DEVELOPMENT AND COMPARATIVE ANALYSIS OF QUOTIENTS REGRESSION BASED EMPIRICAL AND ARTIFICIAL NEURAL NETWORK BASED MODELS FOR PATH LOSS PREDICTION

BY

ABRAHAM CHUWANG DEME

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY,
ZARIA, NIGERIA

NOVEMBER, 2015

DEVELOPMENT AND COMPARATIVE ANALYSIS OF QUOTIENTS REGRESSION BASED EMPIRICAL AND ARTIFICIAL NEURAL NETWORK BASED MODELS FOR PATH LOSS PREDICTION

BY

ABRAHAM CHUWANG DEME
B.Sc (RIGA TECH UNI OF AVIATION, LATVIA) 1994
M.Sc ((RIGA TECH UNI OF AVIATION, LATVIA) 1995
PhD/ENG/14783/2007-2008

A THESIS PRESENTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A DOCTOR OF PHILOSOPHY (PhD) DEGREE IN ELECTRICAL ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,
FACULTY OF ENGINEERING
AHMADU BELLO UNIVERSITY,
ZARIA, NIGERIA

NOVEMBER, 2015

DECLARATION

I, Abraham Chuwang DEME, hereby declare that the work in this thesis entitled "**DEVELOPMENT AND COMPARATIVE ANALYSIS OF QUOTIENTS REGRESSION BASED EMPIRICAL MODELS AND ARTIFICIAL NEURAL NETWORK BASED MODELS FOR PATH LOSS PREDICTION**" has been carried out by me in the Department of Electrical & Computer Engineering, Ahmadu Bello University, Zaria. The information derived from literature has been duly acknowledged in the text and a list of references provided. No part of this thesis was previously presented for another degree or diploma at this or any other institution.

Abraham Chuwang DEME

_____	_____	
(Student)	Signature	Date

CERTIFICATION

This thesis entitled "**DEVELOPMENT AND COMPARATIVE ANALYSIS OF QUOTIENTS REGRESSION BASED EMPIRICAL MODELS AND ARTIFICIAL NEURAL NETWORK BASED MODELS FOR PATH LOSS PREDICTION, A CASE STUDY: JOS-ABUJA RURAL AREA, ABUJA AND MAIDUGURI**", by Abraham Chuwang **DEME**, meets the regulations governing the award of the degree of Doctor of Philosophy (PhD) in Telecommunications, of the Ahmadu Bello University, Zaria, and is approved for its contribution to knowledge and literary presentation.

Chairman, Supervisory Committee (Dr. D. D. Dajab)	..... Signature	..... Date
Member, Supervisory Committee (Prof. B.G. Bajoga)	..... Signature	..... Date
Member, Supervisory Committee (Prof. M.B. Mu'azu)	..... Signature	..... Date
Head of Department (Dr. Y. Jubril)	..... Signature	..... Date
Dean, School of Post-graduate Studies (Prof. Kabir Bala)	..... Signature	..... Date

DEDICATION

This work is dedicated to the Almighty God and my late mother (Mrs. Deme Rebecca).

ACKNOWLEDGEMENT

I remain eternally grateful to the Almighty and Eternal God for giving me the grace to surmount all obstacles encountered during the course of this research, and for all He has done in my life.

My special thanks go to my supervisory committee comprising of Dr. D. D. Dajab, Prof. B.G. Bajoga and Prof. M.B. Mu'azu, as well as Dr.S. M. Sani, for their guidance and immense contributions towards making this research a success. My thanks also go to all those who helped me in one way or the other.

DEME, Abraham Chuwang
Department of Electrical and Computer Engineering
Ahmadu Bello University, Zaria, Nigeria

TABLE OF CONTENTS

CHAPTER TWO: LITERATURE REVIEW

CHAPTER FOUR: RESULTS AND ANALYSES

CHAPTER FIVE: SUMMARY, RECOMMENDATION AND CONCLUSION

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

1G	First Generation Network
2G	Second Generation Network
2.5G	Second and Half Generation
3G	Third Generation Network
4G	Fourth Generation Network
5G	Fifth Generation Network
3GPP	3rd Generation Partnership Project
AMC	Adaptive Modulation and Coding
AMPS	Advanced Mobile Phone Service
$A_{MU}(f,d)$	median attenuation
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
APN	Access Point Name
AuC	Authentication Centre
BS/BST	Base Stations
BSC	Base Station Controller
BSS	Base Station Subsystem
BTS	Base Transceiver Station
CDMA	Code Division Multiple Access
CH_{QRT}	Quotients Regression Technique Adapted COST 231 Hata path loss
CN	Core Network
COST	European Cooperation in the field of Scientific and Technical research

CWI_{RAT}	Root Mean squared Error adapted COST 231 Walfisch-Ikegami
CWI_{QRT}	QRT Adapted COST 231 Walfisch-Ikegami
D2D	Device-to-Device
DCS	Digital Communication System
EIR	Equipment Identity Register
EIRP	Effective Isotropic Radiated Power
eNodeB	evolved NodeB
EPC	Evolved Packet Core
EPS	Evolved Packet System
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FDMA	Frequency Division Multiple Access
FT	File Transfer
FTP	File Transfer Protocol
FWA	Fixed Wireless Access
G_{AREA}	Gain due to type of environment (correction factor for suburban areas)
GIWU	GSM Interworking Unit
GPRS	General Packet Radio Service
GRBF-NN	Generalized Radial Basis Function Neural Network
GSM	Global System for Mobile communications
G_T	Transmitter gain
GTD	Geometrical Theory of Diffraction
HARQ	Hybrid Automatic Request
H_{BG}	Base Station Antenna Height correction factor

HLR	Home Location Register
H_{MG}	Mobile Station Antenna Height correction factor
HO_{RAT}	Root Mean Squared Error Adapted Hata Okumura model
HO_{QRT}	Quotients Regression Adapted Hata Okumura model
HSDPA	High-Speed Downlink Packet Access
HSPA	High-Speed Packet Access
HSS	Home Subscriber Server
HSUPA	High-Speed Uplink Packet Access
IEEE	Institute of Electrical and Electronic Engineers
IMS	IP Multimedia Subsystem
IMT-A	International Mobile Telecommunications- Advanced
IP	Internet Protocol
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union Radio-communication
L_{FSL}	Free Space path loss
L_{LS}	Least Squares function through computed path loss
$L_{RAT\_adapted}$	Root Mean Squared Error adapted path loss
L_{OKM}	Okumura Path loss
L_{QRT}	Quotients Regression Technique Adapted path loss
LOS	Line-Of-Sight
LTE	Long-Term Evolution
MIMO	Multiple-Input Multiple-Output
MLP-NN	Multilayer Perceptron Neural Network

MME	Mobility Management Entity
MMS	Multimedia Messaging Service
MS	Mobile Station
MSC	Mobile Switching Centre
NAS	Non Access Stratum
NLOS	Non-Line-Of-Sight
NSS	Network and Switching Subsystem
OFDM	Orthogonal Frequency Division Multiplexing
PCRF	Policy Control and Charging Rules Function
PCS	Personal Communication System
PDN	Packet Data network
P-GW	PDN Gateway
PSTN	Public Switched Telephone Network
P_T	Transmitted power
QAM	Quadrature amplitude modulation
Q(d)	Adaptation function
QCI	QoS Class Identifier
Q_i	Quotients computed at various intervals
QoS	Quality-of-Service
QPSK	Quadrature Phase Shift Keying
QRT	Quotients Regression Technique
R^2	Coefficient of Determination
RAT	Root Mean Squared Error (RMSE) Adaptation Technique

RLS	Recursive Least Squares
RMSE	Root Mean Square Error
S-GW	Serving Gateway
SUI	Stanford University Interim
TDMA	Time Division Multiple Access
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTD	Uniform Theory of Diffraction
UTRAN	Universal Terrestrial Radio Access Network
VLR	Visitor Location Register
VoIP	Voice over IP
WAN	Wide-Area-Networks
WAP	Wireless Application Protocol
W-CDMA	Wideband Code Division Multiple Access
μ	Absolute Mean Error
σ	Standard Deviation

ABSTRACT

The aim of this study is to develop and analytically compare Quotients Regression based empirical models with artificial neural network (ANN) based models for path loss prediction. The terrains considered as case study include i) the rural terrain between Jos and Abuja ii) the Urban terrain (Abuja), and iii) the semi-urban terrain (Maiduguri). The empirical models considered include the Okumura, Hata-Okumura, COST 231 Hata and the COST 231Walfisch-Ikegami, while the two types of ANN include the Multilayer Perceptron Neural Network (MLP-NN) and the Generalised Radial Basis Function Neural Network (GRBF-NN). A Quotients Regression Technique (QRT) for empirical model adaptation was developed and used to selectively adapt these empirical models to the terrains, based on path loss measurements obtained from Base Transceiver Stations (BTS) situated within the terrains. The adaptation accuracy of the QRT was determined through comparisons with two existing adaptation techniques: i) The Okumura G_{AREA} (Gain due to type of environment) correction factor, and ii) The Root Mean Squared Error (RMSE) Adaptation Technique (RAT). The comparative analysis of the QRT adapted empirical models with ANN-based models was based on three distinct approaches: i) Splitting path loss data into 60% Training, 10% Validation and 30% Testing, ii) Splitting path loss data into 50% Training Set and 50% Testing Set. iii) Random training with path loss data from one BTS and testing with data from another. The following results were obtained: i) The QRT has the highest adaptation accuracy, based on combined RMSE (Root Mean Squared Error) value of **2.1dB** across the three terrains, the RAT technique has **5.66dB** across the three terrains, while the Okumura G_{AREA} has **8.95dB** across the rural terrain alone, ii) The ANN-based models have the highest path loss prediction accuracy, based on an RMSE value of **3.98dB**, followed by the QRT adapted empirical models with **4.49dB**. The RAT adapted empirical models have **5.83dB**, while the empirical models are the least accurate with **7.07dB**. By implication, the ANN-based models only slightly outperform the QRT adapted empirical models by **0.51dB**. However, the QRT adapted empirical models have the highest R^2 (coefficient of determination) value of **0.81**, and by implication, the best fit resulting from the best correlation with the measured path loss data. On the other hand, the QRT adapted empirical models offer an improvement of about **1.34dB** over the RAT adapted empirical models, as well as an improvement of **2.58dB** over existing Empirical Models. The proximity in performance of the QRT adapted empirical models to the ANN-based models can be attributed to the efficiency of the QRT.

CHAPTER ONE

INTRODUCTION

1.1 Background

Since the introduction of cellular systems for mobile communications decades ago, they have undergone tremendous growth and evolution. Over the decades the number of mobile subscribers around the globe has also undergone tremendous growth and will continue to grow. Hence, the availability of high capacity mobile networks with quality delivery of service has become very important especially in recent times. As a result, the creation of a mobile network with good network coverage demands proper planning. One of the greatest challenges faced by radio systems engineers is the accurate determination of the radio propagation characteristics of a particular terrain, with a view to designing a well engineered radio path.

When planning a cellular communications network, it is necessary to define the optimum number of base stations within a coverage area and to also resolve other network related problems. Hence, it is necessary to determine the characteristics of radio signals within the limits of the service area. Since various terrains create specific conditions for propagation of the radio waves within a service area, detailed knowledge of channels propagation characteristics of radio signals is a necessary precondition for the development of effective communication systems operating in any environment. This stems from the fact that as radio signals propagate from the base station to the mobile station, some power is lost, and this power loss is known as path loss. Path loss is dependent on the carrier frequency, antenna height and distance between mobile station and base station (Ashis, 2012). There is a variety of models used by researchers and radio engineers to predict path loss across a particular terrain. A model that suits a given terrain (or environment) may not necessarily be suitable to another terrain. Hence, radio

engineers and researchers have been evolving new techniques for accurate prediction of path loss.

Researchers for many years have made attempts to either develop new propagation models or modify existing ones. Empirical models are some of the most widely used radio propagation models. However, it is pertinent to note that these empirical models are simple to implement but not universally applicable due to terrain clutter differences across the globe. As such, researchers over the years have developed various techniques for adapting empirical models to suite terrains. The most popular technique used by researchers in recent times have had to do with the use of computed correction factors to achieve adaptation as demonstrated by Nadir et al (2010), Ubom et al (2011), Ogbulezie et al (2013), etc. However, in certain scenarios, this technique may not provide the best possible fit to the empirical model if the slope component of the empirical model significantly differs from that of the function of the best fit curve through measured path loss points. Hence, it necessary to create a technique that does not only guarantee the best possible fit for the empirical model, but also ensures the adapted model is robust when tested using new data.

Structural impairments along radio propagation paths demand that non-linear approximation functions are used for path loss prediction. Empirical models are linear approximation based functions; hence they not very accurate in predicting path loss along terrains with well diversified structural impairments, demanding the implementation of non-linear functions that can lead to greater prediction accuracy. As described in (Faria et al., 2009), neural networks can learn to approximate any function to a given accuracy and behave like associative memories by using just example data that is representative of the desired task, operating then as model free estimators. This gives them a key advantage over traditional

approaches to function estimation such as the statistical methods. Hence, computational intelligence techniques have been applied recently to predict path loss with greater accuracy as demonstrated by Ostlin (2010), Ignacio et al. (2012), Abraham et al. (2014), Joseph et al. (2014), Callistus et al. (2015) etc.

1.1.1 The Cellular Network Concept

The cellular network has become the most common platform for wireless communications. With a mobile phone, one can make calls and access data from almost any location across the globe. A cellular network is a communications system whose service area is divided into operating zones called cells, inside which communication between mobile and base stations is carried out by a radio channel. Switching equipment is used to interconnect different parts of a mobile network and to also allow access to the fixed Public Switched Telephone Network (PSTN) (Tarun et al, 2013).

The introduction of the cellular concept played a great role in resolving the spectral congestion and user's capacity problems (Ashis, 2012). The cellular concept splits a given service area into cells, each served by a Base Station (BS), thereby enabling the frequency reuse concept.

The high network capacity of modern cellular networks stems from the frequency reuse concept as frequency spectrum allocation to cellular networks is very limited (Rakibul et al., 2011). As such, the coverage area is divided into cells, each of which is served by a BS. Each BS (or cell) is assigned a group of frequency bands or channels. These are assigned to subscribers on demand. In or to avoid radio co-channel interference, the group of channels assigned to neighbouring cells must be different. However, distant cells with insignificant co-channel interference between them can be assigned the same group of channels. Typically, seven

neighbouring cells are grouped together to form a cluster. Cells are group in sevens to form a cluster, and hence and the total available channels are divided into seven groups, each of which is assigned to a cell. As shown in Figure 1.1, cells marked with the same identifier have the same group of channels assigned to them. Furthermore, the cells marked with different identifiers must be assigned different groups of channels.

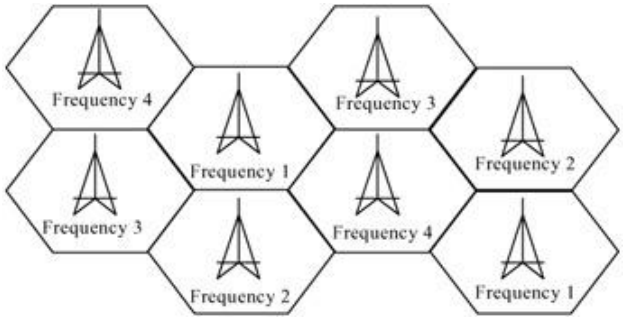
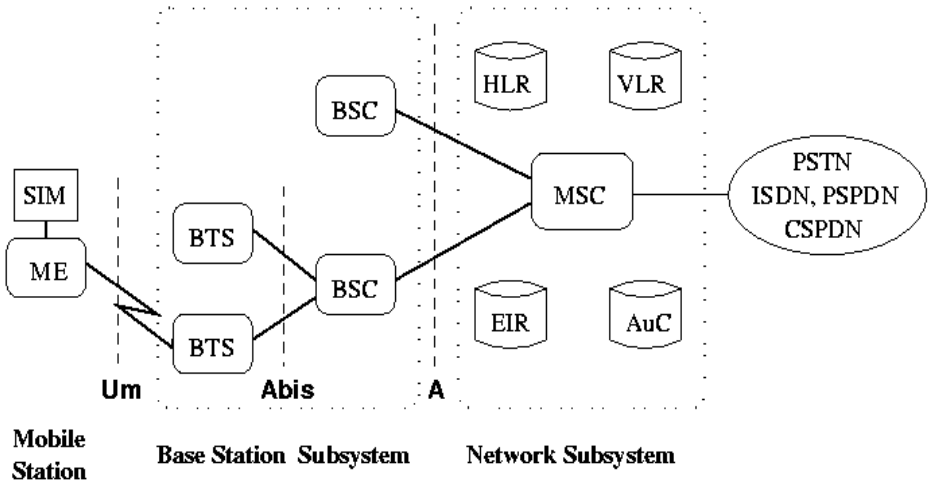


Figure 1.1: A Cellular Network Structure showing Frequency Reuse (Lei et al., 2004)

Jingyuan and Ivan (2005) further explain that during communication a Mobile Station (MS) registers with the nearest BS while the corresponding Mobile Services Switching Center (MSC) stores the information about that MS and its position. This information is used to direct incoming calls to the MS. The concept behind mobile communication is based on the hand-over process. Hand-over occurs during mobile communication when the MS moves from its serving cell to a neighboring cell, thereby forcing a change of frequency since neighboring cells use different channels. The BS constantly monitors any decrease in signal power as the MS approaches the edge of a cell, and compares the signal strength with the signals coming from adjacent cells, after which the call is handed over to the cell with the strongest signal. During the switch, the line is lost for about 400ms. When the MS moves to a different location, it registers itself at the new MSC, and the information on the MS is constantly is updated, such that the MS can be used outside of its original location.

1.1.2 GSM Network Architecture

The GSM (Global System for Mobile Communications) is a typical cellular network. Introduced in the 1990's the GSM is currently the most popular mobile phone across the globe especially the third world countries. The GSM network architecture shown in figure 1.2, as described by Sneha et al. (2014) includes the following components:



SIM Subscriber Identity Module BSC Base Station Controller MSC Mobile services Switching Center
ME Mobile Equipment HLR Home Location Register EIR Equipment Identity Register
BTS Base Transceiver Station VLR Visitor Location Register AuC Authentication Center

Figure 1.2: GSM (3G) Architecture (Sneha et al., 2014)

i) The **Mobile Station (MS)** is basically a mobile phone that contains the Subscriber Identity Module (SIM), which contains relevant user information.

ii) The **Base Station Subsystem (BSS)** provides the interface between the MS and the NSS. It handles transmission and reception. The BSS comprises of the following:

1. **Base Transceiver Station (BTS) or Base Station:** the BTS is found in the centre of a cell and it is mapped to transceivers and antennas used in given cell within the network. Its transmitting power defines the size of a cell.

2. Th**e Base Station Controller (BSC):** It is responsible for controlling a group of BTSs and also managing their radio resources. A BSC handles handoffs, frequency hopping, exchange functions and power control over each managed BTSs.

iii) The **Network and Switching Subsystem (NSS):** The NSS manages communications between a set of mobile users and other mobile users, Integrated Systems Digital Network (ISDN) users, fixed telephony users, etc. It also contains a database for storing information about subscribers and mobility management. Details on the components that make up the NSS can be obtained from Sneha et al., (2014).

The GSM network as shown in figure 1.2 comprises three interfaces: the Um, the A-bis and the A. The **Um** is the Radio interface between MS and BTS. The **A-bis** is the interface between BTS and BSC. Its primary functions include traffic channel transmission, terrestrial channel management, and radio channel management. The interface between the BSS and the NSS is called the **A** interface.

1.1.3 Evolution of Cellular Technology

With technological advancement, the cellular technology has evolved through generations, namely 1G, 2G, 2.5G, 3G and 4G. The fifth generation (5G) is the next wireless communication standard.

a) First Generation (1G)

The First Generation (1G) of cellular technology was essentially an analogue wireless telecommunication system used for voice calls using cell phones (Mudit and Anand, (2010). The

1G standards used included the NMT (Nordic Mobile Telephone), used in Nordic countries, Eastern Europe and Russia. Others include AMPS (Advanced Mobile Phone System) used in the United States, TACS (Total Access Communications System) in the United Kingdom, C-Netz in West Germany, Radiocom 2000 in France, and RTMI in Italy.

b) Second Generation (2G)

As described in Mudit and Anand, (2010), the second generation (2G) was digital and made use of the Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) concepts to increase the network capacity. The 2G network had improved security features and accommodated voice coding and encryption. Popular 2G standards included the GPRS (General Packet Radio Service) classified as 2.5G and EDGE (Enhanced Data rates for GSM Evolution) known as 2.75G. EDGE is an upgrade over GPRS and can function on any network with GPRS deployed on it, provided the carrier implements the necessary upgrades. EDGE technology carries packet switch data and circuit switch data and at a faster rate than GSM. GPRS could provide data rates ranging from 56 kbit/s to 115 kbit/s. 2G provided services such as Wireless Application Protocol (WAP) access, Multimedia Messaging Service (MMS), and for Internet communication services such as email and World Wide Web access.

c) Second and Half Generation (2.5G)

The Second and Half Generation as the name implies came into existence between the second generation (2G) and third generation (3G). As described in Mudit and Anand (2010), the term 2.5G refers to 2G-systems that combine a packet switched domain a circuit switched domain. The term "2.5G" is an informal term, invented solely for marketing purposes, unlike "2G" or

"3G" which are officially defined standards based on those defined by the International Telecommunication Union (ITU). The standard used on 2.5G was the GPRS, which provided data rates from 56 kbit/s to 115 kbit/s. It provided services such as Wireless Application Protocol (WAP) access, Multimedia Messaging Service (MMS), and for Internet communication services such as email and World Wide Web access. 2.5G networks also provided services such as WAP, MMS, SMS mobile games, and search and directory.

d) Third Generation (3G)

As described in Mudit and Anand (2010), the third generation (3G) was aimed at providing high-speed packet-switching data transmission in addition to circuit-switching voice transmission across the Internet. Common 3G network services include wireless voice telephony, video calls/ teleconferencing, broadband wireless data, GPS (global positioning system), mobile television etc. The standard implemented is the High-Speed Packet Access (HSPA) capable of providing data transmissions speeds up to 14.4Mbit/s on the downlink and 5.8Mbit/s on the uplink. 3G technologies make use of TDMA and CDMA. 3G Technology was developed for fast data transfer rates. High-Speed Downlink Packet Access (HSDPA) is a 3.5G is a packet-based Wireless CDMA (W-CDMA) downlink data service with data transmission up to 8-10 Mbit/s and 20 Mbit/s for Multiple-Input Multiple-Output (MIMO) systems over at 5MHz. HSDPA is implemented as Adaptive Modulation and Coding (AMC), MIMO, Hybrid Automatic Request (HARQ), fast cell search, and advanced receiver design. Another standard is the 3.75G High-Speed Uplink Packet Access (HSUPA) capable of higher transfer rates.

e) Fourth Generation 4G

4G refers to the fourth generation of cellular wireless standards. It is basically the extension of the 3G technology with more bandwidth and services and offers high quality audio/video streaming over end to end Internet Protocol (Mudit and Anand, 2010). According to Shin et al. (2013), the fourth generation long-term evolution Advanced (4G LTE Advanced) is the latest standard in the development of 4G mobile networks and has been designed to offer users of 4G mobile devices much faster data speeds than those on offer from existing 4G LTE networks, making it a true 4G standard.

According to Arun et al., (2013), 4G networks are essentially based on packet switching technology. The fourth generation mobile systems use orthogonal frequency division multiplexing (OFDM), MIMO, software defined radio (SDR) technologies (Patil et al, 2012). The modulation techniques implemented enhance efficiency by dividing 5, 10 or 20 MHz wide channels into smaller sub channels or subcarriers each 15 kHz wide. Each is modulated with part of the data. The modulation techniques used are QPSK (Quadrature Phase Shift Keying) or 16QAM *(Quadrature amplitude modulation).* With the aid of MIMO operation that uses several transmitter-receiver-antennas, the data stream is divided amongst the antennas to boost speed and to make the link more reliable. With the help of OFDM and MIMO, LTE can deliver data rates up to 100 Mb/s downstream and 50 Mb/s upstream under the best conditions. In 4G the theoretical upper data rate is 1Gb/s.

In contrast to the GSM network architecture, the LTE- Advanced Architecture comprises of essentially two layers: the access network known as the Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the core network known as the Evolved Packet Core (EPC) network as shown in Figure 1.3 (Alcatel, 2013). As described in (Alcatel, 2013), the term “LTE”

encompasses the evolution of the Universal Mobile Telecommunications System (UMTS) radio access through the E-UTRAN, and it is accompanied by an evolution of the non-radio aspects under the term "System Architecture Evolution" (SAE), which includes the EPC network.

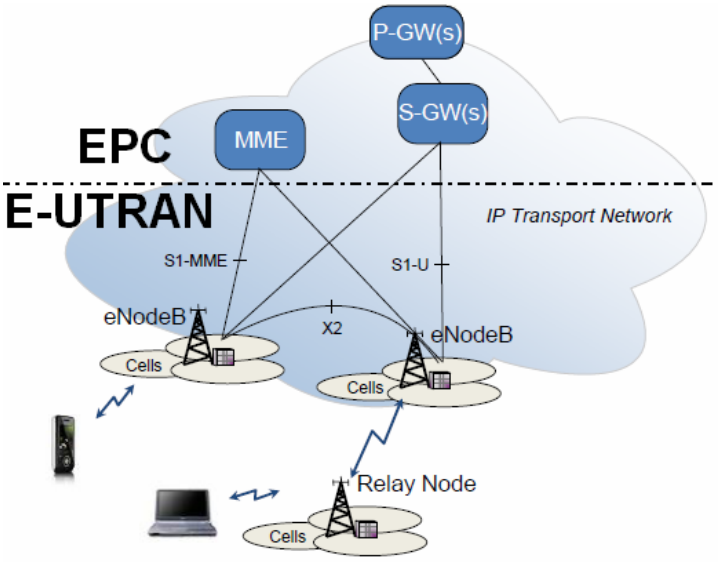


Figure 1.3: LTE-Advanced E-UTRAN network architecture (Ghassan et al., 2012)

Together LTE and SAE comprise the Evolved Packet System (EPS). EPS uses the concept of EPS bearers to route IP traffic from a gateway in the PDN to the UE (User Equipment). A bearer is an IP packet flow with a defined quality of service (QoS) between the gateway and the UE. The E-UTRAN and EPC together set up and release bearers as required by applications. Details on the LTE-Advanced Architecture can be obtained from Alcatel (2013).

f) Fifth Generation (5G)

As described in Ekram et al., (2014), the evolving fifth generation (5G) cellular wireless networks are targeted towards providing higher data rates, excellent end-to-end performance and better user-coverage in hot-spots and crowded areas with lower latency, lower energy

consumption and cost per information transfer, compared with existing cellular networks. In order to address these challenges, 5G systems will be based on a multi-tier architecture consisting of macrocells, different types of licensed small cells, relays, and device-to-device (D2D) networks to serve users with different quality-of-service (QoS) requirements in a spectrum and energy-efficient manner. The features of 5G will include wireless networks network tiers of different sizes, transmit powers, backhaul connections, different radio access technologies that are accessed by an unprecedented number of smart and heterogeneous wireless devices. The 5G architectural upgrade along with the advanced physical communications technology such as high-order spatial multiplexing multiple-input multiple-output (MIMO) communications will provide higher network capacity for more simultaneous users, or higher level spectral efficiency, when compared to the 4G networks. Radio resource and interference management will be a key research challenge in multi-tier and heterogeneous 5G cellular networks. Details on the prospects and challenges of 5G cellular networks are presented in Ekram et al., (2014).

1.2 Aim and Objectives

The aim of this study is to develop the Quotients Regression Technique (QRT) for adapting empirical models, analytically compare the QRT with existing adaptation techniques, and to also perform a path loss prediction comparison of the QRT adapted empirical models with artificial neural network (ANN) based models.

The objectives of the study are as follows:

i) Determination of reliabilities of empirical models for path loss prediction across the terrains under investigation.

ii) Development of the QRT for empirical model adaptation.

iii) Determination the accuracy of the QRT relative to existing techniques.

iv) Determination of path loss prediction accuracy of QRT adapted empirical models relative to empirical models adapted by other techniques.

v) Creation of the ANN-based prediction models.

vi) Determination of path loss prediction accuracy of QRT adapted empirical models relative to ANN-based prediction models.

vii) Determination of the most suitable path loss prediction models for the terrains under investigation.

1.3 Statement of Problem

Adequate knowledge of radio propagation characteristics across a specific terrain is an essential requirement in the planning of a wireless telecommunications network. Since path loss serves as the dominant factor for the characterization of a radio link, there is need for accurate path loss prediction so that the radio path can be optimally engineered. Path loss varies from one environment to the other according to the physical nature, dimensions and geometries of the various obstacles that perturb radio propagation. Hence, it is of high necessity to create prediction models that are not only very accurate, but also computationally efficient.

Although empirical models are quite simple to implement, they are not universally applicable due to terrain diversity across the globe, in spite of the availability of correction factors. The most popular technique for adapting empirical models has to do with introducing computed errors as correction factors into empirical model expressions. However, these correction factors in most cases only modify the constant within an expression, disregarding the

slope coefficient, which is a dominant factor in determining how well an empirical model fits (or is adapted to a given terrain). By implication, if the slope of the best fit curve through measured path loss points significantly differs from that of the empirical model expression, such a technique will be highly inaccurate. As a result, these techniques are limited in terms of ability to accurately adapt empirical models to terrains due to terrain diversities. Hence, it is necessary to develop a technique that does not only provide the best possible fit for the empirical model, but also ensures that the adapted empirical model is robust when tested with new data.

In this study, a novel technique for adapting empirical models, termed the Quotients Regression Technique (QRT) is proposed. The QRT provides the best possible fit for an empirical model by directly fitting the empirical model onto the best fit curve for measured path loss points, thereby ensuring greater correlation with the measured data.

Existing literature have revealed that computational intelligence techniques are the recent alternative approaches used to predict the path loss at a particular location in an investigated area (Ostlin, 2010). Such techniques include Artificial Neural Networks (ANNs). ANNs have the ability to handle non-linear function approximation with greater accuracy than those techniques which are based on linear regression.

Hence, further in this study, QRT adapted empirical models are analytically compared for path loss prediction accuracy with ANN-based models, as well as with empirical models adapted by existing techniques. As case study, at an operating frequency of 900MHz, three Nigerian terrains are considered: the rural area between Jos and Abuja, the urban terrain (Abuja), and the semi-urban terrain (Maiduguri).

1.4 Methodology

The methodology adopted consists of the following series of activity.

i) Received power measurement and path loss Computation.

ii) Determination of reliabilities of empirical models for path loss prediction within the terrains under investigation.

iii) Development of the Quotients Regression Technique (QRT) for empirical model adaptation.

iv) Adaptation accuracy comparison of the QRT with the Okumura G_{AREA} (Gain due to type of environment) Technique and the RMSE (Root Mean Squared Error) Adaptation Technique (RAT). The Okumura G_{AREA} technique involves adapting the Okumura model to a given rural terrain using a G_{AREA} value obtained from the Okumura Curves, as correction factor. On the other hand, the RAT has to with either subtracting a computed RMSE from the model expression if path loss is overestimated, or vice versa.

v) Performance Comparison of the QRT adapted empirical models with the Okumura G_{AREA} and RAT adapted empirical models.

vi) Creating the artificial neural network predictors. The two types of Feed-forward ANNs considered are the Multilayer Perceptron Neural Network (MLP-NN) and the Radial Basis Function Neural Network (RBF-NN).

vii) Comparative analysis of QRT adapted empirical models with ANN based predictors.

1.5 Significance of the Study

As wireless telecommunication becomes ever increasingly popular and indispensable, it is of paramount importance to ensure quality delivery of service to subscribers within a service area. This is achievable through proper network planning and optimization. Radio propagation models are widely used in coverage determination during network planning. A model that satisfies a given terrain may not be suitable to a different terrain due to clutter differences across the globe. Hence, it is necessary to come up with efficient and effective techniques for formulating terrain - specific radio propagation models that can accurately predict path loss within a given area. This is what this study aims to achieve.

1.6 Thesis Outline

This thesis consists of five chapters. The introductory part comprising of the background, aim and objectives, Statement of Problem, methodology, as well as the thesis outline are presented in chapter one. The review of fundamental concepts and review of similar works are contained in chapter two. Chapter three comprises of the materials and methods which describe the measurements procedure, path loss computation, development of the proposed Quotients Regression Technique (QRT), empirical model adaptation, neural network models creation and the comparative analysis of QRT adapted empirical models with neural network based models for path loss prediction. Results and discussions of empirical model adaptation, comparison of adaptation techniques, as well as the comparison of QRT adapted empirical with neural network based models, are presented in Chapter four. Chapter 5 contains summary, significant contributions, conclusions, recommendations for further works, as well as limitations.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Extensive studies have been carried out on the modelling of radio propagation characteristics across various terrains. For this purpose, a wide range of radio propagation models have been used globally. Some of the most widely used propagation models are the empirical models. Unfortunately, existing empirical models though easier to implement, are less sensitive to the environment's physical and geometrical structures and not so accurate, while the deterministic models on the other hand, though more accurate, are computationally inefficient and require more detailed site-specific information which is often difficult to come by (Abhayawardhana et al. 2005).

In recent times, computational intelligent techniques have been used to model radio propagation such as in Ostlin (2010) and Abaraham (2013). This can be attributed to their ability to handle complex non-linear function approximation with a greater accuracy than those techniques which are based on linear regression. As model free estimators, Neural Networks have the capability to handle complex problems based on the availability of training data. For instance, one of the key benefits of Neural Networks is their ability to approximate functions with greater accuracy than statistical methods. Hence, radio propagation models created on the bases of non-linear function approximation have been proven to predict path loss with greater accuracy than those that are based on linear regression. This can be attributed to the heterogeneity of terrain clutters resulting from varying obstacles that perturb radio propagation. Belonging to the category of models that perform non-linear function approximation are feed-

forward neural network, the most commonly used being the Multilayer Perceptron Neural Network (MLP-NN) and the Generalised Radial Basis Function Neural Network (GRBF-NN).

2.2 Review of Fundamental Concepts

The fundamental concepts that form the bases of the study have to do with Radio propagation concepts, empirical propagation models, the techniques for adapting empirical models to suit various terrains, as well as artificial neural networks.

2.2.1 Radio Propagation Mechanisms

Radio propagation mechanisms are quite complicated and diverse. This is as a result of the various sizes, shapes and composition of obstacles situated between the receiver and the transmitter. As the waves travel from transmitter to receiver, attenuation (reduction or loss) of the signal strength occurs. There is free space attenuation and the excess attenuation due to obstacles such as gas, rain, vegetation, wall, dust, etc. In addition, the signal propagates in different manners, known as diffraction, scattering, reflection, transmission, and refraction.

2.2.1.1 Free Space Attenuation

When a signal is transmitted from an antenna, its strength reduces as it travels through free space. This is called attenuation. Free Space attenuation occurs in direct line- of-sight (LOS) situations. The power received at any given point in space is inversely proportional to the square of the distance covered by the signal. As power is isotropically radiated from an antenna, it can be assumed that a ‘sphere’ of power is formed (Mishra, 2004). The Free space Loss expression is obtained as follows (Mishra, 2004):

The surface area of this power sphere is:

$$A = 4\pi R^2 \quad (2.1)$$

The power density S at any point at a distance R from the antenna can be expressed as:

$$S = P*G/A \quad (2.2)$$

Where P is the power transmitted by the antenna, and G is the antenna gain. Thus, the received power P_r at a distance R is:

$$P_r = P*G_t*G_r (\lambda/4\pi R)^2 \quad (2.3)$$

Where G_t is the transmitting antenna gain and G_r is the receiving antenna gain. On converting this to decibels we have:

$$Pr(dB) = P(dB) + G_t(dB) + G_r(dB) + 20\log (\lambda/4\pi) - 20\log d \quad (2.4)$$

Last two terms in equation (2.4) are together called the path loss in free space, or the free space loss. The first two terms (P and G_t) combined are called the effective isotropic radiated power (EIRP).

Thus:

$$\text{Free-space loss (dB)} = EIRP + G_r(dB) - P_r(dB) \quad (2.5)$$

The free-space loss can then be given as:

$$L= 92.5 + 20\log f + 20\log d \quad (2.6)$$

Where f is the frequency in GHz and d is the distance between transmitter and receiver in kilometres.

For radio applications, **f** is measured in MHz. Therefore, the equation becomes:

$$L= 32.45 + 20\log f + 20\log d \quad (2.7)$$

2.2.1.2 Diffraction (or Shadowing)

Diffraction refers to the change in direction of propagation of a radio wave when it strikes a surface that cannot absorb it. It occurs when direct line-of-sight (LOS) propagation between the transmitter and the receiver is obstructed by an opaque obstacle (usually with a sharp edge) whose dimensions are considerably smaller than the signal wavelength (Tapan et al, 2003). The diffraction occurs at the obstacle edges where the radio waves are scattered and, as a result, they are additionally attenuated. The diffraction mechanism often allows the reception of weakened radio signals when the LOS conditions are not satisfied (NLOS case), whether in urban or rural environments.

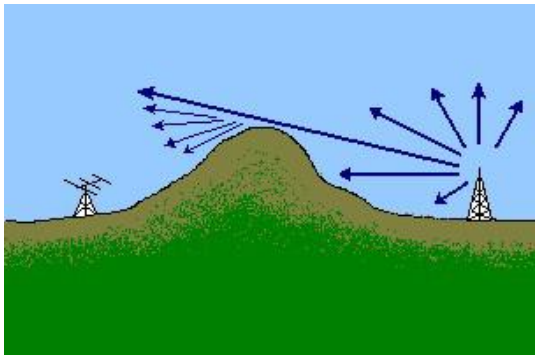


Figure 2.1: Radio Wave Diffraction (Li Qing, 2005).

The loss due to diffraction depends upon the kind of obstruction in the path. In practice, the mobile antenna is at a much lower height than the base station antenna, and there may be high buildings or hills in the area as shown in Figure 2.1. Thus, the signal undergoes diffraction in reaching the mobile antenna. In other words, in NLOS situations, waves travel from transmitter to receiver through diffraction. This phenomenon is also known as 'shadowing' because the mobile receiver is in the shadow of these structures.

a) Knife-Edge Diffraction Geometry

The Knife-Edge Diffraction Geometry as described in (Abhijit, 2009), is as follows:

Taking into consideration an impenetrable obstruction of height h, at a distance d_1 from the transmitter, and d_2 from the receiver as shown in Figure 2.2. The path difference between direct path and the diffracted path is given by (Abhijit, 2009)

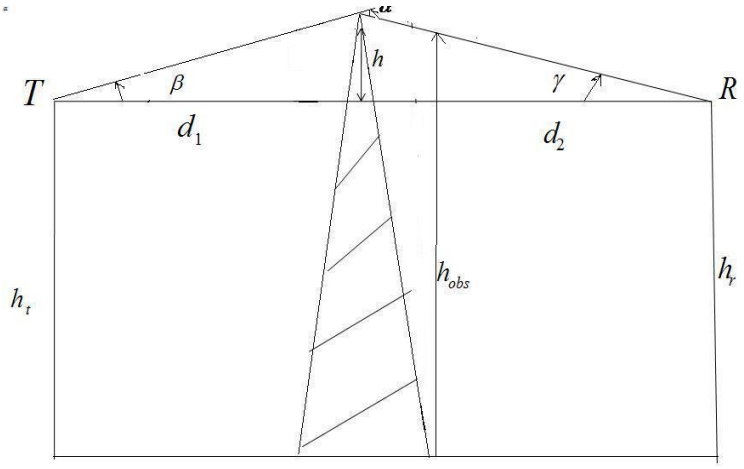


Figure 2.2: Diffraction through a sharp edge (Abhijit, 2009)

$$\delta = \sqrt{d_1^2 + h^2} + \sqrt{d_2^2 + h^2} - (d_1 + d_2)$$

$$= h^2(d_1 + d_2)/(2d_1d_2) \tag{2.8}$$

Thus the phase difference equals (Abhijit, 2009)

$$\varphi = \frac{2\pi\delta}{\lambda} = 2\pi h^2(d_1 + d_2)/\lambda 2d_1d_2 \tag{2.9}$$

With the considerations that $\alpha = \beta + \gamma$ and $\alpha \approx tan\alpha$

$$\alpha tan\alpha = tan\beta + tan\gamma = \frac{h}{d_1} + \frac{h}{d_2} = h(d_1 + d_2)/d_1d_2 \tag{2.10}$$

In order to normalize this, the Fresnel-Kirchoff diffraction parameter

v is used, which is expressed as (Abhijit, 2009)

$$v = h\sqrt{2(d_1 + d_2)/\lambda d_1d_2} + \sqrt{2d_1d_2/\lambda(d_1 + d_2)} \tag{2.11}$$

Therefore, the phase difference becomes

$$\varphi = \pi v^2/2 \tag{2.12}$$

From this, it can be observed that (Abhijit, 2009):

(i) Phase difference is a function of the height of the obstruction, and also,

(ii) Phase difference is a function of the position of the obstruction from transmitter and receiver.

b) Diffraction Losses

As described in Popescu (2003), the estimation of signal attenuation caused by radio wave diffraction of radio over hills and buildings is key to predicting the field strength in a given service area. Generally, it is impossible to make very precise estimates of the diffraction losses, and in practice prediction is based on approximations using empirical corrections. The calculation of diffraction losses over complex and irregular terrains is quite a difficult task. However, expressions for diffraction losses for many simple cases have been derived.

Consider an opaque screen positioned between receiver and transmitter. It will have little effect when the top of the screen is well below the line-of-sight (LOS). The field at the receiver will then be equivalent to the free space value E_0. As the height of the screen is increased, blocking more of the Fresnel zones below the LOS path, this field strength will begin to oscillate, with the amplitude of the oscillations increasing until the obstruction edge is just in line with Transmitter and Receiver at which point the field strength is exactly half the unobstructed value i.e. the loss is 6 dB. As the height is increased above this value the oscillations cease and the field strength decrease steadily. In order to express this in a quantitative way, any obstruction between transmitter and receiver is replaced by an absorbing plane placed at the same position

and the classical diffraction theory is used. The plane is normal to the direct path and extends to infinity in all directions except vertically where it stops at the height of the original obstruction.

Considering a receiver point R, located in the shadow region (also called the diffraction zone) the field strength at receiver is determined as the sum of all secondary Huygens sources in the plane above the obstruction and can be expressed as (Popescu, 2003),

$$\frac{E}{E_0} = F(v) = \frac{1+j}{2}\int_v^\infty \exp\left(-j\frac{\pi}{2}t^2\right)dt \tag{2.13}$$

where F(v) is known as the complex Fresnel integral with v being the value given by equation (2.11) for the height of the obstruction under consideration. If the obstruction lies below the LOS then h and v are negative. If the path is actually obstructed then h and v are positive.

In practice, the Fresnel integral is commonly evaluated using tables or graphs for given values of v. The diffraction gain due to the presence of a knife-edge as compared to the free space E_0, is given by (Popescu, 2003),

$$G_d[dB] = 20\log\ |\mathrm{F}(v)| \tag{2.14}$$

In practice, graphical and numerical solutions are relied upon to compute diffraction gain. The approximate solution is expressed as follows (Popescu, 2003),:

$$\mathrm{L}(v) = 0 \qquad v \leq -1 \tag{2.15}$$

$$\mathrm{L}(v) = 20\log(0.5 - 0.62\,v) \qquad -1 \leq v \leq 0 \tag{2.16}$$

$$\mathrm{L}(v) = 20\log(0.5\exp(-0.95\,v)) \qquad 0 \leq v \leq 1 \tag{2.17}$$

$$\mathrm{L}(v) = 20\log(0.4 - \sqrt{(0.1184 - (0.38 - 0.1\,v))^2} \qquad 1 \leq v \leq 2.4 \tag{2.18}$$

$$\mathrm{L}(v) = 20\log(0.225/v) \qquad v > 2.4 \tag{2.19}$$

2.2.1.3 Scattering

Scattering occurs when the medium through which the wave travels consists of objects whose dimensions are smaller than the wavelength of the waves (Tapan et al, 2003). Scattering is similar to diffraction, except that the radio waves are scattered in a greater number of directions compared with diffraction. Scattered waves are produced by small objects or by other irregularities in the environment. In practice, street signs, street-lamps and even trees can cause scattering. Of all the mentioned effects, scattering is the most difficult to predict. An illustration of scattering is indicated in Figure 2.3.

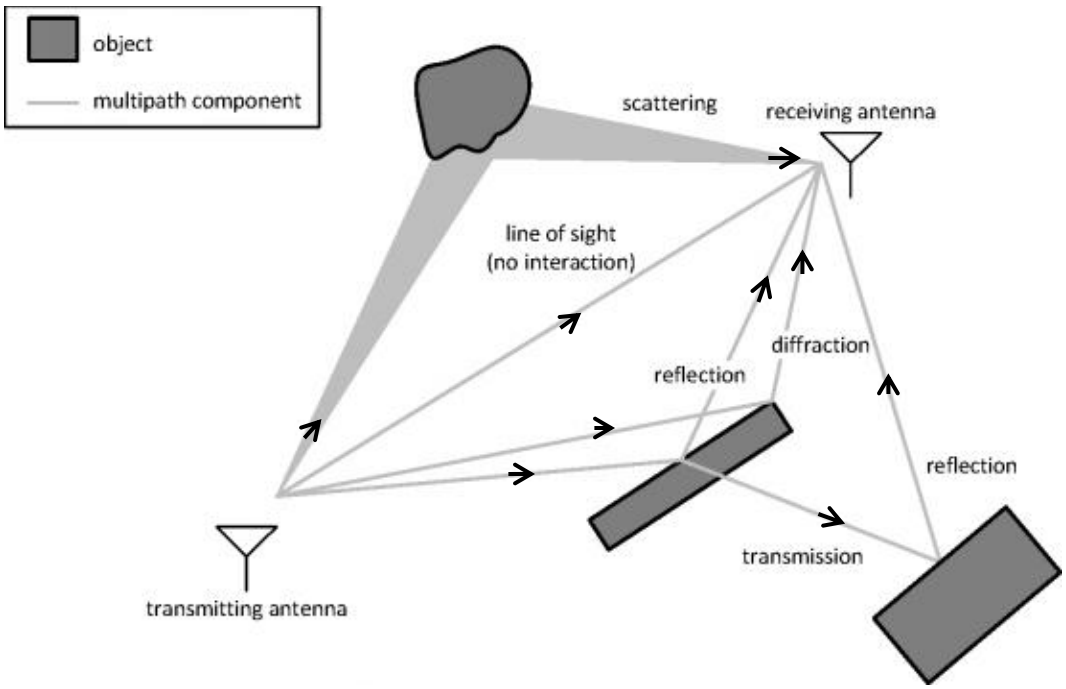


Figure 2.3: Multipath propagation (Wireless and Cable, 2014)

2.2.1.4 Reflection

Reflection occurs when a radio wave strikes an object whose dimensions are big compared with the wavelength of the wave (Li Qing, 2005). Reflection can decrease the signal strength if part of it is absorbed by the reflecting surface. The signal level that reaches the receiving antenna is

the sum of all the components of the signal transmitted by the transmitting antenna. At locations where many reflected waves exist, the received signal level tends to fluctuate. This phenomenon is commonly referred to as multipath fading. An illustration of reflection is indicated in Figure 2.3.

2.2.1.5 Transmission

Transmission occurs when the radio wave encounters an obstacle that is to some extent transparent to the radio waves, like a glass window is to visible light (Li Qing, 2005). This mechanism allows the reception of radio signals inside buildings in cases in which the actual transmitter locations are either outdoors or indoors. An illustration of transmission is indicated in Figure 2.3.

2.2.1.6 Refraction

Refraction refers to the change in direction (i.e. deviation from original path) of a wave as it moves between atmospheres of different densities. It is very important in a macro-cell (which usually consists of many smaller cells) radio system design. Due to an inconstant refractive index of the atmosphere, radio waves do not propagate along a straight line, but rather along a curved one. Therefore, the coverage area of an actual transmitter is usually larger than that predicted by LOS.

2.2.1.7 Multipath Propagation and Fading

Multipath propagation is a propagation phenomenon by which radio signals arrive at the receiving antenna by two or more paths (Abhijit, 2009). The radio waves arrive at the mobile receiver from different directions with different amplitudes, phases and time delays (Tapan et al, 2003). As illustrated in Figure 2.3, causes of multipath propagation include reflections, refractions, scattering, diffraction, atmospheric ducting, reflections from water bodies and terrestrial objects such as mountains and buildings (Abhijit, 2009). The effects of multipath include constructive and destructive interference, and phase shifting of the signal. Destructive interference causes fading. In addition, multipath fading can cause periodic fluctuations in signal strength. Because of wireless challenges such as multipath fading, wireless receivers must be tested through a combination of channel emulation and drive testing in the final environment.

As described in (Abhijit, 2009), a mobile radio channel may be modeled as a linear filter with time varying impulse response in continuous time. The channel impulse response can be expressed as h(d,t). Let x(t) represent the transmitted signal, then the received signal y(d,t) at position d can be expressed as (Abhijit, 2009)

$$y(d,t) = x(t) \otimes h(d,t) = \int_{-\infty}^{\infty} x(\tau)\, h(d,\ t - \tau)\, d\tau \qquad (2.20)$$

where $\otimes$ is the tensor product.

For a causal system (Abhijit, 2009)

$$y(d,t) = \int_{-\infty}^{t} x(\tau)\, h(d,\ t - \tau)\, d\tau \qquad (2.21)$$

Since the receiver moves along the ground at a constant velocity v, the position of the receiver is d = vt, i.e., the position of the receiver can be expressed as d = vt.

That gives (Abhijit, 2009),

$$y(vt,t) = \int_{-\infty}^{t} x(\tau)\, h(d, t - \tau)\, d\tau \quad (2.22)$$

Since v is a constant, y(vt, t) is just a function of t. Therefore the above equation can be expressed as (Abhijit, 2009),

$$y(t) = \int_{-\infty}^{t} x(\tau)\, h(d,\ t - \tau)\, d\tau = x(t)*h(vt, t) = x(t)*h(d, t) \quad (2.23)$$

In general, the channel impulse response can be expressed

i) t : time variation due to motion

ii) τ : channel multipath delay for a fixed value of t.

It is useful to discretize the multipath delay axis τ of the impulse response into equal time delay segments called excess delay bins, each bin having a time delay width equal to (Abhijit, 2009),

$$(\tau_{i+1} - \tau_i) = \Delta\tau \ \text{ for } i \in \{0, 1, 2, \dots, N-1\} \quad (2.24)$$

where N represents the total number of possible equally-spaced multipath components, including the first arriving component. The useful frequency span of the model is $2/\Delta\tau$. The model may be used to analyze transmitted RF signals having bandwidth less than $2/\Delta\tau$.

If there are N multi-paths, maximum excess delay is given by $N\Delta\tau$ (Abhijit, 2009), .

$$\{y(t) = x(t)*h(t, \tau_i) |\ i=0, 1, \dots, N-1\} \quad (2.25)$$

Bandpass channel impulse response model is (Abhijit, 2009),

$$x(t) \rightarrow h(t,\tau) = Re\{h_b(t,\tau)e^{j\omega_c t}\} \rightarrow y(t) = Re\{r(t)e^{j\omega_c t}\} \quad (2.26)$$

Baseband equivalent channel impulse response model is given by (Abhijit, 2009),

$$c(t) \rightarrow 1/2h_b(t,\tau) \rightarrow r(t) = c(t) * 1/2h_b(t,\tau) \quad (2.27)$$

Average power is (Abhijit, 2009),

$$\overline{x(t)^2} = 1/2(|c(t)|^2) \quad (2.28)$$

The baseband impulse response of a multipath channel can be expressed as (Abhijit, 2009),

$$h_b(t,\tau) = \sum_{i=0}^{N-1} a_i(t,\tau)\exp\left[j(2\pi f_c \tau_i(t) + \varphi_i(t,\tau))\right]\delta(\tau - \tau_i(t)) \quad (2.29)$$

where $a_i(t,\tau)$ and $\tau_i(t)$ are the real amplitudes and excess delays, respectively, of the i-th multipath component at time t. The phase term $2\pi f_c \tau_i(t) + \varphi_i(t,\tau)$in the above equation represents the phase shift due to free space propagation of the i-th multipath component, plus any additional phase shifts which are encountered in the channel.

If the channel impulse response is wide sense stationary over a small-scale time or distance interval, then (Abhijit, 2009)

$$h_b(\tau) = \sum_{i=0}^{N-1} a_i \exp\left[j\theta_i\right]\delta(\tau - \tau_i) \quad (2.30)$$

For measuring $h_b(\tau)$, we use a probing pulse to approximate $\delta(t)$ i.e., (Abhijit, 2009),

$$p(t) \approx \delta(t - \tau) \quad (2.31)$$

Power delay profile is taken by spatial average of $|h_b(t,\tau)|^2$over a local area. The received power delay profile in a local area is given by (Abhijit, 2009)

$$p(t) \approx k|h_b(t;\tau)|^2 \quad (2.32)$$

a) Fading

As described in Abhijit (2009), the term fading, or, small-scale fading, means rapid fluctuations of the amplitudes, phases, or multipath delays of a radio signal over a short period or short travel distance. This might be so severe that large scale radio propagation loss effects might be ignored.

b) Multipath Fading Effects

In principle, the following are the main multipath effects (Abhijit, 2009):

i) Rapid changes in signal strength over a small travel distance or time interval.

ii) Random frequency modulation due to varying Doppler shifts on different multipath signals.

iii) Time dispersion or echoes caused by multipath propagation delays.

c) Types of Small-Scale Fading

As described in Abhijit (2009), the type of fading experienced by the signal through a mobile channel depends on the relation between the signal parameters (bandwidth, symbol period) and the channel parameters (rms delay spread and Doppler spread). There are two types of fading due to the time dispersive nature of the channel.

d) Fading Effects due to Multipath Time Delay Spread

The types of fading Effects due to Multipath Time Delay Spread include flat fading and frequency selective fading (Abhijit, 2009):.

Flat Fading: this occurs when the signal bandwidth (B_S) of the transmitted signal is much less than the coherence bandwidth (B_C) of the channel, i.e., $B_S \ll B_C$. Likewise, flat fading occurs if the symbol period of the signal (T_S) is much more than the rms delay spread of the channel (σ_τ), i.e., $T_S \gg \sigma_\tau$. When flat fading occurs, the mobile channel has a constant gain and linear phase response over its bandwidth.

Frequency selective fading: This occurs when $B_S \gg B_C$ and $T_S \ll \sigma_\tau$. In such situations multiple copies of the transmitted signal arrive at the receiver, all attenuated and delayed in time, thereby causing the channel to introduce inter symbol interference.

e) Fading Effects due to Doppler Spread

The types of fading Effects due to Doppler Spread include fast fading and slow fading (Abhijit, 2009):

Fast Fading: In a fast fading channel, the channel impulse response changes rapidly within the symbol duration of the signal. Due to Doppler spreading, the signal undergoes frequency dispersion leading to distortion. Therefore a signal undergoes fast fading if $T_S \gg T_C$ where T_C is the coherence time and $B_S \gg B_D$ where B_D is the Doppler spread. Transmissions involving very low data rates suffer from fast fading.

Slow Fading: In such a channel, the rate of the change of the channel impulse response is much less than the transmitted signal. A channel can be considered slow faded if the channel impulse response is almost constant over at least one symbol duration. Hence $T_S \ll T_C$ and $B_S \gg B_D$. Thus, the velocity of the user plays an important role in deciding whether the signal experiences fast or slow fading.

2.2.1.8 Absorption

Absorption occurs when radio waves are transmitted from one medium to another, with a resultant loss of energy (Subcourse Number SS0130, 2005). For example, if a radio signal is propagated through trees the foliage will absorb some of the energy of the signal. A receiving antenna should be placed in the best position possible in order to ensure greater absorption of incoming electromagnetic energy.

2.2.2 Radio Propagation Models

In cellular communication systems radio waves are used to convey information between transmitting and receiving antennas. However, it is a well known fact that as radio signals travel from transmitter to receiver, they undergo attenuation. As a result, one of the greatest challenges in establishing a cellular network is the determination of radio propagation characteristics of the terrain. For this purpose, radio propagation models are used. A radio propagation model, also known as the Radio Wave Propagation Model or the Radio Frequency Propagation Model, is a mathematical formulation for the characterization of radio wave propagation as a function of frequency, distance and other conditions (Alim et al., 2010). Radio propagation models do not point out the exact behaviour of a link, rather, they predict the most likely behaviour the link may exhibit under the specified conditions. Propagation models can be empirical, deterministic or stochastic.

2.2.2.1 Deterministic Models

The deterministic models (Sumit and Vishal, 2012) make use of the laws governing electromagnetic wave propagation to determine the received signal power at a particular location. The field strength is calculated using the Geometrical Theory of Diffraction (GTD) as a component comprising of direct, reflected and diffracted rays at the required position. Deterministic models often require a complete 3-D map of the propagation environment. An example of a deterministic model is ray tracing model (Govind and Sonika, 2014). Semi-deterministic models are half-way between deterministic and empirical models (Piacentini and Rinaldi, 2010). Deterministic ones, on the other hand, are more accurate but require detailed information about the environment and more computational effort.

2.2.2.2 Stochastic Models

Stochastic models are used for modelling an environment as a series of random variables. Although these models are the least accurate, they require the least information about the environment and use much less processing power to generate predictions (Abhayawardhana, 2005).

2.2.2.3 Empirical Models

Empirical models are those models that are formulated based on observations and measurements alone (Akpado et al, 2013). They are mathematical formulations used to predict radio propagation behaviour across a terrain. Although empirical models are easier to implement and usually require less computational effort, they are less sensitive to the environment. Empirical models can be split into two subcategories namely, time dispersive and non-time dispersive (Abhayawardhana, 2005). Time dispersive models provide information on the time dispersive characteristics of a channel, such as the channel's delay spread during multipath propagation. An example of this type is the Stanford University Interim (SUI) (Sami, 2013). Examples of non-time dispersive empirical models are ITU-R (Sami, 2013), Hata Model (Yuvraj, 2012) and the COST-231 Hata model (Chhaya Dalela, 2013). All these models predict mean path loss as a depending of various parameters such as distance, antenna heights etc.

a) The Okumura Model

The Okumura model (Yuvraj, 2012; Obot et al., 2011) was formulated based on measurements obtained from the city of Tokyo, Japan. It is one of the most widely used models for path loss prediction across various terrain types, classified as urban, suburban, quasi-open area and open areas. This model is applicable for frequencies in the range 150 MHz to 1920 MHz (although it

is typically extrapolated up to 3000 MHz) and distances of 1 km to 100 km. The Okumura model path loss equation is given by (Yuvraj, 2012):

$$L = L_{FSL} - A_{MU} - H_{MG} - H_{BG} - G_{AREA} \quad (2.33)$$

where,

L - Median path loss in Decibels (dB)

L_{FSL}- Free Space Loss in Decibels (dB)

A_{MU}- Median attenuation in Decibels (dB)

H_{BG}- Base Station antenna height gain factor given by $20\log(h_b/200)$ for $30m<h_b<100m$

H_{MG}- Mobile Station antenna height gain factor given by $10\log(h_m/3)$ for $h_m<3m$

G_{AREA}- Gain due to type of environment

b) The Hata-Okumura Model

The Hata-Okumura Model (Yuvraj, 2012; Isabona and Konyeha, 2013) incorporates the graphical information from the Okumura Model. The Hata Model for Urban Areas is a widely used propagation model for predicting path loss in urban areas, and also has formulations for predicting path loss in Suburban and Open Areas. The Hata Model for Urban Areas is valid for the following parameters:

Frequency Range: 150 MHz to 1500 MHz

Transmitter Height: 30 m to 200 m

Link distance: 1 km to 20 km

Mobile Station (MS) height: 1 m to 10 m

Hata Model for Urban Areas is formulated as (Neskovic, 2002)

$$L_U = 69.55 + 26.16logf - 13.82logh_B - C_H + (44.9 - 6.55logh_B)logd \qquad (2.34)$$

For small or medium sized cities (where the mobile antenna height is not more than 10 meters),

$C_H = 0.8 + (1.1logf - 0.7)h_M - 1.56logf$ and for large cities,

$$C_H \begin{cases} 8.29(log(1.54h_M))^2 - 1.1\,, & for\ 150MHz \le f \le 200MHz \\ 3.2(log\,(11.75h_M\,))^2 - 4.97\,, & for\ 200MHz \le f \le 1500MHz \end{cases}$$

where,

L_U - Path loss in Urban Areas

h_B - Height of base station antenna in meters (m)

h_M - Height of mobile station antenna in meters (m)

f - Frequency of Transmission in megahertz (MHz).

C_H - Antenna height correction factor

D - Distance between the base and mobile stations in kilometres (km).

The Hata Model for Suburban Areas is given by (Neskovic, 2002)

$$L_{SU} = L_U - 2(log\frac{f}{28})^2 - 5.4 \qquad (2.35)$$

Hata model for open areas is formulated as (Neskovic, 2002)

$$L_O = L_U - 4.78(logf\,)^2 + 18.33logf - 40.94 \qquad (2.36)$$

c) The Cost 231 Hata Model

The COST 231 Hata (Purnima, 2010; Mardeni, 2010) Model is an extension of the Hata Model, which is also an extension of the Okumura Model. It was formulated to suit the European environments taking into consideration a wide range of frequencies (150MHz to 2000MHz). The COST 231 Hata Model is one of the most widely used radio propagation models because of

suitability for urban, semi-urban, suburban and rural areas. The COST 231 Hata Model is given by (Purnima, 2010)

$$L = 46.3 + 33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd + C \quad (2.37)$$

where,

L - Median path loss in Decibels (dB)

C is 0dB for medium cities and suburban areas

C is 3dB for metropolitan areas

f - Frequency of Transmission in Megahertz (MHz)(150MHz to 2000MHz)

h_B - Base Station Antenna effective height in Meters (30m to 100m)

d - Link distance in Kilometers (km) (up to 20kilometers)

h_R - Mobile Station Antenna effective height in Meters (m) (1 to 10metres)

$a(h_R)$ - Mobile station Antenna height correction factor as described in the Hata Model for Urban Areas.

For urban areas, $a(h_R) = 3.20(\log10(11.75hr))2-4.97$, for f >400 MHz

For sub-urban and rural areas, $a(h_R) = (1.1\log(f) - 0.7)h_R - 1.56\log(f) -0.8$

d) The Cost 231 Walfisch-Ikegami Model

As described in Chhaya Dalela (2013), this empirical propagation model was created on the bases of the models from J. Walfisch and F. Ikegami (Walfisch and Bertoni, 1988) and further developed by the COST 231 project. Now referred to as the Empirical COST-Walfisch-Ikegami Model, it was developed and used in Europe. The model has high prediction accuracy in urban environments because it considers multiple diffraction losses over rooftops of buildings in the

vertical plane between the Base and Mobile Stations. However, the model does not take into account path loss due to multiple reflections. The Model is valid for the following parameters:

Frequency Range: 800 MHz to 2000 MHz

Transmitter Height (h_b): 4m to 50 m

Link distance: 0.02km to 5km

Mobile Station (MS) height (h_m): 1m to 3m

Mean height of buildings (h_{roof})

Mean Street Width (w)

Mean building separation (b)

The Line of Sight (LOS) path loss equation is given by (Mardeni and Kwan, 2010):

$$PL = 42.64 + 20logf + 26logd \tag{2.38}$$

However, when there is No Line of Sight (NLOS) the equation is (Mardeni and Kwan, 2010)

$$PL = L_{FS} + L_{RTS} + L_{MSD} \tag{2.39}$$

Where,

L_{FS} is free-space path loss and is expressed as:

$$L_{FS} = 32.45 + 20logf + 20logd \tag{2.40}$$

L_{RTS} is path loss due to rooftop to street diffraction and is expressed as:

$$L_{RTS} = -16.9 - 10logw + 10logf + 20\,log(h_b - h_m) + L_{ori} \tag{2.41}$$

L_{ori} in (2.41) is path loss due to orientation angle φ (in degrees), between incident wave and street. It is expressed as:

$$L_{ori} = \begin{cases} -10 + 0.354\varphi for \;\; 0 \le \varphi < 35 \\ 2.5 + 0.075(\varphi - 35) for \;\; 35 \le \varphi < 55 \\ 4 - 0.114(\varphi - 55) for \;\; 55 \le \varphi < 90 \end{cases} \tag{2.42}$$

L_{MSD} is path loss due to multi-screen diffraction, and is expressed as:

$$L_{MSD} = L_{BSH} + k_a + k_d log d + k_f log f - 9 \log b \quad (2.43)$$

Where,

$$L_{BSH} = \begin{cases} -18\log(1 + h_b - h_{roof}) & for h_b > h_{roof} \\ 0 & for h_b \le h_{roof} \end{cases}$$

$$k_a = \begin{cases} 54 & for h_b > h_{roof} \\ 54 - 0.8(h_b - h_{roof}) & for d \ge 0.5km \text{ and } h_b \le h_{roof} \\ 54 - \dfrac{0.8(h_b - h_{roof})}{0.5} & for d < 0.5km \text{ and } h_b \le h_{roof} \end{cases}$$

$$k_d = \begin{cases} 18 & for h_b > h_{roof} \\ 18 - 15(h_b - h_{roof}) & for h_b \le h_{roof} \end{cases}$$

$$k_f = \begin{cases} -4 + 0.7\left(\dfrac{f}{925} - 1\right) & for \text{ medium size city and suburban area} \\ -4 + 1.5\left(\dfrac{f}{925} - 1\right) & for \text{ metropolitan area } (i.e. \text{ large city}) \end{cases}$$

2.2.3 Soft Computing

As described in Jang et al., (1997), Soft Computing was defined by Lofti Zadeh as an emerging approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. Soft Computing is a term that encompasses a collection of computing techniques (Santosh, 2014). Zadeh described Soft Computing as a multidisciplinary system that represents a fusion of the fields of Neural Networks, Fuzzy Logic, Probabilistic Computing, Evolutionary and Genetic Computing. Soft Computing is an integration of techniques that were designed to model solutions to real world problems, which have never been modeled or too difficult to model mathematically. Soft Computing is a fusion of techniques that works with great synergy, providing flexible information processing capability for handling real-life ambiguous situations. Its aim is to

exploit the tolerance for imprecision, uncertainty, approximate reasoning and partial truth in order to achieve tractability, robustness and low-cost solutions (Santosh et al, 2014). The basic concept on which soft computing is based, has to do with creating synergic computation techniques that lead to an acceptable solution to a problem. Soft computing differs from conventional (hard) computing in that, unlike hard computing, it can accommodate imprecision, uncertainty, partial truth and approximation. In a nut shell, Soft Computing is basically a collection of optimization techniques which can be used to find solutions to complex problems.

Soft computing techniques have been applied to many real-world problems in attempts to find acceptable solutions. These techniques have been implemented in signal processing, pattern recognition, quality assurance and industrial inspection, business forecasting, speech processing, credit rating, adaptive process control, robotics control, natural language understanding, etc. Possible new application areas are programming languages, user-friendly application interfaces, automaticized programming, computer networks, database management, fault diagnostics and information security (Santosh et al, 2014).

Hybrid systems have been created by combining different soft computing techniques in order to explore the properties of the different techniques. A typical example is the Adaptive Neuro-Fuzzy Inference System (ANFIS).

ANFIS was first proposed by Jang (1993) to combine the learning ability of NNs with the ability of fuzzy systems to interpret imprecise information, and it was based on the first-order Takagi–Sugeno–Kang (TSK) model. An ANFIS is a combination of ANN and FIS (Fuzzy Inference System) to form an intelligent adaptive system capable of solving complex non-linear problems. ANNs are quite useful in modelling systems where there is no mathematical relationship between input and output patterns. This stems from the fact that, as systems that

mimic the human brain, ANNs can be trained using input patterns and target output, and then used to predict a result given new set of inputs. Based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning, FIS is a computational network capable of modelling human knowledge and reasoning.

2.2.3.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a mathematical model that tries to simulate the structure and functionalities of biological neural networks (Andrej et al, 2011). The human brain is considered a highly complex, nonlinear, and parallel computer (information processing system) (George and Michael, 2006). The Artificial neural network is basically a system of interconnected artificial neurons that mimic the human brain to form a complex programming structure for neural processing (organization and learning). Each neuron has thousands of connections with other neurons, constantly receiving incoming signals to reach the cell body. Each neuron is created in such a way that, if the resulting sum of the signals surpasses a certain threshold, a response is sent through the axon. ANNs have become a popular technique for the application in most of the data mining fields including classification, forecasting, functional approximation, rule extraction, pattern recognition and medical applications (Janmenjoy et al, 2015). Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques.

As described in Faria et al (2009), neural networks can learn to approximate any function to a given accuracy and behave like associative memories by using just example data that is representative of the desired task. ANNs are model-free estimators capable of solving complex

problems based on the presentation of a large numbers of training data. As such, They have a key advantage over traditional approaches to function estimation such as the statistical methods. Neural networks have the ability to estimate a function in the absence of any mathematical formulation of how the outputs are derived from inputs. Hence, a key distinguishing feature of ANNs is their ability to adapt and provide acceptable solutions to problems.

2.2.3.2 The Fundamental Unit of Neural Networks

The neuron is the information processing unit of any ANN. The neuron model in Figure 2.4 shows that the neuron has the following three basic elements:

(i) Inputs (i.e. synapses or connecting links). Each input has an associated weight.

(ii) An adder (or linear combiner) for summing up weighed inputs.

(iii) An activation (or transfer) function that limits the amplitude of the output signal.

Mathematically, the neuron can be defined by the following equation (Andrej et al, 2011):

$$Y = \sum_{i=1}^{N} X_i W_i + b \qquad (2.44)$$

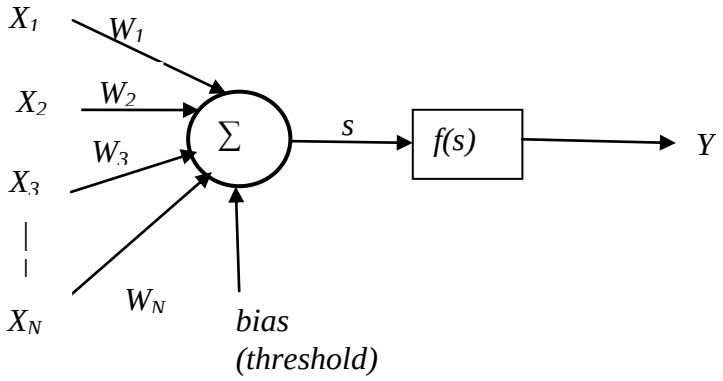


Figure 2.4: The Neuron Architecture

The inputs X_i and weights W_i are vectors while the bias b, is a scalar quantity. Biases are values that are added to the sums calculated at each node (except input nodes) during the feed-forward

phase. That is, the bias associated with a particular node is added to the weighted sum, prior to the use of the activation function at that same node. Therefore, the neuron can be described as a processing element that receives a number of inputs, obtains the weighted sum of the inputs together with an additional scalar bias parameter, and then uses the result as argument for a single valued function (svf) called the activation function, f(s) (Abraham et al., 2013).

2.2.3.3 Activation Functions

The term activation function is used to refer to the function, f that converts the net input value (net) to the node's output value (Nordstrom and Svensson, 1998). In other words, an activation function is a function that determines the output of an artificial neuron given a set of inputs. The activation functions commonly used in neural networks include the unit step (or threshold), sigmoid and Gaussian (Fadzilah et al, 2012).

a) Unit step (threshold)

This activation function sets the output at one of two levels, depending on the magnitude of the total input compared to some threshold value. As shown in Figure 2.5, the output is 1 if a given threshold is attained, else 0. It is often used in the last layer of neural networks. It is also used in Perceptrons, to create neurons that make classification decisions.

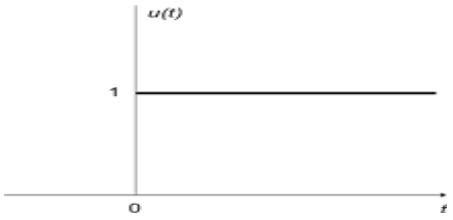


Figure 2.5: Unit step (threshold) function

b) Sigmoid

The sigmoid as in Figure 2.6 is used in processes that progress from initial minimal values to a climax with time. A typical example of such a process is the learning process of a neuron. The sigmoid function is a non-linear activation function used in back-propagation multi-layer networks. The Sigmoid activation function is of two types: logistic and tangential. The logistic has a range 0 and 1 and the tangential has values ranging -1 to +1.

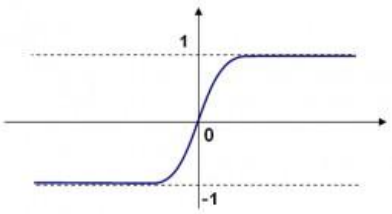


Figure 2.6: Sigmoid function

c) Gaussian

The Gaussian function is a continuous bell-shaped curve as shown in Figure 2.7. The node output (high/low) is interpreted in terms of class membership (1/0), depending on how close the net input is to a chosen value of average. Such functions are used in Radial Basis Function Networks.

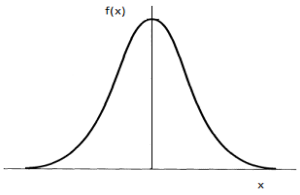


Figure 2.7: Gaussian function

2.2.3.4 Learning

The ability to learn is associated with intelligence. Just like the human brain can be trained to perform a certain function, ANNs can be trained to solve a problem. As described in (Neha, 2013), a learning can be described as a process of updating network architecture and connection weights so that a network can efficiently perform a specific task. The network usually learns the

connection weights from available training patterns. As connection weights are updated repeatedly, network performance is improved over time. A key advantage of neural networks over traditional expert systems is that ANNs have the ability to learn underlying rules from available data instead of following a set of rules created by individuals. A learning algorithm is a procedure that determines how weights are continuously adjusted as dictated by the set of rules. Learning algorithms are aimed at obtaining optimal set of weights for a given problem. The task of obtaining optimal weights is dependent on any of the following:

1. Number of iterations (or computation time).
2. Network error minimization.
3. Generalization ability of the network.

For every learning rule there is a variety of learning algorithms. Most algorithms can only be used with a single learning rule. Therefore, training can take on many different forms, using a combination of learning paradigms, learning rules, and learning algorithms. After the training phase, an ANN goes into the production phase. An ANN is considered static if it has distinct learning and production phases, whereas ANNs that continue learning during the production phase are deemed dynamic.

As described in (Neha, 2013), the three major learning paradigms, each corresponding to a particular abstract learning task include supervised learning, unsupervised learning and reinforcement learning. In supervised learning the network is provided with the desired output for every input pattern. Weights are adjusted to enable the network produce outputs that are close enough to the desired outputs. A supervised learning process is shown in Figure 2.8. Unlike supervised learning, unsupervised learning does not require desired output to any corresponding input pattern available in the training data set. Rather, it explores the underlying

structure in the data, or correlations between the input patterns in the data, and then organizes these patterns into categories from thee correlations.

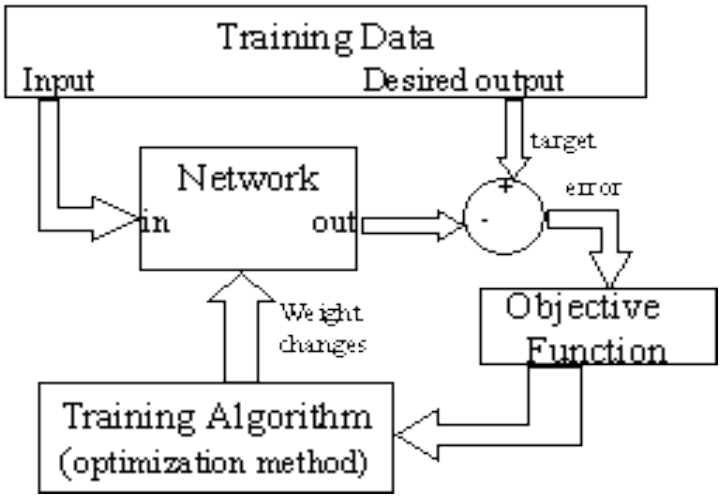


Figure 2.8: Supervised learning model
(http://www.yukool.com/nn/fundamental.htm#_Toc499362382, May, 2014**)**

Hybrid learning incorporates supervised and unsupervised learning paradigms. Parts of the weights are usually determined through supervised learning, while the others are obtained through unsupervised learning. Reinforcement learning is a type of supervised learning in which the network is provided with only a critique on the desired outputs, not the desired output itself. The BFGS quasi-Newton method (trainbfg), the conjugate gradient back propagation (traincgb) with Powell-Beale restarts and the Levenberg-Marquardt back propagation algorithm are typical examples of supervised learning methods.

2.2.3.5 Training Algorithms

As described in (Neha, 2013), the neurons in feed forward ANNs are organized in layers, hence they send their signals "forward", while errors are propagated backwards. The back propagation algorithm is used in such layered feed-forward ANNs. These networks receive inputs through input layer neurons, process the inputs using one or more intermediate hidden layers, and then

give out output through output layer neurons. The back propagation algorithm is the most common training algorithm for adjusting weights during the training phase on ANNs. Weights can be adjusted using the gradient descent method after computing the error between the network output and desired output. For practical reasons, ANNs that implement the back-propagation algorithm do not have too many layers, since the time for training the networks grows exponentially.

The initial structure of a neural network is such that it is not likely to produce the desired output. The errors generated for all input patterns are propagated backwards, from the output layer towards the input layer. The weights are adjusted in order to minimize the errors between actual and desired outputs.

According to Menaka (2014), the implementation of the back propagation algorithm is described in each training iteration as follows:

1. A particular case of training data is fed through the network in a forward direction, producing results at the output layer.
2. Error is calculated at the output nodes based on known target information, and the necessary changes to the weights that lead into the output layer are determined based upon this error calculation, in order to minimize the error
3. The changes to the weights that lead to the preceding network layers are determined as a function of the properties of the neurons to which they directly connect until all necessary weight changes are calculated for the entire network.

2.2.3.6 Artificial Neural Network Architectures

The architecture of an ANN plays an important role in attaining desired performance. Depending on pattern of connections, ANNs are usually classified into two major groups: feed-forward networks and recurrent (or feed-back) networks. As described by Padmapriya (2015), a feed-forward network is a non-recurrent network which contains an input layer, one or more hidden layers, and an output layer. In such networks, signals can only travel in the forward direction while errors propagate in the opposite direction. As input data moves from one layer to the other, each processing unit performs computation based on a weighted sum of its inputs, and the new values then serve as input to the next layer, until an output is produced at the output layer. A threshold transfer function is sometimes used to quantify the output of a neuron in the output layer. Feed-forward networks include Single-layer Perceptron or simply perceptron, the Multi-Layer Perceptron Neural Network (MLP-NN) and the Radial Basis Function networks (RBFN).

a) The Multi-Layer Perceptron Neural Network

Gaurang et al (2011) describe the MLP-NN as a supervised feed forward neural network trained with the standard back propagation algorithm. Their ability to learn how to transform input data into a desired output makes them suitable for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map. They have been shown to approximate the performance of optimal statistical classifiers in difficult problems.

Multilayer feed forward neural networks are used for nonlinear function approximation. As the name implies, a MLP-NN is a network that comprises of an input layer, one or more hidden layers and an output layer. However, according to Östlin (2004), a neural network with only one hidden layer can approximate any function with finitely many discontinuities to an arbitrary precision, provided the activation functions of the hidden units are non-linear. Problems

that require two or more hidden layers are rarely encountered in practice. Even for problems requiring more than one hidden layer theoretically, most of the time, using one hidden layer performs much better than using two hidden layers in practice (Syed, 2010). Some reasons why multiple hidden layers are not recommended are as follows:

i) Training becomes slow.

ii) Gradient becomes unstable as errors are propagated across layers, since the training algorithms for the MLP-NN are gradient based.

iii) As the number of hidden layers increases, the number of local minima also increases, thereby introducing a possibility of optimization algorithm missing the global minima.

However, there are certain situations that demand the implementation of multiple hidden layers. A typical example is a function with infinite discontinuities.

When designing a MLP-NN, major consideration is the number of neurons in the hidden layer. As described in (Syed, 2010), depending on the level of complication of the data set, too few neurons can lead to an occurrence called under-fitting, while too many neurons can lead to over-fitting. The optimal number of hidden neurons depends on many factors which include the following: the numbers of input and output units, the number of training cases, the amount of noise in the targets, the complexity of the error function, the network architecture, as well as the training algorithm. In most situations, the best approach to find the optimal number of hidden units is trial and error since there is no easy way to determine the optimal number of hidden units without training using different numbers of hidden units and estimating the generalization error of each.

A neural network is said to be fully connected if every node in each layer of the network is connected to every other node in the adjacent forward layer. If, however, some of the communication links (synaptic connections) are missing from the network, we say that the network is partially connected. Figure 2.9 shows the configuration of a fully connected MLP-NN with one hidden layer and one output layer.

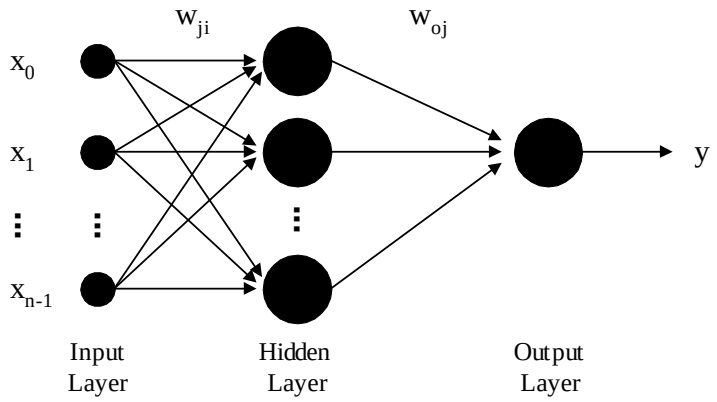


Figure 2.9: Multilayer Perceptron Neural Network with one hidden layer (Popescu et al, *2001)*

Figure 2.9 shows that each neuron of the input layer is connected to each neuron of the hidden layer, and in turn, each neuron of the hidden layer is connected to the single neuron of the output layer. As a result, signal transmission across the entire network can only be in the forward direction, i.e, from the input layer, through the hidden layer and eventually to the output layer. Signals arriving at the inputs propagate forward from neuron to neuron, until they finally arrive at the output neuron and emerge as output signals. Error signals propagate in the opposite direction from the output neuron across the network. The output of the MLP-NN is describe by the following expression: (Popescu et al, 2001)

$$y = F_0\left(\sum_{j=0}^{M} W_{0j}\left(F_h\left(\sum_{i=0}^{N} W_{ji}X_{ij}\right)\right)\right) \tag{2.45}$$

where:

$\mathbf{w_{oj}}$ represents the synaptic weights from neuron j in the hidden layer to the single output neuron

$\mathbf{x_i}$ represents the i^{th} element of the input vector,

$\mathbf{F_h}$ and F_0 are the activation function of the neurons from the hidden layer and output layer, respectively,

$\mathbf{w_{ji}}$ are the connection weights between the neurons of the hidden layer and the inputs.

During the training phase, the MLP-NN iteratively updates the weights of the system based on the mean squared error E, described by equation (2.46) (Popescu et al, 2001), between the generated output and the desired output,

$$E = \frac{1}{2}\sum_{i=1}^{m}(y_i - d_i)^2 \qquad (2.46)$$

where, y_i is the output value calculated by the network and d_i represents the desired output.

When the error between generated output and the desired output is minimized to a tolerable value, the learning process is terminated and the network can be used in a testing phase with test vectors. At this stage, the neural network said to be in a state that ensures it produces output with tolerable errors.

b) Generalised Radial Basis Function Neural Network

Radial basis functions (Powell, 1997) are used to approximate multivariable functions or data which are only known at a finite number of points or too difficult to evaluate otherwise. A radial basis function is a function whose value depends either on the distance from the origin or a given

central point c. If the value of the function depends only on the distance from the origin x, then function is given by

$$\varphi(x) = \varphi(\|x\|) \tag{2.47}$$

On the other hand, if the function value is dependent only on the distance from a point c, called the centre, then the value is given by

$$\varphi(x, c) = \varphi(\|x - c\|) \tag{2.48}$$

Summation of Radial basis functions can be used in function approximation, expressed as

$$f(x) = \sum_{i=1}^{N} w_i \, \varphi(\|x - c_i\|) \tag{2.49}$$

Where, $f(x)$ is the sum of N radial basis functions, each based on different centres c_i, with an associated weight w_i. In artificial neural networks, the summation of radial basis Functions $f(x)$, is considered a single layer radial basis function neural network, with the radial basis functions serving as activation functions.

The Generalized Radial Basis Function Neural Network (GRBF-NN) can be used to solve problems that demand the implementation of any function approximation. During the training phase, the GRBF-NN attempts to find an interpolation surface that provides a best fit to the training, within the multidimensional space (Popescu et al, 2002). Hence, ability of the network to generalization is determined through the use of this multi-dimensional surface to interpolate test data.

As described in Anwesha et al. (2012), GRBF networks have three layers: input layer, hidden layer and output layer. One neuron in the input layer corresponds to each predictor variable. Hidden layer has a variable number of neurons. Each neuron consists of a radial basis

function centred on a point with the same dimensions as the predictor variables. The output layer has a weighted sum of outputs from the hidden layer to form the network outputs.

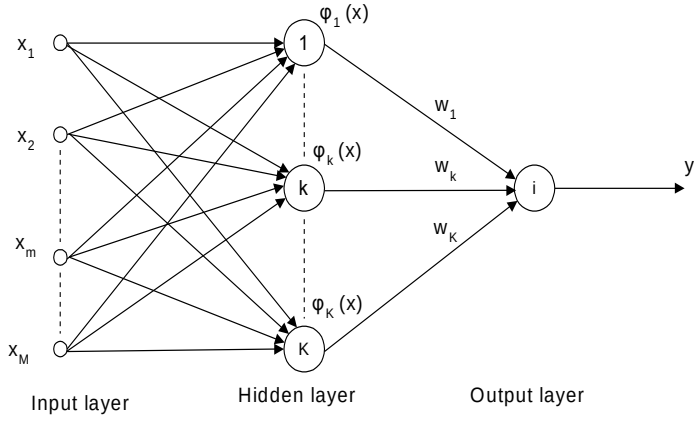


Figure 2.10: The Generalized Radial Basis Function Neural Network (Popescu et al., 2002)

Unlike the MLP, the output of hidden-nodes are not calculated using the weighted-sum activation function; rather the output of each hidden-node, φ_k is obtained by the closeness of input X to an M-dimensional parameter vector μ_k associated with the k^{th} hidden node (Popescu et al., 2002).

As shown in Figure 2.10, the GRBF–NN consists of three layers as follows:

i. The input layer
ii. The hidden layer, where input data undergoes nonlinear transformation
iii. The linear output layer, where the outputs are produced

The output of a Radial Basis Function Neural Network is given by (John, 2004)

$$Y_i(X) = \sum_{i=1}^{K} w_{ik}\varphi_k(X) \tag{2.50}$$

where,

X is the input vector

W_{ik} is the connection weight in the second layer (from hidden to output layer)

k is the number of hidden nodes

i denotes the i-th hidden node

φ_k is the radial basis activation function.

As described in Tsung-Ying et al. (2014), the radial basis function is a multi-dimensional function that describes the distance between a given input vector and a pre-defined centre vector. The Gaussian function is a type of radial basis function given by

$$\varphi_k = \exp\left(-\frac{\left\| X - \mu_k \right\|^2}{2\sigma_k^2} \right) \tag{2.51}$$

Where,

μ_k denotes the centre vector

σ_k denotes the spread (width) of the function

The training of a RBFNN is in two stages:

1. Determination of radial basis function parameters, i.e., Gaussian centre and spread width
2. Determination of output weight by supervised learning.

2.2.3.7 Artificial Neural Network Parameters

The generalization performance of an ANN depends on the following parameters (Lahiri et al, 2009):

i) Number of neurons in the hidden layers

The number of nodes in hidden layer has a profound effect on ANN performance. Too few nodes could not learn the relationship in data properly and too large number of nodes increases the network complexity and execution time. The optimal number of nodes in a hidden layer is normally calculated by the trial and error method. Such an approach, apart from consuming enormous time, may not really obtain the best possible performance.

ii) Learning Rate

This controls the size of weights and bias changes during learning.

iii) Momentum

Momentum simply adds a fraction **m** of the previous weight update to the current one. The momentum parameter is used to prevent the system from converging to a local minimum or saddle point. A high momentum parameter can also help to increase the speed of convergence of the system. However, setting the momentum parameter too high can create a risk of overshooting the minimum, which can cause the system to become unstable. A momentum coefficient that is too low cannot reliably avoid local minima, and can also slow down the training of the system.

iv) Training type

0 = train by epoch, 1 = train by minimum error

v) Epoch

This is the number of iterations. Training stops when this number is exceeded.

vi) Minimum Error

This refers to the minimum mean square error of the epoch. It is the square root of the sum of squared differences between the network targets and actual outputs divided by number of patterns (only for training by minimum error).

2.2.4 Discrete Least Squares Approximation

As described in Okonuga (2004), the concept on which least square approximation is based involves fitting a polynomial function P(x) to a set of data points (x_i, y_i) having a theoretical solution

$$y = f(x) \tag{2.52}$$

The aim is to minimize the squares of the errors, and in so doing, consider that a set of data satisfying the theoretical solution to equation (2.52) are (x_1, y_1), (x_2, y_2), . . . , (x_n, y_n). Let the polynomial to be fitted to these set of data points be denoted by P(x) or Pn(x) to denote a polynomial of degree n. The curve or line represented by P(x) is considered the best fit to f(x), if the difference between $P(x_i)$ and $f(x_i)$, i = 1, 2, . . . , n is least. That is, the sum of the differences $e_i = f(x_i) – P(x_i)$, i = 1, 2, . . . , n should be the minimum.

The sum of the differences e_i may add up to zero, thereby given the wrong error for the approximating polynomial. As such, the square of these differences are preferable. In other words, the sum of the squares of the deviations to get the best fitted curve is considered. Thus the required equation for the sum of squares error that requires minimization is then written as

$$S = \sum_{i=1}^{n}[f(x_i) - P(x_i]^2 \tag{2.53}$$

where

$$P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \tag{2.54}$$

In order to derive the discrete function that best fits the given data points, equations (2.52) and (2.54) are substituted in equation (2.53) to give

$$S = \sum_{i=1}^{n}\left[y_i - (\,a_0 + a_1 x_i + a_2 {x_i}^2 + \cdots + a_k {x_i}^k)\right]^2 \tag{2.55}$$

To minimize S, equation (2.55) is differentiated with respect to a_i and equated to zero. Therefore, differentiating equation (2.55) partially with respect to $\mathbf{a_0, a_1,\ldots, a_k}$, and equating each to zero, gives:

$$\left.\begin{aligned}
\frac{dS}{da_0} &= -2\sum_{i=1}^{n}\left[y_i - (\,a_0 + a_1 x_i + a_2 {x_i}^2 + \cdots + a_k {x_i}^k)\right] \quad = 0 \\
\frac{dS}{da_1} &= -2\sum_{i=1}^{n}\left[y_i - (\,a_0 + a_1 x_i + a_2 {x_i}^2 + \cdots + a_k {x_i}^k)\right] \quad x_i = 0 \\
\frac{dS}{da_2} &= -2\sum_{i=1}^{n}\left[y_i - (a_0 + a_1 x_i + a_2 {x_i}^2 + \cdots + a_k {x_i}^k)\right] \quad {x_i}^2 = 0 \\
&\vdots \\
\frac{dS}{da_k} &= -2\sum_{i=1}^{n}\left[y_i - (\,a_0 + a_1 x_i + a_2 {x_i}^2 + \cdots + a_k {x_i}^k)\right] \quad {x_i}^k = 0
\end{aligned}\right\} \tag{2.56}$$

The system can be rewritten as follows:

$$\left.\begin{aligned}
\sum y_i &= n a_0 + a_1 \sum x_i + a_2 \sum x_i^2 + \ldots + a_k \sum x_i^k \\
\sum x_i y_i &= a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \ldots + a_k \sum x_i^{k+1} \\
\sum x_i^2 y_i &= a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \ldots + a_k \sum x_i^{k+2} \\
&\vdots \\
\sum x_i^k y_i &= a_0 \sum x_i^k + a_1 \sum x_i^{k+1} + a_2 \sum x_i^{k+2} + \ldots + a_k \sum x_i^{2k}
\end{aligned}\right\} \tag{2.57}$$

Solving equation (2.57) to determine $\mathbf{a_0, a_1,\ldots,a_k}$, and substituting into equation (2.54) gives the best fitted curve to equation (2.52). The set of equations (2.57) are called the Normal Equations of the Least Squares Method. Equation (2.57) can only be used by creating a table of values corresponding to each sum and the sum is found for each column.

2.2.5 Types of Prediction Area

According to Sylvain Ranvier (2004), three types of prediction areas include:

i) **Open Area:** open space, no tall trees or building in path

ii) **Suburban Area:** Village Highway scattered with trees and houses and some obstacles near the mobile station but not very congested

iii) **Urban Area:** Built up city or large town with large buildings and houses, or a village with close houses and tall trees. The term "Large City" (or metropolitan Area) used by Hata is an urban area with buildings above 15meters, while "Small City" refers to an urban area with buildings below 15 meters.

2.2.6 Performance Evaluation Statistics

Some of the most widely used statistical indices for performance evaluation include the Absolute Mean Error (μ), Standard Deviation (σ), Root Mean Squared Error (RMSE) and the coefficient of determination (R^2).

2.2.6.1 Absolute Mean Error

As stated in Wackley et al., (2008), the absolute mean error is a measure that determines the closeness of predicted values to observed values or a desired outcome. The smaller the absolute mean error is, the closer the prediction. The absolute mean error is given by (Popescu et al, 2001)

$$\mu = \frac{1}{N}\sum_{i=1}^{N} E_i \tag{2.58}$$

where

- N is the total number of measured samples.
- i represents the index of the measured sample.

- E_i is the absolute error between the target output and predicted output computed using the formula $E_i = |PL_{Predicted} - PL_{target}|$

2.2.6.2 Standard Deviation

Standard deviation is a quantity that determines the variation of a value from the desired value or mean value (Olasunkanmi et al., 2014). The smaller the standard deviation, the closer the set of values is to the desired value or mean. Popescu et al., (2001) obtained the standard deviation from the absolute error and the mean absolute error as:

$$\sigma = \sqrt{\frac{1}{N-1}\left(\sum_{i=1}^{N} E_i^2 - N\mu^2\right)} \tag{2.59}$$

2.2.6.3 Root Mean Squared Error (RMSE)

Root Mean Square Error (RMSE) gives a measure of the differences between values predicted by a model and values actually observed (Olasunkanmi et al., 2014). The smaller the RMSE, the more accurate the prediction is. The RMSE has been used as a standard statistical metric to measure model performance in meteorology, air quality, and climate research studies (Chai1 and Draxler, 2014). The RMSE of a model prediction with respect to the predicted variable X_{model} is defined as the square root of the mean squared error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}} \tag{2.60}$$

where X_{obs} is observed values and X_{model} is the modelled values at time/place i.

Popescu et al (2001) define the root mean square error as

$$RMSE = \sqrt{\mu^2 + \sigma^2} \tag{2.61}$$

2.2.6.4 Coefficient of Determination (Goodness of Fit)

As described in Abraham (2013), R-square (R^2) measures how successful the fit is in explaining the variation of the data- the goodness of fit. It is also called the square of the multiple correlation coefficients or the coefficient of multiple determinations and given by

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{2.62}$$

where y_i is the measured path loss, $\hat{y}_i$ is the predicted path loss and $\bar{y}_i$ is the mean of the measured path loss. R^2 can take on any value between 0 and 1, but can be negative for models without a constant, which indicates that the model is not appropriate for the data. A value closer to 1 indicates that a greater proportion of variance is accounted for by the model.

2.3 Review of Similar Works

The following are critically reviewed similar research works on the modification of existing empirical models for terrain-specific path loss prediction suitability, and also on the application of computational intelligence techniques for path loss prediction:

Wolfle and Landstorfer, (1998) developed a model for field strength prediction in indoor environments with neural networks. The predictor was an ANN based model for the prediction of electric field strength for mobile communication networks in indoor environment. In contrast to other neural prediction models a good generalization is achieved, so the prediction results are also very accurate in buildings not used for the training of the neural network. However, the

study does not take into account factors that greatly influence radio propagation in indoor environments. These include concrete penetration losses, reflection and transmission coefficients of such materials.

Popescu et al., (2001) conducted a study on field strength prediction in indoor environment with a neural model. The proposed model consists of a multilayer perceptron neural network model trained with field strength measurements conducted in the 1890 MHz frequency band. Although results of the prediction show a good agreement with the measurements, greater accuracy would have been attained if the study had taken into account the impact of angle of incidence, as well as the impact of frequency on penetration losses.

Turkan et al., (2010) adopted a Fuzzy adaptive neural network approach to path loss prediction in urban areas at GSM-900 band in which a new algorithm based ANFIS for tuning the path loss model is introduced. The performance of the path loss model which was obtained from proposed algorithm was compared to the Bertoni-Walfisch model, which is one of the best studied for propagation analysis involving buildings. This comparison was based on the mean square error between predicted and measured values. According to the indicated error criterion, the errors related to the predictions that were obtained from the algorithm were less than the errors that were obtained from the Bertoni-Walfisch Model. The propagation measurements were carried out in the 900 MHz band in the city of Istanbul, Turkey. However, the efficiency of the proposed algorithm remains to be tested at higher frequencies.

Nadir, (2011) carried out a seasonal pathloss modeling at 900MHz for suburban Oman by investigating the variation in pathloss for the months of summer and winter, between the measured and predicted values, according to the Okumura-Hata propagation model for different cells. The Okumura-Hata model was modified according to the results obtained in the investigation, hence obtaining two modified Okumura-Hata equations. The root mean square error (RMSE) was calculated between measured path loss values and those predicated on basis of Okumura-Hata model for this area. The RMSE was found to be up to 6dB. The obtained experimental data is compared and analyzed further using different cells and from the months of January, the low rain season and August, which is a high rainy season and a difference of approximately 9dB, is observed between two seasons. Although the study demonstrates the impact of humidity on radio propagation, it clearly demonstrates the shortcoming of modification by computed RMSE resulting from the high prediction errors.

Obot et al., (2011) carried out a comparative analysis of path loss prediction models for urban macrocellular environments. Specifically, three path loss prediction models namely free space, Hata and Egli were used to predict path losses. The calculated path loss values were compared with practical measured data obtained from a Visafone base station located in Uyo, Nigeria. The comparative analysis reveals that the RMSE for free space, Hata and Egli were 16.24dB, 2.37dB and 8.40dB respectively. The results showed that Hata's model is the most accurate and reliable path loss prediction model for macrocellular urban propagation environments, since its RMSE value of 2.37dB is smaller than the acceptable minimum RMSE value of 6dB for good signal propagation. However, the conclusion that the Hata model is suitable for path loss prediction

across Nigerian urban areas is unacceptable. This is as a result of terrain diversities resulting from differences in clutter, vegetation, atmospheric conditions, etc.

Shoewu and Edeko, (2011) performed an analysis of radio wave propagation in Lagos environs. Field measurements were carried out at different locations within Epe town and its environs from live radio base stations transmitting at 900MHz and 1800MHz. The respective path loss values were estimated and compared with the Okumura-Hata model for rural, suburban and urban areas. The result indicated an appreciable consistency with the empirical models except for rural areas. For rural areas, there is an appreciable deviation from the measured results. This is because the rural area in this study differs from that of Okumura. However, the conclusion that the research shows that the Okumura-Hata model for radio wave propagation is very effective for radio wave propagation pathloss prediction in sub-urban and urban areas in Western part of Nigeria, is unacceptable. This also stems from terrain diversities resulting from differences in clutter across Western Nigeria.

Ubom et al., (2011) performed a path loss characterization of Wireless Propagation for South – South region of Nigeria. Statistical path loss models derived from experimental data collected in Port Harcourt in South-South region of Nigeria from 10 existing microcells operating at 876 MHz. Variations in pathloss between the measured and the predicted values from the Okumura-Hata model were calculated by finding the root mean square errors (RMSE) to be 10.7dB and 13.4dB for the urban and suburban terrains respectively. These variations (errors) were used to modify the Okumura-Hata models for the two terrain categories. Comparing the modified Hata model with the measured values for the two categories showed a better result. However, the

developed models cannot be acceptable for path loss prediction across the entire South-South region without tests for generalization across the entire region.

Ignacio et al., (2012) demonstrated the influence of training set selection in Artificial Neural Network-Based Propagation Path Loss Prediction. The study analyzes the use of artificial neural networks (ANNs) for predicting the received power/path loss in both outdoor and indoor links. The approach followed has been a combined use of ANNs and ray-tracing, the latter allowing the identification and parameterization of the so-called dominant path. A complete description of the process for creating and training an ANN-based model is presented with special emphasis on the training process. From experimental analyses, results show that the error parameters, mean, root mean square, and standard deviation were always below 7 dB. However, in order to further demonstrate the impact of training set selection, the prediction pattern of the ANN-based model on each route should have been analysed separately through randomization of the traininig-testing or appropriate classification of traininig-testing of measured data.

Abraham, (2013) applied artificial intelligence techniques to Macro-cell path loss prediction. In that study, the adaptive neuro-fuzzy inference systems (ANFISs), multiple layer perceptron neural network (MLP-NN) and radial basis function neural network (RBF-NN) were trained using actual signal strength measurement taken at certain suburban areas of Bauchi metropolis, Nigeria. The trained networks were then used to predict propagation losses at the stated areas under differing conditions. The predictions were compared with the prediction accuracy of the popular Hata model. The two comparative approaches adopted were as follows: (i) splitting path loss data into 60% training and 40% testing, (ii) a generalization test involving training with data

from one base station and testing with data from another. It was observed that ANFIS model gave a better fit in all cases having higher R^2 values in each case and on average is more robust than MLP and RBF models as it generalizes better to a different data. This study, however, does not take into account the need for neural network validation. Part of the data should have been set aside for validation in order to further refine neural network construction.

Akingbade and Olorunnibi, (2013) conducted a study on a path loss prediction model for UHF radio waves propagation in Akure metropolis. In the study, the Friis and Okumura-Hata models were evaluated for path loss prediction accuracy through comparisons with measured path loss values obtained from three routes within the Akure metropolis. It was discovered that on each route the Okumura-Hata model gave a better prediction than the Friis. Using the prediction errors, the Okumura-Hata model was modified to suit the routes. However, the modified Okumura-Hata models were not tested for suitability for path loss prediction along other routes within the Akure metropolis, which implies that were excluded zones within the metropolis.

Likewise, **Famoriji and Olasoji, (2013)** conducted a study UHF Radio Frequency Propagation Model for Akure Metropolis. The study involved using the Friis and Okumura-Hata and to predict television broadcast signal strength in Akure Metropolis through comparisons with measured data obtained from three different routes within the metropolis. On the average, the Okumura-Hata model gave a better prediction with mean error of 21.97dB, which was used to modify the model. However, as acknowledged by the researchers, the height of 325meters used in the calculations was beyond the limit specified by Hata. Therefore, a slightly erroneous model based the analysis of three routes within an entire semi-urban area is insufficient for total

acceptability. The modified models should have been tested for greater capacity to generalize using data not encountered.

Ogbulezie et al., (2013) conducted a study on site specific measurements and propagation models for GSM in three cities in Northern Nigeria. From the measurements and analysis carried out in this work, it is shown that the classical models for path loss determination over estimate the path loss in all the cities. The standard deviations varied from 6.9dB to 45.87dB for the Okumura Hata model at 900MHz whereas for the COST231 Hata model it was from 6.51dB to 44.79dB at that same frequency. At 1800MHz the standard deviation for COST 231 ranged from 9.42dB to 24.37dB.Similarly, the mean square errors (μe) ranged from 2.18dB to 14.52dB for Okumura Hata at 900MHz. For COST 231 Hata at this frequency, it was from 2.06dB to 14.18dB. The mean square error at 1800MHz varied from 2.98dB to 14.18dB. However, the COST 231 Walfisch-Ikegami model should have been considered in the metropolitan areas. Apart from that, choosing sites assumed to represent the most common propagation characteristics of an area under study may not necessarily yield accurate results.

Joseph et al., (2014) applied artificial neural network for path loss prediction in urban macro-cellular environment. The model consists of a multilayer perceptron trained with measured data using Scaled Conjugate Gradient algorithm. A mean squared error comparison between the ANN-based model and the free space, Hata and Egli models shows that the ANN model has the least prediction error of the order of 1.68dB result. The free space, Hata and Egli models have mean squared errors of 16.24dB, 2.37dB and 8.4dB respectively. However, it was stated the ANN model generated different results each time it was trained as a result of different initial

weight and bias values, and different divisions of measured data into training, validation and test sets. In order to avoid such indiscriminate randomizations, an initial random seed should have been set using the rand( ) function. Furthermore, the Walfisch-Ikegami should have been considered due to its suitability for path loss prediction in urban environments.

Olasunkanmi et al., (2014) carried out a comparative analysis of received signal strength (RSS) Prediction Models for Radio Network Planning of GSM 900 MHz in Ilorin, Nigeria. The suitability of Okumura-Hata model, COST 231-Hata model and Standard Propagation Model for radio coverage prediction on terrains of Ilorin City, Nigeria was investigated. The predictions of Standard Propagation Model gave the minimum Root Mean Square Error (RMSE) of 5.52 dB, 12.73 dB and 18.4 dB on BS2501, BS2502 and BS2503 respectively. The deviation of the mean RSS predicted by Okumura-Hata was found to be the highest when compared with that of the actual data collected. However, with a mean RMSE value of about 12 dB across the three terrains the use of Standard Propagation Model in radio network planning at 900 MHz may not deliver a better Quality of Service (QoS). Modifying the models for greater prediction accuracy should have been considered.

Callistus et al., (2015) carried out a performance evaluation of a generalized regression neural network (GRNN) for path loss prediction model in macrocellular environment. Compared with some conventional path loss prediction models like the Free Space, Hata and Egli models, the proposed model shows superior prediction results. The GRNN model gives accurate prediction even in environments different from where the training samples were obtained. However, instead of the aforementioned empirical models, the study should have considered models that have

been proven to have high prediction accuracy in urban environments, such as the COST 231 Wlsfisch-Ikegami model, for appropriate comparison.

Akinyemi et al., (2015) conducted a study on the determination of the most suitable propagation model for mobile Communications in South-South Nigeria urban-terrain. Two Global Systems for Mobile Communications (GSM) base stations operating at 900MHz and 1800MHz bands were used for the experiment in a typical urban area. The field measurement results were compared with SUI model, Ericsson model, Friis model and Walficsh-Bertoni model for urban area. The results obtained indicate the least variation with Walficsh-Bertoni model for urban area. However, the study does not take into account the fact that it is statistically unacceptable to consider the result of path loss analysis of one Base Station within a city, as applicable to an entire state.

In view of the limitations observed from the reviewed literature, a different approach to empirical model adaptation will be adopted in this study. This stems from the fact that the use of correction factors to modify an empirical model to suit a given terrain may not provide the best possible fit for the empirical model, resulting from the fact that such correction factors only influence the constant in an empirical model expression. By implication, if the slope of the best fit curve through measured path loss points significantly differs from that of the empirical model expression, such a technique will be highly inaccurate. The Quotients Regression Technique (QRT) developed in this study provides the best possible fit for the empirical model by directly fitting it onto the best fit curve for measured path loss points. With regards to the ANN-based models, the need for neural network validation will be taken into account. This is to ensure that

the neural network construction is further refined before testing. In addition to that, each Base Station will be analysed separately through randomization of the training-validation-testing of measured data. Moreover, in order to prevent the ANN-based models from generating different results each time they are trained as a result of different initial weight and bias values, an initial random seed would be set using the rand( ) function. Finally, where applicable, terrain-suitable propagation models will be considered for appropriate comparisons.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

The development of the proposed Quotients Regression Technique (QRT) and the performance comparison of QRT adapted empirical models with empirical models adapted using the Okumura G_{AREA} and RMSE techniques, as well as with ANN based models are hereby presented.

3.2 Received Power Measurement and Path Loss Computation

From each of the terrains under investigation, received power measurements were obtained from multiple Base Stations of the mobile network service provider, Mobile Telecommunications Network (MTN), Nigeria. The instrument used was a hand-held Cellular Mobile Network Analyser (SAGEM OT 290) capable of measuring signal strength in decibel milliwatts (dBm). A description of the device is presented in Appendix C, while the measurement set-up is shown in Figure 3.1. For operator network identification, the instrument was equipped with an MTN Subscriber Identification Module (SIM) card. In the process of taking readings, the instrument was held at a height of 1.5meters above the ground. Using the Broadcast Control Channel feature, the instrument was fixed to a given frequency in order to avoid hopping.

For each of the terrains, received power (P_R) readings were obtained at specific intervals away from the Base Station, starting from the reference distance d_0, within the radiating far field (propagation region). The Fraunhofer far field radius (R_{ff}) is given by $R_{ff} > \frac{2D^2}{\lambda}$, where D is the transmitting antenna length in meters and λ is the wavelength of the transmitted signal derived from $\lambda = \frac{c}{f}$, where c is the velocity of light and f, the propagation frequency. For the rural terrain, R_{ff} at 900MHz was found to be greater than 54 meters for an antenna length (D) of 3meters, given by the service provider (MTN). Hence, the reference distance of 100meters was

adopted since it falls within the radiating far field of the transmitting antenna. Readings were then taken at intervals of 300 meters away from the Base Station. For the urban terrains, R_{ff} was found to be greater than 6m based on antenna length of 1 meter (obtained from MTN), with the reference distance adopted being 50 meters, after which measurements were taken at intervals of 100 meters away from the Base Station.

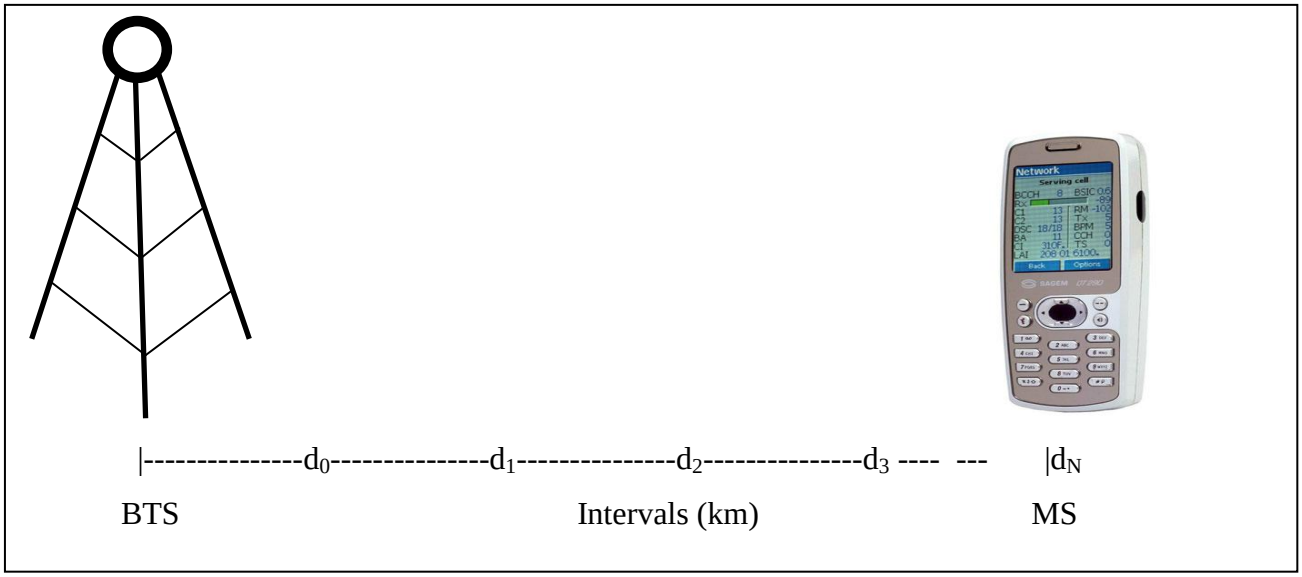


Figure 3.1: Measurement Set-up

For the Rural Terrain between Jos and Abuja, readings were obtained from Base Stations situated within the rural areas between Jos, Riyom, Gidan Waya, and Keffi. For Abuja, the Urban terrain, readings were obtained from Garki, Maitama, and Wuse. For Maiduguri, the semi-urban terrain, readings were obtained from various parts of the city.

Path loss was computed using equation (3.1) as follows:

$$L_P = EIRP - P_R \quad (3.1)$$

where,

EIRP is the Effective Isotropic Radiated Power, determined from the expression

$$EIRP=P_T-L_F+G_T \quad (3.2)$$

Where,

P_T - Transmitted power

L_F - Feeder Loss

G_T – Transmitter gain

Table I of Appendix A contains the rural area measured received power values taken at intervals of 0.3km away from the Base Station after a reference distance of 0.1km, corresponding path loss values (L_P), as well as mobile network parameters obtained from the Service Provider. Table II and Table III of Appendix A contain similar data for the urban and semi-urban terrains respectively. However, readings were taken at intervals of 0.1km away from the Base Station after a reference distance of 0.05km, as stated earlier.

3.3 Determination of Reliabilities of Empirical Models for Path Loss Prediction

For each of the terrains, the reliability of each of the considered empirical models was determined by computing the RMSE (equation (2.60)) between the empirical model prediction and the geometric mean of measured path loss values presented in Tables I to III of Appendix A, described in section 3.2. The criterion for acceptability as stated in (Wu and Yuan, 1998; Obot et al., 2011) is computed RMSE not greater than 6dB, which is adopted in this study. Based on that, any empirical model deemed not reliable is adapted to the given terrain using the following techniques:

i) The Okumura G_{Area} as correction factor (Rural Terrain only).

ii) The Root Mean Squared Error Adaptation Technique (RAT)

iii) The proposed Quotients Regression Technique (QRT) discussed in section 3.4.

3.4 Development of the Proposed Quotients Regression Technique

In this study, a novel technique, termed the Quotients Regression Technique is proposed (QRT). The QRT, which is based on the least squares approximation technique described in section 2.2.4, employs the use of a quotients function as a multiplication factor, to adapt an empirical model to a given terrain. The technique has to do with multiplying the empirical model expression with a quotients function that is dependent on transmitter – receiver separation. This distance dependent function represents the best fit least squares curve through quotient points, obtained by dividing the best fit Least Squares expression for measured path loss points, by the empirical model expression at various transmitter-receiver separations. The algorithm is as follows:

Step1: Obtain the best fit Least Squares quadratic function for computed path loss points by solving the system of equations (3.4), derived from the system of equations (2.57), while replacing y_i with measured path loss L_i, and x_i with transmitter- receiver separation (in kilometres) d_i, where N is the number of path loss values.

$$\left.\begin{array}{c} \sum_{i=1}^{N} L_i = N a_0 + a_1 \sum_{i=1}^{N} d_i + a_2 \sum_{i=1}^{N} {d_i}^2 \\ \sum_{i=1}^{N} d_i L_i = a_0 \sum_{i=1}^{N} d_i + a_1 \sum_{i=1}^{N} {d_i}^2 + a_2 \sum_{i=1}^{N} {d_i}^3 \\ \sum_{i=1}^{N} {d_i}^2 L_i = a_0 \sum_{i=1}^{N} {d_i}^2 + a_1 \sum_{i=1}^{N} {d_i}^3 + a_2 \sum_{i=1}^{N} {d_i}^4 \end{array}\right\} \quad (3.4)$$

Solving the system of equations, we obtain the coefficients $\boldsymbol{a_0}$, $\boldsymbol{a_1}$, and $\boldsymbol{a_2}$ to formulate the best fit function (3.5).

$$L(d) = a_0 + a_1 d + a_2 d^2 \quad (3.5)$$

Step2: Using equation (3.6), obtain the quotients Q_1, Q_2, ...,Q_N at intervals d_1, d_2, ..., d_N respectively, by dividing $L(d_i)$ with the empirical model expression.

$$Q(d_i) = \frac{Measured\ Path\ Loss\ best\ fit\ curve\ Expression\ at\ d_i}{Empirical\ Model\ Predicted\ Path\ Loss\ Expression\ at\ d_i} = \frac{L(d_i)}{L_P(d_i)} \quad (3.6)$$

Step3: Obtain the optimal Least Squares function Q(d) through the points Q_1, Q_2, ..., Q_N depending on the following scenarios:

i) The empirical model may require linear adaptation. In such a case, the system of equations to be used to obtain the linear coefficients $\mathbf{a_0}$ and $\mathbf{a_1}$for the linear equation $Q(d) = a_0 + a_1 d$ is equation (3.7).

$$\left.\begin{array}{c} \sum_{i=1}^{N} Q_i = N a_0 + a_1 \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} d_i Q_i = a_0 \sum_{i=1}^{N} d_i + a_1 \sum_{i=1}^{N} d_i^2 \end{array}\right\} \quad (3.7)$$

ii) The empirical model may require quadratic adaptation, hence the system of equations to be used to obtain the coefficients $\mathbf{a_0}$, $\mathbf{a_1}$and $\mathbf{a_2}$ for the quadratic equation $Q(d) = a_0 + a_1 d + \mathbf{a_2} d^2$ is the system of equations (3.8)

$$\left.\begin{array}{c} \sum_{i=1}^{N} Q_i L_i = N a_0 + a_1 \sum_{i=1}^{N} d_i + a_2 \sum_{i=1}^{N} d_i^2 \\ \sum_{i=1}^{N} d_i Q_i = a_0 \sum_{i=1}^{N} d_i + a_1 \sum_{i=1}^{N} d_i^2 + a_2 \sum_{i=1}^{N} d_i^3 \\ \sum_{i=1}^{N} d_i^2 Q_i = a_0 \sum_{i=1}^{N} d_i^2 + a_1 \sum_{i=1}^{N} d_i^3 + a_2 \sum_{i=1}^{N} d_i^4 \end{array}\right\} \quad (3.8)$$

In any case, Q(d) is the Quotients Function for adaptation.

Step4: Finally, multiply the empirical model expression by the Quotients Function Q(d) to obtain the adapted expression.

The solutions to equations (3.4), (3.7) and (3.8) are implemented in the MATLAB code in Appendix B, by creating a table of values corresponding to each sum, and then finding the sum for each column as earlier described in section 2.2.4.

3.5 Adaptation Accuracy Comparison of the Quotients Regression Technique with other Adaptation Techniques

The efficiency of the novel QRT relative to existing techniques is evaluated using geometric mean of computed path loss figures presented in Tables I to III of appendix A. The two existing techniques considered as earlier stated are as follows:

i) Okumura Adaptation using G_{AREA} obtained from the Okumura curves

ii) RMSE Adaptation Technique (RAT)

3.5.1 The Okumura Adaption Technique using G_{AREA}

The Okumura adaption technique using G_{AREA}(Gain due to type of area) has to do with subtracting the G_{AREA} value obtained from the Okumura Correction Factor curves (Obot et al., 2011), from the Okumura Model expression (2.8).

3.5.2 The Root Mean Squared Error Adaption Technique (RAT)

A widely used technique for adapting an empirical propagation model to a given terrain involves the use of a computed Root Mean Squared Error (RMSE) to compensate for differences between measurements and the empirical propagation model expression. Such a technique has been put to use by Nadir et al (2010), Ubom et al (2011), Ogbulezie et al (2013), etc. As stated in (Nadir et al, 2010), the RMSE error is given by

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(M-P)^2}{N-1}} \quad (3.3)$$

where,

M – Measured Path Loss

P – Predicted Path Loss

N- Number of paired values

Adaptation is achieved by subtracting the RMSE from the model expression if the model overestimates the path loss, else it is added to the model expression if the reverse is the case.

3.6 Creating the Artificial Neural Network Predictors

The two types of ANN considered in this study are the Multi-Layer Perceptron neural network (MLP-NN) and the Generalised Radial Basis Function Neural Network (GRBF-NN). The creation of the ANN-based prediction models is usually aimed at ensuring minimized errors for the training data set and also ensuring that the network generalises well when fed with sets of data not used when training the network. Selection of training data set is the key to ensuring good generalization of ANNs (Steve et al., 1997). Generalization property here refers to the ability of an ANN to perform well given encountered data. ANNs learn by iteratively adjusting the weights until the desired output is obtained (Kandilli, 2007). These form the bases for the formulation of the ANN based models in this study.

3.6.1 Creating the MLP-NN Based Model

The MLP-NN architecture is defined by establishing the number of hidden layers to be used, the number of neurons contained in each layer, the activation function type, etc. In this study, a

MLP-NN with 1 hidden layer and 1 output layer is adopted. The number of neurons in the input layer is dependent on path loss vector size created from terrain data. The number of neurons in the hidden layer and other parameters such as the number of training iterations and the desired error goals are all determined by trial and error. The adjustable weights are based on the Mean Square Error (MSE). The supervised learning algorithms considered is the Levenberg-Marquardt (trainlm) algorithm, which has been proven to perform better than other algorithms in function approximation. Other parameters are based on MATLAB default settings. The MLP-NN is created using the MATLAB Neural Network ToolBox function *newff*, and simulated using the function *sim*.

In this study, the MLP-NN model parameters adopted as follows:

i) Number of neurons in input layer is variable, depending input vector size

ii) Number of neurons in hidden layer

iii) 1 linearly activated output layer

iv) Training Algorithm: Levenberg-Marquardt

v) Number of iterations

vi) Error goal

vii) Stopping condition: dependent on error goal or number of iterations

3.6.2 Creating the GRBN-NN Based Model

In this study, the well-known design consisting of one input layer, a hidden layer of variable number of neurons and an output layer is adopted. As described in MATLAB (Neural Network Toolbox™), GRBN-NN can require more neurons than standard feed-forward back-propagation networks, but often they can be designed in a fraction of the time it takes to train standard feed-forward networks. They work best when many training vectors are available. The Radial Basis

network was created using the function `newrb`. The function `newrb` iteratively creates a radial basis network one neuron at a time. Neurons are added to the network until the sum-squared error falls beneath an error goal or a maximum number of neurons has been reached. The network is simulated using the function *sim*. The key elements in the creation of a GRBF-NN are basically the following arguments:

i) The spread constant

ii) Sum-squared error goal

As described in (Abraham et al, 2013), the GRBF-NN training parameter spread controls the smoothness and generalization of the approximation. The larger the spread is, the smoother the function approximation will be. Too large a spread means a lot of neurons will be required to fit a fast changing function. Too small a spread means many neurons will be required to fit a smooth function, and the network may not generalize well.

3.7 Comparison of Quotients Regression adapted Empirical Models with Artificial Neural Network Predictors

The adapted empirical models and the neural network models were compared based on the absolute mean error (μ), standard deviation (σ), root mean squared error (RMSE) and the coefficient of determination (R^2) using three techniques as follows:

a) Splitting Base Station Data into 60% Training, 10% Validation and 30% Testing

This basically involves separately analyzing each base station data by splitting path loss data obtained from it into 60% training, 10% validation and 30% testing. This is to ensure that the neural networks are trained for optimum performance. It is pertinent to note the split is specified in the MATLAB application. This technique simultaneously carries out a performance comparison of the ANN-based models with the QRT adapted empirical models on each base station.

b) **Splitting entire data into 50% training and 50% testing**

This is basically a test for generalization. Generalization refers to the capacity of neural network predictors to produce reasonable outputs for inputs not encountered during training (learning). Neural network learning is considered successful only if the system can perform well on test data on which the system has not been trained (Lahiri and Ghanta, 2011). The aim of this technique is to determine how well the neural networks can generalize. The generalization property of a neural network model determines how robust and hence how suitable the model is for path loss prediction.

This technique involves splitting measured path loss data obtained from Base Stations within a given terrain into two sets: 50% training and 50% testing as presented in Appendix A. The geometric mean of all path loss values at each receiver-transmitter separation is obtained from the training set using equation (3.9), and then used to train the neural network models, and to also adapt the empirical models using the QRT. From the testing set, each Base station set of data is statistically compared with the trained neural network models and the QRT adapted empirical models.

$$\text{GM}=\sqrt[n]{X_1.X_2.X_{3,}\ldots,X_n} \tag{3.9}$$

In performance evaluation, the geometric mean is preferred to the arithmetic mean because it is less sensitive to extreme values (Nicholas, 2002).

c) Training with one Base Station data set and testing with a set from another

This is another test for generalization. The technique involves randomly training with a data set obtained from one Base Station and testing with a data set from another Base Station (Abraham et al. 2013). Simultaneously, the testing set is used to test the QRT adapted empirical models. By implication, a given data set can both be used for training and testing.

CHAPTER FOUR
RESULTS AND DISCUSSIONS

4. 1 Introduction

The results generated by the MATLAB code in Appendix B are hereby presented. Each of the aforementioned terrains is analysed separately.

For built-up terrains, the COST 231 Hata and COST 231 Walfisch-Ikegami are considered. The Hata-Okumura model is suitable for mega-cells, hence it is applied along with the COST 231 Hata, to the Rural Area. In all cases, model predictions presented here are based on the mobile network configurations of the base stations for appropriate comparison.

For the rural terrain, the selected empirical models are tested for acceptability, and then adapted to the terrain using the three techniques described in section 3.4. For the urban and semi-urban terrains, the adaptation techniques applied are the RAT and the QRT. Sequel to comparison for adaptation accuracy of the adaptation techniques, the adapted empirical models are tested for capacity to generalise. Finally, the QRT adapted empirical models are statistically compared for path loss prediction accuracy with the ANN-based models.

As stated in section 3.7, the first comparative technique adopted involves randomly splitting path loss data obtained from each Base Station into 60% training, 10% validation and 30% testing. This is in contrast to the technique adopted by Abraham (2013), where the split is 60% training and 40% testing, disregarding the need for validation. The importance of validation is highlighted by Brian et al.(2013), who describe the training-validating-testing approach as the first, and often the only, option system developers consider for the assessment of a neural network. The assessment is accomplished by the repeated application of neural network training data, followed by an application of neural network testing data to determine whether the neural network is acceptable. The training set is used as the primary set of data that is applied to the neural network for learning and adaptation. The validation set is used to further refine the neural network construction. The testing set is then used to determine the performance of the neural network by computation of an error metric.

A major observation made in the Abraham (2013) approach is that the 40% of the data supposedly set aside for testing was actually used for validation and not testing. Furthermore, the same set was actually used in simulating the neural networks after training. By implication, no data was actually used for testing.

Table 4.1: RMSE Comparison of Abraham's (2013) Approach with this Study

Base Stations	**Approach**	**Models**			
		MLP-NN	**RBF-NN**	**ANFIS**	**Hata**
BST1	*Abraham*	*4.264*	*4.529*	*3.444*	*10.663*
	This study	3.4989	2.0097	3.0682	9.773
BST2	*Abraham*	*3.915*	*3.807*	*3.762*	*8.442*
	This study	4.9924	3.6817	2.6817	20.9574
BST3	*Abraham*	*5.828*	*5.043*	*4.967*	*11.458*
	This study	6.9077	6.566	6.9164	11.9227
BST4	*Abraham*	*9.789*	*5.759*	*5.373*	*13.341*
	This study	5.8018	4.0906	4.0345	13.0011
AVERAGE	***Abraham***	***5.949***	***4.785***	***4.387***	***10.976***
	This study	**5.3002**	**4.087**	**4.175**	**13.91**

A Root Mean Square Error comparison of the approach adopted in this study with that of Abraham (2013), based on the Abraham (2013) data obtained at 900MHz, is presented in Table 4.1, which shows that the improvements in prediction accuracy offered by the approach adopted in this study over that of Abraham (2013) are as follows:

i) 10.91% on the MLP-NN

ii) 14.59% on the RBF-NN

iii) 4.83% on the ANFIS.

Based on the technique adopted in this study, it can be observed from Table 4.1 that the RBF-NN slightly outperforms the ANFIS model, justifying the adoption of the RBF-NN model.

4.2 Comparison of Adaptation Techniques Using the Okumura Model

This comparative analysis is aimed at ascertaining the performance of the proposed QRT relative to the use of established Okumura terrain specific correction factors, and also the use of RMSE for adaptation. The choice of the Okumura model for comparative analysis is based on the fact that apart from its suitability for path loss prediction in urban areas, it has correction factors for open, quasi-open and sub-urban areas. In this comparative analysis the rural area between Jos and Abuja is considered. This terrain matches Hata's description of a suburban area. Therefore, the mobile network parameters and the path loss values in Table I of the appendix A are considered.

4.2.1 Adapting the Okumura Model using Curve Correction Factors

From the Okumura Median Attenuation curves (Obot et al 2011), the median attenuation $A_{MU}(f,d)$ for this terrain within the range 1 to 4.3km at 900MHz is 20dB. Likewise, the correction factor G_{AREA} for suburban areas at 900MHz is 9dB according to the Okumura G_{AREA} Correction Factor curves. Therefore, from equation (2.33), the G_{AREA} adapted Okumura model path loss equation for this terrain is given by:

$$L_{OKM} = L_{FSL} + 20 - H_{MG} - H_{BG} - 9$$
$$= L_{FSL} + 11 - H_{MG} - H_{BG} \quad (4.1)$$

where,

- L_{FSL}=32.45+20logf+20logd

- H_{BG}=10log(hm/3)
- H_{HG}=20log(hb/200)

4.2.2 Adapting the Okumura Model using the RMSE Adaptation Technique (RAT)

For this terrain, the Okumura model prediction error based on RMSE, computed by the MATLAB code based on the path loss geometric mean in Table I of appendix A, was found to be 8.95dB. Therefore, the RAT adapted Okumura model expression from (4.1) is given by:

$$L_{RAT\_adapted} = L_{FSL} + 11 - H_{MG} - H_{BG} + 8.95$$
$$= L_{FSL} + 19.95 - H_{MG} - H_{BG} \quad (4.2)$$

4.2.3 Adapting the Okumura Model using the Quotients Regression Technique (QRT)

Following the steps described in section 3.4, the Okumura model is adapted to the terrain as follows:

Step1: The ***coefficients*** of the best fit Least Squares function through geometric mean of the computed path loss points in Table I of Appendix A, were obtained by solving the system of equations (3.4),

where:

N is the number of samples

d_i represents the intervals

L_i represents the path loss values.

The coefficients obtained are a_0=94.33, a_1=23.13, a_2=-2.55, and hence the best fit least squares equation for computed path loss is

$$L_{LS} = 94.33 + 23.13d - 2.55d^2 \quad (4.3)$$

Step2: Quotients computed at various intervals using equation (4.4) are shown in Table 4.2.

$$Q_i = \frac{L_{LS}(d_i)}{L_{OKM}} \tag{4.4}$$

Table 4.2: Quotients for Okumura Model Adaptation (Jos-Abuja Rural Area)

Intervals (km)	0.1	0.4	0.7	1.0	1.3	1.6	1.9	2.2	2.5	2.8	3.1	3.4	3.7	4.0	4.3
Quotients	0.96	0.98	0.93	0.95	0.98	1	1.02	1.04	1.06	1.08	1.09	1.09	1.1	1.1	1.1

Step3: Equation (3.7) was used to determine the coefficients of the best fit Least Squares function $Q(d) = a_0 + a_1 d$ through quotient points, where Q_i represents quotients obtained at intervals d_i.

Coefficients obtained for best fit curve through quotients are a_0=0.9238 and a_1=0.0475.

Therefore, the adaptation function is given by:

$$Q(d) = 0.9238 + 0.0475d \tag{4.5}$$

Step 4: The QRT Adapted Okumura Model for this terrain is obtained by multiplying the Okumura model expression by the adaptation function Q(d) as follows:

$$L_{QRT} = (0.9238 + 0.0475d) \, . L_{OKM} \tag{4.6}$$

4.2.4 Comparison of the Adaption Techniques as applied to the Okumura Model

Figure 4.1 generated by the MATLAB code in Appendix B1 presents a comparison of the three adaptation techniques as applied to the Okumura model. The G_{AREA} adapted Okumura plot is based on equation (4.1). Figure 4.1 shows that the G_{AREA} adapted Okumura model underestimates the path loss, while Table 4.3 shows that the model does so by a RMSE value of 8.95dB. Likewise, the RAT adapted Okumura model is based on equation (4.2). It can be observed from Figure 4.1 that the RAT adapted Okumura model overestimates the path loss,

while Table 4.3 shows that the model does so by an RMSE value of 9.41dB. Finally, the QRT adapted Okumura model is based on equation (4.6). It can be observed from figure 4.1 that there is a close convergence of the QRT adapted model plot with the path loss data. This is an indication of the effectiveness of the technique. Results in Table 4.3 buttress this fact, with the QRT adapted model having the lowest prediction errors and the best fit. The QRT-adapted model outperforms the G_{AREA} adapted model and RAT-adapted model by RMSE values of 6.78dB and 7.23dB respectively. It can also be observed from Table 4.3 that the G_{AREA} adapted Okumura model outperforms the RAT-adapted counterpart by an RMSE value of about 0.45dB. This is a typical scenario where the slope component of an empirical model expression significantly differs from that of the best fit curve through measured path loss point, in which the inefficiencies of the G_{AREA} and RAT adaptation techniques are exposed.

Table 4.3: Adaptation Accuracy Comparison (Jos-Abuja Rural Area)

PERFORMANCE STATISTICS	G_{AREA}	RAT	QRT
μ(dB)	7.5037	7.6857	2.0712
σ(dB)	5.0557	5.6143	1.2599
RMSE(dB)	8.9533	9.4069	2.1773
R^2	0.6948	0.6631	0.9719

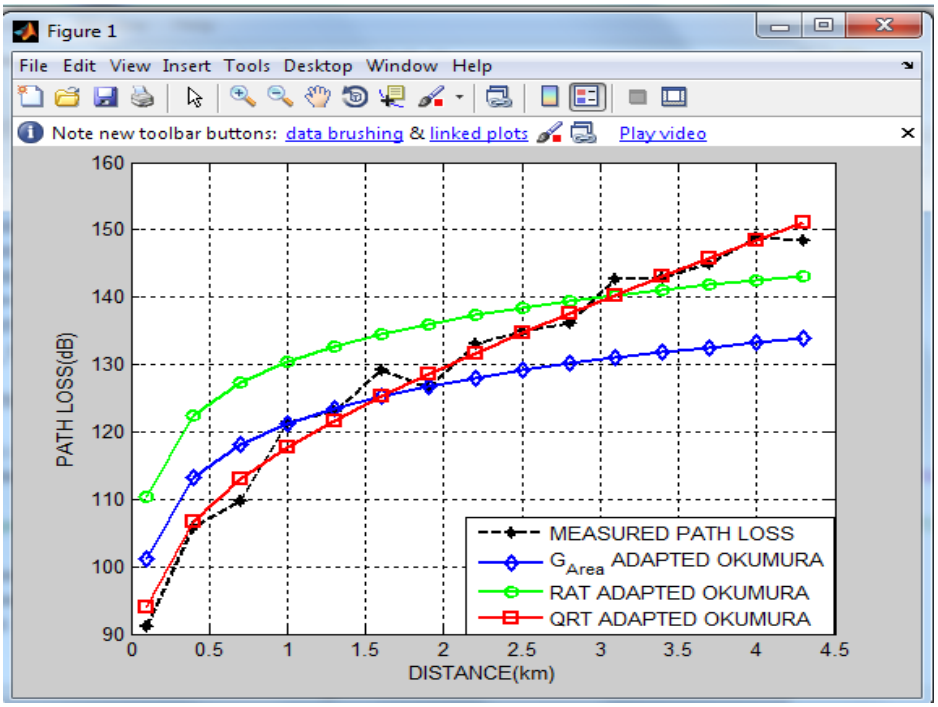
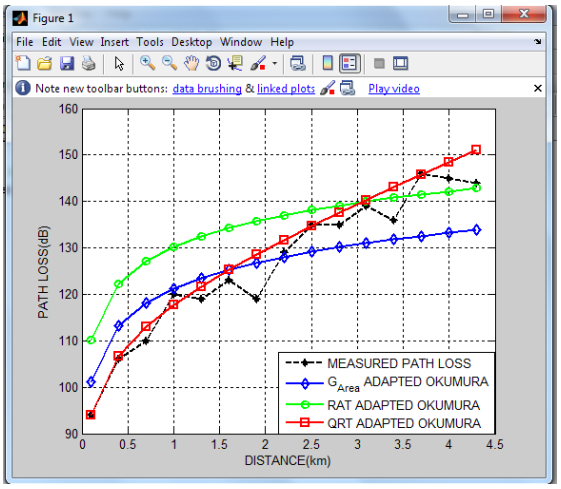


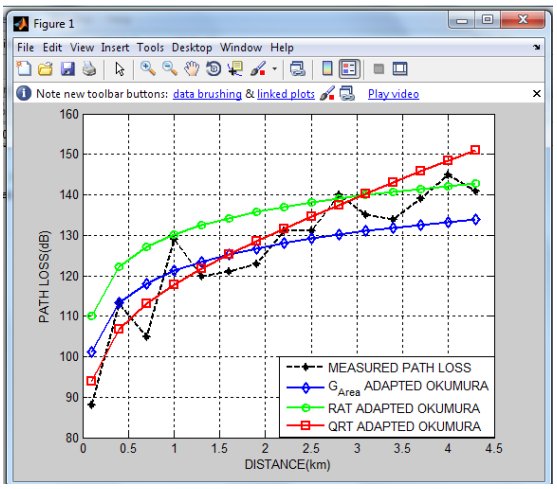
Figure 4.1: Comparison of Adaptation Techniques (Jos-Abuja Rural Area)

4.2.5 Generalization Test for Adapted Okumura Models

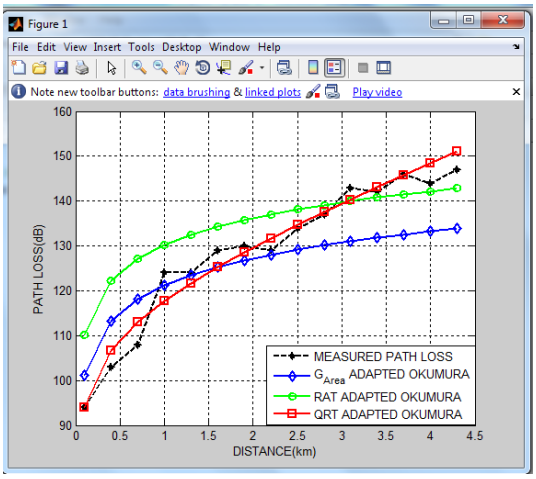
Each of the three adapted models is tested for prediction accuracy using data from the generalization set (Base Stations 6 to 10) of Table I. Figures 4.2 a) to 4.2 e) graphically show the performance of each of the three adapted models on each of the Base Stations. It can be observed that the QRT adapted model exhibits the closest prediction on each of the Base Stations. Results in Table 4.4 clearly show that the QRT adapted Okumura model gives the most accurate predictions across all the Base Stations with the least prediction errors. On the average, the trend continues with the QRT Adapted Okumura model outperforming its counterparts with an acceptable RMSE value of 4.53dB. By implication, the QRT adapted model offers an improvement of about 3.63dB over the Okumura G_{AREA} adapted model, and also an improvement of 4.91dB over the RAT-adapted model.



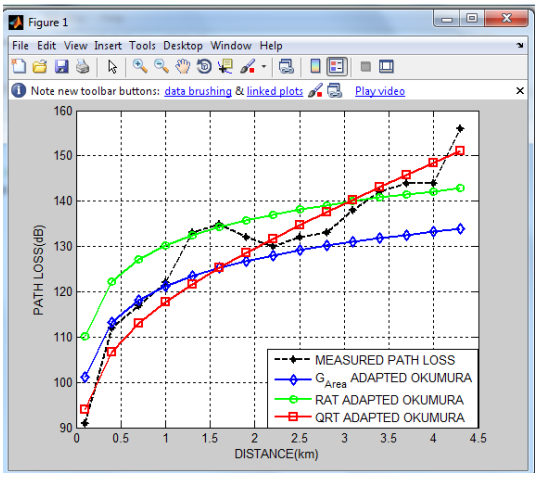
a): BST 6 Comparison



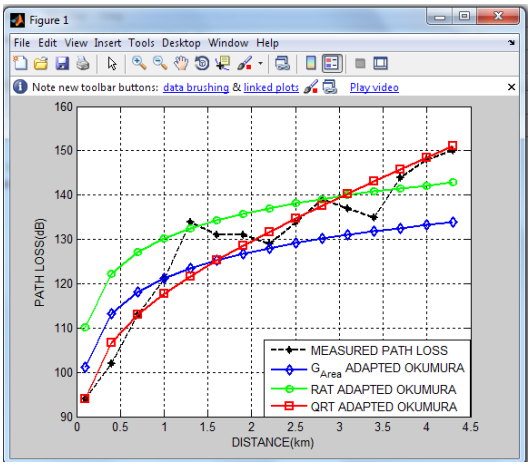
b): BST 7 Comparison



c): BST 8 Comparison



d): BST 9 Comparison



e): BST 10 Comparison

Figure 4.2: Generalisation Comparison of Adapted Okumura Models

Table 4.4: Path Loss Prediction Accuracies of Adapted Okumura Models

MODEL	STATS.	BST6	BST7	BST8	BST9	BST10	GEOM. MEAN
G_{AREA} Adapted Okumura	μ(dB)	6.49	6.13	7.33	7.13	7.36	6.87
	σ(dB)	3.64	4.20	4.41	5.77	4.72	4.50
	RMSE(dB)	7.39	7.36	8.48	9.05	8.65	8.16
	R^2	0.76	0.75	0.71	0.64	0.68	0.71
RAT Adapted Okumura	μ(dB)	8.67	8.37	7.26	6.16	7.03	7.44
	σ(dB)	6.06	6.94	6.02	5.31	5.72	5.99
	RMSE(dB)	10.47	10.74	9.30	8.02	8.94	9.44
	R^2	0.51	0.47	0.65	0.72	0.66	0.59
QRT Adapted Okumura	μ(dB)	3.01	5.62	2.59	4.29	3.2	3.60
	σ(dB)	2.78	3.06	1.91	2.84	3.89	2.82
	RMSE(dB)	4.04	6.36	3.18	5.10	4.58	4.53
	R^2	0.93	0.82	0.96	0.89	0.91	0.90

4.3 Comparison of Quotients Regression Adapted Empirical Models with Artificial Neural Network Predictors

In this section the QRT adapted empirical models are compared for path loss prediction accuracy with the ANN-based models, as well as with the RAT-adapted empirical models. Each of the three terrains is analysed separately using the three techniques described in section 3.7.

4.3.1 The Rural Area between Jos and Abuja

The terrain under investigation is situated between Jos and Abuja. This terrain matches the description of Hata's suburban area as presented in section 2.2.5. Received power values obtained from Base Stations across the terrain and the corresponding computed path loss values are presented in Table I of Appendix A. The empirical models considered are the COST 231 Hata and the Hata-Okumura models due to availability of correction factors for rural areas.

The MLP-NN parameters determined suitable for this terrain are as follows:

i) Number of neurons in hidden layer (with sigmoid type tang. activation) is 3.

ii) 1 linearly activated output layer.

iii) Training Algorithm: Levenberg-Marquardt back-propagation.

iv) Number of iteration = 105

v) Error goal: 0.1

For the GRBF-NN, the parameters are as follows:

i) Spread constant is 0.5

ii) Error goal: 0.1

4.3.1.1 Testing the COST 231 Hata and the Hata-Okumura models for acceptability

Figure 4.3, generated by the MATLAB code in Appendix B2 shows a graphical comparison of the geometric mean of computed path loss values presented in Table I of appendix A, with the COST 231 Hata and the Hata-Okumura models.

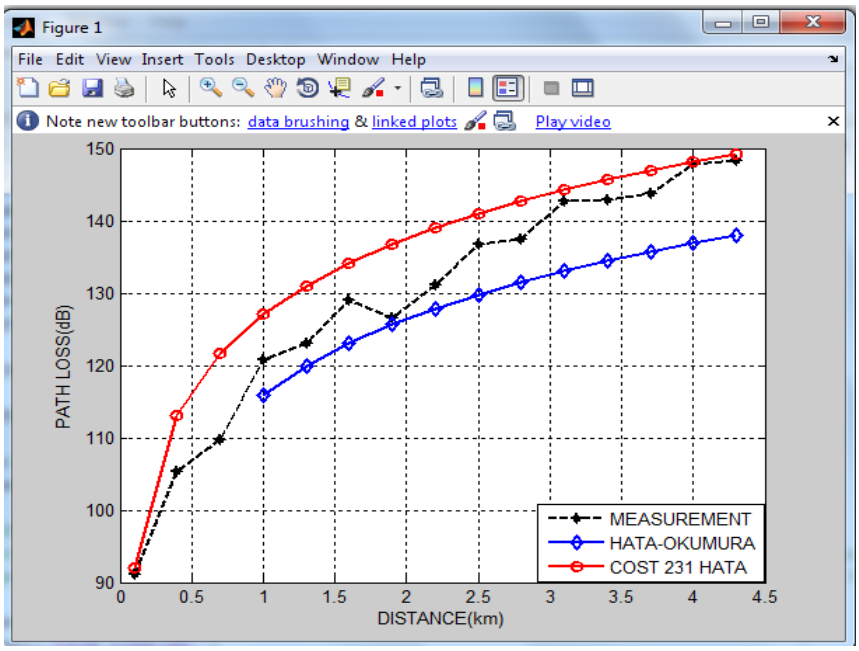


Figure 4.3: Comparison of Empirical Models with Mean Computed Path Loss (Jos - Abuja Rural)

The COST 231 Hata plot was derived from equation (2.37) while the Hata-Okumura from equation (2.35). It is obvious that neither of the two empirical models is quite reliable: the COST 231 Hata model overestimates the path loss, while the Hata-Okumura model does the opposite. Results in Table 4.5, generated by the MATLAB code in Appendix B2 show that the COST 231 Hata model slightly overestimates the path loss by a RMSE of 6.13dB, while the Hata-Okumura model underestimates the path loss by a RMSE of 7.13dB. According to Obot et al. (2011), any model that overestimates or underestimates path loss by a RMSE value not greater than 6dB is acceptable.

Table 4.5: Performance Evaluation of Empirical Models (Jos - Abuja Rural Area)

PERFORMANCE STATISTICS	COST 231 Hata	Hata-Okumura
μ(dB)	5.56	6.09
σ(dB)	4.00	3.53
RMSE(dB)	6.24	7.71
R^2	0.83	0.44

4.3.1.2 Adapting the COST 231 Hata and the Hata-Okumura Models

In adapting the two empirical models to the geometric mean of computed path loss, a comparative analysis is made between the two techniques described in Chapter 3, i.e., the RAT and the QRT.

i) Adapting the COST 231 Hata Model using the RAT

For this terrain, the COST 231 Hata model RMSE computed by the MATLAB code Appendix B2, based on the path loss geometric mean was found to be 6.23dB. Therefore, the modified COST 231 Hata model expression based on this technique is given by

$$L_{RAT\ Adptd} = 46.3 + 33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd\ - 6.24$$

$$= 40.07 + 33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd \quad (4.7)$$

ii) Adapting the COST 231 Hata Model using the Quotients Regression Technique

The COST 231 Hata model was adapted to the terrain using the steps described in section 3.4.1.2.

Step1: The best fit Least Squares function through geometric mean of the computed path loss points in Table I of Appendix A, is equation (4.3).

Step2: Quotients computed at various intervals using equation (4.8) are shown in Table 4.6.

$$Q_i = \frac{L_{LS}(d_i)}{46.3+33.9logf-13.82logh_B-a(h_R)+(44.9-6.55logh_B)logd_i} \quad (4.8)$$

Table 4.6: Quotients for COST 231 Hata Adaptation (Jos - Abuja Rural Area)

Intervals (km)	0.1	0.4	0.7	1.0	1.3	1.6	1.9	2.2	2.5	2.8	3.1	3.4	3.7	4.0	4.3
Quotient	1.05	0.91	0.9	0.9	0.92	0.93	0.94	0.96	0.97	0.98	0.99	0.98	0.99	0.98	0.98

Step3: Equation (3.7) was used to determine the ***coefficients*** of the best fit Least Squares function $Q(d)=a_0+a_1d$ through quotient points. Coefficients obtained for best fit curve through quotients are a_0=0.93 and a_1=0.0126. Therefore, the adaptation function is given by:

$$Q(d)=0.93+0.0126d \quad (4.9)$$

Step 4: The QRT adapted COST 231 Hata Model for this terrain was obtained by multiplying the COST 231 Hata expression by the adaptation function Q(d) as follows:

$$CH_{QRT} = (0.93 + 0.0126d)(46.3 + 33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd)$$

$$= 43.06 + 0.58d + (31.53 + 0.43d)logf - (12.85 + 0.174d)logh_B - (0.93a + 0.0126d)(h_R) + (41.76 - 6.09logh_B)logd + (0.566 - 0.083logh_B)dlogd \quad (4.10)$$

iii) Adapting the Hata Okumura Model using the RAT

The Hata-Okumura computed RMSE was found to be 7.71dB. Therefore, the RAT-adapted Hata-Okumura model expression is given by

$$HO_{RAT} = L_U - 2(logf)^2 - 5.4 + 7.71$$
$$= L_U - 2(logf)^2 + 2.31 \quad (4.11)$$

iv) Adapting the Hata Okumura Model using the Quotients Regression Technique

The Hata-Okumura Quotients Function obtained is Q(d)=1+0.0156d. Therefore, the QRT adapted Suburban Area Hata-Okumura model for this terrain is given by

$$HO_{QRT}=\text{Q(d) x } L_{SU}=(1+0.0156\text{d}) \text{ x } (L_U - 2(logf)^2 - 5.4)$$
$$=(1 + 0.0156d)L_U - (2 + 0.0312d)(logf)^2 - (5.4 + 0.08424d) \quad (4.12)$$

4.3.1.3 Comparison of Adaptation Techniques

Figure 4.4 generated by the MATLAB code in Appendix B2 shows a graphical comparison of the two adaptation techniques relative to the geometric mean of measurements, as applied to the COST 231 Hata Model. The RAT-adapted COST 231 Hata model and the QRT adapted COST 231 Hata model plots were derived from equations (4.7) and (4.10) respectively. Figure 4.4 shows which of the two techniques gives a closer adaptation to the geometric mean. The differences between the two techniques can be clearly observed at distances beyond 2 kilometres away from the Base Station. It is obvious that the QRT gives a closer adaptation to the mean computed path loss than its counterpart. This fact is buttressed by the performance statistics presented in Table 4.7, generated by the MATLAB code in Appendix B2, which shows that the QRT is more accurate than the RAT technique by an RMSE value of 1.2dB. The apparent convergence of the QRT adapted COST 231 Hata with the computed path loss is highlighted by the high coefficient of determination of 0.97.

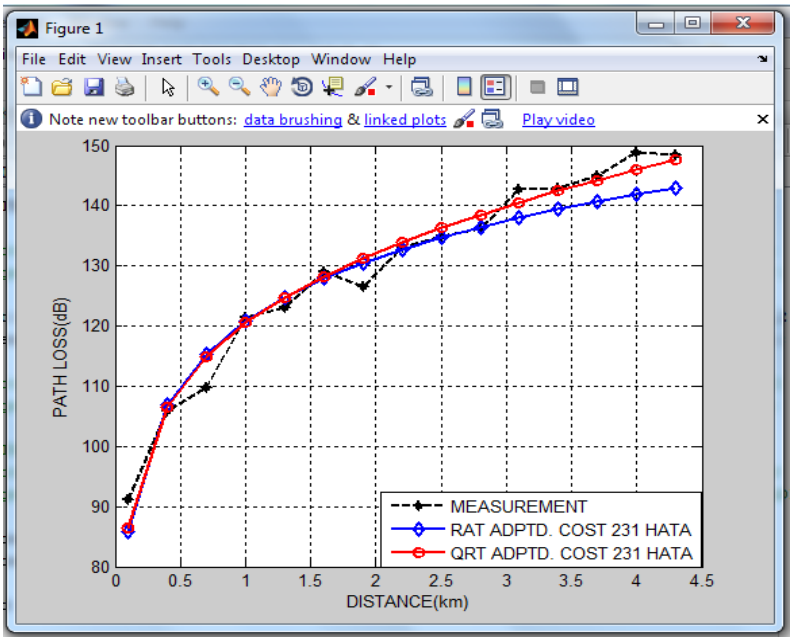


Figure 4.4: Comparison of Adaptation Techniques for the COST 231 Hata Model (Jos and Abuja Rural)

Table 4.7: Performance Comparison of Adaptation Techniques (Jos-Abuja Rural)

PERFORMANCE STATISTICS	Adapted COST 231 Hata		Adapted Hata-Okumura	
	RAT Adaptation	QRT Adaptation	RAT Adaptation	QRT Adaptation
μ(dB)	3.02	1.98	2.85	1.58
σ(dB)	2.34	1.70	1.63	1.21
RMSE(dB)	3.77	2.57	3.25	1.96
R^2	0.95	0.97	0.88	0.95

Likewise, Figure 4.5 shows that the QRT adapted Hata plot based on equation (4.12) is closer to the geometric mean of computed path loss than the RAT-adapted counterpart derived from equation (4.11). Results in Table 4.7 indicate that the QRT is more accurate by about 1.29dB.

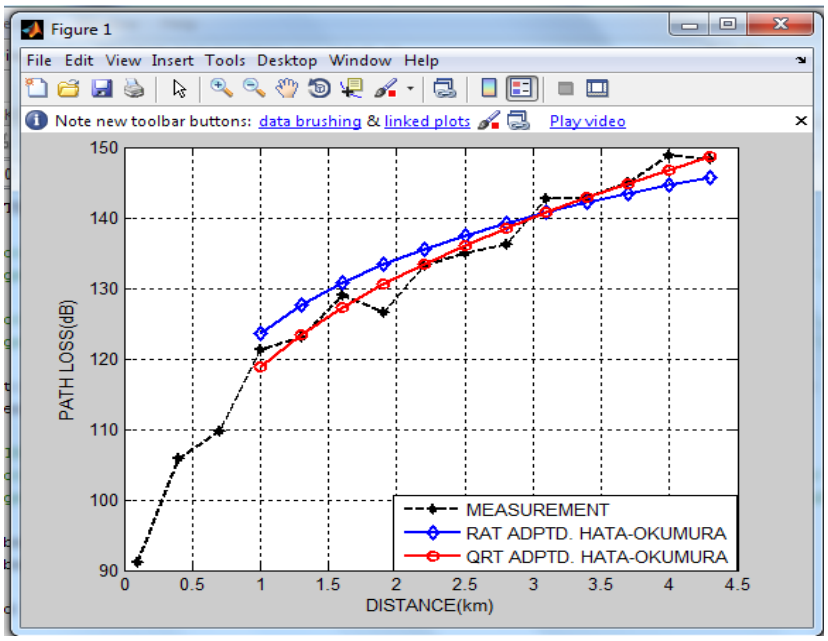


Figure 4.5: Comparison of Adaptation Techniques for the Hata-Okumura Model (Jos-Abuja Rural)

4.3.1.4 Generalisation test for RAT Adapted and QRT Adapted Empirical Models

Figure 4.6 generated by the MATLAB code in Appendix B2 shows a graphical interpolation comparison of the RAT-Adapted and QRT Adapted COST 231 models relative to Base Station 6 data. The distinction between the two adapted models can be observed at distances farther away from the base station, with the QRT adapted counterpart showing a slightly better interpolation.

Likewise, Figure 4.7 shows a graphical interpolation comparison of the RAT-adapted and QRT adapted Hata-Okumura models relative to Base Station 6 data. However, the distinction between the two adapted models in this case is quite evident, with the QRT adapted counterpart exhibiting much better interpolation. Table 4.8 presents the performance of each of the adapted models relative to Base Stations 6 to 10. Results in Table 4.8 indicate that on the average, the QRT adapted COST 231 Hata and QRT adapted Hata-Okumura models outperform the RAT-adapted counterparts by 0.23dB and 0.42dB respectively.

Table 4.8: Performance Comparison of RAT and QRT adapted Empirical Models (Jos-Abuja Rural)

MODEL	STATS.	BST6	BST7	BST8	BST9	BST10	MEAN
RAT-Adapted COST 231 Hata	μ(dB)	3.80	4.69	3.28	3.99	3.82	3.89
	σ(dB)	3.17	2.67	2.44	3.35	2.82	2.87
	RMSE(dB)	4.88	5.35	4.04	5.13	4.69	4.80
	R^2	0.89	0.87	0.93	0.88	0.91	0.90
QRT Adapted COST 231 Hata	μ(dB)	3.55	5.18	2.49	3.93	3.42	3.61
	σ(dB)	3.23	2.73	2.39	2.94	2.84	2.81
	RMSE(dB)	4.72	5.81	3.40	4.85	4.39	4.57
	R^2	0.90	0.85	0.95	0.90	0.92	0.90
RAT-Adapted Hata-Okumura	μ(dB)	5.01	5.70	2.35	3.69	3.42	3.85
	σ(dB)	3.92	3.09	1.78	3.05	2.41	2.75
	RMSE(dB)	6.26	6.42	2.90	4.71	4.13	4.69
	R^2	0.59	0.32	0.87	0.69	0.73	0.61
QRT Adapted Hata-Okumura	μ(dB)	3.08	4.70	2.00	4.04	3.51	3.33
	σ(dB)	2.75	2.98	1.79	3.51	3.08	2.76
	RMSE(dB)	4.05	5.49	2.64	5.25	4.59	4.27
	R^2	0.83	0.50	0.90	0.61	0.66	0.69

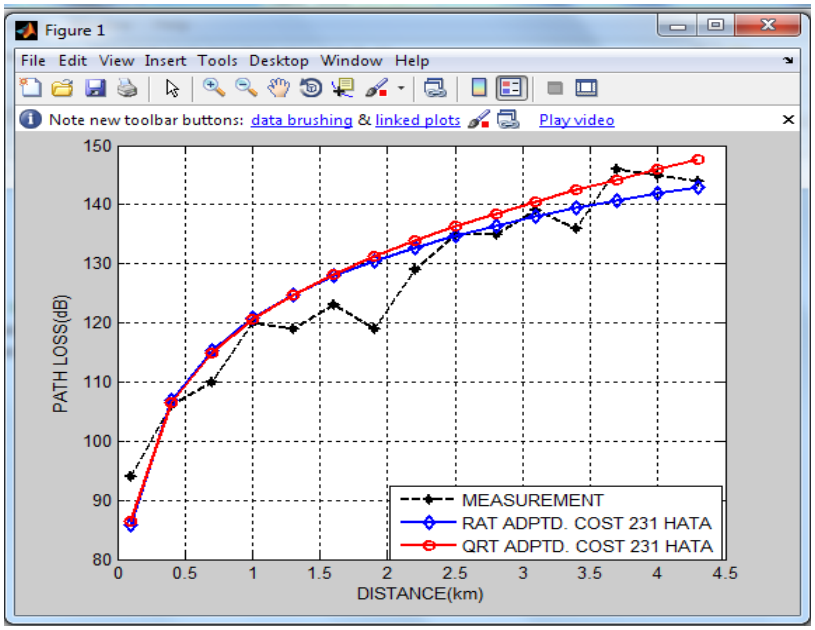


Figure 4.6: Performance Comparison of RAT and QRT Adapted COST 231 Hata Models (Jos-Abuja Rural)

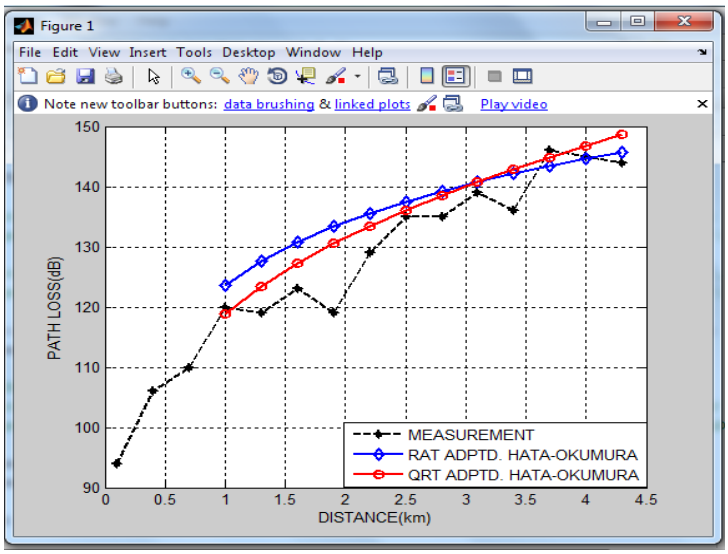


Figure 4.7: Performance Comparison of RAT and QRT Adapted Hata-Okumura Models (Jos-Abuja Rural)

4.3.1.5 Comparison of ANN-based and QRT Adapted Empirical Models using the Training-Validation-Testing Technique (Technique A) (Jos-Abuja Rural)

As stated earlier, this technique involves analysing each base station data separately by randomly splitting the data into 60% training, 10% validation and 30% testing. The split is specified in MATLAB code by the function dividerand(pn,0.6,0.1,0.3), where *pn* is the path loss input vector. Figure 4.8 shows a sample MATLAB code (Appendix B3) generated graphical comparison of the trained neural network models, the adapted empirical models and the standard empirical models, based on Base Station 1 data. The plots were derived from the following:

(i) The neural network model plots are based on training with 60%, validating with 10% and testing with 30% of computed path loss in Table I of appendix A.

(ii) The COST 231 Hata plot derived from equation (2.37)

(iii) The Hata-Okumura plot derived from equation (2.35)

(iv) The Quotients Function Adapted COST 231 Hata plot derived from equation (4.10)

(v) The Quotients Function Adapted Hata-Okumura plot derived from equation (4.12)

Table 4.9: Performance Evaluation using Training-Validation-Testing Technique (Jos – Abuja Rural)

MODEL	STAT.	BST 1	BST 2	BST 3	BST 4	BST 5	BST 6	BST 7	BST 8	BST 9	BST 10	GEOM. MEAN
MLP-NN	μ(dB)	2.86	5.22	1.58	5.64	3.90	2.86	4.70	1.78	3.62	4.03	3.36
	σ(dB)	1.90	1.94	1.51	4.31	1.86	2.82	4.88	1.41	3.22	1.94	2.36
	RMSE(dB)	3.33	5.50	2.08	6.83	4.24	3.81	6.41	2.18	4.63	4.39	4.05
	R^2	0.84	0.80	0.89	0.69	0.69	0.85	0.21	0.89	0.57	0.75	0.67
GRBF-NN	μ(dB)	4.27	4.93	3.53	4.90	4.07	3.73	4.86	3.35	5.07	3.14	4.13
	σ(dB)	2.20	2.96	2.86	4.34	2.88	1.45	3.81	1.49	2.63	2.20	2.54
	RMSE(dB)	4.70	5.60	4.36	6.25	4.82	3.95	5.93	3.61	5.59	3.70	4.77
	R^2	0.68	0.79	0.52	0.74	0.60	0.83	0.33	0.71	0.37	0.82	0.61
QRT Adapted COST 231 Hata	μ(dB)	3.57	5.11	4.10	3.45	2.95	3.55	5.18	2.49	3.93	3.42	3.69
	σ(dB)	2.43	2.74	2.83	2.51	1.96	3.23	2.73	2.39	2.94	2.84	2.64
	RMSE(dB)	4.28	5.76	4.93	4.22	3.51	4.72	5.81	3.40	4.85	4.39	4.52
	R^2	0.94	0.88	0.90	0.94	0.96	0.90	0.85	0.95	0.90	0.92	0.91
QRT Adapted Hata-Okm	μ(dB)	4.39	4.48	4.31	2.97	2.62	3.08	4.70	2.00	4.04	3.51	3.49
	σ(dB)	2.88	2.86	3.53	2.29	1.79	2.75	2.98	1.79	3.51	3.08	2.68
	RMSE(dB)	5.19	5.25	5.48	3.69	3.13	4.05	5.49	2.64	5.25	4.59	4.35
	R^2	0.61	0.80	0.38	0.90	0.86	0.83	0.50	0.90	0.61	0.66	0.68
COST 231 Hata	μ(dB)	3.19	8.63	2.83	6.43	6.19	7.76	8.10	5.49	4.72	5.46	5.55
	σ(dB)	3.08	5.65	2.70	5.22	4.16	4.43	4.84	3.68	2.94	3.58	3.92
	RMSE(dB)	4.36	10.21	3.85	8.17	7.38	8.86	9.35	6.54	5.51	6.47	6.76
	R^2	0.93	0.61	0.94	0.76	0.80	0.65	0.60	0.83	0.87	0.82	0.77
Hata-Okm	μ(dB)	9.85	4.98	9.76	6.00	5.93	4.44	3.95	6.43	7.43	6.76	6.29
	σ(dB)	2.96	4.32	2.67	4.12	3.27	3.15	3.90	2.72	4.91	4.20	3.55
	RMSE(dB)	10.25	6.47	10.09	7.18	6.70	5.37	5.44	6.94	8.79	7.87	7.34
	R^2	-0.52	0.69	-1.12	0.62	0.35	0.70	0.51	0.27	-0.10	0.01	0.10

It can be observed that the MLP-NN model and the COST 231 Hata are the closest to the test output. Table 4.9 also shows that the MLP-NN model has the best fit (0.89) the least RMSE (3.33dB) on Base Station 1.

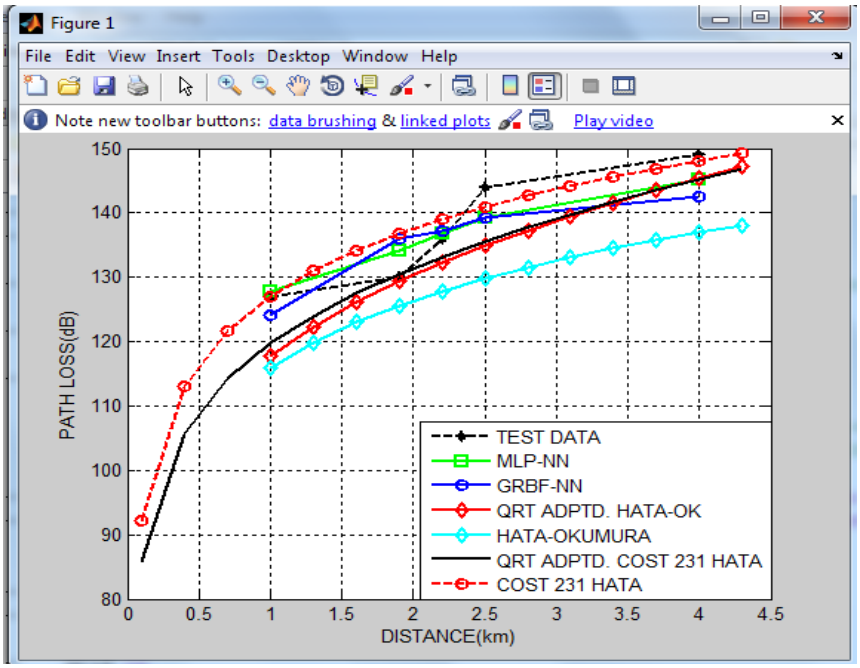


Figure 4.8: Comparison of the Neural Network Models on BST1 using Training-Validation-Testing Technique (Jos - Abuja Rural)

The performance indicators across all the Base Stations based on this comparative technique are presented in Table 4.9. Results indicate that the MLP-NN predictor performs better than the two QRT adapted empirical models, which in turn outperform the GRBF-NN. On the average, the ANN–based models are slightly more accurate by an RMSE value of 0.25dB. However, the QRT adapted empirical models have better fit.

4.3.1.6 Comparison of ANN-based with Quotients Regression Adapted Empirical Models using the 50% Training and 50% Testing Technique

As stated in section 3.7, this technique is aimed at determining the generalization capacities of the prediction models. A model that generalises well is considered more robust and effective in path loss prediction. Computed path loss values in Table I of appendix A are classified as follows: Base stations 1 to 5 - training set, while Base Stations 6 to 10 - testing set.

The geometric mean of the training set at each receiver-transmitter separation is obtained using equation (3.9), and then used to train the neural networks predictors.

Figure 4.9 shows a sample MATLAB code (Appendix B4) generated graphical comparison of the trained neural network models, the adapted empirical models and the standard empirical models, with the test data from Base Station 6. The plots were based on the following:

(vi) The neural network model plots are based on training with the training set geometric mean in Table I of appendix A, and testing with testing set.

(vii) The COST 231 Hata plot derived from equation (2.37)

(viii) The Hata-Okumura plot derived from equation (2.35)

(ix) The QRT Adapted COST 231 Hata plot derived from equation (4.10)

(x) The QRT Adapted Hata-Okumura plot derived from equation (4.12)

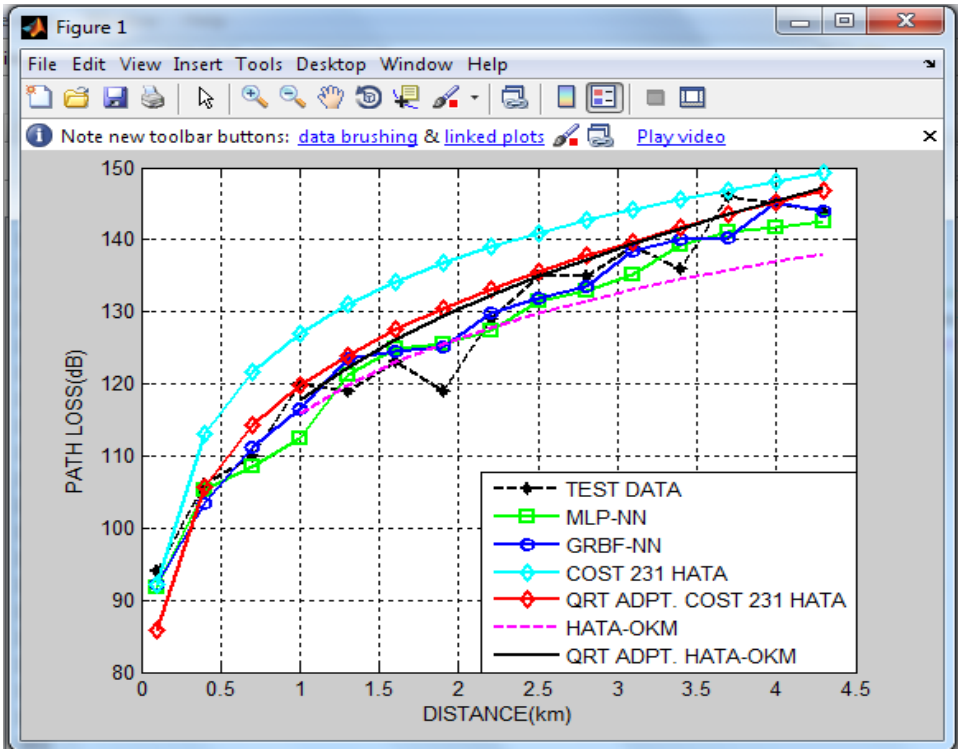


Figure 4.9: Comparison of Predictors with BST6 data using 50% Training and 50% Testing (Jos - Abuja Rural)

Table 4.10: Performance Evaluation using the 50% Training and 50% Testing Technique (Jos- Abuja Rural Area)

MODEL	STAT.	GM / BST6	GM/ BST7	GM/ BST8	GM/ BST9	GM/ BST10	GEOM. MEAN
MLP-NN	μ(dB)	3.12	4.11	2.24	3.87	3.43	3.28
	σ(dB)	1.95	4.21	2.53	2.49	2.82	2.71
	RMSE(dB)	3.64	5.79	3.31	4.55	4.38	4.26
	R^2	0.94	0.85	0.96	0.91	0.92	0.91
GRBF-NN	μ(dB)	2.51	4.12	2.32	4.03	2.97	3.10
	σ(dB)	1.96	3.53	1.64	1.98	2.50	2.24
	RMSE(dB)	3.14	5.35	2.81	4.46	3.83	3.81
	R^2	0.96	0.87	0.97	0.91	0.94	0.93
QRT Adapted COST 231 Hata	μ(dB)	3.55	5.18	2.49	3.93	3.42	3.61
	σ(dB)	3.23	2.73	2.39	2.94	2.84	2.81
	RMSE(dB)	4.72	5.81	3.40	4.85	4.39	4.57
	R^2	0.90	0.85	0.95	0.90	0.92	0.90
QRT Adapted Hata-Okm	μ(dB)	3.08	4.70	2.00	4.04	3.51	3.33
	σ(dB)	2.75	2.98	1.79	3.51	3.08	2.76
	RMSE(dB)	4.05	5.49	2.64	5.25	4.59	4.27
	R^2	0.83	0.50	0.90	0.61	0.66	0.69
COST 231 Hata	μ(dB)	7.76	8.10	5.49	4.72	5.46	6.17
	σ(dB)	4.43	4.84	3.68	2.94	3.58	3.84
	RMSE(dB)	8.86	9.35	6.54	5.51	6.47	7.20
	R^2	0.65	0.60	0.83	0.87	0.82	0.74
Hata-Okm	μ(dB)	4.44	3.95	6.43	7.43	6.76	5.63
	σ(dB)	3.15	3.90	2.72	4.91	4.20	3.70
	RMSE(dB)	5.37	5.44	6.94	8.79	7.87	6.75
	R^2	0.70	0.51	0.27	-0.10	0.0056	0.14

It can be observed that the QRT adapted empirical models are closer to the test data than the standard empirical models. The neural network plots are zigzag in appearance, just like the test data, as demanded by the terrain clutter. However, there is no clear indication of which of the models is the closest to the test data. Nevertheless, results of comparison with Base 6 data as indicate in Table 4.10, show that the ANN-based models are slightly more accurate. The performance indicators across Base Stations 6 to 10 are presented in Table 4.10, generated by the

MATLAB code Appendix B4. Table 4.9 results show that on the average, the ANN-based models are slightly more accurate.

4.3.1.7 Comparison of Model Predictors by Random Training with one Base Station and Testing with another

The sequence is as follows: in a random manner, each of the neural network models is trained using data from one Base Station, and then tested using data obtained from a different Base Station.

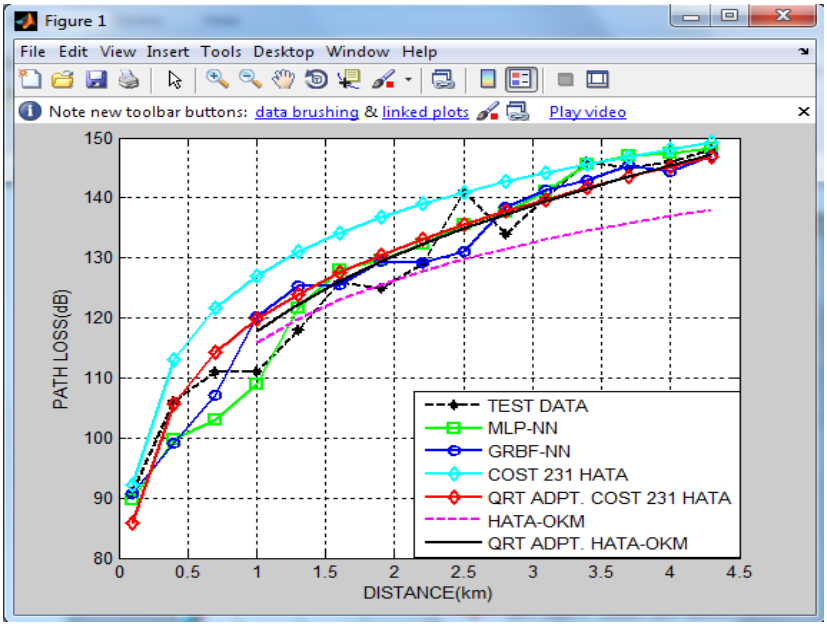


Figure 4.10: Training with BST8 and Testing with BST4 (Jos-Abuja Rural)

Figure 4.10, produced by Appendix B4 code, shows a graphical comparison for generalization of all the Model predictors, as a result of training the neural network models with Base Station 8 data and testing all models with Base Station 4 data. It is obvious that the existing empirical models are further away from the test data plot than the other models. However, there is no clear indication of the best predictor in terms of generalisation.

Table 4.11: Performance Comparison by Random Training with one Base Station and

Testing with another (Jos-Abuja Rural Area)

MODEL	STAT.	BST8 /BST4	BST1 /BST7	BST6 /BST2	BST3 /BST9	BST2 /BST10	BST8 /BST7	GEOM MEAN
MLP-NN	μ(dB)	3.00	5.06	3.81	3.42	4.03	5.01	3.98
	σ(dB)	2.25	3.52	3.97	3.13	2.90	4.89	3.34
	RMSE(dB)	3.71	6.10	5.41	4.57	4.90	6.89	5.16
	R^2	0.95	0.83	0.89	0.91	0.90	0.78	0.88
GRBF-NN	μ(dB)	3.63	3.78	2.76	3.43	4.15	4.65	3.69
	σ(dB)	3.32	2.62	2.26	2.20	3.02	2.73	2.66
	RMSE(dB)	4.84	4.55	3.52	4.04	5.07	5.34	4.52
	R^2	0.92	0.91	0.95	0.93	0.89	0.87	0.91
QRT Adapted COST 231 Hata	μ(dB)	3.53	5.47	5.40	3.78	3.33	5.47	4.39
	σ(dB)	2.72	2.76	2.91	2.79	2.88	2.76	2.80
	RMSE(dB)	4.40	6.08	6.09	4.64	4.34	6.08	5.21
	R^2	0.93	0.83	0.86	0.90	0.92	0.83	0.88
QRT Adapted Hata-Okm	μ(dB)	2.98	5.11	4.76	4.02	3.36	5.11	4.13
	σ(dB)	2.51	2.86	3.05	3.23	3.09	2.86	2.92
	RMSE(dB)	3.82	5.80	5.58	5.07	4.47	5.80	5.04
	R^2	0.89	0.45	0.77	0.63	0.68	0.45	0.62
COST 231 Hata	μ(dB)	6.43	8.10	8.63	4.72	5.46	8.10	6.74
	σ(dB)	5.22	4.84	5.65	2.94	3.58	4.84	4.40
	RMSE(dB)	8.17	9.35	10.21	5.51	6.47	9.35	7.99
	R^2	0.76	0.60	0.61	0.87	0.82	0.60	0.70
Hata-Okm	μ(dB)	6.00	3.95	4.98	7.43	6.76	3.95	5.35
	σ(dB)	4.12	3.90	4.32	4.91	4.20	3.90	4.21
	RMSE(dB)	7.18	5.44	6.47	8.79	7.87	5.44	6.75
	R^2	0.62	0.51	0.69	-0.01	0.0056	0.51	0.20

Table4.11 performance indicators for training with Base Station 8 and testing with Base Station 4 data show that the QRT adapted models are more accurate. Table 4.11 results show that on the average, the ANN-based models are slightly more accurate than the QRT technique by 0.25 dB.

4.3.1.8 Combined Performance Analysis based on Three Comparative Techniques

Table 4.12 presents the overall combined performance indicators of all predictors across the three techniques. Results show that the most accurate of all the predictors is the GRBF-NN model with the following indicators: $\mu=3.62dB$, $\sigma=2.47dB$, $RMSE=4.35dB$ and $R^2=0.80$. In contrast, the predictor with the worst performance is the COST 231 Hata with $\mu=6.13dB$, $\sigma=4.05dB$, $RMSE=7.3dB$ and $R^2=0.74$.

Table 4.12: Overall Performance Statistics based on all techniques (Jos-Abuja Rural)

MODEL	STATS.	Technique A	Technique B	Technique C	GEOM MEAN.
MLP-NN	μ(dB)	3.36	3.28	3.98	3.53
	σ(dB)	2.36	2.71	3.34	2.77
	RMSE(dB)	4.05	4.26	5.16	4.47
	R^2	0.67	0.91	0.88	0.81
GRBF-NN	μ(dB)	4.13	3.10	3.69	3.62
	σ(dB)	2.54	2.24	2.66	2.47
	RMSE(dB)	4.77	3.81	4.52	4.35
	R^2	0.61	0.93	0.91	0.80
QRT Adapted COST 231 Hata	μ(dB)	3.69	3.72	4.39	3.92
	σ(dB)	2.64	2.78	2.80	2.74
	RMSE(dB)	4.52	4.62	5.21	4.77
	R^2	0.91	0.90	0.88	0.90
QRT Adapted Hata-Okm	μ(dB)	3.49	3.42	4.13	3.67
	σ(dB)	2.68	2.66	2.92	2.75
	RMSE(dB)	4.35	4.29	5.04	4.55
	R^2	0.68	0.67	0.62	0.66
COST 231 Hata	μ(dB)	5.55	6.17	6.74	6.13
	σ(dB)	3.92	3.84	4.40	4.05
	RMSE(dB)	6.76	7.20	7.99	7.30
	R^2	0.77	0.74	0.70	0.74
Hata-Ok	μ(dB)	6.29	5.63	5.35	5.74
	σ(dB)	3.55	3.70	4.21	3.81
	RMSE(dB)	7.34	6.75	6.75	6.94
	R^2	0.10	0.14	0.20	0.14

On the average, the performance difference between the neural network predictors and the QRT adapted empirical models is negligible: the neural network predictors outperform the QRT

adapted empirical models by about 0.25dB. The closeness in performance is a testimony of the accuracy of the QRT.

4.3.2 The Urban Terrain (Abuja)

This is a metropolitan environment that matches Hata's description of an urban area as described in section 2.2.5. As a result, the empirical models considered are the COST 231 Walfisch-Ikegami, which has good prediction accuracy in urban areas, and the COST 231 Hata, which has correction factors for urban areas. Mobile network parameters obtained from the service provider, received power values obtained from Base Stations across the terrain and the corresponding computed path loss values are presented in Table II of appendix A.

The MLP-NN parameters determined suitable for this terrain are as follows:

i) Number of hidden layer neurons (with sigmoid type tangential activation) is 3

ii) 1 linearly activated output layer

iii) Training Algorithm: Levenberg-Marquardt back-propagation

iv) Number of iterations = 105

v) Error goal is 0.01

For the GRBF-NN, the parameters are as follows:

i) Spread constant is 0.501.

ii) ii) Error goal is 0.1

4.3.2.1 Testing the COST 231 Walfisch-Ikegami and the COST 231 Hata models for acceptability

The COST 231 Walfisch-Ikegami parameters considered for this terrain are as follows:

i) Average road width = 30 meters

ii) Average inter-building separation=2 x 30 = 60 meters

iii) Average number of floors= 7

iv) Average roof height=3 x number of floors + 3 = 24 meters

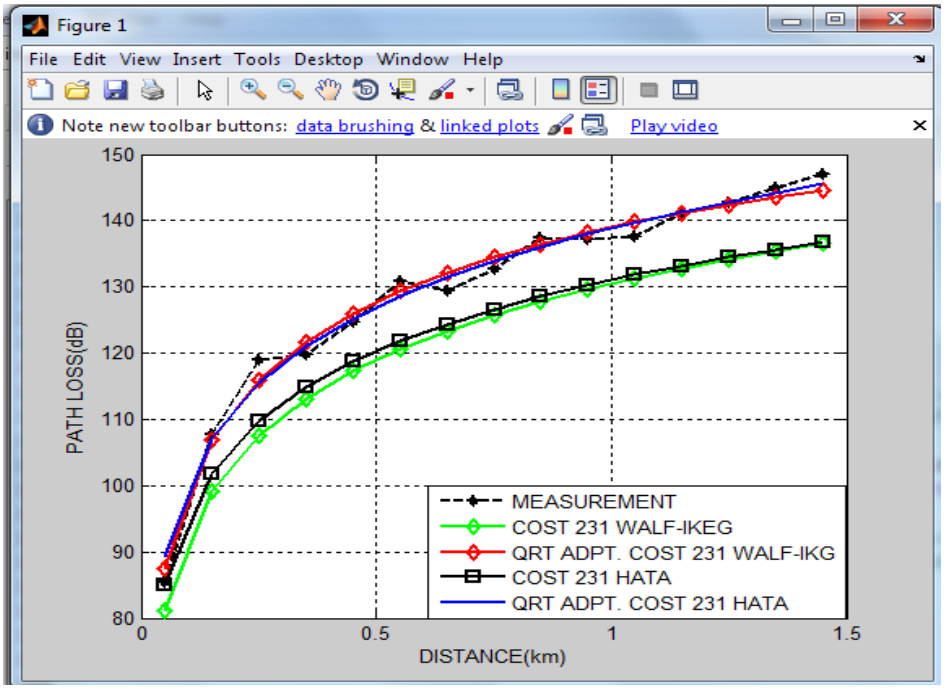


Figure 4.11: Comparison of the Empirical Models with Mean Measurements (Abuja)

Figure 4.11, based on Code in Appendix B5 shows a graphical performance evaluation of the COST 231 Walfisch-Ikegami (equation 2.39) and the COST 231 Hata (equation 2.37) models relative to the geometric mean of measurements. It is obvious that the two empirical models underestimate the path loss. Results in Table 4.13 show that the COST 231 Walfisch-Ikegami underestimates the path loss by a RMSE value of 8.64dB, while the COST 231 Hata does likewise by 7.6dB.

Table 4.13: Performance Evaluation of Empirical Models (Abuja)

PERFORMANCE STATISTICS	COST 231 Walfisch-Ikegami	COST 231 Hata
μ(dB)	8.08	6.95

σ(dB)	1.89	2.47
RMSE(dB)	8.64	7.60
R^2	0.72	0.78

4.3.2.2 Adapting the COST 231 Walfisch-Ikegami and the COST 231 Hata Models

i) Adapting the COST 231 Walfisch-Ikegami Model

The adapted COST 231 Walfisch-Ikegami Model expression based on the RAT is given by

$$\text{CWI}_{\text{RAT}}=L_{fs}-L_{msd}+L_{rts}+8.64 \quad (4.13)$$

The least squares expression for the geometric mean is

$$L_{LS}=93.11+78.68d-30.61d^2 \quad (4.14)$$

The quotients function obtained is Q(d)=1.0805-0.0155d

Therefore, the Quotients Function Modified COST 231 Walfisch-Ikegami Model expression is

$$\text{CWI}_{\text{QRT}}=\text{Q(d) x L}_{\text{p}}=(1.0805-0.0157\text{d})\ \text{x}(L_{fs}-L_{msd}+L_{rts}) \quad (4.15)$$

ii) Adapting the COST 231 Hata Model

The RAT-adapted COST 231 Hata Model expression is given by

$$\text{CH}_{\text{\_RAT}}=46.3+33.9logf-13.82logh_B-a(h_R)+(44.9-6.55logh_B)logd+7.6$$

$$=53.9+33.9logf-13.82logh_B-a(h_R)+(44.9-6.55logh_B)logd \quad (4.16)$$

The quotients function obtained is Q(d)=1.05+0.0097d=1.05 since 0.0097d is negligible. Therefore, the QRT adapted COST 231 Hata Model expression is given by

$$\text{CH}_{\text{QRT}}=\text{Q(d) x L}_{\text{p}}$$

$$=1.05\ \text{x}\ (46.3+33.9logf-13.82logh_B-a(h_R)+(44.9-6.55logh_B)logd$$

$$=48.61+35.6logf-14.51logh_B-1.05a(h_R)+(47.15-6.88logh_B)logd \quad (4.17)$$

4.3.2.3 Comparison of Adaptation Techniques

A performance comparison of the two Adaptation techniques based on code in Appendix B5, is presented in Table 4.14. Results show that the QRT outperforms the RAT by 0.13dB in COST 231 Walfisch-Ikegami Model Adaptation, and by 0.67dB in COST 231 Hata Model Adaptation.

Table 4.14: Comparison of Adaptation Techniques (Abuja)

PERFORMANCE STATISTICS	Adapted COST 231 Walfisch-Ikegami		Adapted COST 231 Hata	
	RAT Adaptation	QRT Adaptation	RAT Adaptation	QRT Adaptation
μ(dB)	1.52	1.55	1.93	1.45
σ(dB)	1.05	0.85	1.51	1.11
RMSE(dB)	1.89	1.76	2.47	1.80
R^2	0.99	0.99	0.98	0.99

4.3.2.4 Performance Comparison of RAT-Adapted and QRT Adapted Empirical Models

Figure 4.12 shows a convergence of the RAT-adapted Walfisch-Ikegami (equation 4.13), the QRT adapted Walfisch-Ikegami (equation 4.15), the RAT-adapted COST 231 Hata (equation 4.16), and the QRT adapted COST 231 Hata (equation 4.17).

Table 4.15 presents the performance each of the adapted models relative to Base Stations 6 to 10. Results in Table 4.15 indicate that on the average, the QRT adapted COST 231 Walfisch-Ikegami and the QRT adapted COST 231 Hata models outperform the RAT-adapted counterparts by 0.17dB and 0.49dB respectively.

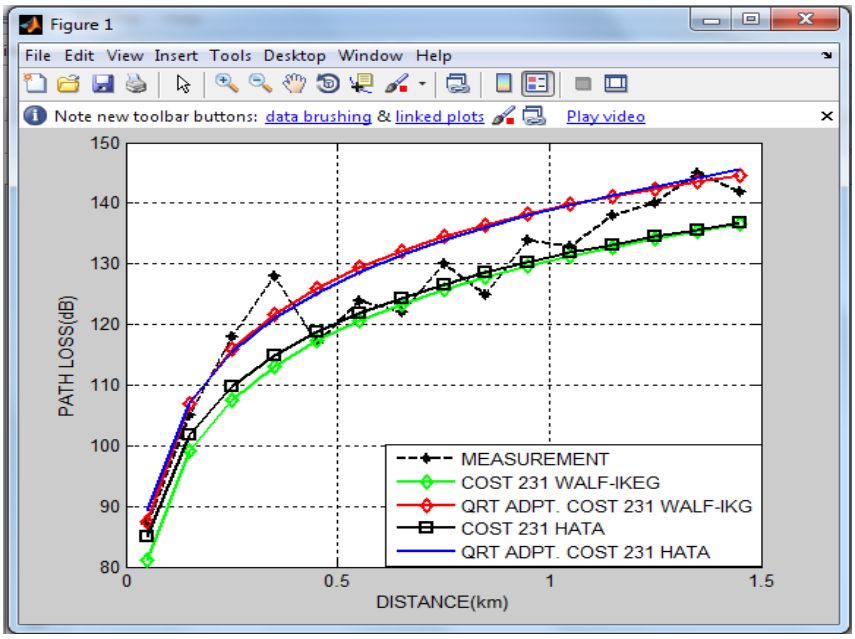


Figure 4.12: Comparison of RAT and QRT Adapted Empirical models with BST 6 data (Abuja)

Table 4.15: Performance Comparison of RAT and QRT adapted Empirical Models (Abuja)

MODEL	STATS	BST6	BST7	BST8	BST9	BST10	MEAN
RAT-Adapted COST231 Walf-Ikeg	μ(dB)	5.04	3.81	4.56	3.90	4.96	4.43
	σ(dB)	3.13	2.69	3.91	2.67	3.02	3.05
	RMSE(dB)	5.88	4.62	5.92	4.68	5.76	5.34
	R^2	0.84	0.92	0.85	0.90	0.84	0.87
RAT-Adapted COST 231 Hata	μ(dB)	5.03	4.11	4.76	3.93	4.98	4.54
	σ(dB)	3.12	3.07	3.89	2.73	2.93	3.12
	RMSE(dB)	5.87	5.07	6.07	4.73	5.72	5.47
	R^2	0.84	0.90	0.84	0.90	0.84	0.86
QRT Adapted COST 231 Walf-Ikeg	μ(dB)	4.80	3.52	4.23	3.62	4.80	4.16
	σ(dB)	3.33	2.65	4.13	2.75	3.06	3.14
	RMSE(dB)	5.77	4.35	5.82	4.49	5.63	5.17
	R^2	0.84	0.93	0.86	0.91	0.84	0.88
QRT Adapted COST 231 Hata	μ(dB)	4.80	3.39	4.12	3.68	4.70	4.10
	σ(dB)	2.98	2.45	3.77	2.56	2.83	2.88
	RMSE(dB)	5.60	4.13	5.50	4.43	5.44	4.98
	R^2	0.85	0.94	0.87	0.91	0.85	0.88

4.3.2.5 Comparison of ANN-based models with QRT Adapted Empirical Models using the

Training-Validation-Testing Technique

Figure 4.13, generated by the code in Appendix B6, presents a sample graphical analysis of the Model predictors using Base Station 1 data. The results in Table 4.16 show that the MLP-NN model is just slightly more accurate than the QRT adapted models. On the average, the ANN-based models are more accurate than the QRT adapted models by an RMSE value of about 0.52dB.

Table 4.16: Performance Comparison using the Training-Validation-Testing Technique (Jos-Abuja Rural)

MODEL	STAT.	BS1	BS2	BS3	BS4	BS5	BS6	BS7	BS8	BS9	BS10	GEOM. MEAN
MLP-NN	μ(dB)	3.17	6.06	4.20	2.20	2.13	5.43	2.05	4.94	3.67	3.76	3.51
	σ(dB)	2.75	5.15	3.30	1.93	0.67	3.05	0.86	2.85	2.55	3.74	2.31
	RMSE(dB)	4.01	7.62	5.14	2.80	2.21	6.08	2.19	5.56	4.32	5.03	4.16
	R^2	0.84	0.25	0.58	0.64	0.83	0.42	0.95	0.48	0.70	0.28	0.55
GRBF-NN	μ(dB)	0.76	4.64	4.30	2.81	4.14	5.37	2.42	5.13	3.88	3.88	3.35
	σ(dB)	0.49	2.30	2.70	1.76	2.57	4.55	1.36	2.82	2.52	2.68	2.09
	RMSE(dB)	0.88	5.08	4.93	3.23	4.74	6.74	2.71	5.72	4.48	4.56	3.85
	R^2	0.99	0.67	0.61	0.51	0.24	0.29	0.92	0.45	0.68	0.41	0.53
QRT Adapted COST 231 Walf-Ikeg	μ(dB)	2.66	4.40	2.44	3.62	2.82	4.80	3.52	4.23	3.62	4.80	3.60
	σ(dB)	2.59	4.17	1.67	2.48	2.82	3.33	2.65	4.13	2.75	3.06	2.88
	RMSE(dB)	3.66	5.97	2.93	4.35	3.92	5.77	4.35	5.82	4.49	5.63	4.57
	R^2	0.95	0.86	0.97	0.92	0.94	0.84	0.93	0.86	0.91	0.84	0.90
QRT Adapted COST 231 Hata	μ(dB)	2.48	4.21	2.58	3.74	3.08	4.80	3.39	4.12	3.68	4.70	3.59
	σ(dB)	2.34	4.30	1.37	2.74	2.80	2.98	2.45	3.77	2.56	2.83	2.71
	RMSE(dB)	3.36	5.91	2.90	4.58	4.10	5.60	4.13	5.50	4.43	5.44	4.49
	R^2	0.96	0.87	0.97	0.91	0.93	0.85	0.94	0.87	0.91	0.85	0.91
COST 231 Walf-Ikeg	μ(dB)	7.50	8.87	7.23	8.90	10.36	5.38	4.90	5.04	5.65	4.53	6.57
	σ(dB)	3.61	4.85	2.87	4.55	3.54	3.77	2.79	2.49	3.32	3.44	3.45
	RMSE(dB)	8.27	10.03	7.74	9.92	10.91	6.50	5.59	5.59	6.50	5.62	7.43
	R^2	0.76	0.61	0.76	0.57	0.52	0.80	0.88	0.87	0.81	0.84	0.73
COST 231 Hata	μ(dB)	6.46	7.97	6.09	7.89	9.23	4.65	3.94	4.19	4.68	3.62	5.59
	σ(dB)	4.00	4.76	3.28	4.35	3.82	3.30	3.20	2.63	3.10	3.09	3.50
	RMSE(dB)	7.53	9.20	6.87	8.93	9.94	5.64	5.01	4.90	5.56	4.69	6.58
	R^2	0.80	0.67	0.81	0.66	0.60	0.85	0.91	0.90	0.86	0.89	0.79

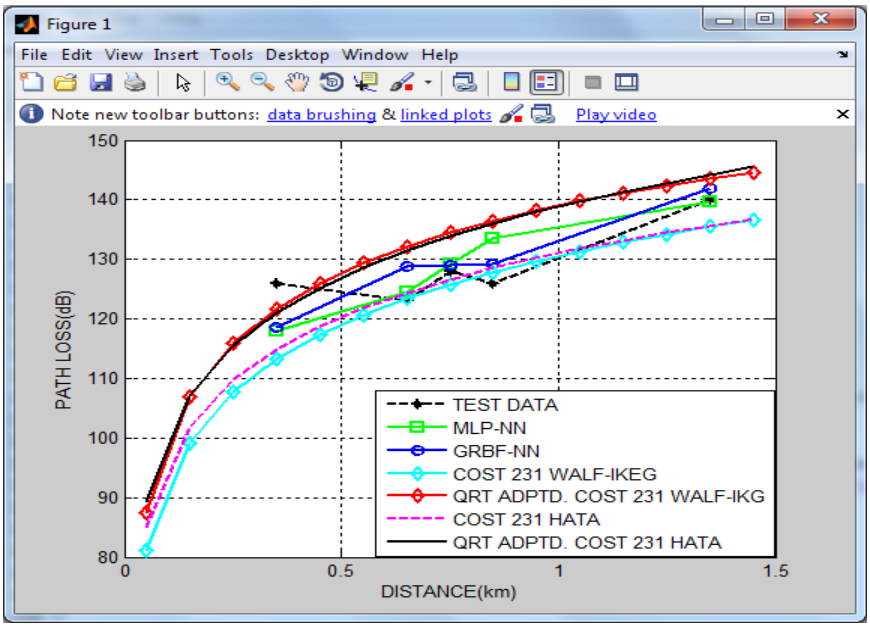


Figure 4.13: Graphical Comparison of Model Predictors on BST1using the Training-Validation-Testing Technique (Abuja)

4.3.2.6 Comparison of ANN-based models with the QRT Adapted Empirical Models using the 50%Training and 50% Testing Technique

Data from base stations 1 to 5 is classified as training set while base stations 6 to 10 data as testing set. The geometric mean of the training set at each receiver-transmitter separation is obtained using equation (3.9), and then used to train the neural networks predictors.

Figure 4.14, based on code in Appendix B7, is a sample graphical comparison of the Model predictors with test data from Base Station 6. The plots show that the neural network models are closer to the test data than the empirical models. A summary of the results obtained based on this comparative technique is presented in Table 4.17. The table shows the statistical performance of each of the Model Predictors relative to the test data from Base Stations 6 to 10. Results indicate that the ANN-based models are more accurate than the QRT adapted empirical models by more than 1.5dB.

Table 4.17: Performance Comparison using the 50% Training and 50% Testing Technique (Abuja)

MODEL	STAT.	GM /BST6	GM/ BST7	GM/ BST8	GM/ BST9	GM/ BST10	GEOM. MEAN
MLP-NN	μ(dB)	2.85	2.06	2.67	2.56	2.45	2.50
	σ(dB)	2.37	1.38	2.03	1.75	1.96	1.87
	RMSE(dB)	3.66	2.45	3.31	3.07	3.10	3.09
	R^2	0.94	0.98	0.95	0.96	0.95	0.96
GRBF-NN	μ(dB)	3.07	1.90	3.50	2.20	2.84	2.64
	σ(dB)	3.23	1.46	2.50	1.38	2.61	2.12
	RMSE(dB)	4.37	2.37	4.26	2.57	3.80	3.36
	R^2	0.91	0.98	0.92	0.97	0.93	0.94
QRT Adapted COST 231 Walf-Ikeg.	μ(dB)	4.80	3.52	4.23	3.62	4.80	4.16
	σ(dB)	3.33	2.65	4.13	2.75	3.06	3.14
	RMSE(dB)	5.77	4.35	5.82	4.49	5.63	5.17
	R^2	0.84	0.93	0.86	0.91	0.84	0.88
QRT Adapted COST 231 Hata	μ(dB)	4.80	3.39	4.12	3.68	4.70	4.10
	σ(dB)	2.98	2.45	3.77	2.56	2.83	2.88
	RMSE(dB)	5.60	4.13	5.50	4.43	5.44	4.98
	R^2	0.85	0.94	0.87	0.91	0.85	0.88
COST 231 Walf-Ikeg.	μ(dB)	5.38	4.90	5.04	5.65	4.53	5.08
	σ(dB)	3.77	2.79	2.49	3.32	3.44	3.13
	RMSE(dB)	6.50	5.59	5.59	6.50	5.62	5.94
	R^2	0.80	0.88	0.87	0.81	0.84	0.84
COST 231 Hata	μ(dB)	4.65	3.94	4.19	4.68	3.62	4.20
	σ(dB)	3.30	3.20	2.63	3.10	3.09	3.05
	RMSE(dB)	5.64	5.01	4.90	5.56	4.69	5.15
	R^2	0.85	0.91	0.90	0.86	0.89	0.88

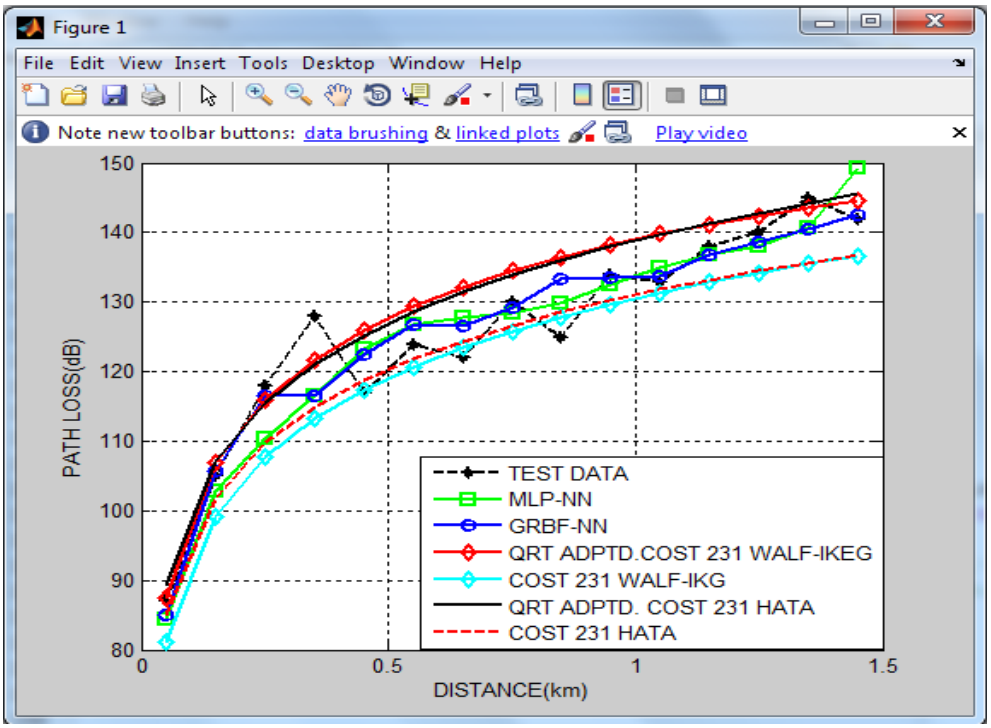


Figure 4.14: Graphical Comparison of Models with BST6 using the 50% Training and 50% Testing Technique *(Abuja)*

4.3.2.7 Comparison of ANN-based models with the QRT Adapted Empirical Models by Random Training with one Base Station and Testing with another

Figure 4.15, based on code in Appendix B7, shows a graphical comparison for generalization of all the Model predictors, as a result of training the neural network models with Base Station 5 data and testing with Base Station 10 data. It is obvious that the ANN-based models are closer to the test output than the rest. A summary of the generalisation based performances of the predictors compared with the target test patterns is presented in Table 4.18. On the average, the neural network models generalise better than the QRT adapted empirical models.

Table 4.18: Performance Comparison by Random Training with one Base Station and

Testing with another (Abuja)

MODEL	STAT.	BST7 /BST1	BST5 /BST10	BST8 /BST4	BST2 /BST6	BST3 /BST9	BST10 /BST6	GEOM MEAN
MLP-NN	μ(dB)	2.06	3.48	4.49	3.63	3.36	3.57	3.35
	σ(dB)	1.80	2.92	5.17	2.55	2.25	2.49	2.70
	RMSE(dB)	2.69	4.48	6.72	4.39	4.00	4.30	4.28
	R^2	0.97	0.90	0.80	0.91	0.93	0.91	0.90
GRBF-NN	μ(dB)	2.66	2.84	4.33	2.99	3.16	2.07	2.94
	σ(dB)	2.19	2.24	3.39	2.77	1.63	0.90	2.02
	RMSE(dB)	3.40	3.57	5.43	4.01	3.53	2.24	3.58
	R^2	0.96	0.94	0.87	0.92	0.94	0.98	0.93
QRT Adapted COST 231 Walf-Ikeg.	μ(dB)	2.66	4.80	3.62	4.80	3.62	4.80	3.96
	σ(dB)	2.59	3.06	2.48	3.33	2.75	3.33	2.90
	RMSE(dB)	3.66	5.63	4.35	5.77	4.49	5.77	4.87
	R^2	0.95	0.84	0.92	0.84	0.91	0.84	0.88
QRT Adapted COST 231 Hata	μ(dB)	2.48	4.70	3.74	4.80	3.68	4.80	3.93
	σ(dB)	2.34	2.83	2.74	2.98	2.56	2.98	2.73
	RMSE(dB)	3.36	5.44	4.58	5.60	4.43	5.60	4.76
	R^2	0.96	0.85	0.91	0.85	0.91	0.85	0.89
COST 231 Walf-Ikeg	μ(dB)	7.50	4.53	8.90	5.38	5.65	5.65	6.11
	σ(dB)	3.61	3.44	4.55	3.77	3.32	3.32	3.65
	RMSE(dB)	8.27	5.62	9.92	6.50	6.50	6.50	7.08
	R^2	0.76	0.84	0.57	0.80	0.81	0.81	0.76
COST 231 Hata	μ(dB)	6.46	3.62	7.89	4.65	4.68	4.68	5.16
	σ(dB)	4.00	3.09	4.35	3.30	3.10	3.10	3.46
	RMSE(dB)	7.53	4.69	8.93	5.64	5.56	5.56	6.17
	R^2	0.80	0.89	0.66	0.85	0.86	0.86	0.81

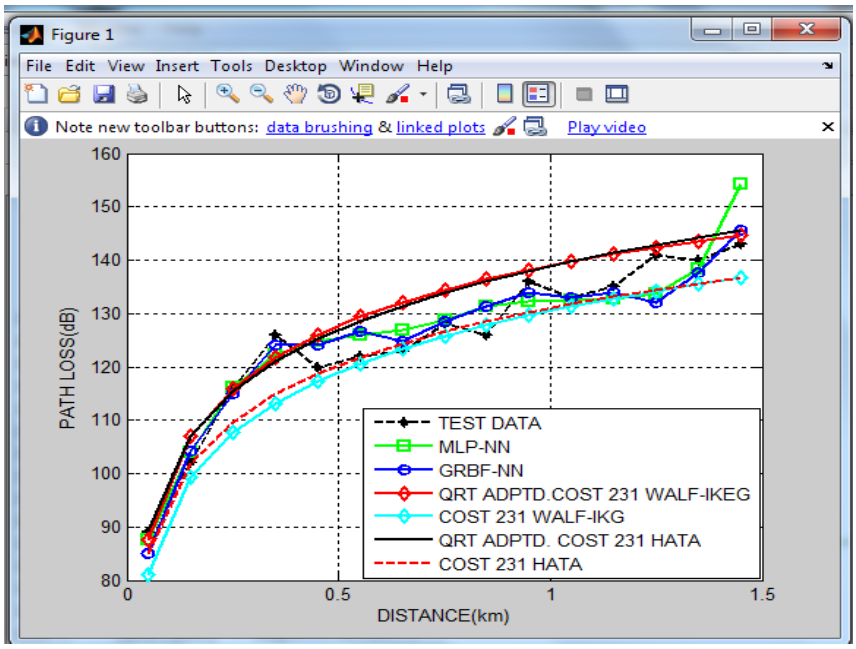


Figure 4.15: Comparison by Training with BST5 and Testing with BST10 (Abuja)

4.3.2.8 Combined Performance Analysis across the three Techniques

The combined performance indicators of the Model Predictors across the three techniques are presented in Table 4.19. Results indicate that MLP-NN predictor is more accurate than the QRT adapted empirical models by about 0.45dB. On the average, the ANN-based models are more accurate than QRT adapted empirical models by about 1.1dB.

Table 4.19: Mean Performance based on three Techniques (Abuja)

MODEL	STAT.	TECHNIQUE A	TECHNIQUE B	TECHNIQUE C	GEOM MEAN
MLP-NN	μ(dB)	3.51	2.50	3.35	3.09
	σ(dB)	2.31	1.87	2.70	2.27
	RMSE(dB)	4.16	3.09	4.28	3.80
	R^2	0.55	0.96	0.90	0.78
GRBF-NN	μ(dB)	3.35	2.64	2.94	2.96
	σ(dB)	2.09	2.12	2.02	2.08
	RMSE(dB)	3.85	3.36	3.58	3.59
	R^2	0.53	0.94	0.93	0.77
QRT Adapted COST 231 Walf-Ikeg	μ(dB)	3.60	4.16	3.96	3.90
	σ(dB)	2.88	3.14	2.90	2.97
	RMSE(dB)	4.57	5.17	4.87	4.86
	R^2	0.90	0.88	0.88	0.89
QRT Adapted COST 231 Hata	μ(dB)	3.59	4.10	3.93	3.87
	σ(dB)	2.71	2.88	2.73	2.77
	RMSE(dB)	4.49	4.98	4.76	4.74
	R^2	0.91	0.88	0.89	0.89
COST 231 Walf-Ikeg	μ(dB)	6.57	5.08	6.11	5.89
	σ(dB)	3.45	3.13	3.65	3.40
	RMSE(dB)	7.43	5.94	7.08	6.79
	R^2	0.73	0.84	0.76	0.78
COST 231 Hata	μ(dB)	5.59	4.20	5.16	4.95
	σ(dB)	3.50	3.05	3.46	3.33
	RMSE(dB)	6.58	5.15	6.17	5.94
	R^2	0.79	0.88	0.81	0.83

4.3.3 The Semi-Urban Terrain (Maiduguri)

This terrain matches Hata's description of "Small City" as described in section 2.2.5. As a result, the empirical models considered are as follows: the COST 231 Walfisch-Ikegami, which has good prediction accuracy in urban areas, and the COST 231 Hata, which has correction factors for semi-urban areas. Mobile network parameters obtained from the service provider, received

power values obtained from Base Stations across the terrain and the corresponding computed path loss values are presented in Table III of appendix A

The MLP-NN parameters determined suitable for this terrain are as follows:

i) Number of neurons in hidden layer (with sigmoid type tang. activation) is 3

ii) 1 linearly activated output layer

i) Training Algorithm: Levenberg-Marquardt back-propagation

ii) Number of iteration = 105

iii) Error goal=0.001

For the GRBF-NN, the parameters are as follows:

i) Spread constant is 0.51

ii) Error goal=0.01

4.3.3.1 Acceptability Test for the COST 231 Walfisch-Ikegami and COST 231 Hata

The COST 231 Walfisch-Ikegami parameters considered for this terrain are as follows:

i) Average road width = 20 meters

ii) Average inter-building separation=2 x 20 = 40 meters

iii) Average number of floors= 3

iv) Average roof height=3 x number of floors + 3 = 12 meters

Figure 4.16, generated by the code in Appendix B8, presents a graphical comparison of the above mentioned empirical models with the geometric mean of measurements in Table III of Appendix A. It can be clearly observed that, while there is a convergence of the COST 231 Hata plot (equation 2.37) with that of the measured data at some points, the COST 231 Walfisch-Ikegami (equation 2.39) is far off the mark. Results in Table 4.20 show that the COST 231

Walfisch-Ikegami model with a RMSE of 11.99dB is beyond the acceptable limit, while the COST 231 Hata is acceptable with an RMSE value of 4.47dB and a good fit value, R^2 of 0.93.

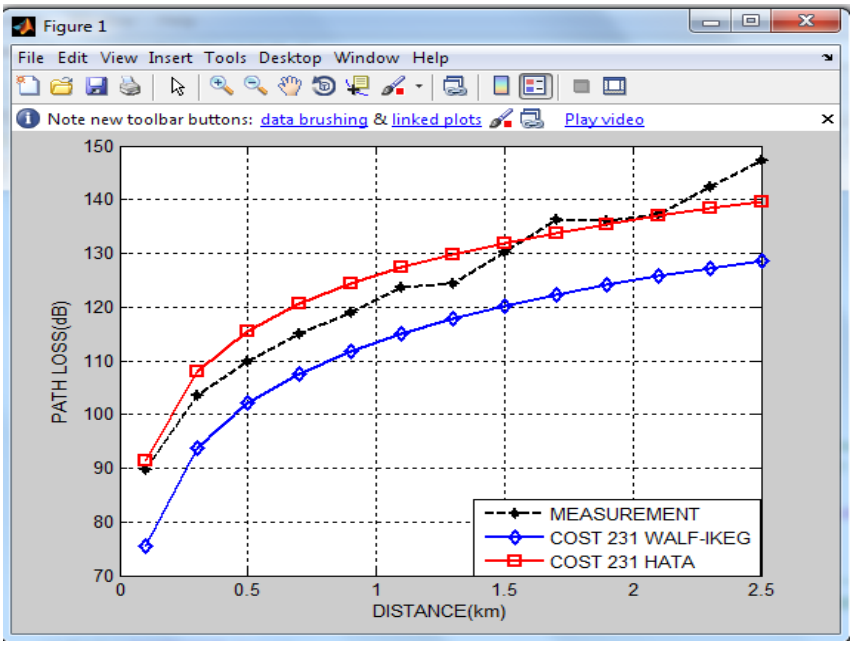


Figure 4.16: Comparison of Empirical Models with Mean Measurements (Maiduguri)

Table 4.20: Performance Test for reliability Empirical Models (Maiduguri)

PERFORMANCE STATISTICS	COST Walfisch-Ikegami	COST 231 Hata
μ(dB)	10.97	3.74
σ(dB)	3.66	2.21
RMSE(dB)	11.99	4.47
R^2	0.47	0.93

4.2.3.2 Adapting the COST 231 Walfisch-Ikegami and the COST 231 Hata Models

The RAT adapted COST 231 Walfisch-Ikegami expression is given by

$$CWI_{RAT} = L_{fs} - L_{msd} + L_{rts} + 11.99 \quad (4.18)$$

The COST 231 Walfisch-Ikegami QRT adaptation factor obtained is Q(d)=1.1. Therefore, the QRT adapted COST 231 Walfisch-Ikegami model for this terrain is given by

$$CWI_{QRT} = Q(d) \times L_p = 1.1 \times (L_{fs} - L_{msd} + L_{rts})$$

$$=1.1L_{fs} - 1.1L_{msd} + 1.1L_{rts} \quad (4.19)$$

Although the COST 231 Hata is acceptable, its performance is further enhanced through adaptation for the sake of adaptation comparisons. The RAT-adapted, expression is given by

$$CH_{\_RAT}=46.3 + 33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd + 4.47$$

$$=50.77+33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd \quad (4.20)$$

The quotients function obtained is Q(d)= 0.9436+ 0.0336d. Therefore, the QRT adapted COST 231 Hata Model expression is given by

$$CH_{QRT}=Q(d) \text{ x } L_p \quad (4.21)$$

where,

$$Lp=46.3 + 33.9logf - 13.82logh_B - a(h_R) + (44.9 - 6.55logh_B)logd$$

4.3.3.3 Comparison of Adaptation Techniques

A performance comparison of the two adaptation techniques is presented in Table 4.21. Results show that the QRT outperforms the RAT by 0.13dB in COST 231 Walfisch-Ikegami adaptation, and by 2.83 in COST 231 Hata.

Table 4.21: Comparison of Adaptation Techniques (Maiduguri)

PERFORMANCE STATISTICS	Adapted COST 231 Walfisch-Ikegami		Adapted COST 231 Hata	
	RAT Adaptation	QRT Adaptation	RAT Adaptation	QRT Adaptation
μ(dB)	3.16	3.01	3.6	1.85
σ(dB)	1.91	1.92	3.56	1.11
RMSE(dB)	3.66	3.53	4.97	2.14
R^2	0.95	0.95	0.9	0.98

4.3.3.4 Performance Comparison of RAT Adapted and QRT Adapted Empirical Models

Figure 4.17, based on code in Appendix B8, shows the performance of all adapted empirical models relative to Base Station 8 data. Table 4.22 presents the performance of each of the adapted models relative to Base Stations 5 to 8. Results in Table 4.22 indicate that on the average, the RAT-adapted COST 231 Walfisch-Ikegami (equation 4.18) outperforms the QRT adapted counterpart (equation 4.19) by 0.32dB, while QRT adapted COST 231 Hata outperforms the RAT-adapted COST 231 Hata 1.45dB.

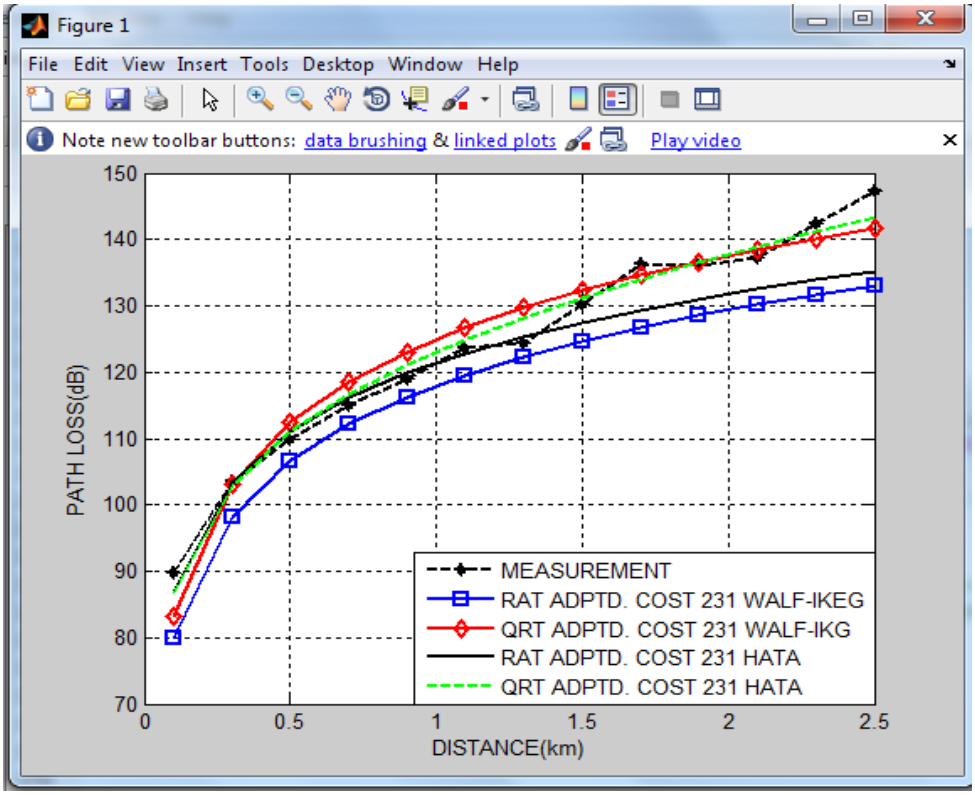


Figure 4.17: Comparison of RAT and QRT Adapted Empirical models relative to BST 8 data (Maiduguri)

Table 4.22: Performance Comparison of RAT and QRT adapted Empirical Models (Maiduguri)

MODEL	STATS	BST5	BST6	BST7	BST8	MEAN

RAT-Adapted COST231 Walf-Ikeg	μ(dB)	2.01	2.54	3.83	3.75	2.93
	σ(dB)	1.62	2.04	4.12	3.46	2.62
	RMSE(dB)	2.54	3.21	5.51	5.01	3.87
	R^2	0.97	0.96	0.89	0.90	0.93
RAT-Adapted COST 231 Hata	μ(dB)	4.35	3.30	4.57	4.73	4.20
	σ(dB)	1.81	3.54	5.28	4.37	3.49
	RMSE(dB)	4.69	4.74	6.83	6.33	5.57
	R^2	0.89	0.91	0.84	0.84	0.87
QRT Adapted COST 231 Walf-Ikeg	μ(dB)	2.59	2.68	3.62	3.76	3.12
	σ(dB)	2.30	1.92	4.09	3.75	2.87
	RMSE(dB)	3.40	3.25	5.34	5.20	4.19
	R^2	0.94	0.96	0.90	0.89	0.92
QRT Adapted COST 231 Hata	μ(dB)	μ(dB)	2.98	2.01	2.87	3.28
	σ(dB)	σ(dB)	1.71	1.91	3.26	2.61
	RMSE(dB)	RMSE(dB)	3.41	2.72	4.25	4.12
	R^2	R2	0.94	0.97	0.94	0.93

4.3.3.5Comparison of ANN-based models with the QRT Adapted Empirical Models using the Training-Validation-Testing Technique

Figure 4.18, based on code in Appendix B9, is a sample graphical comparison of all model predictors on Base Station 1. It is obvious that there is a convergence between the two neural network predictors, which are also convergent with the test data at certain points. Figure 4.18 also shows a convergence of the QRT adapted COST 231 Walfisch-Ikegami (equation 4.19) and the COST 231 Hata model (equation 2.37). Results of Base Station 1 analysis presented in Table 4.23 indicate that the QRT adapted COST 231 Walfisch-Ikegami is close in performance to the MLP-NN based model. The GRBF-NN is the better of the two neural network models.

Table 4.23: Performance Comparison using the Training-Validation-Testing Technique (Maiduguri)

MODEL	STAT.	BST1	BST2	BST3	BST4	BST5	BST6	BST7	BST8	GEOM. MEAN

MLP-NN	μ(dB)	4.02	4.68	3.40	5.31	2.41	2.91	3.70	4.12	3.71
	σ(dB)	2.59	1.23	1.19	3.71	1.61	2.11	3.22	1.55	1.98
	RMSE(dB)	4.60	4.80	3.55	6.20	2.78	3.44	4.63	4.33	4.18
	R^2	0.63	0.62	0.81	0.06	0.71	0.69	0.31	0.43	0.43
GRBF-NN	μ(dB)	3.36	3.01	6.77	5.10	2.85	2.94	2.75	1.89	3.33
	σ(dB)	3.29	0.91	3.93	3.06	2.33	2.32	1.07	1.30	2.01
	RMSE(dB)	4.41	3.11	7.58	5.74	3.50	3.56	2.91	2.20	3.84
	R^2	0.66	0.84	0.14	0.20	0.54	0.67	0.73	0.85	0.49
QRT Adapted COST 231 Walf-Ikeg.	μ(dB)	3.95	3.78	3.40	3.89	2.59	2.68	3.62	3.76	3.42
	σ(dB)	2.95	2.29	2.65	3.78	2.30	1.92	4.09	3.75	2.87
	RMSE(dB)	4.86	4.38	4.25	5.32	3.40	3.25	5.34	5.20	4.43
	R^2	0.91	0.92	0.94	0.88	0.94	0.96	0.90	0.89	0.92
COST 231 Walf-Ikeg	μ(dB)	10.08	10.24	12.08	11.06	11.93	11.08	11.31	11.55	11.15
	σ(dB)	5.00	4.16	4.61	4.47	2.64	3.20	5.69	5.20	4.25
	RMSE(dB)	11.17	10.99	12.87	11.86	12.20	11.50	12.57	12.58	11.95
	R^2	0.54	0.49	0.41	0.39	0.25	0.44	0.44	0.35	0.40
COST 231 Hata	μ(dB)	4.97	4.11	3.93	3.86	1.67	3.03	4.28	3.95	3.57
	σ(dB)	3.06	2.54	3.33	3.78	1.92	2.33	4.42	3.62	3.02
	RMSE(dB)	5.78	4.78	5.07	5.30	2.49	3.77	6.03	5.26	4.65
	R^2	0.88	0.90	0.91	0.88	0.97	0.94	0.87	0.89	0.90

The performance indicators of the predictors across all the Base Stations based on this comparative technique are presented in Table 4.23. The results show the difference in performance between the QRT adapted COST 231 Walfisch-Ikegami and the MLP-NN based model is just about 0.25dB, with the QRT adapted COST 231 Walfisch-Ikegami model having the best fit of all. The GRBF-NN predictor is the best of all with an average RMSE value of 3.84dB and R^2 of 0.49.

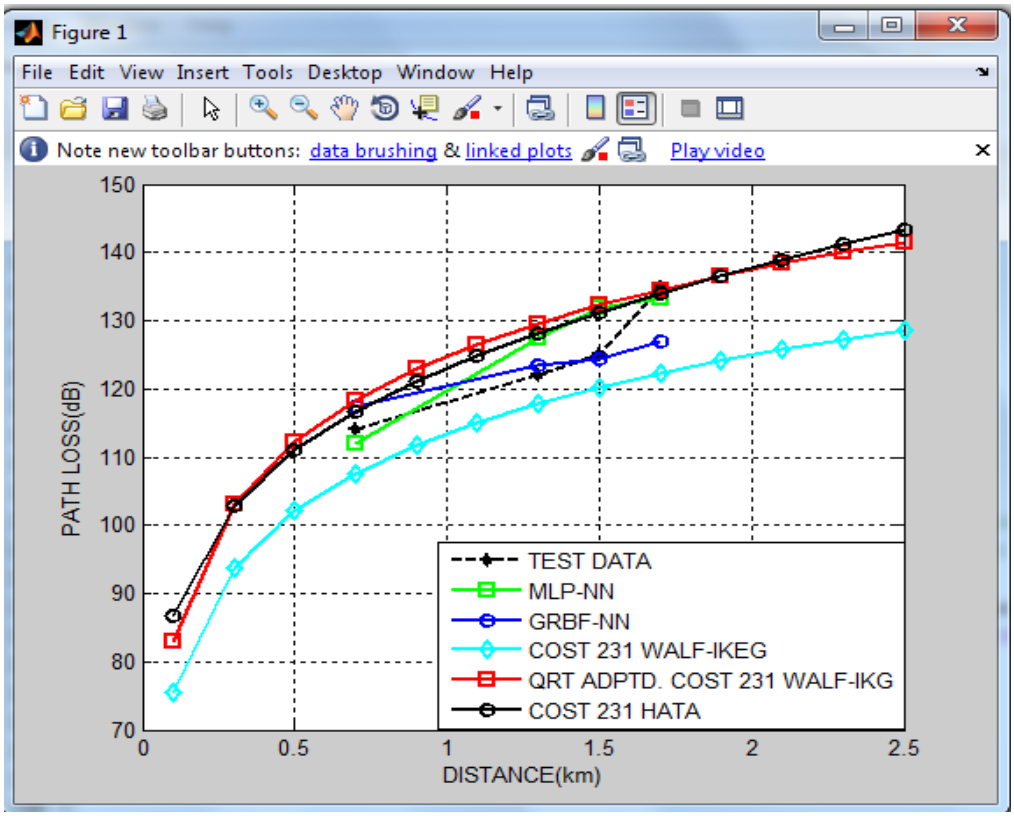


Figure 4.18: Comparison of Neural Network Models on BST1 Data using the Training-Validation-Testing Technique *(Maiduguri)*

4.3.3.6 Comparison of ANN-based models with the QRT Adapted Empirical Models using the 50% Training and 50% Testing Technique

Data from base stations 1 to 4 is classified as training set while base stations 5 to 8 data as testing set. The geometric mean of the training set at each receiver-transmitter separation is obtained using equation (3.9), and then used to train the neural network models. Figure 4.19, generated by the code in Appendix B10, is a sample graphical comparison of the geometric mean trained neural network models and the empirical models, with test data from Base station 5. It can be observed that the standard COST 231 Walfisch-Ikegami model (equation 2.39) is further away from the test data than the other models. The performance indicators in Table 4.24 show that the QRT adapted COST 231 Walfisch-Ikegami model has the best performance on

The mean performance across Base Stations 5 to 8 shows that the difference in performance between the QRT adapted COST 231 Walfisch-Ikegami and the MLP-NN based model is just about 0.15dB.

Table 4.24: Performance Comparison using the 50% Training and 50% Testing Technique (Maiduguri)

MODEL	STATS.	GM /BST5	GM/ BST6	GM/ BST7	GM/ BST8	GEOM MEAN
MLP-NN	μ(dB)	4.29	3.11	2.68	2.94	3.20
	σ(dB)	2.61	3.35	2.15	2.07	2.50
	RMSE(dB)	4.96	4.47	3.38	3.55	4.04
	R^2	0.88	0.92	0.96	0.95	0.92
GRBF-NN	μ(dB)	3.41	2.21	2.41	2.67	2.64
	σ(dB)	1.83	1.75	2.03	1.44	1.75
	RMSE(dB)	3.84	2.78	3.10	3.00	3.15
	R^2	0.93	0.97	0.97	0.96	0.96
QRT Adapted COST 231 Walf-Ikeg.	μ(dB)	2.59	2.68	3.62	3.76	3.12
	σ(dB)	2.30	1.92	4.09	3.75	2.87
	RMSE(dB)	3.40	3.25	5.34	5.20	4.19
	R^2	0.94	0.96	0.90	0.89	0.92
COST 231 Walf-Ikeg	μ(dB)	11.93	11.08	11.31	11.55	11.46
	σ(dB)	2.64	3.20	5.69	5.20	3.98
	RMSE(dB)	12.20	11.50	12.57	12.58	12.20
	R^2	0.25	0.44	0.44	0.35	0.36
COST 231 Hata	μ(dB)	1.67	3.03	4.28	3.95	3.04
	σ(dB)	1.92	2.33	4.42	3.62	2.91
	RMSE(dB)	2.49	3.77	6.03	5.26	4.15
	R^2	0.97	0.94	0.87	0.89	0.92

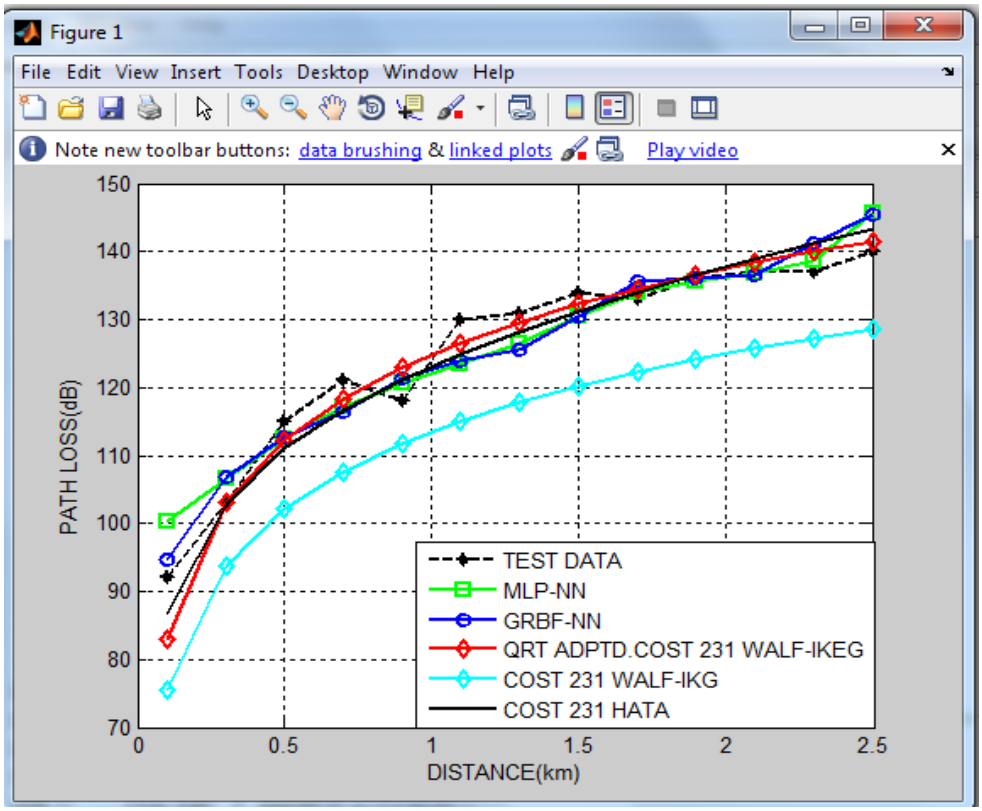


Figure 4.19: Graphical Comparison of Models with BST5 Measurements using the 50% Training and 50% Testing Technique *(Maiduguri)*

4.3.3.7 Comparison of of ANN-based models with the QRT Adapted Empirical Models by Random Training with one Base Station and Testing with another

Figure 4.20, generated by the code in Appendix B10, shows a scenario where the neural network predictors are trained with data from Base Station 2 and tested with that of Base Station 7. The QRT adapted COST 231 Walfisch-Ikegami model appears to be the more convergent with the test data. Results of Base Stations 2/7 pairing presented in Table 4.25 further shows that the QRT adapted COST 231 Walfisch-Ikegami model outperforms the MLP-NN based model by 0.24dB.

Table 4.25: Performance Comparison by Random Training with one Base Station and

Testing with another (Maiduguri)

MODEL	STAT.	BST2 /BST7	BST6/ BST3	BST4 /BST6	BST8/ BST2	BST1 /BST8	BST7 /BST5	GEOM MEAN
MLP-NN	μ(dB)	4.56	2.46	1.72	2.74	5.30	4.44	3.28
	σ(dB)	3.36	2.13	1.08	1.67	6.03	3.73	2.57
	RMSE(dB)	5.58	3.20	2.01	3.17	7.86	5.71	4.15
	R^2	0.89	0.96	0.98	0.96	0.75	0.84	0.89
GRBF-NN	μ(dB)	3.02	2.59	3.82	3.12	2.94	3.30	3.11
	σ(dB)	2.72	2.27	3.53	2.14	2.09	3.30	2.62
	RMSE(dB)	4.00	3.38	5.11	3.73	3.56	4.58	4.02
	R^2	0.94	0.96	0.89	0.94	0.95	0.89	0.93
QRT Adapted COST 231 Walf-Ikeg	μ(dB)	3.62	3.40	2.68	3.78	3.76	2.59	3.27
	σ(dB)	4.09	2.65	1.92	2.29	3.75	2.30	2.73
	RMSE(dB)	5.34	4.25	3.25	4.38	5.20	3.40	4.23
	R^2	0.90	0.94	0.96	0.92	0.89	0.94	0.92
COST 231 Walf-Ikeg	μ(dB)	11.31	12.08	11.08	10.24	11.55	11.93	11.35
	σ(dB)	5.69	4.61	3.20	4.16	5.20	2.64	4.11
	RMSE(dB)	12.57	12.87	11.50	10.99	12.58	12.20	12.10
	R^2	0.44	0.41	0.44	0.49	0.35	0.25	0.39
COST 231 Hata	μ(dB)	4.28	3.93	3.03	4.11	3.95	1.67	3.34
	σ(dB)	4.42	3.33	2.33	2.54	3.62	1.92	2.91
	RMSE(dB)	6.03	5.07	3.77	4.78	5.26	2.49	4.39
	R^2	0.87	0.91	0.94	0.90	0.89	0.97	0.91

Table 4.25 presents the performance indicators of all the models based on this comparative technique. The results show that the difference in performance between the QRT adapted COST 231 Walfisch-Ikegami and the MLP-NN based model is a negligible 0.08dB. However, the best in terms of generalisation is the GRBF-NN model with a mean RMSE value of 4.02dB and R^2=0.93.

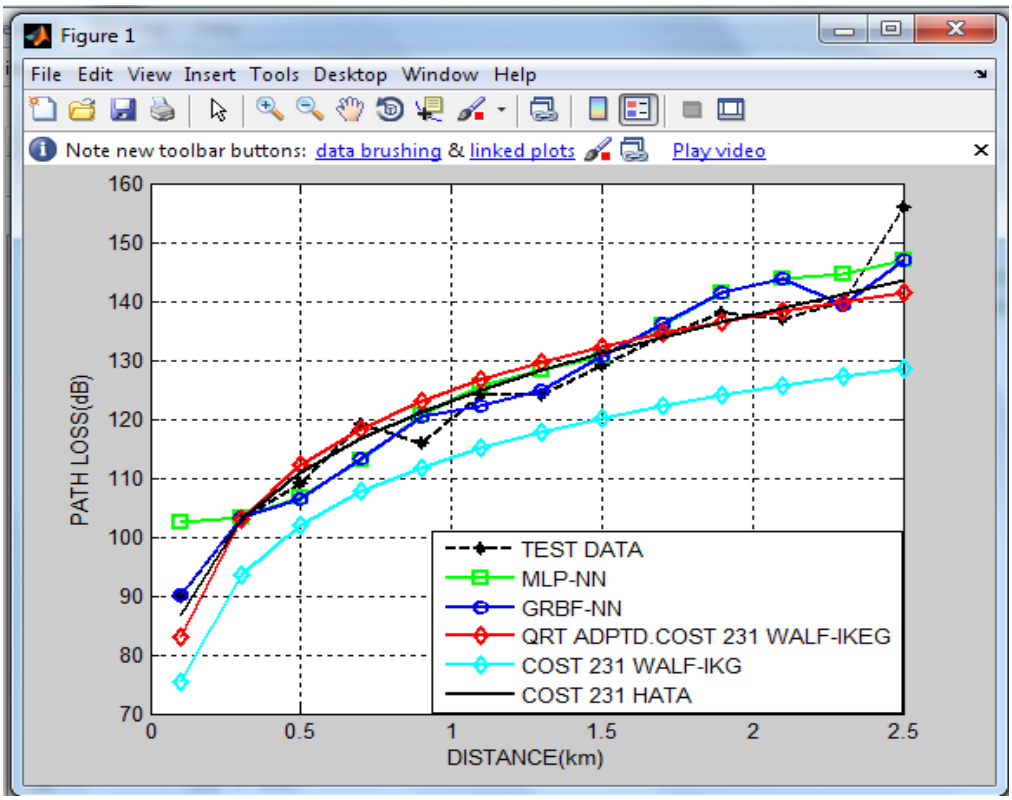


Figure 4.20: Generalisation Comparison by Training with BST2 and Testing with BST7 (Maiduguri)

4.3.3.8 Combined performance Analysis across three Techniques

The overall performance indicators of the Model Predictors across the three techniques are presented in Table 4.26. Results indicate that the Model Predictor with the best performance across the three techniques is the GRBF-NN predictor with the following indicators: $\mu=3.01dB$, $\sigma=2.1dB$, $RMSE=4.0dB$ and $R^2=0.76$. However, the difference in performance between the QRT adapted COST 231 Walfisch-Ikegami and the MLP-NN based model is just about 0.16dB.

Table 4.26: Overall Performance based on three Techniques (Maiduguri)

	STAT.	Technique A	Technique B	Technique C	GEOM MEAN.
MLP-NN	μ(dB)	3.71	3.20	3.28	3.39
	σ(dB)	1.98	2.50	2.57	2.33
	RMSE(dB)	4.18	4.04	4.15	4.12
	R^2	0.43	0.92	0.89	0.71
GRBF-NN	μ(dB)	3.33	2.64	3.11	3.01
	σ(dB)	2.01	1.75	2.62	2.10
	RMSE(dB)	3.84	3.15	4.02	3.65
	R^2	0.49	0.96	0.93	0.76
QRT Adapted COST 231 Walf-Ikeg.	μ(dB)	3.42	3.12	3.27	3.27
	σ(dB)	2.87	2.87	2.73	2.82
	RMSE(dB)	4.43	4.19	4.23	4.28
	R^2	0.92	0.92	0.92	0.92
COST 231 Walf-Ikeg	μ(dB)	11.15	11.46	11.35	11.32
	σ(dB)	4.25	3.98	4.11	4.11
	RMSE(dB)	11.95	12.20	12.10	12.08
	R^2	0.40	0.36	0.39	0.38
COST 231 Hata	μ(dB)	3.57	3.04	3.34	3.31
	σ(dB)	3.02	2.91	2.91	2.95
	RMSE(dB)	4.65	4.15	4.39	4.39
	R^2	0.90	0.92	0.91	0.91

4.4 Overall Performance Comparison of Adaptation Techniques across the Three Terrains

The overall geometric mean values of adaptation accuracies of the RAT and the proposed QRT across the three terrains are presented in Table 4.27. Results show that the QRT has an overall RMSE value of 2.1dB, while the RAT has 5.66dB. By implication, the QRT is more accurate in adaptation accuracy than the RAT counterpart by an RMSE value of 3.56dB. The comparison between the Okumura G_{AREA} technique and the QRT is based on the rural area only since the Okumura G_{AREA} technique is not applicable to urban areas. Results in Table 4.3 show that the difference in adaptation accuracy between the Okumura G_{AREA} technique and the QRT is much wider in the Okumura model adaptation to the rural terrain: the QRT has an RMSE value of 2.18

dB while the Okumura G_{AREA} technique 8.95dB, implying that the QRT outperforms the Okumura G_{AREA} technique by 6.77dB on the rural terrain only. The efficiency of the QRT can be attributed to the fact that, while the RAT and Okumura G_{AREA} adaptation techniques only modify the constant in a model expression, the QRT provides the best fit for the model. These results show that the QRT is the best of the three techniques in terms of empirical model adaptation.

Table 4.27: Overall Adaptation Accuracies of the RAT Technique and the QRT across the three Terrains

PERFORMANCE STATISTICS	Adaptation Techniques	
	RAT Adaptation	QRT Adaptation
μ(dB)	2.91	1.88
σ(dB)	2.00	1.29
RMSE(dB)	5.66	2.10
R^2	0.89	0.97

4.5 Overall Performance Comparison of Model Categories across the three Terrains

Table 4.28 presents the Overall path loss prediction accuracies of all model categories used in this study. On the geometric mean, the ANN-based models have the highest path loss prediction accuracy based on an RMSE value of 3.98dB. The QRT adapted empirical models are the next with 4.49dB, the RAT-adapted empirical models have 5.83dB based on three techniques across the three terrains, while the empirical models are the least accurate with 7.07dB.

Table 4.28: Overall Performance Comparison of Model Categories across three terrains

PERFORMANCE INDICIES	NEURAL NETWORK MODELS	QRT ADAPTED EMPIRICAL MODELS	RAT- ADAPTED EMPIRICAL MODELS	EMPIRICAL MODELS
μ(dB)	3.25	3.71	4.01	5.95
σ(dB)	2.33	2.80	2.94	3.70
RMSE(dB)	3.98	4.49	5.83	7.07
R^2	0.77	0.81	0.69	0.56

By implication, the ANN-based models slightly outperform the QRT adapted empirical models by 0.51dB. In fact, the MLP-NN based model alone just slightly outperforms the QRT adapted empirical models by about 0.37dB. However, the QRT adapted empirical models have the highest R^2 value of 0.81, and by implication, the best fit resulting from the best correlation with the measured path loss data. This shows that the QRT is quite efficient in empirical model adaptation. The high accuracy of the ANN-based models can be attributed to the fact that they are based on non-linear approximation functions, as demanded by the non-linear nature of the measured path loss values, caused by the various terrain clusters across radio propagation paths. The proximity in performance of the QRT adapted empirical models to the neural network predictors can be attributed to the efficiency of the technique.

On the other hand, the QRT adapted empirical models offer an improvement of about 1.34dB over the RAT-adapted empirical models, as well as an improvement of 2.58dB over existing Empirical Models.

CHAPTER FIVE

SUMMARY AND CONCLUSIONS

5.1 Introduction

This chapter presents the research summary, significant contributions, conclusions made, recommendations for further work, and limitations.

5.2 Summary

In this study, a novel Quotient Regression Technique for adapting an empirical model to a given terrain is developed and compared for adaptation accuracy with the Okumura G_{AREA} technique and the commonly used RAT adaptation technique. Sequel to that, the path loss prediction accuracies of the QRT adapted empirical models were evaluated relative to empirical models adapted by the other two adaptation techniques, as well as to ANN-based models. Results show that the QRT offers significant improvements over the Okumura G_{AREA} technique and the RAT, in terms of empirical model adaptation. Furthermore, the QRT adapted empirical models offer significant improvements over the Okumura G_{AREA} and the RAT adapted empirical models. As further demonstration of the effectiveness of the QRT, the QRT adapted empirical models are only fractionally less accurate than the ANN-based models, implying that the QRT adapted models can serve as computationally efficient alternatives to the ANN-based models.

5.3 Significant Contributions

The significant contributions of this study to the existing body of knowledge include the following:

i) A new and more accurate technique for adapting empirical models, termed the Quotients Regression Technique (QRT) was developed and implemented, from which a Root Mean Squared Error (RMSE) value of 2.1dB and R^2 of 0.97 were obtained. This is an improvement on both the commonly used RAT, which has an RMSE value of 5.66dB and R^2 of 0.7, and the Okumura G_{AREA} technique which has an RMSE value of 8.95dB and R^2 of 0.69.

ii) Implemented QRT adapted empirical models for improved path loss prediction across different terrains, by which an improvement of about 1.34dB (RMSE) was achieved compared with RAT adapted empirical models.

iii) Implemented QRT adapted empirical models for improved path loss prediction across different terrains, by which an improvement of about 2.58dB (RMSE) was achieved compared with existing empirical models.

iv) Implemented Artificial Neural Network techniques for improved path loss prediction across different terrains, by which an improvement of about 3.09dB (RMSE) was achieved compared with existing empirical models.

v) Comparative analysis shows a slight convergence in performance between the QRT adapted empirical models and artificial neural network models based on RMSE difference of 0.51dB.

vi) QRT Adapted empirical and ANN based models will be useful to mobile network operators in network planning, in order to ensure quality delivery of service within the areas under investigation.

5.4 Conclusions

The following conclusions were arrived at based on a case study of the following Nigerian terrains: the rural area between Jos and Abuja, the urban terrain (Abuja), and the semi-urban terrain (Maiduguri).

i) Overall, the QRT was found to be more accurate in empirical model adaptation than the RAT and the Okumura correction factors Technique by 3.56dB and 6.85dB respectively.

ii) QRT adapted empirical models, on the average, found more accurate in path loss prediction than the RAT adapted counterparts by an RMSE value of 1.34dB, and with a better fit of 0.81 compared to 0.7 for the RAT adapted counterparts.

iii) QRT adapted Okumura model found more accurate in path loss prediction than the G_{AREA} adapted counterpart by an RMSE value of 3.67dB

iv) QRT adapted empirical models found more accurate in path loss prediction than the existing empirical models by about 2.58dB

v) ANN based models found slightly more reliable than the QRT adapted empirical models by an RMSE value of 0.51dB. However, QRT adapted models can serve as computationally efficient alternatives to the ANN based models.

vi) ANN based models found more reliable than existing empirical models by 3.09dB.

vii) The proximity in performance of the adapted empirical models to the neural network predictors can be attributed to their adaptation to the geometric mean

which is insensitive to extreme values, as well as the effectiveness of the proposed QRT.

5.5 Recommendations for further work

The following are some recommendations that can offer improvement on this study:

i) The efficiency of the new QRT can further be verified using data from other terrains.

ii) The efficiency of the new QRT can further be verified using data obtained at other frequencies.

iii) The efficiency of the new QRT can further be verified using other empirical models.

iv) Other adaptation techniques can be used in order to improve results.

v) The study can serve as a basis for seasonal path loss modelling of the terrains under investigation.

vi) Other ANN training techniques can be used in order to improve results.

vii) Other Soft Computing Techniques can be applied in order to improve results.

viii) Other empirical models apart from the ones used in this study can be tested for path loss prediction reliability across the terrains.

5.5 Limitations

The path loss prediction models developed in this study are of great significance for network planning across the environments under investigation. However, the limitations of the study include the following:

i) The measurement equipment used may not be the most accurate. Hence, results obtained are limited based on the data obtained.

ii) Models developed in this study are based on data obtained from the terrains under investigation, hence may yield inaccurate results when applied to other terrains.

iii) Models were developed based on data obtained between the months of April and June hence may not be accurate when implemented in other seasons.

iv) Study was conducted strictly at an operating frequency of 900MHz. Therefore, models may not be accurate at other frequencies.

v) Artificial Neural Networks may not be as accurate as other Soft Computing Techniques; hence Other Soft Computing Techniques may improve results.

REFERENCES

Abhayawardhana V.S., WasselI.J., Crosby D., Sellers M.P., Brown M.G. (2005). *Comparison of empirical propagation path loss models for fixed wireless access systems. 61th IEEE Technology Conference*,Stockholm, pp. 73-77. Retrieved from http://www.scribd.com/doc/61233911/Comparison-of-Empirical-Propagation-Path-Loss#scribd,in February 2012.

Abhijit M. (2009). Lecture Notes on Mobile Communication. Department of Electronics and Communication Engineering Indian Institute of Technology Guwahati Guwahati - 781039, India. Pp. 80-82. Retrieved from http://www.iitg.ernet.in/scifac/qip/public_html/cd_cell/EC632.pdf, in July 2014.

Abraham U. U. (2013). Radio Propagation Modelling of GSM at 900MHz and 1800MHz for Bauchi. A Doctoral Thesis submitted to the Department of Electrical Engineering, Abubakar Tafawa Balewa University, Bauchi, Nigeria.

Abraham U. U., Okereke O. U. and Omizegba E. E. (2013). Macrocell path loss prediction using artificial intelligence techniques: *International Journal of Electronics*, p. 9. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/00207217.2013.792040?journalCode=tetn20#.VQ9UcPzF8oE, in October 2013.

Alcatel Lucent Strategic White Paper. (2013). The LTE Network Architecture: A comprehensive tutorial, pp. 1-23. Retrieved from http://www.cse.unt.edu/~rdantu/FALL_2013_WIRELESS_NETWORKS/LTE_Alcatel_White_Paper.pdf, in March 2015.

Alim M. A. , Rahman M. M., Hossain M. M., Al-Nahid A.. (2010). Analysis of Large-Scale Propagation Models for Mobile Communications in Urban Area. *International Journal of Computer Science and Information Security*, Vol. 7, No. 1, pp. 135-139. Retrieved from http://arxiv.org/ftp/arxiv/papers/1002/1002.2187.pdf, in September, 2014.

Alotaibi, F. D., Abdennour, A., & Ali A. A. (2008). A robust prediction model using ANFIS based on recent TETRA outdoor RF measurements conducted in Riyadh city – Saudi Arabia. AEU – *International Journal of Electronics and Communications*, 62(9), 674–682. Retrieved from http://www.researchgate.net/publication/222301425, in November 2013.

Akingbade K. F. and Olorunnibi E. D. (2013). Path Loss Prediction Model for UHF Radio waves Propagation In Akure Metropolis. *International Journal of Engineering (IJE)*, Volume (8). Pp. 30-37.

Akinyemi P., Azi S.O., Ojo J.S., Abiodun C.I. (2015). Evaluation for suitable propagation model to mobile Communications in South-South Nigeria urban-terrain. *American Journal of Engineering Research (AJER)*e-ISSN: 2320-0847 p-ISSN : 2320-0936 Volume-4, Issue-4, pp-01-05.

Akpado K.A., Oguejiofor O.S., Abe A., Ejiofor A. C. (2013). Pathloss Prediction for a typical mobile communication system in Nigeria using empirical models. IRACST – *International Journal of Computer Networks and Wireless Communications (IJCNWC),* ISSN: 2250-3501 Vol.3, No2. Pp. 207-211. Retrieved from http://www.ijcnwc.org/papers/vol3no22013/24vol3no2.pdf

Andrej K., Janez B. and Andrej K. (2011). Introduction to the Artificial Neural Networks, Artificial Neural Networks - Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN: 978-953-307-243-2, p.1. Retrieved from http://cdn.intechopen.com/pdfs-wm/14881.pdf, in December 2013

Anwesha B., Shounak D., Amit K., D. N. T. (2012). Development strategy of eye movement controlled rehabilitation aid using Electrooculogram. *International Journal of Scientific & Engineering Research*, Volume 3, Issue 6, June-2012 1 ISSN 2229-5518. Pp 1-6.

Arun K., Suman and Renu. (2013). Comparison of 3G Wireless Networks and 4G Wireless Networks. *International Journal of Electronics and Communication Engineering*. ISSN 0974-2166 Volume 6, Number 1, pp. 1-8. Retrieved from http://www.ripublication.com/irph/ijece/ijecev6n1_01.pdf, in February 2015.

Ashish E. (2012). Pathloss Determination Using Okumura-Hata Model For Rourkela. A Thesis submitted to the Department of Electronics & Communication Engineering, National Institute of Technology Rourkela, p. 9. Retrieved from http://ethesis.nitrkl.ac.in/3774/1/ashish.thesis.pdf, in July, 2014.

Brian T., Marjorie D., Christina M. (2003). Verification and validation of neural networks: a sampling of research in progress. Intelligent Computing: Theory and Applications, Kevin L. Priddy, Peter J. Angeline, Editors, Proceedings of SPIE Vol. 5103, pp. 8-16. Retrieved from http://people.cs.umass.edu/~btaylor/publications/PSI000008.pdf, October 2014.

Callistus O. M., Joseph M. M., Gabriel A. I. (2015). Performance Evaluation of Generalized Regression Neural Network Path loss Prediction Model in Macrocellular Environment. *Journal of Multidisciplinary Engineering Science and Technology* (JMEST) ISSN: 3159-0040 Vol. 2 Issue 2, pp 204-208.

Chai1 T. and Draxler R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE). Geosci. Model Dev., 7, 1247–1250, Retrieved fromwww.geosci-model-dev.net/7/1247/2014/in November 2014.

Chang P., Yang W. (1997). Environment-adaptation mobile radio propagation prediction using radial basis function neural networks. *Vehicular Technologies, IEEE Transactions* on Vol. 46, Issue 1. Retrieved from http://140.98.202.196/xpl/login.jsp?tp=&arnumber=554747&url=http%3A%2F%2F140.98.202.196%2Fstamp%2Fstamp.jsp%3Farnumber%3D554747, in July 2014.

Chhaya D. (2013). Comparative Study of Radio Channel Propagation and Modeling for 4G Wireless Systems. *International Journal of Engineering and Advanced Technology* (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-5, pp. 296-299. Retrieved from http://journaldatabase.info/articles/comparative_study_radio_channel.html, in November 2014

Ekram H., Mehdi R., Hina T., and Amr A. (2014). Evolution Towards 5G Multi-tier Cellular Wireless Networks: An Interference Management Perspective. arXiv:1401.5530v2 [cs.NI], pp. 1-10. Retrieved from http://arxiv.org/pdf/1401.5530.pdf, in March 2015.

Fadzilah S., Ehab A. Omer A. Omer and Rajib H. (2012). Data Mining and Neural Networks, pp. 98-116. Retrieved from http://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications in May, 2014.

Famoriji J.O. and Olasoji Y.O. (2013). UHF Radio Frequency Propagation Model for Akure Metropolis. *Research Journal of Engineering Sciences*. ISSN 2278 – 9472 Vol. 2(5), pp. 6-10.

Farhad E. (2012). Mobile Radio Propagation Prediction for Two Different Districts in Mosul-City. MATLAB – A Fundamental Tool for Scientific Computing and Engineering Applications – Volume 2, pp. 20-35. Retrieved from http://cdn.intechopen.com/pdfs-wm/39339.pdf, in January 2015.

Faria P. , Zita V., Joao P., Khodr H. (2009). ANN Based Day-Ahead Spinning Reserve Forecast

for Electricity Market Simulation. Center of the Electr. Eng., Polytech. Inst. of Porto (IPP), Porto, Portugal DOI: 10.1109/ISAP.2009.5352930 Conference: Intelligent System Applications to Power Systems, 2009. ISAP '09. p. 2.

Gaurang P., Amit G., Y P Kosta and Devyani P. (2011) "Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers", *International Journal of Computer Theory and Engineering*, Vol. 3, No. 2, pp. 332-337. Retrieved from https://www.academia.edu/2490802/Behaviour_analysis_of_multilayer_perceptrons_with_multiple_hidden_neurons_and_hidden_layers, in July 2014.

George D. M and Michael N. V. (2006). Adaptive Algorithms For Neural Network Supervised Learning: A Deterministic Optimization Approach. *International Journal of Bifurcation and Chaos*, Vol. 16, No. 7,pp. 1929–1950. Retrieved from http://www.dcs.bbk.ac.uk/~gmagoulas/AdaptAlgorithms.pdf, in October 2013.

Ghassan A. A., Mahamod I.& Kasmiran J. (2012). The Evolution to 4G Cellular Systems: Architecture and Key Features of LTE-Advanced Networks. IRACST – *International Journal of Computer Networks and Wireless Communications* (IJCNWC), ISSN: 2250-3501 Vol. 2, No. 1, pp. 21-26. Retrieved from https://www.academia.edu/9078401/The_Evolution_to_4G_Cellular_Systems_Architecture_and_Key_Features_of_LTE-Advanced_Networks, in March 2015.

Govind S. and Sonika S. (2014). A Review on Outdoor Propagation Models on Radio Communication. *International Journal of Computer Engineering & Science*. ISSN: 2231–6590. Volume 4, Issue 2, pp. 64-68. Retrieved from http://www.ijces.org/media/4Iss8-IJCES0402604_v4_iss2_64-68.pdf, in March 2015.

Ignacio F. A., Juan A. R., and Fernado P. F. (2012), Influence of Training Set Selection in Artificial Neural Network-Based Propagation Path Loss Prediction: *International Journal of Antennas and Propagation*.Volume 2012, Article ID 351487, 7 pages, pp. 122-128. Retrieved from http://www.hindawi.com/journals/ijap/2012/351487/, in October 2014.

Isabona J., Konyeha. C.C. (2013). Urban Area Path loss Propagation Prediction and Optimisation Using Hata Model at 800MHz. IOSR *Journal of Applied Physics* (IOSR-JAP) e-ISSN: 2278-4861. Volume 3, Issue 4, PP 08-18. www.iosrjournals.org

Jang J.S., Sun C.T., Mizutani. E. (1997). Neuro-Fuzzy and Soft Computing. A

Computational to Learning and Machine Intelligence. Prentice-Hall Inc, pp1-328. Retrieved from http://www.csbdu.in/pdf/Neuro-uzzy%20And%20Soft%20Computing%20Jang.pdf, in October 2013.

Jang J.S. (1993). Adaptive Network Based Fuzzy Inference System", IEEE Trans. On Systems Man. and Cybemetics, Vol.23, No.3, (1993), Pp 665-685. Retrieved from http://www.dca.ufrn.br/~meneghet/FTP/anfis%2093.pdf, in November, 2013.

Janmenjoy N., Bighnaraj N. (2015). A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges. *International Journal of Database Theory and Application* Vol.8, No.1 (2015), pp.169-186. Retrieved from http://www.sersc.org/journals/IJDTA/vol8_no1/18.pdf, in April, 2015.

John A. B. (2004). Radial Basis Function Networks: Introduction. Introduction to Neural Networks: Lecture 12. Retrieved from http://www.cs.bham.ac.uk/~jxb/NN/l12.pdf, in July, 2014.

Joseph M. Mom, Callistus O. Mgbe, and Gabriel A. Igwue. (2014). Application of Artificial Neural Network For Path Loss Prediction In Urban Macrocellular Environment. *American Journal of Engineering Research*-ISSN: 2320-0847. ISSN : 2320-0936 Volume-03, Issue-02, pp-270-275. Retrieved from http://www.ajer.org/papers/v3%282%29/ZI32270275.pdf,in September 2014.

Kandilli. I., M. Sönmez, H. M. Ertunc, B. Çakir. (2007). *Online Monitoring of Tool Wear In Drilling and Milling By Multi-Sensor Neural Network Fusion*. Proceedings of 2007 IEEE International Conference on Mechatronics and Automation, 1388-1394.Retrieved from http://www.researchgate.net/publication/224720856_Online_Monitoring_Of_Tool_Wear_In_Drilling_and_Milling_By_Multi-Sensor_Neural_Network_Fusion, in April 2014.

Lahiri S. K., Ghanta K. C.(2009). Artificial neural network model with the parameter tuning assisted by a differential evolution technique: the study of the hold up of the slurry flow in a pipeline. *Chemical Industry & Chemical Engineering Quarterly* 15 (2) 103−117. Retrieved from http://www.doiserbia.nb.rs/img/doi/1451-9372/2009/1451-93720902103L.pdf, in March, 2014.

Lei P.W., Chatwin C.R., Young R.C.D., Ting S.H. (2004). Chapter 4: Opportunities and Limitations in M-Commerce, p. 1. Retrieved from http://flylib.com/books/en/3.97.1.38/1/, in December, 2014.

Li Q. (2005). GIS Aided Radio Wave Propagation Modeling and Analysis.Master of Science

In Geography. Blacksburg, Virginia, p 7-9.A Masters thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University.Retrieved fromhttp://scholar.lib.vt.edu/theses/available/etd-05272005-140752/unrestricted/Thesis_LiQing.pdf, in March, 2012.

Mardeni.R. (2010). Optimised COST-231 Hata Models for WiMAX Path Loss Prediction in Suburban and Open Urban Environments. *Published by Canadian Center of Science and Education. Modern Applied Science* Vol. 4, No. 9, pp. 75-89. Retrieved from http://www.ccsenet.org/journal/index.php/mas/article/viewFile/6462/5745, in January, 2015.

Mardeni K. and Kwan K. F. (2010). Optimization of Hata Propagation Prediction Model in Suburban Area in Malaysia. Progress in Electromagnetics Research C, Vol. 13, 91-106. Retrieved from http://www.jpier.org/PIERC/pierc13/08.10011804.pdf, in June 2012.

Menaka.K.,Devil S. (2014). "A Novel Approach for Optimal Allocation and Sizing of Distributed Energy Storage System in Smart Grid".IOSR *Journal of Electrical and Electronics Engineering* (IOSR-JEEE) e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 9, Issue 1, Page 25-33. Retrieved from http://www.iosrjournals.org/iosr-jeee/Papers/Vol9-issue1/Version-5/E09152533.pdf, in July 2014.

Mishra A. R. (2004). Second-generation Network Planning and Optimisation (GSM), pp. 19-53. Retrieved from http://media.johnwiley.com.au/product_data/excerpt/7X/04708626/047086267X.pdf, in February, 2011.

Mudit R. B., Anand V. B. (2010). Generations of Mobile Wireless Technology: A Survey. *International Journal of Computer Applications* (0975 – 8887)Volume 5– No.4, pp. 26-32. Retrieved from http://ijcaonline.net/volume5/number4/pxc3871282.pdf, in November, 2014.

Nadir Z., Muhammad I. A. (2010). *Pathloss Determination Using Okumura-Hata Model and Cubic Regression for Missing Data for Oman*; Proceedings of International MultiConference of Engineers and Computer Scientists Vol II 2010, IMECS, Hong Kong. Retrieved from http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp804-807.pdf,2013

Nadir Z. (2011). Seasonal Pathloss Modeling at 900MHz for Oman. *International Conference*

on Telecommunication Technology and Applications. Proc .of CSIT vol.5. IACSIT Press, Singapore.

Neha G. (2013). Artificial Neural Networks. *Network and Complex Systems* www.iiste.org ISSN 2224-610X (Paper) ISSN 2225-0603 (Online) Vol.3, No.1, pp. 24-28. Retrieved from http://iiste.org/Journals/index.php/NCS/article/view/6063/6019, in July 2014.

Nicholas C. (2002). Computer Architecture, The Mc-Graw Hill Companies, p. 5.

Nisirat M. A., Ismail M., Nissirat L., and Al-Khawaldeh S. (2011). A Terrain Roughness Correction Factor for Hata Path Loss Model at 900MHZ; *Progress in Electromagnetics Research* C, Vol. 22, pp. 11-22. Retrieved from http://www.jpier.org/PIERC/pierc22/02.11041402.pdf, in August 2012

Neskovic, A., Neskovic, N., and Paunovic, D. (2002). Macrocell electric field strength prediction model based upon artificial neural networks. *IEEE Journal on Selected Areas in Communications*, 20(6), 1170–1177. Retrieved from. http://www.researchgate.net/publication/3234845_Macrocell_electric_field_strength_prediction_model_based_upon_artificial_neural_networks, in October 2013.

Nordstrom, T. and Svensson, B. (1998). Using and Designing Massively Parallel Computers for Artificial Neural Networks", *Journal of Parallel and Distributed Processing*, no.3, pp. 260-285. Retrieved from http://ac.els-cdn.com/074373159290068X/1-s2.0-074373159290068X-main.pdf?_tid=bb4c380a-d577-11e4-9e9a-00000aab0f02&acdnat=1427567274_8122311df549d99d5c1fa350a6486d4b, in November 2013.

Obot A, Simeon O., Afolayan J.. (2011). Comparative Analysis of Path Loss Prediction Models for Urban Macrocellular Environments. Nigerian Journal of Technology Vol. 30, No. 3, pp 50-59.Retrieved from http://www.nijotech.com/index.php/nijotech/article/view/20/10, in July 2012.

Ogbulezie, J.C. ,Onuu, M.U., Bassey, D.E and Etienam-Umoh, S. (2013). Site specific measurements and propagation models for GSM in three cities in Northern Nigeria. *American Journal of Scientific And Industrial Research*. Science Huβ. pp. 238-245. Retrieved from http://www.scihub.org/AJSIR ISSN: 2153-649X, in September 2014.

Okunuga, S. A. Numerical Analysis II. National Open University. Lecture Notes, pp. 8-9.

Retrieved from http://www.nou.edu.ng/NOUN_OCL/pdf/pdf2/MTH%20307%20Numerical%20Analysis%20II%20.pdf, June, 2014.

Olasunkanmi F. O, Segun I. P., Robert O. A., Oluwole A. A. (2014). Comparative Analysis of Received Signal Strength Prediction Models for Radio Network Planning of GSM 900 MHz in Ilorin, Nigeria. *International Journal of Innovative Technology and Exploring Engineering* (IJITEE) ISSN: 2278-3075, Volume-4 Issue-3, pp. 45-50. Retrieved from http://www.researchgate.net/publication/264672680_Comparative_Analysis_of_Received_Signal_Strength_Prediction_Models_for_Radio_Network_Planning_of_GSM_900_MHz_in_Ilorin_Nigeria, in October 2014.

Östlin, E., Zepernick, H.J., and Suzuki, H. (2010). Macrocell Path loss Prediction Using Artificial Neural Networks: IEEE Transactions on Vehicular Technology, vol. 59, No. 6, pp. 2735-2744. Retrieved from http://www.researchgate.net/publication/224138702_Macrocell_Path-Loss_Prediction_Using_Artificial_Neural_Networks, in November 2013.

Östlin, E., Zepernick, H.J., and Suzuki, H. (2004). Macrocell radio wave propagation prediction using an artificial neural network. IEEE Semiannual Vehicular Technology Conference, 1, 57–61. Retrieved from http://www.researchgate.net/publication/4127410_Macrocell_radio_wave_propagation_prediction_using_an_artificial_neural_network, in November 2013.

Patil C. S., Karhe R. R., Aher M. A. (2012). Review on Generations in Mobile Cellular Technology. *International Journal of Emerging Technology and Advanced Engineering* Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 10, pp. 614-619.

Padmapriya S. (2015). A Study on Algorithmic Approaches and Mining Methodologies In Data Mining. *International Journal of Computer Science Trends and Technology* (IJCST) – Volume 3 Issue 1, ISSN: 2347-8578. www.ijcstjournal.org, pp. 105-109. Retrieved from http://www.ijcstjournal.org/volume-3/issue-1/IJCST-V3I1P21.pdf, in February 2015.

Piacentini M., RinaldiF. (2010). Path loss prediction in urban environment using learning machines and dimensionality reduction techniques. Retrieved from http://www.math.unipd.it/~rinaldi/papers/path_loss.pdf, in June, 2014.

Popescu I., Naforni I., Gavriloaia G. (2001). Field Strength Prediction in Indoor Environment

with a Neural Network Model:*FACTA UNIVERSITATIS (NIS), Series: Electronics and Energetics*. Vol. 14, No. 3, pp 329-336. Retrieved from http://facta.junis.ni.ac.rs/eae/fu2k13/fu02.pdf, October 2013.

Popescu, I., Kanatas, A., Angelou, E., Nafornita, I., & Constantinou, P. (2002). *Applications of generalized RBF-NN for path loss prediction*. 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2002), 1, 484–488. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.5719&rep=rep1&type=pdf, in September 2013.

Popescu I. (2003). Neural Network Applications for Radio coverage Studies in Mobile Communication Systems. A Doctoral Thesis submitted to the Politehnica University Timisoara, Electronics and Telecommunications Faculty. Pp. 73-75. Retrieved from http://www.tc.etc.upt.ro/docs/cercetare/teze_doctorat/nnarcsmcs.pdf, in September, 2013.

Popescu, I.,Nikitopoulos, D.,Nafornita, I.,Constantinou P. (2006). *ANN Prediction Models for Indoor Environment.* Wireless and Mobile Computing, Networking and Communications. IEEE International Conference on 19-21 June 2006. Page(s): 366 – 371. Retrieved from http://www.researchgate.net/publication/221508201_ANN_Prediction_Models_for_Indoor_Environment, in August 2013.

Purnima K. S. (2010). Comparative Analysis of Propagation Path loss Models with Field Measured Data. *International Journal of Engineering Science and Technology* Vol. 2(6), 2010, pp. 2008-2013. Retrieved from http://www.ijest.info/docs/IJEST10-02-06-85.pdf, in in June 2012.

Powell M. J. (1977). Restart procedures for the conjugate gradient method. *Mathematical Programming.* (Springer) 12 (1): 241—254. Retrieved from http://link.springer.com/article/10.1007%2FBF01593790#page-1, in January 2014

Rakibul H., Piyal H., Shahidul I., Syed M. H. (2011). Handover Management in GSM Cellular Systems. A dissertation is submitted to the Department of Electrical, Electronics & Telecommunication Engineering of Dhaka International University, Bangladesh. Pp. 13. http://www.researchgate.net/profile/Utsho_A_Arefin/publication/271600639_Handover_Management_in_GSM_Cellular_System/links/54cdd76a0cf298d6565e36f5.pdf

Sami A. M. (2013). Path Loss Propagation Model Prediction for GSM Network Planning.

International Journal of Computer Applications (0975 – 8887) Volume 84 – No 7, pp. 30-33. Retrieved from http://research.ijcaonline.org/volume84/number7/pxc3892830.pdf, in Auguust 2014.

Santosh K. D., Sachin T., Burnwal A.P. (2014). Some Relevance Fields of Soft Computing Methodology. International Journal of Research In Computer Applications And RoboticsISSN 2320–7345, pp. 1-6. Retrieved from http://www.ijrcar.com/Volume_2_Issue_6/v2i601.pdf, in October 2014.

Sarumi J., Awodele O., Okolie S. O, and Adekunle Y.A. (2013). Problems and Prospects of Mobile Computing in Nigeria. *The International Journal of Computer Science & Applications* (TIJCSA). Volume 2, No. 06, August 2013 ISSN – 2278-1080, pp. 73-82. Retrieved from http://www.journalofcomputerscience.com/2013Issue/Aug13/V2No06Aug13P015.pdf, in June 2014.

Saveeda P., Ayyappan K. and Kumar R. (2014). Coverage and Link Budget Calculation for GSM Cellular System using Various Parameters. *International journal of Computer Networking and Communication* (IJCNAC)Vol. 2, No. 1, pp. 13-24. Retrieved from http://www.arpublication.org/jl/jc/vol2/c1140202.pdf, in November 2014.

Shin O., Salah E. E.i, Yeon K. J., and Yoan S. (2013). Advanced technologies for LTE Advanced. EURASIP Journal on Wireless Communications and Networking 2013, 2013:25,pp.1-3. Retrieved from http://jwcn.eurasipjournals.com/content/2013/1/25 in March 2015.

Shoewu O. and Edeko F.O. (2011). Analysis of radio wave propagation in Lagos environs. *American Journal of Scientific and Industrial Research*, 2(3): 438-455. Retrieved from http://www.scihub.org/AJSIR/PDF/2011/3/AJSIR-2-3-438-455.pdf, in June 2014.

Sneha V., Sushmitha P., Sheik M. (2014). An Idea of Mobile Network Portability to minimize Radiation effect. International Journal of Advanced Trends in Computer Science and Engineering, Vol. 3 , No.1, Pages : 575– 580. Retrieved from http://www.warse.org/pdfs/2014/icetetssp110.pdf, in November 2014.

Steve L., C. Lee G., Ah C. T.(1997). *Lessons in Neural Network Training: Overfitting May be*

Harder than Expected. Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97, AAAI Press, Menlo Park, California, pp. 540–545. Retrieved from http://clgiles.ist.psu.edu/papers/AAAI-97.overfitting.hard_to_do.pdf, in November 2013.

Subcourse Number SS0130. (2005). Principles Of Radio Wave Propagation. Edition B United States Army Signal Center and Fort Gordon Fort Gordon, Georgia 30905-5000, p. 1-12. Retrieved from http://www.freeinfosociety.com/media/pdf/4611.pdf, in June 2013.

Sumit J. and Vishal G. (2012). A Review on Empirical Data Collection and Analysis of Bertoni's Model at 1.8 GHz. *International Journal of Computer Applications* (0975 – 8887) Volume 56– No.6, pp 17- 23. Retrieved from http://www.researchgate.net/profile/Sumit_Joshi4/publication/232769092_A_Review_on_Empirical_Data_Collection_and_Analysis_of_Bertonis_Model_at_1.8GHz/links/02bfe51014d158d141000000.pdf, in June 2012.

Syed S. H. (2010). Simplified Neural Networks Algorithms for Function Approximation And Regression Boosting On Discrete Input Spaces. Thesis, School of Computer Science. p. 42. Retrieved from https://www.escholar.manchester.ac.uk/api/datastream?publicationPid=uk-ac-man-scw:125559&datastreamId=FULL-TEXT.PDF, August, 2013.

Sylvain R. (2004). Path loss Models S-72.333. Physical layer methods in wireless communication systems Helsinki University Of Technology Smarad Centre of Excellence, slides 1-36. Retrieved from http://www.comlab.hut.fi/opetus/333/2004_2005_slides/Path_loss_models, in July 2011.

Tapan K.S., Zhong J.,Kyungjung K., Abdellatif M., Magdalena S. (2003). A Survey of Various Propagation Models for Mobile Communication. *IEEE Antennas and Propagation Magazine*, Vol. 45, No. 3, pp. 51-82, Retrieved from http://wiki.unik.no/media/Master/SarkaretalPropagationModels.pdf, in June 2013.

Tarun K., Pradeep K. S., Purabi S., Amarnath P. (2013). Mobile Computing - An Introduction with Ad Hoc Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*. Volume 3, Issue 2, pp. 334-340. Retrieved from http://www.ijarcsse.com/docs/papers/Volume_3/2_February2013/V3I2-0145.pdf, in August 2014.

T¨urkan E. D., Berna Y. H., Aysen A. (2010). Fuzzy adaptive neural network approach to path

loss prediction in urban areas at GSM-900 band. *Turk J Elec Eng & Comp Sci*, Vol.18, pp. 1077-1081. Retrieved from http://journals.tubitak.gov.tr/elektrik/issues/elk-10-18-6/elk-18-6-12-0904-18.pdf, in September 2014.

Tsung-Ying S., Chan-Cheng L., Chung-Ling L., Shent-Ta H., Cheng-Sen H. (2014). A Radial Basis Function Neural Networks with Adaptive Structure via Particle Swarm Optimisation, pp.423-437. Retrieved from www.intechopen.com, in October, 2014.

Ubom, E.A., Idigo, V. E., Azubogu, A.C., Ohaneme, C.O., Alumona, T. L. (2011).Path loss Characterization of Wireless Propagation for South – South Region of Nigeria.*International Journal of Computer Theory and Engineering*, Vol. 3, No. 3, pp. 360-364. Retrieved from http://www.ijcte.org/papers/332-G961.pdf, in August 2012

Vanith. S, Tamilselvan V. (2014). Wireless Mobile Computing. *International Journal of Science and Engineering Research* (IJ0SER), Vol 2 Issue 6.
Retrieved from http://ijoser.org/Files/59.pd, in August 2013.

Wireless and Cable. (2014). Physical radio channel models. Universitiet Gent. Retrieved from http://www.wica.intec.ugent.be/research/propagation/physical-radio-channel-models, in February 2015.

Wolfle G. and Landstorfer F. M.. (1998). *Dominant Paths for the Field Strength Prediction:* 48th IEEE Conference on Vehicular Technology, vol. 1, May 1998, pp 552-556. Retrieved from http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-14581/Paper/6b_4.pdf, in July 2012.

Wu J. and Yuan D. (1998). *Propagation Measurements and Modeling in Jinan City*. IEEE International Symposium on Personal, indoor and Mobile Radio Communications, Boston, MA, USA, Vol. 3, pp. 1157-1159. Retrieved from http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=731360&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D731360, in July 2011.

Yuvraj S. (2012). Comparison of Okumura, Hata and COST-231 Models on the Basis of Path Loss and Signal Strength. *International Journal of Computer Applications* (0975 – 8887) Volume 59– No.11, pp. 37-41. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.303.4057&rep=rep1&type=pdf, in January 2015.

APPENDIX B: MATLAB CODE

APPENDIX B1: Comparison of Adaptation Techniques

```
% BASED ON JOS-ABUJA RURAL AREA DATA
% COMPARISON OF ADAPTATION TEHCNIQUES AT 900MHZ

close all
clear all
clc

%GSM NETWORK PARAMETERS
     hb = 33;
     f = 900;
     hm = 1.5;

dist= [0.1:0.3:4.3];  %  Transmitter-Receiver separations

%   Path loss data
pldata= [91.18
105.95
109.79
121.36
123.11
129.12
126.54
133.17
135.00
136.17
142.79
142.93
144.95
148.79
148.39
]';

%G_area Adapted OKUMURA

d=dist;
L50=32.45+20*log10(f)+20*log10(d)+20-20*log10(hb/200)-10*log10(hm/3)-9;
trd=pldata;
T=pldata;
N=length(T);


%G_area Adapted OKUMURA Prediction Errors

OK=L50';
OK_PE = T-OK';  % Prediction Error
OK_MSE = mse(OK_PE); %   Mean Square Error
OK_RMSE=sqrt(OK_MSE);   %Root mean square error computation
N=length(T);
ABSER=abs(T-OK');
OK_ABSERR=1/N*sum(ABSER);   % Absolute Error
ERRSUM=sum((T-OK').^2);
ERRSUMPL=sum((T-mean(T)).^2);
OK_RS=1-(ERRSUM/ERRSUMPL);  %R-Squared computation
TEMP=(sum(ABSER.^2)-N*OK_ABSERR^2);
```

```
OK_STD=sqrt(TEMP/(N-1));    % Standard Deviation
OK_OUTPUT=[OK_ABSERR;OK_STD;OK_RMSE;OK_RS]  %Prediction Errors


%RAT Adapted OKUMURA Prediction Errors

RMSE=sqrt(sum((T-OK').^2)/(N-1))
%OK1=L50+RMSE;
OK1=L50+9.27;
OK=OK1';
OK_PE = T-OK';  % Prediction Error
OK_MSE = mse(OK_PE); %   Mean Square Error
OK_RMSE=sqrt(OK_MSE);   %Root mean square error computation
N=length(T);
ABSER=abs(T-OK');
OK_ABSERR=1/N*sum(ABSER);   % Absolute Error
ERRSUM=sum((T-OK').^2);
ERRSUMPL=sum((T-mean(T)).^2);
OK_RS=1-(ERRSUM/ERRSUMPL);  %R-Squared computation
TEMP=(sum(ABSER.^2)-N*OK_ABSERR^2);
OK_STD=sqrt(TEMP/(N-1));    % Standard Deviation
OK1_OUTPUT=[OK_ABSERR;OK_STD;OK_RMSE;OK_RS]  %Prediction Errors


%QRT Adapted OKUMURA

Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
        Ncc = sum(d.^4);
        Dcc = sum(trd);
        Hcc = sum(d.*trd);
        Pcc = sum((d.^2).*trd);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;

            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc * Lcc)
+ Ccc * (Ecc * Mcc - Fcc * Lcc);
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc - Gcc *
Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc * Lcc)
+ Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc * Lcc)
+ Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
            AA01 = DT1 / dt
            AA02 = DT2 / dt

           L=AA00+AA01*d+AA02*(d.*d);
           Q=L./L50;
           QUOTS=[d Q]
```

```
     bl = sum(d);
     cl = sum(Q);
     el = sum(d.^2);
     fl = sum(d.*Q);

            al = N;
            dl = bl;
            dt = al * el - dl * bl;
            DT0 = cl * el - fl * bl;
            DT1 = al * fl - dl * cl;

            A0 = DT0 / dt
            A1 = DT1 / dt

%OK2=L50.*(0.9238+0.0475.*d);
OK2=L50.*(A0+A1.*d);
OK=OK2';
OK_PE = T-OK';  % Prediction Error
OK_MSE = mse(OK_PE); %   Mean Square Error
OK_RMSE=sqrt(OK_MSE);   %Root mean square error computation
N=length(T);
ABSER=abs(T-OK');
OK_ABSERR=1/N*sum(ABSER);   % Absolute Error
ERRSUM=sum((T-OK').^2);
ERRSUMPL=sum((T-mean(T)).^2);
OK_RS=1-(ERRSUM/ERRSUMPL);  %R-Squared computation
TEMP=(sum(ABSER.^2)-N*OK_ABSERR^2);
OK_STD=sqrt(TEMP/(N-1));    % Standard Deviation
OK2_OUTPUT=[OK_ABSERR;OK_STD;OK_RMSE;OK_RS]  %Prediction Errors

STATS=[OK_OUTPUT OK1_OUTPUT OK2_OUTPUT]

plot(d,pldata,'*--k',d,L50,'d-b',d,OK1,'o-g',d,OK2,'s-r','LineWidth',2);
legend('MEASURED PATH LOSS','G_A_r_e_a ADAPTED OKUMURA','RAT ADAPTED
OKUMURA','QRT ADAPTED OKUMURA');

xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')
grid on
```

APPENDIX B2: Acceptability Test and Adaptation of Empirical Models (Rural)

```
% TEST FOR ACCEPTABILITY AND ADAPTATION OF EMPIRICAL MODELS FOR RURAL
THE RURAL TERRAIN
close all
clear all
clc
```

```
%GSM NETWORK PARAMETERS
     hb = 33;
     f = 900;
     hm = 1.5;
     EIRP=46;

%HATA-OKUMURA CONSTANTS
    H_CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2+18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=0; % For small and medium city

dtrain= (0.1:0.3:4.3);  %  Transmitter-Receiver separations

%   Training Path loss data

pltrain= [91.18
105.95
109.79
121.36
123.11
129.12
126.54
133.17
135.00
136.17
142.79
142.93
144.95
148.79
148.39
]';




%   Hata Path loss data
hatad= (1.0:0.3:4.3);
plhata = [121.36
123.11
129.12
126.54
133.17
135.00
```

```
136.17
142.79
142.93
144.95
148.79
148.39
]';
T=pltrain;


% COST 231 HATA QRT ADAPTATION

d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;

trd=pltrain;
N=length(T);

Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
        Ncc = sum(d.^4);
        Dcc = sum(trd);
        Hcc = sum(d.*trd);
        Pcc = sum((d.^2).*trd);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;

            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Mcc - Fcc * Lcc);
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc -
Gcc * Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc
* Lcc) + Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
            AA01 = DT1 / dt
            AA02 = DT2 / dt
            L=AA00+AA01*d+AA02*(d.*d);

           Q=L./CHPL;
           QUOTS=[d Q]

           bl = sum(d);
```

```
      cl = sum(Q);
      el = sum(d.^2);
      fl = sum(d.*Q);

             al = N;
             dl = bl;

             dt = al * el - dl * bl;
             DT0 = cl * el - fl * bl;
             DT1 = al * fl - dl * cl;

             A0 = DT0 / dt
             A1 = DT1 / dt

             CH2=CHPL.*(A0+A1.*d);

CH=CHPL;
T=pltrain';
N=length(pltrain);

CH_PE = T-CH';  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH');
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH').^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation
CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];

%RAT ADAPTED COST 231 HATA
RMSE=sqrt(sum((T-CH').^2)/(N-1))

% QRT ADAPTION OF THE HATA-OKUMURA MODEL
d=hatad;

AHATA=69.55+26.16*log10(f)-13.82*log10(hb)-H_CH;
BHATA=(44.9-6.55*log10(hb))*log10(d);
PLu=AHATA+BHATA;
PLsu=PLu-5.4-2*(log10(f/28))^2;

T=plhata;
N=length(T);

        Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
```

```
        Ncc = sum(d.^4);
        Dcc = sum(plhata);
        Hcc = sum(d.*plhata);
        Pcc = sum((d.^2).*plhata);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;

            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Mcc - Fcc * Lcc);
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc -
Gcc * Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc
* Lcc) + Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
            AA01 = DT1 / dt
            AA02 = DT2 / dt

           L=AA00+AA01*d+AA02*(d.*d);

           Q=L./PLsu;

     bl = sum(d);
     cl = sum(Q);
     el = sum(d.^2);
     fl = sum(d.*Q);
           al = N;
           dl = bl;

           dt = al * el - dl * bl;
           DT0 = cl * el - fl * bl;
           DT1 = al * fl - dl * cl;

           A0 = DT0 / dt
           A1 = DT1 / dt

           HO2=PLsu.*(A0+A1.*d);

HO=PLsu';
HO_PE = T-HO';  %  HATA Prediction Error
HO_MSE = mse(HO_PE); % HATA Mean Square Error
HO_RMSE=sqrt(HO_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-HO');
HO_ABSERR=1/N*sum(ABSER);%  Absolute Error
ERRSUM=sum((T-HO').^2);%Rsquared computation
```

```
ERRSUMPL=sum((T-mean(T)).^2);
HO_RS=1-(ERRSUM/ERRSUMPL); % R-Squared computation
TEMP=(sum(ABSER.^2)-N*HO_ABSERR^2);
HO_STD=sqrt(TEMP/(N-1));% Standard Deviation

HO_OUTPUT=[HO_ABSERR;HO_STD;HO_RMSE;HO_RS]; % HATA

%RAT ADAPTED HATA-OKUMURA
RMSE2=sqrt(sum((T-HO').^2)/(N-1))

STATS=[CH_OUTPUT HO_OUTPUT]

plot(dtrain, pltrain, '*--k',dtrain,CH,'d-b',hatad,HO,'o-
r','LineWidth',2)

legend('MEASUREMENT','RAT ADPTD COST 231 HATA','QF ADPTD. COST 231
HATA');

%plot(dtrain, pltrain, '*--k',dtrain,CH3,'d-b',dtrain,CH2,'o-
r','LineWidth',2)
%legend('MEASUREMENT','RAT ADPTD COST 231 HATA','QF ADPTD. COST 231
HATA');

%plot(dtrain, pltrain, '*--k',hatad,HO,'d-b',dtrain,CH,'o-
r','LineWidth',2)
%legend('MEASUREMENT','HATA-OKUMURA','COST 231 HATA');

%MAIN
%plot(dtrain, pltrain, '*--k',hatad,HO,'d-c',hatad,HO2,'d-
b',dtrain,CH,'o-k',dtrain,CH2,'-r','LineWidth',2)
%legend('MEASUREMENT','HATA-OKUMURA','OPT. HATA-OKUMURA','COST 231
HATA','ADPTD. COST 231 HATA');

xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')

grid on
```

APPENDIX B3: Technique A (Rural Terrain)

```
% RURAL TERRAIN

%TECHNIQUE A: SPLITTING DATA INTO 60% TRAINING, 10% VALIDATION AND 30%
TESTING

close all
clear all
```

```
clc

%GSM NETWORK PARAMETERS
     hb = 33;
     f = 900;
     hm = 1.5;
     EIRP=46;

%HATA-OKUMURA CONSTANTS
    H_CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2-18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=0; % For small and medium city

   dtrain= (0.1:0.3:4.3);  %  Transmitter-Receiver separations

%   Training Path loss data
pltrain= [90
106
111
127
125
133
130
136
144
141
144
147
144
149
150
]';




%   Hata Path loss data
hatad= (1.0:0.3:4.3);
plhata = [127
125
133
130
136
144
141
```

```
144
147
144
149
150
]';

% FORMATTING BASE STATION DATA TO HAVE MEAN=0, AND STD=1
[pn,ps1] = mapstd(dtrain);
[tn,ts] = mapstd(pltrain);

rand('seed',6273432);
[trainP,valP,testP,trainInd,valInd,testInd] =
dividerand(pn,0.6,0.1,0.3);
[trainT,valT,testT] = divideind(tn,trainInd,valInd,testInd);


% MLP-NN
net = newff(pn,tn,[3],{},'trainlm');
net.trainParam.epochs = 105;
net.trainParam.goal = 0.1;
[net,tr] = train(net,trainP,trainT);

simulate = sim(net,testP);  %  STEP4:  SIMULATING NETWORK RESPONSE TO
INPUTS
simulate_rev = mapstd('reverse',simulate,ts);

pltrain_rev=mapstd('reverse',trainT,ts);
dtrain_rev=mapstd('reverse',trainP,ps1);
%plval_rev=mapstd('reverse',valT,ts);
%dval_rev=mapstd('reverse',valP,ps1);

plval_rev=mapstd('reverse',testT,ts);
dval_rev=mapstd('reverse',testP,ps1);

T=plval_rev; MLP=simulate_rev;
MLP_PE = T-MLP;  % MLP-NN Prediction Error
MLP_MSE = mse(MLP_PE); % MLP-NN Mean Square Error
MLP_RMSE=sqrt(MLP_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-MLP);
MLP_ABSERR=1/N*sum(ABSER);% MLP-NN Absolute Error
ERRSUM=sum((T-MLP).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
MLP_RS=1-(ERRSUM/ERRSUMPL); %MLP-NN R-Squared computation
a1=(sum(ABSER.^2)-N*MLP_ABSERR^2);
MLP_STD=sqrt(a1/(N-1));%MLP-NN Standard Deviation

MLP_OUTPUT=[MLP_ABSERR;MLP_STD;MLP_RMSE;MLP_RS]
```

```
% GRBF-NN
eg = 0.1; % sum-squared error goal(default 0.0)
sc = 0.5;

[net,tr]= newrb(trainP,trainT,eg,sc);
%SIM_RB = sim(net,valP);
SIM_RB = sim(net,testP);
REV_SIM = mapstd('reverse',SIM_RB,ts);

RBF=REV_SIM;
RBF_PE = T-RBF;  % RBF-NN Prediction Error
RBF_MSE = mse(RBF_PE); % RBF-NN Mean Square Error
RBF_RMSE=sqrt(RBF_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-RBF);
RBF_ABSERR=1/N*sum(ABSER);% RBF-NN Absolute Error
ERRSUM=sum((T-RBF).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
RBF_RS=1-(ERRSUM/ERRSUMPL); %RBF-NN R-Squared computation
a1=(sum(ABSER.^2)-N*RBF_ABSERR^2);
RBF_STD=sqrt(a1/(N-1));%RBF-NN Standard Deviation
RBF_OUTPUT=[RBF_ABSERR;RBF_STD;RBF_RMSE;RBF_RS];

% ADAPTED COST 231 HATA PREDICTION
d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;
CH2=CHPL';
CH3=CHPL.*(0.93+0.0126.*d);%Optimised
CH=CH3';
%CH=CH2;
T=pltrain';
N=length(pltrain);

CH_PE = T-CH;  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH);
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation
CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];

% ADAPTED HATA-OKUMURA PREDICTION
d=hatad;
```

```
AHATA=69.55+26.16*log10(f)-13.82*log10(hb)-H_CH;
BHATA=(44.9-6.55*log10(hb))*log10(d);
PLu=AHATA+BHATA;
PLsu=PLu-5.4-2*(log10(f/28))^2;
T=plhata;
N=length(T);
HO2=PLsu';
HO3=PLsu.*(1+0.0156.*d);% Adapted
HO=HO3';
%HO=HO2;
HO_PE = T-HO';  % COST 231 HATA Prediction Error
HO_MSE = mse(HO_PE); % HATA Mean Square Error
HO_RMSE=sqrt(HO_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-HO');
HO_ABSERR=1/N*sum(ABSER);%  Absolute Error
ERRSUM=sum((T-HO').^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
HO_RS=1-(ERRSUM/ERRSUMPL); % R-Squared computation
TEMP=(sum(ABSER.^2)-N*HO_ABSERR^2);
HO_STD=sqrt(TEMP/(N-1));% Standard Deviation

HO_OUTPUT=[HO_ABSERR;HO_STD;HO_RMSE;HO_RS]; % HATA Performance
statistics
STATS=[HO_OUTPUT CH_OUTPUT]

STATS=[MLP_OUTPUT RBF_OUTPUT CH_OUTPUT HO_OUTPUT]
plot(dval_rev,plval_rev,'*--k',dval_rev,MLP,'s-g',dval_rev,RBF,'o-
b',hatad,HO,'d-r',hatad,HO2,'d-c',dtrain,CH,'-k',dtrain,CH2,'o--
r','LineWidth',2);

legend('TEST DATA','MLP-NN','GRBF-NN','ADPTD. HATA-OK','HATA-
OKUMURA','ADPTD. COST 231 HATA','COST 231 HATA');
xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')

grid on
```

APPENDIX B4: Techniques B and C (Rural)

```
% RURAL TERRAIN
%   TECHNIQUE B: TRAINING WITH GEOM. MEAN AND TESTING WITH TESTING SET
%   TECHNIQUE C: TRAINING WITH ONE BST AND TESTING WITH ANOTHER

close all
clear all
clc
```

```
%GSM NETWORK PARAMETERS
     hb = 33;
     f = 900;
     hm = 1.5;
     EIRP=46;

%HATA-OKUMURA CONSTANTS
    CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2+18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=0; % For small and medium city


%   STEP 1: ASSEMBLING TRAINING DATA

dtrain= [0.1:0.3:4.3];  %  Transmitter-Receiver separations

%   Training Path loss data
pltrain= [91.18
105.95
109.79
121.36
123.11
129.12
126.54
133.17
135.00
136.17
142.79
142.93
144.95
148.79
148.39
]';



%   Testing Path loss data
dtest= [0.1:0.3:4.3];

pltest = [94
102
113
121
134
131
131
```

```
129
134
139
137
135
144
148
150
]';

% HATA-OKUMURA TESTING DATA
phata = [1.0:0.3:4.3]; %  Hata-Okumura Transmitter-Receiver
separations

plhata = [121
134
131
131
129
134
139
137
135
144
148
150
]';


% FORMATTING BST1 AND BST2 DATA TO HAVE MEAN=0, AND STD=1

[pn,ps1] = mapstd(dtrain);
[tn,ts] = mapstd(pltrain);
[pn2,ps2] = mapstd(dtest);
[tn2,ts2] = mapstd(pltest);

[hpn1,hps1] = mapstd(phata);
[htn2,hts2] = mapstd(plhata);
[hpn2,hps2] = mapstd(phata);
 rand('seed',62734347);

% MLP-NN TRAINING OF ONE BST AND TESTING WITH ANOTHER

net = newff(pn,tn,5,{},'trainlm');% STEP 2: CREATING NETWORK OBJECT
% Default Training Algorithm: Levenberg-Marquardt backpropagation
net.trainParam.epochs = 105; % Number of iterations
net.trainParam.goal = 0.001;
[net,tr] = train(net,pn,tn); %  STEP3:  TRAINING NETWORK

an = sim(net,pn2);  %  STEP4:  SIMULATING NETWORK RESPONSE TO INPUTS
```

```
pltrain_rev=mapstd('reverse',trainT,ts);
dtrain_rev=mapstd('reverse',trainP,ps1);
%plval_rev=mapstd('reverse',valT,ts);
%dval_rev=mapstd('reverse',valP,ps1);

plval_rev=mapstd('reverse',testT,ts);
dval_rev=mapstd('reverse',testP,ps1);

%HATA-OKUMURA VECTORS

v1=mapstd('reverse',hpn1,hps1);
kk=mapstd('reverse',htn2,hts2);
vv=mapstd('reverse',hpn2,hps2);

T=plval_rev; MLP=simulate_rev;
MLP_PE = T-MLP;  % MLP-NN Prediction Error
MLP_MSE = mse(MLP_PE); % MLP-NN Mean Square Error
MLP_RMSE=sqrt(MLP_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-MLP);
MLP_ABSERR=1/N*sum(ABSER);% MLP-NN Absolute Error
ERRSUM=sum((T-MLP).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
MLP_RS=1-(ERRSUM/ERRSUMPL); %MLP-NN R-Squared computation
a1=(sum(ABSER.^2)-N*MLP_ABSERR^2);
MLP_STD=sqrt(a1/(N-1));%MLP-NN Standard Deviation

MLP_OUTPUT=[MLP_ABSERR;MLP_STD;MLP_RMSE;MLP_RS]

% RBF-NN TRAINING OF BST1 AND COMPARING WITH BST2

eg = 0.01; % Sum-squared error goal(default 0.0)
sc = 0.322;  % Spread constant
 [net,tr]= newrb(pn,tn,eg,sc);   %   STEP2:  CREATING NETWORK OBJECT
SIM_RB = sim(net,testP);
REV_SIM = mapstd('reverse',SIM_RB,ts);

RBF=REV_SIM;
RBF_PE = T-RBF;  % RBF-NN Prediction Error
RBF_MSE = mse(RBF_PE); % RBF-NN Mean Square Error
RBF_RMSE=sqrt(RBF_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-RBF);
RBF_ABSERR=1/N*sum(ABSER);% RBF-NN Absolute Error
ERRSUM=sum((T-RBF).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
RBF_RS=1-(ERRSUM/ERRSUMPL); %RBF-NN R-Squared computation
a1=(sum(ABSER.^2)-N*RBF_ABSERR^2);
RBF_STD=sqrt(a1/(N-1));%RBF-NN Standard Deviation
```

```
RBF_OUTPUT=[RBF_ABSERR;RBF_STD;RBF_RMSE;RBF_RS];


% QRT ADAPTED COST 231 HATA PREDICTION
d=p;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
PLch=A+B;
t5=toc;
T=pltrain';
N=length(pltrain);

CH_PE = T-CH;  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH);
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation
CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];


% QRT ADAPTED  HATA-OKUMURA PREDICTION

d=phata;
A=69.55+26.16*log10(f)-13.82*log10(hb)-CH;
B=(44.9-6.55*log10(hb))*log10(d);
PLu=A+B;
%PLs=PLu-SU;
PLo=PLu-4.78*(log10(f))^2+18.33*log10(f)-40.94;
PLsu=PLu-5.4-2*(log10(f/28))^2;
 T=plhata;
N=length(T);


HO2=PLsu';
HO3=PLsu.*(1+0.0156.*d);% Adapted
HO=HO3';
%HO=HO2;
HO_PE = T-HO';  % COST 231 HATA Prediction Error
HO_MSE = mse(HO_PE); % HATA Mean Square Error
HO_RMSE=sqrt(HO_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-HO');
HO_ABSERR=1/N*sum(ABSER);%  Absolute Error
ERRSUM=sum((T-HO').^2);%Rsquared computation
```

```
ERRSUMPL=sum((T-mean(T)).^2);
HO_RS=1-(ERRSUM/ERRSUMPL); % R-Squared computation
TEMP=(sum(ABSER.^2)-N*HO_ABSERR^2);
HO_STD=sqrt(TEMP/(N-1));% Standard Deviation

HO_OUTPUT=[HO_ABSERR;HO_STD;HO_RMSE;HO_RS]; % HATA Performance
statistics

STATS=[HO_OUTPUT CH_OUTPUT]
 STATS=[MLP_OUTPUT RBF_OUTPUT CH_OUTPUT HO_OUTPUT]

plot(v,k,'*--k',v,Y1,'s-g',v,Y2,'o-b',v,Y52,'d-c',v,Y5,'d-r',vv,Y42,'-
--m',vv,Y4,'-k','LineWidth',2);

legend('TEST DATA','MLP-NN','GRBF-NN','COST 231 HATA','ADPT. COST 231
HATA','HATA-OKM','ADPT. HATA-OKM');

grid on
```

APPENDIX B5: Applicability Test and Adaptation of Empirical Models (Abuja)

```
%  ADAPTATION OF EMPIRICAL MODELS FOR URBAN TERRAIN

close all
clear all
clc

%GSM NETWORK PARAMETERS
```

```
     hb = 28;
     f = 900;
     hm = 1.5;
     EIRP=45;

%HATA-OKUMURA CONSTANTS
    CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2-18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=3; % For small and medium city

%WALFISCH-IKEGAMI CONSTANTS
    rw = 30;
    B1 = 2 * rw;
    roof = 3;
    floors = 7;
    Hroof = 3 * floors + roof;
    dH = hb - Hroof;

% Lbsh
if hb > Hroof
        Lbsh = -18 * log10(1 + dH);
else
        Lbsh = 0;
end


if hb > Hroof
        Kd = 18;
else
        Kd = 18 - 15 * dH / Hroof;
end

    Kf = -4 + 1.5 * (f / 925 - 1); %for Large City
%Kf = -4 + 0.7 * (f / 925 - 1);
%Lori
    Lsum = 0;
    teta=10;
    jt=0;

while 90 > teta,
if teta >= 0 && teta < 35
            Lteta = -10 + 0.35 * teta;
elseif teta >= 35 && teta < 55
            Lteta = 2.5 + 0.075 * (teta - 35);
else
            Lteta = 4 - 0.114 * (teta - 55);
```

```
end
        Lsum = Lsum + Lteta;
        teta=teta+10;
        jt=jt+1;
end

   Lori=Lsum/jt;
   Lrts=-16.9-10*log10(rw)+10*log10(f)+20*log10(Hroof - hm)+Lori

%   STEP 1: ASSEMBLING TRAINING DATA

dtrain= (0.05:0.1:1.45);  %  Transmitter-Receiver separations

%   Training Path loss data
pltrain= [85.5
107.7
119.0
119.8
124.8
130.8
129.5
132.7
137.4
137.2
137.6
141.0
142.8
144.9
147.1
]';

T=pltrain;



% COST 231 HATA ADAPTATION

d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;
trd=pltrain;
N=length(T);

Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
        Ncc = sum(d.^4);
        Dcc = sum(trd);
        Hcc = sum(d.*trd);
```

```
        Pcc = sum((d.^2).*trd);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;

            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Mcc - Fcc * Lcc);
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc -
Gcc * Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc
* Lcc) + Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
            AA01 = DT1 / dt
            AA02 = DT2 / dt

           L=AA00+AA01*d+AA02*(d.*d);

           Q=L./CHPL;

           QUOTS=[d Q]

           bl = sum(d);
     cl = sum(Q);
     el = sum(d.^2);
     fl = sum(d.*Q);

            al = N;
            dl = bl;

            dt = al * el - dl * bl;
            DT0 = cl * el - fl * bl;
            DT1 = al * fl - dl * cl;

            A0 = DT0 / dt
            A1 = DT1 / dt

            CH2=CHPL.*(A0+A1.*d);
%Y=PLch.*Q;
%CH=CHPL.*(0.9364+0.0126.*d);
CH=CHPL;
%CH=CHPL.*(1.05+0.0097.*d);';
%CH2=CHPL.*(1.05+0.0097.*d);%Optimised
T=pltrain';
%T=pltrain';
N=length(pltrain);
```

```
CH_PE = T-CH';  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH');
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH').^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];


% QRT ADAPTION OF THE COST 231 WALFISCH-IKEGAMI

   d=dtrain;
if hb > Hroof
                Ka = 54;
else
if  d>= 0.5
                    Ka = 54 - 0.8 * dH;
else
                    Ka = 54 - 0.8 * dH * d / 0.5;
end
end

            Lmsd = Lbsh + Ka + Kd * Log10(d)+ Kf * Log10(f)- 9 *
Log10(B1)

            Lfs = 32.45 + 20 * Log10(d)+ 20 * Log10(f)

            PL = Lfs + Lmsd + Lrts;

        Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
        Ncc = sum(d.^4);
        Dcc = sum(pltrain);
        Hcc = sum(d.*pltrain);
        Pcc = sum((d.^2).*pltrain);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;

            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Mcc - Fcc * Lcc);
```

```
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc -
Gcc * Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc
* Lcc) + Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
            AA01 = DT1 / dt
            AA02 = DT2 / dt

           L=AA00+AA01*d+AA02*(d.*d);

           Q=L./PL;

     bl = sum(d);
     cl = sum(Q);
     el = sum(d.^2);
     fl = sum(d.*Q);
           al = N;
           dl = bl;

           dt = al * el - dl * bl;
           DT0 = cl * el - fl * bl;
           DT1 = al * fl - dl * cl;

           A0 = DT0 / dt
           A1 = DT1 / dt

           WI2=PL.*(A0+A1.*d);

WI=PL;
%WI=WI.*(1.0805-0.0155.*d);
%WI2=WI.*(1.0805-0.0155.*d);% Optimised
WI_PE = T-WI';  % COST 231 HATA Prediction Error
WI_MSE = mse(WI_PE); % CH Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI');
WI_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-WI').^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

WI_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

RMSE1=sqrt(sum((T-CH').^2)/(N-1))
RMSE2=sqrt(sum((T-WI').^2)/(N-1))
```

```
STATS=[WI_OUTPUT CH_OUTPUT]

plot(dtrain, pltrain, '*--k',dtrain,WI,'d-c',dtrain,WI2,'d-
r',dtrain,CH,'--m',dtrain,CH2,'--r','LineWidth',2)
legend('MEASUREMENT','COST 231 WALF-IKEG','OPT. COST 231 WALF-
IKG','COST 231 HATA','OPT. COST 231 HATA');

xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')

grid on
```

APPENDIX B6 : Technique A (Abuja)

```
% URBAN TERRAIN (ABUJA)

%TECHNIQUE A: SPLITTING DATA INTO 60% TRAINING, 10% VALIDATION AND 30%
TESTING


close all
clear all
```

```
clc

%GSM NETWORK PARAMETERS
     hb = 28;
     f = 900;
     hm = 1.5;
     EIRP=45;

%HATA-OKUMURA CONSTANTS
    CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2-18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=3; % For small and medium city

%WALFISCH-IKEGAMI CONSTANTS
    rw = 30;
    B1 = 2 * rw;
    roof = 3;
    floors = 7;
    Hroof = 3 * floors + roof;
    dH = hb - Hroof;

% Lbsh
if hb > Hroof
        Lbsh = -18 * log10(1 + dH);
else
        Lbsh = 0;
end


if hb > Hroof
        Kd = 18;
else
        Kd = 18 - 15 * dH / Hroof;
end
    Kf = -4 + 1.5 * (f / 925 - 1); %for Large City
    Kf = -4 + 0.7 * (f / 925 - 1);

%Lori
    Lsum = 0;
    teta=10;
    jt=0;
while 90 > teta,
if teta >= 0 && teta < 35
            Lteta = -10 + 0.35 * teta;
elseif teta >= 35 && teta < 55
            Lteta = 2.5 + 0.075 * (teta - 35);
```

```
else
            Lteta = 4 - 0.114 * (teta - 55);
end
        Lsum = Lsum + Lteta;
        teta=teta+10;
        jt=jt+1;
end

   Lori=Lsum/jt;
   Lrts=-16.9-10*log10(rw)+10*log10(f)+20*log10(Hroof - hm)+Lori;


   dtrain= (0.05:0.1:1.45);  %  Transmitter-Receiver separations

%   Training Path loss data
pltrain= [89
102
116
126
120
122
123
128
126
136
133
135
141
140
143
]';

% FORMATTING BASE STATION DATA TO HAVE MEAN=0, AND STD=1

[pn,ps1] = mapstd(dtrain);
[tn,ts] = mapstd(pltrain);
rand('seed',627343);
rand('seed',6273432);
[trainP,valP,testP,trainInd,valInd,testInd] =
dividerand(pn,0.6,0.1,0.3);
[trainT,valT,testT] = divideind(tn,trainInd,valInd,testInd);


% MLP-NN
net = newff(pn,tn,[3],{},'trainlm');
net.trainParam.epochs = 105;
net.trainParam.goal = 0.01;
[net,tr] = train(net,trainP,trainT);
```

```
simulate = sim(net,testP);  %  STEP4:  SIMULATING NETWORK RESPONSE TO
INPUTS
simulate_rev = mapstd('reverse',simulate,ts);

pltrain_rev=mapstd('reverse',trainT,ts);
dtrain_rev=mapstd('reverse',trainP,ps1);
%plval_rev=mapstd('reverse',valT,ts);
%dval_rev=mapstd('reverse',valP,ps1);

plval_rev=mapstd('reverse',testT,ts);
dval_rev=mapstd('reverse',testP,ps1);

T=plval_rev; MLP=simulate_rev;
MLP_PE = T-MLP;  % MLP-NN Prediction Error
MLP_MSE = mse(MLP_PE); % MLP-NN Mean Square Error
MLP_RMSE=sqrt(MLP_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-MLP);
MLP_ABSERR=1/N*sum(ABSER);% MLP-NN Absolute Error
ERRSUM=sum((T-MLP).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
MLP_RS=1-(ERRSUM/ERRSUMPL); %MLP-NN R-Squared computation
a1=(sum(ABSER.^2)-N*MLP_ABSERR^2);
MLP_STD=sqrt(a1/(N-1));%MLP-NN Standard Deviation

MLP_OUTPUT=[MLP_ABSERR;MLP_STD;MLP_RMSE;MLP_RS]

% GRBF-NN
eg = 0.1; % sum-squared error goal(default 0.0)
sc = 0.502;

[net,tr]= newrb(trainP,trainT,eg,sc);
%SIM_RB = sim(net,valP);
SIM_RB = sim(net,testP);
REV_SIM = mapstd('reverse',SIM_RB,ts);

RBF=REV_SIM;
RBF_PE = T-RBF;  % RBF-NN Prediction Error
RBF_MSE = mse(RBF_PE); % RBF-NN Mean Square Error
RBF_RMSE=sqrt(RBF_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-RBF);
RBF_ABSERR=1/N*sum(ABSER);% RBF-NN Absolute Error
ERRSUM=sum((T-RBF).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
RBF_RS=1-(ERRSUM/ERRSUMPL); %RBF-NN R-Squared computation
a1=(sum(ABSER.^2)-N*RBF_ABSERR^2);
RBF_STD=sqrt(a1/(N-1));%RBF-NN Standard Deviation

RBF_OUTPUT=[RBF_ABSERR;RBF_STD;RBF_RMSE;RBF_RS];
```

```
% QRT ADAPTED COST 231 HATA PREDICTION
d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;
CH2=CHPL';
CH3=CHPL.*(1.05+0.0097.*d);%Optimised
CH=CH3';
%CH=CH2;

T=pltrain';
N=length(pltrain);

CH_PE = T-CH;  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH);
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];

% QRT ADAPTED COST 231 WALFISCH-IKEGAMI PREDICTION

   d=dtrain;
if hb > Hroof
                Ka = 54;
else
if  d>= 0.5
                    Ka = 54 - 0.8 * dH;
else
                    Ka = 54 - 0.8 * dH * d / 0.5;
end
end

            Lmsd = Lbsh + Ka + Kd * Log10(d)+ Kf * Log10(f)- 9 *
Log10(B1);

            Lfs = 32.45 + 20 * Log10(d)+ 20 * Log10(f);

            PL = Lfs + Lmsd + Lrts;

WI3=PL.*(1.0805-0.0155.*d);% Optimised
WI=WI3';
```

```
WI2=PL';
WI_PE = T-WI;  % COST 231 HATA Prediction Error
WI_MSE = mse(WI_PE); % CH Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI);
WI_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-WI).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));%CH Standard Deviation
WI_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

STATS=[MLP_OUTPUT RBF_OUTPUT WI_OUTPUT CH_OUTPUT]

%plot(dval_rev,plval_rev,'*--k',dval_rev,MLP,'s-g',dval_rev,RBF,'o-
b',dtrain,WI,'d-r',dtrain,CH,'--r','LineWidth',2)
%legend('TEST DATA','MLP-NN','GRBF-NN','OPT. COST 231 WALF-IKG','COST
231 HATA');

plot(dval_rev,plval_rev,'*--k',dval_rev,MLP,'s-g',dval_rev,RBF,'o-
b',dtrain,WI2,'d-c',dtrain,WI,'d-r',dtrain,CH2,'--m',dtrain,CH,'-
k','LineWidth',2);

legend('TEST DATA','MLP-NN','GRBF-NN','COST 231 WALF-IKEG','ADPTD.
COST 231 WALF-IKG','COST 231 HATA','ADPTD. COST 231 HATA');
xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')
grid on
```

APPENDIX B7: Techniques B and C (Abuja)

```
% URBAN TERRAIN
%   TECHNIQUE B: TRAINING WITH GEOM. MEAN AND TESTING WITH TESTING SET
%   TECHNIQUE C: TRAINING WITH ONE BST AND TESTING WITH ANOTHER

close all
clear all
clc

%GSM NETWORK PARAMETERS
     hb = 28;
```

```
     f = 900;
     hm = 1.5;
     EIRP=45;

%HATA-OKUMURA CONSTANTS
    CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2-18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=3; % For small and medium city

%WALFISCH-IKEGAMI CONSTANTS
    rw = 30;
    B1 = 2 * rw;
    roof = 3;
    floors = 7;
    Hroof = 3 * floors + roof;
    dH = hb - Hroof;

% Lbsh
if hb > Hroof
        Lbsh = -18 * log10(1 + dH);
else
        Lbsh = 0;
end


if hb > Hroof
        Kd = 18;
else
        Kd = 18 - 15 * dH / Hroof;
end

    Kf = -4 + 1.5 * (f / 925 - 1); %for Large City
    Kf = -4 + 0.7 * (f / 925 - 1);

%Lori
    Lsum = 0;
    teta=10;
    jt=0;
while 90 > teta,
if teta >= 0 && teta < 35
            Lteta = -10 + 0.35 * teta;
elseif teta >= 35 && teta < 55
            Lteta = 2.5 + 0.075 * (teta - 35);
else
            Lteta = 4 - 0.114 * (teta - 55);
end
```

```
        Lsum = Lsum + Lteta;
        teta=teta+10;
        jt=jt+1;
end

   Lori=Lsum/jt;
   Lrts=-16.9-10*log10(rw)+10*log10(f)+20*log10(Hroof - hm)+Lori;

%   STEP 1: ASSEMBLING TRAINING DATA

dtrain= (0.05:0.1:1.45);  %  Transmitter-Receiver separations

%   Training Path loss data
pltrain= [84
111
120
135
123
128
126
120
134
135
140
143
143
146
150
]';

%   Testing Path loss data
dtest= (0.05:0.1:1.45);
pltest = [87
105
118
128
117
124
122
130
125
134
133
```

```
138
140
145
142
]';

% FORMATTING TRAINING AND TESTING DATA TO HAVE MEAN=0, AND STD=1

[pn,ps1] = mapstd(dtrain);
[tn,ts] = mapstd(pltrain);
[pn2,ps2] = mapstd(dtest);
[tn2,ts2] = mapstd(pltest);

%rand('seed',62734347);
rand('seed',6273434);

% MLP-NN TRAINING AND TESTING

net = newff(pn,tn,4,{},'trainlm');% STEP 2: CREATING NETWORK OBJECT
% Default Training Algorithm: Levenberg-Marquardt backpropagation
net.trainParam.epochs = 100; % Number of iterations
net.trainParam.goal = 0.0001;
[net] = train(net,pn,tn); %  STEP3:  TRAINING NETWORK

simulate = sim(net,pn2);  %  STEP4:  SIMULATING NETWORK RESPONSE TO
INPUTS
simulate_rev = mapstd('reverse',simulate,ts2);


dtrain_rev=mapstd('reverse',pn,ps1);
pltrain_rev=mapstd('reverse',tn,ts);
dtest_rev=mapstd('reverse',pn2,ps2);
pltest_rev=mapstd('reverse',tn2,ts2);

T=pltest_rev;
MLP=simulate_rev;
MLP_PE = T-MLP;  % MLP-NN Prediction Error
MLP_MSE = mse(MLP_PE); % MLP-NN Mean Square Error
MLP_RMSE=sqrt(MLP_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-MLP);
MLP_ABSERR=1/N*sum(ABSER);% MLP-NN Absolute Error
ERRSUM=sum((T-MLP).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
MLP_RS=1-(ERRSUM/ERRSUMPL); %MLP-NN R-Squared computation
a1=(sum(ABSER.^2)-N*MLP_ABSERR^2);
MLP_STD=sqrt(a1/(N-1));%MLP-NN Standard Deviation

MLP_OUTPUT=[MLP_ABSERR;MLP_STD;MLP_RMSE;MLP_RS];
```

```
% GRBF-NN TRAINING AND TESTING

eg = 0.001; % Sum-squared error goal(default 0.0)
sc = 0.822;  % Spread constant

[net,tr]= newrb(pn,tn,eg,sc);   %   STEP2:  CREATING NETWORK OBJECT

%Max no of neurons(default=length(P)), No of neurons to
add(default(25)

SIM_RB = sim(net,pn2);    %  STEP3:  SIMULATING NETWORK RESPONSE TO
INPUTS
REV_SIM = mapstd('reverse',SIM_RB,ts2);

RBF=REV_SIM;
RBF_PE = T-RBF;  % RBF-NN Prediction Error
RBF_MSE = mse(RBF_PE); % RBF-NN Mean Square Error
RBF_RMSE=sqrt(RBF_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-RBF);
RBF_ABSERR=1/N*sum(ABSER);% RBF-NN Absolute Error
ERRSUM=sum((T-RBF).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
RBF_RS=1-(ERRSUM/ERRSUMPL); %RBF-NN R-Squared computation
a1=(sum(ABSER.^2)-N*RBF_ABSERR^2);
RBF_STD=sqrt(a1/(N-1));%RBF-NN Standard Deviation

RBF_OUTPUT=[RBF_ABSERR;RBF_STD;RBF_RMSE;RBF_RS];


% COST 231 HATA PREDICTION

d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;
CH2=CHPL';
CH3=CHPL.*(1.05+0.0097.*d);%Optimised
CH=CH3;
%CH=CH2';

T=pltest';
N=length(pltest);
CH_PE = T-CH';  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH');
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH').^2);%Rsquared computation
```

```
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];

% ADAPTED COST 231 WALFISCH-IKEGAMI PREDICTION

   d=dtest;
if hb > Hroof
                Ka = 54;
else
if  d>= 0.5
                    Ka = 54 - 0.8 * dH;
else
                    Ka = 54 - 0.8 * dH * d / 0.5;
end
end

            Lmsd = Lbsh + Ka + Kd * Log10(d)+ Kf * Log10(f)- 9 *
Log10(B1);

            Lfs = 32.45 + 20 * Log10(d)+ 20 * Log10(f);

            PL = Lfs + Lmsd + Lrts;

WI3=PL.*(1.0805-0.0155.*d);% Optimised
WI=WI3';
WI2=PL';
%WI=WI2;
WI_PE = T-WI;  % COST 231 HATA Prediction Error
WI_MSE = mse(WI_PE); % CH Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI);
WI_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-WI).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

WI_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

STATS=[MLP_OUTPUT RBF_OUTPUT WI_OUTPUT CH_OUTPUT]

plot(dtest,pltest_rev,'*--k',dtrain,MLP,'s-g',dtrain,RBF,'o-
b',dtrain,WI,'d-r',dtrain,WI2,'d-c',dtrain,CH,'-k',dtrain,CH2,'--
r','LineWidth',2)
```

```
legend('TEST DATA','MLP-NN','GRBF-NN','ADPTD.COST 231 WALF-IKEG','COST
231 WALF-IKG','ADPTD. COST 231 HATA','COST 231 HATA');

%plot(v,k,'*--k',v,Y1,'s-g',v,Y2,'o-b',v,Y6,'d-c',v,Y62,'d-r',v,Y5,'--
m',v,Y52,'--r','LineWidth',2);
%legend('TEST DATA','MLP-NN','GRBF-NN','COST 231 WALF-IKEG','OPT. COST
231 WALF-IKEG','COST 231 HATA','OPT. COST 231 HATA');
xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')

grid on
```

APPENDIX B8: Applicability Test and Adaptation of Empirical Models (Maiduguri)

```
%  ADAPTATION OF EMPIRICAL MODELS FOR SEMI-URBAN TERRAIN

close all
clear all
clc

%GSM NETWORK PARAMETERS
     hb = 40;
     f = 900;
     hm = 1.5;
```

```
     EIRP=46;

%HATA-OKUMURA CONSTANTS
    CH=0.8+(1.1*log10(f)-0.7)*hm-1.56*log10(f);
    SU=5.4+2*(log10(f/28))^2;  %Sub-urban
    OP=4.78*(log10(f))^2-18.33*log10(f)-40.94; %Open Area

%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=0; % For small and medium city

%WALFISCH-IKEGAMI CONSTANTS
    rw = 20;
    B1 = 2 * rw;
    roof = 3;
    floors = 3;
    Hroof = 3 * floors + roof;
    dH = hb - Hroof;

% Lbsh
if hb > Hroof
        Lbsh = -18 * log10(1 + dH);
else
        Lbsh = 0;
end


if hb > Hroof
        Kd = 18;
else
        Kd = 18 - 15 * dH / Hroof;
end

%Kf = -4 + 1.5 * (f / 925 - 1); %for Large City

    Kf = -4 + 0.7 * (f / 925 - 1);

%Lori
    Lsum = 0;
    teta=10;
    jt=0;
while 90 > teta,
if teta >= 0 && teta < 35
            Lteta = -10 + 0.35 * teta;
elseif teta >= 35 && teta < 55
            Lteta = 2.5 + 0.075 * (teta - 35);
else
            Lteta = 4 - 0.114 * (teta - 55);
end
        Lsum = Lsum + Lteta;
```

```
        teta=teta+10;
        jt=jt+1;
end

   Lori=Lsum/jt;
   Lrts=-16.9-10*log10(rw)+10*log10(f)+20*log10(Hroof - hm)+Lori


dtrain= (0.1:0.2:2.5);  %  Transmitter-Receiver separations

%   Path loss data
pltrain= [89.67
103.64
109.83
115.12
118.96
123.60
124.37
130.12
136.41
135.99
137.27
142.30
147.22
]';

T=pltrain;


% COST 231 HATA ADAPTATION

d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;

trd=pltrain;
N=length(T);

Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
        Ncc = sum(d.^4);
        Dcc = sum(trd);
        Hcc = sum(d.*trd);
        Pcc = sum((d.^2).*trd);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;
```

```
            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Mcc - Fcc * Lcc);
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc -
Gcc * Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc
* Lcc) + Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
            AA01 = DT1 / dt
            AA02 = DT2 / dt

           L=AA00+AA01*d+AA02*(d.*d);

           Q=L./CHPL;

           QUOTS=[d Q]

           bl = sum(d);
     cl = sum(Q);
     el = sum(d.^2);
     fl = sum(d.*Q);

            al = N;
            dl = bl;

            dt = al * el - dl * bl;
            DT0 = cl * el - fl * bl;
            DT1 = al * fl - dl * cl;

            A0 = DT0 / dt
            A1 = DT1 / dt

            CH2=CHPL.*(A0+A1.*d);
%Y=PLch.*Q;

CH=CHPL;

T=pltrain';
N=length(pltrain);

CH_PE = T-CH';  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH');
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH').^2);%Rsquared computation
```

```
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];

% QRT ADAPTED COST 231 WALFISCH-IKEGAMI

   d=dtrain;
if hb > Hroof
                Ka = 54;
else
if  d>= 0.5
                    Ka = 54 - 0.8 * dH;
else
                    Ka = 54 - 0.8 * dH * d / 0.5;
end
end

            Lmsd = Lbsh + Ka + Kd * Log10(d)+ Kf * Log10(f)- 9 *
Log10(B1)

            Lfs = 32.45 + 20 * Log10(d)+ 20 * Log10(f)

            PL = Lfs + Lmsd + Lrts;


        Bcc = sum(d);
        Ccc = sum(d.^2);
        Gcc = sum(d.^3);
        Ncc = sum(d.^4);
        Dcc = sum(pltrain);
        Hcc = sum(d.*pltrain);
        Pcc = sum((d.^2).*pltrain);
            Ecc = Bcc;
            Fcc = Ccc;
            Lcc = Ccc;
            Mcc = Gcc;

            dt = N * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Mcc - Fcc * Lcc);
            DT0 = Dcc * (Fcc * Ncc - Gcc * Mcc) - Bcc * (Hcc * Ncc -
Gcc * Pcc) + Ccc * (Hcc * Mcc - Fcc * Pcc);
            DT1 = N * (Hcc * Ncc - Gcc * Pcc) - Dcc * (Ecc * Ncc - Gcc
* Lcc) + Ccc * (Ecc * Pcc - Hcc * Lcc);
            DT2 = N * (Fcc * Pcc - Hcc * Mcc) - Bcc * (Ecc * Pcc - Hcc
* Lcc) + Dcc * (Ecc * Mcc - Fcc * Lcc);

            AA00 = DT0 / dt
```

```
            AA01 = DT1 / dt
            AA02 = DT2 / dt

           L=AA00+AA01*d+AA02*(d.*d);

           Q=L./PL;

     bl = sum(d);
     cl = sum(Q);
     el = sum(d.^2);
     fl = sum(d.*Q);
           al = N;
           dl = bl;

           dt = al * el - dl * bl;
           DT0 = cl * el - fl * bl;
           DT1 = al * fl - dl * cl;

           A0 = DT0 / dt
           A1 = DT1 / dt

           WI2=PL.*(A0+A1.*d);


WI=PL;
WI_PE = T-WI';  % COST 231 HATA Prediction Error
WI_MSE = mse(WI_PE); % CH Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI');
WI_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-WI').^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

WI_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

RMSE1=sqrt(sum((T-CH').^2)/(N-1))
RMSE2=sqrt(sum((T-WI').^2)/(N-1))

STATS=[WI_OUTPUT CH_OUTPUT]

plot(dtrain, pltrain, '*--k',dtrain,WI,'d-b',dtrain,WI2,'d-
r',dtrain,CH,'-k',dtrain,CH2,'--r','LineWidth',2)
legend('MEASUREMENT','COST 231 WALF-IKEG','ADPTD. COST 231 WALF-
IKG','COST 231 HATA','ADPTD. COST 231 HATA');
```

```
%plot(v,k,'*--k',v,Y1,'s-g',v,Y2,'o-b',v,Y6,'d-c',v,Y62,'d-r',v,Y5,'--
m',v,Y52,'--r','LineWidth',2);
%legend('TEST DATA','MLP-NN','GRBF-NN','COST 231 WALF-IKEG','OPT. COST
231 WALF-IKEG','COST 231 HATA','OPT. COST 231 HATA');
xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')

grid on
```

APPENDIX B9: Technique A (Maiduguri)

```
% SEMI-URBAN TERRAIN
%TECHNIQUE A: SPLITTING DATA INTO 60% TRAINING, 10% VALIDATION AND 30%
TESTING
close all
clear all
clc

%GSM NETWORK PARAMETERS
     hb = 40;
     f = 900;
     hm = 1.5;
     EIRP=46;
```

```
%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=0; % For small and medium city

%WALFISCH-IKEGAMI CONSTANTS
    rw = 20;
    B1 = 2 * rw;
    roof = 3;
    floors = 3;
    Hroof = 3 * floors + roof;
    dH = hb - Hroof;

% Lbsh
if hb > Hroof
        Lbsh = -18 * log10(1 + dH);
else
        Lbsh = 0;
end


if hb > Hroof
        Kd = 18;
else
        Kd = 18 - 15 * dH / Hroof;
end

    Kf = -4 + 1.5 * (f / 925 - 1); %for Large City
    Kf = -4 + 0.7 * (f / 925 - 1);

%Lori
    Lsum = 0;
    teta=10;
    jt=0;

while 90 > teta,
if teta >= 0 && teta < 35
            Lteta = -10 + 0.35 * teta;
elseif teta >= 35 && teta < 55
            Lteta = 2.5 + 0.075 * (teta - 35);
else
            Lteta = 4 - 0.114 * (teta - 55);
end
        Lsum = Lsum + Lteta;
        teta=teta+10;
        jt=jt+1;
end

   Lori=Lsum/jt;
   Lrts=-16.9-10*log10(rw)+10*log10(f)+20*log10(Hroof - hm)+Lori;
```

```
   dtrain= (0.1:0.2:2.5);  %  Transmitter-Receiver separations

%   Training Path loss data
pltrain= [93
104
111
119
117
127
121
129
133
140
136
140
152
]';

% FORMATTING BASE STATION DATA TO HAVE MEAN=0, AND STD=1

[pn,ps1] = mapstd(dtrain);
[tn,ts] = mapstd(pltrain);

rand('seed',6273432);

[trainP,valP,testP,trainInd,valInd,testInd] =
dividerand(pn,0.6,0.1,0.3);
%[trainP,valP,testP,trainInd,valInd,testInd] =
dividerand(pn,0.48,0.4,0.12);
%[trainP,valP,testP,trainInd,valInd,testInd] =
dividerand(pn,0.48,0.12,0.4);
[trainT,valT,testT] = divideind(tn,trainInd,valInd,testInd);


% MLP-NN
net = newff(pn,tn,[3],{},'trainlm');
net.trainParam.epochs = 100;
net.trainParam.goal = 0.1;
[net,tr] = train(net,trainP,trainT);

simulate = sim(net,testP);  %  STEP4:  SIMULATING NETWORK RESPONSE TO
INPUTS
simulate_rev = mapstd('reverse',simulate,ts);

pltrain_rev=mapstd('reverse',trainT,ts);
dtrain_rev=mapstd('reverse',trainP,ps1);
%plval_rev=mapstd('reverse',valT,ts);
%dval_rev=mapstd('reverse',valP,ps1);
```

```
plval_rev=mapstd('reverse',testT,ts);
dval_rev=mapstd('reverse',testP,ps1);

T=plval_rev; MLP=simulate_rev;
MLP_PE = T-MLP;  % MLP-NN Prediction Error
MLP_MSE = mse(MLP_PE); % MLP-NN Mean Square Error
MLP_RMSE=sqrt(MLP_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-MLP);
MLP_ABSERR=1/N*sum(ABSER);% MLP-NN Absolute Error
ERRSUM=sum((T-MLP).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
MLP_RS=1-(ERRSUM/ERRSUMPL); %MLP-NN R-Squared computation
a1=(sum(ABSER.^2)-N*MLP_ABSERR^2);
MLP_STD=sqrt(a1/(N-1));%MLP-NN Standard Deviation

MLP_OUTPUT=[MLP_ABSERR;MLP_STD;MLP_RMSE;MLP_RS]

% GRBF-NN
eg = 0.1; % sum-squared error goal(default 0.0)
%sc = 0.3273;
%sc = 0.9115;
sc = 0.51;

[net,tr]= newrb(trainP,trainT,eg,sc);
%SIM_RB = sim(net,valP);
SIM_RB = sim(net,testP);
REV_SIM = mapstd('reverse',SIM_RB,ts);

RBF=REV_SIM;
RBF_PE = T-RBF;  % RBF-NN Prediction Error
RBF_MSE = mse(RBF_PE); % RBF-NN Mean Square Error
RBF_RMSE=sqrt(RBF_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-RBF);
RBF_ABSERR=1/N*sum(ABSER);% RBF-NN Absolute Error
ERRSUM=sum((T-RBF).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
RBF_RS=1-(ERRSUM/ERRSUMPL); %RBF-NN R-Squared computation
a1=(sum(ABSER.^2)-N*RBF_ABSERR^2);
RBF_STD=sqrt(a1/(N-1));%RBF-NN Standard Deviation

RBF_OUTPUT=[RBF_ABSERR;RBF_STD;RBF_RMSE;RBF_RS];

% COST 231 HATA PREDICTION
d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;
%CH=CHPL;
```

```
CH2=CHPL.*(0.9436+0.0336.*d);
CH=CH2';
 T=pltrain';
N=length(pltrain);

% COST 231 HATA Prediction Errors
CH_PE = T-CH;  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH);
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];

% QRT ADAPTED COST 231 WALFISCH-IKEGAMI

   d=dtrain;
if hb > Hroof
                Ka = 54;
else
if  d>= 0.5
                    Ka = 54 - 0.8 * dH;
else
                    Ka = 54 - 0.8 * dH * d / 0.5;
end
end

    Lmsd = Lbsh + Ka + Kd * Log10(d)+ Kf * Log10(f)- 9 * Log10(B1);

            Lfs = 32.45 + 20 * Log10(d)+ 20 * Log10(f);

            PL = Lfs + Lmsd + Lrts;

WI=WI2';

% QRT Adapted Walf-Ikeg Prediction Errors

WI_PE = T-WI;  % Prediction Error
WI_MSE = mse(WI_PE); % CH Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI);
WI_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-WI).^2);%Rsquared computation
```

```
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));%CH Standard Deviation
WI2_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

% QRT Walf-Ikeg Prediction Errors

WI=PL';
WI_PE = T-WI;
WI_MSE = mse(WI_PE); % CH Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI);
WI_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-WI).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

WI_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];
STATS=[MLP_OUTPUT RBF_OUTPUT WI2_OUTPUT WI_OUTPUT CH_OUTPUT]

plot(dval_rev,plval_rev,'*--k',dval_rev,MLP,'s-g',dval_rev,RBF,'o-
b',dtrain,WI,'d-c',dtrain,WI2,'d-r',dtrain,CH,'o-k','LineWidth',2)
legend('TEST DATA','MLP-NN','GRBF-NN','COST 231 WALF-IKEG','ADPTD.
COST 231 WALF-IKG','COST 231 HATA');

xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')
grid on
```

APPENDIX B10: Techniques B and C (Maiduguri)

```
% SEMI URBAN TERRAIN (MAIDUGURI)
%   TECHNIQUE B: TRAINING WITH GEOM. MEAN AND TESTING WITH TESTING SET
%   TECHNIQUE C: TRAINING WITH ONE BST AND TESTING WITH ANOTHER

close all
clear all
clc

%GSM NETWORK PARAMETERS
     hb = 40;
     f = 900;
     hm = 1.5;
     EIRP=46;
```

```
%COST 231 HATA CONSTANTS
    ahR=(1.1*log10(f)-0.7)*hm-1.56*log10(f)-0.8;
    C1=0; % For small and medium city

%WALFISCH-IKEGAMI CONSTANTS
    rw = 20;
    B1 = 2 * rw;
    roof = 3;
    floors = 3;
    Hroof = 3 * floors + roof;
    dH = hb - Hroof;

% Lbsh
if hb > Hroof
        Lbsh = -18 * log10(1 + dH);
else
        Lbsh = 0;
end


if hb > Hroof
        Kd = 18;
else
        Kd = 18 - 15 * dH / Hroof;
end

    Kf = -4 + 1.5 * (f / 925 - 1); %for Large City
    Kf = -4 + 0.7 * (f / 925 - 1);

%Lori
    Lsum = 0;
    teta=10;
    jt=0;
while 90 > teta,
if teta >= 0 && teta < 35
            Lteta = -10 + 0.35 * teta;
elseif teta >= 35 && teta < 55
            Lteta = 2.5 + 0.075 * (teta - 35);
else
            Lteta = 4 - 0.114 * (teta - 55);
end
        Lsum = Lsum + Lteta;
        teta=teta+10;
        jt=jt+1;
end

   Lori=Lsum/jt;
   Lrts=-16.9-10*log10(rw)+10*log10(f)+20*log10(Hroof - hm)+Lori;

%   STEP 1: ASSEMBLING TRAINING DATA
```

```
dtrain=(0.1:0.2:2.5);  %  Transmitter-Receiver separations
%   Training Path loss data
pltrain= [90
103
109
119
116
124
124
129
134
138
137
140
156
]';

%   Testing Path loss data
dtest= (0.1:0.2:2.5);
pltest = [92
103
115
121
118
130
131
134
133
136
137
137
140
]';
% FORMATTING TRAINING AND TESTING DATA TO HAVE MEAN=0, AND STD=1

[pn,ps1] = mapstd(dtrain);
[tn,ts] = mapstd(pltrain);
[pn2,ps2] = mapstd(dtest);
[tn2,ts2] = mapstd(pltest);

%rand('seed',62734347);
rand('seed',6273434);

% MLP-NN TRAINING AND TESTING

net = newff(pn,tn,4,{},'trainlm');% STEP 2: CREATING NETWORK OBJECT
% Default Training Algorithm: Levenberg-Marquardt backpropagation
net.trainParam.epochs = 100; % Number of iterations
net.trainParam.goal = 0.00001;
[net] = train(net,pn,tn); %  STEP3:  TRAINING NETWORK
```

```
simulate = sim(net,pn2);  %  STEP4:  SIMULATING NETWORK RESPONSE TO
INPUTS
simulate_rev = mapstd('reverse',simulate,ts2);

dtrain_rev=mapstd('reverse',pn,ps1);
pltrain_rev=mapstd('reverse',tn,ts);
dtest_rev=mapstd('reverse',pn2,ps2);
pltest_rev=mapstd('reverse',tn2,ts2);

T=pltest_rev;
MLP=simulate_rev;
MLP_PE = T-MLP;  % MLP-NN Prediction Error
MLP_MSE = mse(MLP_PE); % MLP-NN Mean Square Error
MLP_RMSE=sqrt(MLP_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-MLP);
MLP_ABSERR=1/N*sum(ABSER);% MLP-NN Absolute Error
ERRSUM=sum((T-MLP).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
MLP_RS=1-(ERRSUM/ERRSUMPL); %MLP-NN R-Squared computation
a1=(sum(ABSER.^2)-N*MLP_ABSERR^2);
MLP_STD=sqrt(a1/(N-1));%MLP-NN Standard Deviation

MLP_OUTPUT=[MLP_ABSERR;MLP_STD;MLP_RMSE;MLP_RS];

% GRBF-NN TRAINING OF BST1 AND COMPARING WITH BST2

eg = 0.001; % Sum-squared error goal(default 0.0)
sc = 0.822;  % Spread constant

[net,tr]= newrb(pn,tn,eg,sc);   %   STEP2:  CREATING NETWORK OBJECT

%Max no of neurons(default=length(P)), No of neurons to
add(default(25)

SIM_RB = sim(net,pn2);    %  STEP3:  SIMULATING NETWORK RESPONSE TO
INPUTS
REV_SIM = mapstd('reverse',SIM_RB,ts2);

RBF=REV_SIM;
RBF_PE = T-RBF;  % RBF-NN Prediction Error
RBF_MSE = mse(RBF_PE); % RBF-NN Mean Square Error
RBF_RMSE=sqrt(RBF_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-RBF);
RBF_ABSERR=1/N*sum(ABSER);% RBF-NN Absolute Error
ERRSUM=sum((T-RBF).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
RBF_RS=1-(ERRSUM/ERRSUMPL); %RBF-NN R-Squared computation
```

```
a1=(sum(ABSER.^2)-N*RBF_ABSERR^2);
RBF_STD=sqrt(a1/(N-1));%RBF-NN Standard Deviation

RBF_OUTPUT=[RBF_ABSERR;RBF_STD;RBF_RMSE;RBF_RS];


% COST 231 HATA PREDICTION

d=dtrain;
A=46.3+33.9*Log10(f)-13.82*Log10(hb)-ahR;
B=(44.9-6.55*log10(hb))*log10(d)+C1;
CHPL=A+B;
%CH=CHPL;
CH2=CHPL.*(0.9436+0.0336.*d);
CH=CH2;
T=pltest';
N=length(pltest);
CH_PE = T-CH';  % COST 231 HATA Prediction Error
CH_MSE = mse(CH_PE); % CH Mean Square Error
CH_RMSE=sqrt(CH_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-CH');
CH_ABSERR=1/N*sum(ABSER);% CH Absolute Error
ERRSUM=sum((T-CH').^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
CH_RS=1-(ERRSUM/ERRSUMPL); %CH R-Squared computation
TEMP=(sum(ABSER.^2)-N*CH_ABSERR^2);
CH_STD=sqrt(TEMP/(N-1));%CH Standard Deviation

CH_OUTPUT=[CH_ABSERR;CH_STD;CH_RMSE;CH_RS];


% QRT ADAPTED COST 231 WALFISCH-IKEGAMI PREDICTION

   d=dtest;
if hb > Hroof
                Ka = 54;
else
if  d>= 0.5
                    Ka = 54 - 0.8 * dH;
else
                    Ka = 54 - 0.8 * dH * d / 0.5;
end
end

            Lmsd = Lbsh + Ka + Kd * Log10(d)+ Kf * Log10(f)- 9 *
Log10(B1);

            Lfs = 32.45 + 20 * Log10(d)+ 20 * Log10(f);
```

```
            PL = Lfs + Lmsd + Lrts;

WI3=PL.*1.1;% Adapted Walfisch-Ikeg.
WI=WI3';
WI2=WI3';
WI_PE = T-WI;  %  Prediction Error
WI_MSE = mse(WI_PE); %  Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI);
WI_ABSERR=1/N*sum(ABSER);% Absolute Error
ERRSUM=sum((T-WI).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));% Standard Deviation

WI2_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

WI2=PL';
WI=WI2;
WI_PE = T-WI;  %  Prediction Error
WI_MSE = mse(WI_PE); %  Mean Square Error
WI_RMSE=sqrt(WI_MSE);%Root mean square error computation
N=length(T);
ABSER=abs(T-WI);
WI_ABSERR=1/N*sum(ABSER);% Absolute Error
ERRSUM=sum((T-WI).^2);%Rsquared computation
ERRSUMPL=sum((T-mean(T)).^2);
WI_RS=1-(ERRSUM/ERRSUMPL); %R-Squared computation
TEMP=(sum(ABSER.^2)-N*WI_ABSERR^2);
WI_STD=sqrt(TEMP/(N-1));% Standard Deviation

WI_OUTPUT=[WI_ABSERR;WI_STD;WI_RMSE;WI_RS];

STATS=[MLP_OUTPUT RBF_OUTPUT WI2_OUTPUT WI_OUTPUT CH_OUTPUT]

plot(dtest,pltest_rev,'*--k',dtrain,MLP,'s-g',dtrain,RBF,'o-
b',dtrain,WI3,'d-r',dtrain,WI,'d-c',dtrain,CH,'-k','LineWidth',2)
legend('TEST DATA','MLP-NN','GRBF-NN','ADPTD.COST 231 WALF-IKEG','COST
231 WALF-IKG','COST 231 HATA');

%plot(v,k,'*--k',v,Y1,'s-g',v,Y2,'o-b',v,Y6,'d-c',v,Y62,'d-r',v,Y5,'--
m',v,Y52,'--r','LineWidth',2);
%legend('TEST DATA','MLP-NN','GRBF-NN','COST 231 WALF-IKEG','OPT. COST
231 WALF-IKEG','COST 231 HATA','OPT. COST 231 HATA');
xlabel('DISTANCE(km)')
ylabel('PATH LOSS(dB)')

grid on
```