data which means it was not involved in the model training.

5.2. Discussion

The prediction performance achieved by evaluating the mathematical models have significance in the estimation of fault occurrence time of the mobile sites. The performance values 50.18%,58.28%,71.01% obtained by taking -29 to -50 % range error from the actual can be considered as medium achievement of the model. This can save 50% to 70% time before fault occurs. It can averagely improve quality and revenue if the proactive maintenance is performed with in that time ahead.

The performance percentage of predicted over the target obtained by taking + or -10% range error from the actual are 90.71%,102.19%,103.00% and 108.00%. These values have different interpretations. The 90.71% is taken as it can save 90.71% time before fault happens and it is approaching to the actual fault time. This prediction value can bring a significant advantage in managing the proactive maintenance. The other results are above 100% but they are near with 10% deviation which have an interpretation of predicted value is above the actual fault occurrence time. Therefore, this is not considered as an advantage because the predictive values must be equal or less than the actual fault time values.

The best prediction value obtained is 90.71% of the actual value which is highest achievement of the model. This implies that the operators can know the fault occurrence time ahead of 90.71 % to actual fault occurrence time. Therefore, the linear mathematical model obtained is useful to implement the proactive maintenance approach by preparing inspection summary list of all fault reasons of mobile sites. Also, the new fault data can be adapted to the existing data which were used to develop the model so that to develop new model for future. The results of this work are validated by using the actual fault data occurrence time to test the performance of prediction.

6. Conclusion and Future Work

6.1. Conclusion

The trend of telecommunication network advancement in mobile network is making the life of customers easy. Currently telecommunication network interruption is affecting individual customers business and life. Even in the coming generations of cellular mobile communications, the services are very sensitive, and the impact of network faults will be higher unless the proactive maintenance approach is applied instead of following the corrective approach. From this research work, it can be concluded that it is possible to predict the fault occurrence time using historical fault data of mobile sites.

The Nonlinear Autoregressive neural network time series prediction method has achieved a prediction success of 90.71%. The interpretation is that it is possible to manage 90.71% time ahead before fault happens; that means, it helps to estimate that fault is going to occur in certain amount of time. Additionally, the best performing model is achieved by training the neural network using the weekly data categories in which commonly preventive works are scheduled.

Thus, by preventing the fault it is helpful to improve network reliability and availability performance, to reduce maintenance cost, to enhance revenue and improve customer satisfaction. Fault occurrence time prediction is an important technique to implement proactive maintenance approach. Besides, fault prediction helps to schedule resources because it is possible to estimate the next fault occurrence time of sites. Also, it is possible to enhance the availability performance of 3G mobile sites which are currently dominant in number and coverage in Addis Ababa. The work here will not be limited to an existing data but will be made adaptable to include future data.

6.2. Future work

Most fault prediction works are not based on real telecommunication fault data and can be done more on different networks. The prediction efficiency of different techniques can be compared by using telecommunication network fault data. The same method can be applied for each mobile site fault occurrence prediction because it increases the similarity of data. The per site data can be used to localize faults once the predicted fault occurrence time is near. This means the prediction can be more specific to a single mobile site. Also, determination of optimal hidden layer neuron numbers can be a future work.

References

[1]. D.Medved, K.Brlas, D.Saric," Fault Analysis and Prediction in Telecommunication Access Network", IEEE conference paper, Croatia, 2000.

[2]. T.Matsukawa, H.Funakoshi, "Analyzing failure frequency and severity in communication networks", USA,IEEE 2010 Proceedings - Annual Reliability and Maintainability Symposium (RAMS) .

[3]. M. L. Ayers. *Telecommunications System Reliability Engineering, Theory and Practice*. New Jersey, John Wiley & Sons, Inc, 2012. pp. 7-61.

[4]. Y. Kumar, H. Farooq, A. Imran, "Fault Prediction and Reliability Analysis in a Real Cellular Network", 2017 IEEE Wireless Communications and Mobile Computing Conference (IWCMC).

[5]. F.Salfner, M.Schieschke, M.Malek," Predicting Failures of Computer Systems: A Case Study for a Telecommunication System", Berlin, Germany,2011.

[6]. A. Birolini. *Reliability Engineering Theory and Practice*. Berlin Heidelberg, Springer, 2007.

[7]. Wei Wu, Feng Zhang, Min Liu and Weiming Shen "A Multi-agent Based Failure Prediction Method Using Neural Network Algorithm," 2014 IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA.,2014

[8]. H. Farooq, M. S. Parwez and A. Imran, "Continuous Time Markov Chain Based Reliability Analysis for Future Cellular Networks," 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1-6, San Diego, CA, 2015.

[9]. ethiotelecom.et, 'Ethio Telecom Ranked First in Africa', 2017. [Online]. Available: http://www.ethiotelecom.et/press release [Accessed: 23- Dec- 2017].

[10]. ethiotelecom.et, 'Ethiopia - Telecoms, Mobile and Broadband - Statistics and Analyses', 2017. [Online]. Available: http://www.ethiotelecom.et/ [Accessed: 22- Dec- 2017].

[11]. Ethio telecom network performance reports,2017

[12]. Ethio Telecom, "Telecom expansion project tender no: et/F/RT/34/2011", Sep 2011.

[13]. F. Kwabena, J. Quarshie, K. Osei, "Statistical Analysis of Outage Time of Commercial Telecommunication Networks in Ghana", 2013 IEEE

[14]. J. Zhao, L.Xu, L.Liu,'' Equipment Fault Forecasting Based on ARMA Model", IEEE International Conference on Mechatronics and Automation, Harbin, China, August 5 - 8, 2007.

[15]. M.Jaudet, N.Iqbal, A.Hussain, "Neural Networks for Fault-prediction in a Telecommunications Network", IEEE, Scotland, UK, 2004

[16]. A.Abu-Samah, M.K.Shahzad, E.Zamai, A.Ben Said, "Failure Prediction Methodology for Improved Proactive Maintenance using Bayesian Approach", International Federation of Automatic Control(IFAC) conference paper, France, 2015.

[17]. G. Panchal1, A.Ganatra, Y. Kosta and D.Panchal4," Behaviour Analysis of Multilayer Perceptron with Multiple Hidden Neurons and Hidden Layers", IEEE,2012.

[18]. ITU-T Study Group 2 (2005-2008), "ITU-T Recommendation E.800", Quality of telecommunication services: concepts, models, objectives and dependability planning – Terms and definitions related to the quality of telecommunication services, September 2008.

[19]. R.Adhikari and R.K.Agrawal .*An Introductory study on Time Series Modeling and Forecasting*. , New Delhi, AV Akademikervelag GmbH&Co.KG,2013.

[20]. O. Ostashchuk "Time series data prediction and Analysis" Master Thesis, Technical University in Prague, Czech, Jan 2017.

[21]. Mamidi Sai Akash," Artificial intelligence and neural networks" International Journal of Soft Computing and Artificial Intelligence, Volume-3, Issue-2, India, Nov-2015.

[22]. S.Haykin .*Neural Networks and Learning Machines*. 3rd ed., USA, Pearson Education, 2009.

[23]. S. Xu, L. Chen," A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining", 5th International Conference on Information Technology and Applications, Australia, 2008

[24]. K. Gnana and S. N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks", Research article, India,2013.

[25]. D. P. Pitambare," Survey on Optimization of Number of Hidden Layers in Neural Networks", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 11, India, November 2016.

[26]. F.S. Panchal1 and M.Panchal," Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network", International Journal of Computer Science and Mobile Computing,Vol.3 Issue.11, November- 2014, pg. 455-464, India, November- 2014.

[27]. G. Panchal1, A.Ganatra, Y. Kosta and D.Panchal4," Behaviour Analysis of Multilayer Perceptron with Multiple Hidden Neurons and Hidden Layers", International Journal of Computer Theory and Engineering, Vol. 3, No. 2, India,April 2011

[28]. N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi," An artificial neural network model for rainfall forecasting", Published in Hydrology and Earth System Science, Thailand ,2009.

[29]. J.Han , M.Kamber and J.Pei. *Data mining Concepts and Techniques*. 3rd ed.,USA, Morgan Kaufmann, 2012.

[30]. www.mathwork.com [Accessed: 29- Sep- 2018].