UNIVERSAL SIGNAL PROPAGATION PREDICTION MODEL BASED ON PSO-TRAINED MODIFIED ANFIS FOR WIRELESS COMMUNICATION NETWORKS

MALACK OMAE OTERI

DOCTOR OF PHILOSOPHY

(Telecommunication Engineering)

JOMO KENYATTA UNIVERSITY
OF
AGRICULTURE AND TECHNOLOGY

2023

Universal Signal Propagation Prediction Model Based on PSO-Trained Modified ANFIS for Wireless Communication Networks

Malack Omae Oteri

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Telecommunication Engineering of the Jomo Kenyatta University of Agriculture and Technology

2023

DECLARATION

This thesis is my original work and has not been presented for a degree in any other University.

Signature……………………………………… Date ………………………………

Malack Omae Oteri

This thesis has been submitted for examination with our approval as the university supervisors;

Signature……………………………………… Date ………………………………

Prof. Kibet Langat, PhD

JKUAT, Kenya

Signature……………………………………… Date ………………………………

Dr. Peter Kihato, PhD

JKUAT, Kenya

DEDICATION

To my dear wife Everlyne Kerubo, my daughter Michelle Nyambeki and my two sons Philbert Obiero and Gabriel Oteri who gave me humble time and great support as I was undertaking this research.

My mother Florence Nyambeki and in memory of my late father Oteri Mekenye who diligently encouraged and inspired me throughout my learning life.

DEDICATION

First and foremost, I take this opportunity to thank our almighty God for the strength and hope he granted me throughout this research and in my study life. I thank JKUAT for sponsoring this research. I thank my supervisors Prof. Kibet, Dr. Kihato, the late Dr. Ndungu, my examiners and correction supervisor Prof. Nderu for their invaluable advice and support in relation to this work and for believing in me to undertake the research. All the people who contributed to this research in one way or the other, I really appreciate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

ABI	Allied Business Intelligence
AI	Artificial Intelligence
AM	Amplitude Modulation
ANFIS	Adaptive Neuro- Fuzzy Inference Systems
ANN	Artificial Neural Networks
AP	Access Point
AUC	Authentication Center
AWM	Average Wall Model
BS	Base Station
BSC	Base Station Controller
BTS	Base Transceiver Station
CA	Collision Avoidance
CD	Collision Detection
CH	COST231-Hata
CHM	COST231-Hata Metropolitan
CHSU	COST231-Hata Sub-Urban
COST	COperationeuropéennedans le domaine de la recherche Scientifique et Technique

CSMA	Carrier Sense Multiple Access
CW	Continuous Wave
DPM	Dominant Path model
DiSM	Digital Surface Model
DSM	Dual Slope Model
DSSS	Direct Sequence Spread Spectrum
EIR	Equipment Identity Register
EM	Electromagnetic
EVDO	Evolution - Data Optimized
FDTD	Finite Difference Time Domain
FHSS	Frequency Hop Spread Spectrum
FL	Fuzzy Logic
FM	Frequency Modulation
GA	Genetic Algorithm
GD	Gradient Descent
GPUs	Graphic Processing Units
GSM	Global System for Mobile
HF	High Frequency

HLR	Home Location Register
HO	Hata-Okumura
HOR	Hata-Okumura Rural
HOSU	Hata-Okumura Sub-Urban
HOU	Hata-Okumura Urban
HSPA	GPPs High Speed Packet Access
IEEE	Institute of Electrical Electronic Engineers
ITM	Irregular Terrain Model
ITU-R	International Telecommunication Union Radio Communication Sector
LoS	Line-of-Sight
LMS	Least Mean Square
LSE	Least Square Estimator
LTE	Long Term Evolution
MAC	Media Access Control
ME	Mean Error
MF	Medium Frequency
MFs	Membership Functions
MLP-NN	Multilayer Perceptron Neural Networks

MoM	Method of Moment
MS	Mobile Station
MSC	Mobile Switching Center
MSE	Mean Square Error
MWM	Multi Wall model
OSM	One-Slope Model
OSRM	One Slope with Random Input model
PAN	Personal Area Network
PCS	Personal Communication Services
PM	Phase Modulation
PSO	Particle Swarm Optimization
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RBF-NN	Radial Basis Function Neural Networks
RMS	Root Mean Square Error
RO	Ray Optical
RSSI	Received Signal Strength Indication
RT	Ray Tracing

Rx	Receiver
SCADA	Supervisory Control and Data Acquisition
SD	Standard Deviation
SHF	Super High Frequency
SIM	Subscriber Identity Module
SINR	Signal to Interference plus Noise Ratio
SOP	Self Organizing Maps
SPSS	Statistical Package for the Social Sciences
TKS	Takagi and Sugeno's
TLM	Transmission Line Matrix
TRGRM	Two-Ray Ground Reflection model
TV	Television Vision
Tx	Transmitter
USB	Universal Serial Bus
UTD/GTD	Uniform/Geometry Theory of Diffraction
UWB	Ultra Wide Band
UV	Ultra Violet
VCO	voltage controlled oscillator

VLF	Very Low Frequency
VLR	Visitor Location Register
VSATs	Very small aperture terminals
WAN	Wide Area Network
WAP	Wireless Application Protocol
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLANs	Wireless Local Area Networks

NOMENCLATURE

Symbol	**Description**
$\boldsymbol{\mu}$	Prediction error
$\boldsymbol{\sigma}$	Standard deviation
$\boldsymbol{\varphi}$	Multivariate Gaussian function
$\boldsymbol{\alpha}$	Path loss exponent
C	Constant which depends on the scenario parameters such as the carrier frequency
L_{OSM}	OSM Predicted signal loss
L_0	Signal loss at distance *d* from transmitter
n	Power decay factor
d	Distance between antennas
d_0	Reference distance between antennas
h_b	Base station height
h_m	Mobile station height
k_w	Number of penetrated walls
L_{MWM}	OSM Predicted signal loss
L_1	Free space loss at a distance of 1m from transmitter
L_{Walls}	Contribution of walls to total signal loss

L_{Floors}	Contribution of floors to total signal loss
a_{wi}	Transmission loss factor of one wall of i-th kind
k_{wi}	Number of walls of i-th kind
a_{f}	Transmission loss factor of one floor
k_{f}	Number of floors
W_{0j}	Synaptic weights from neuron j in the hidden layer to the single output neuron
X_{i}	i^{th} element of the input vector
F_{h} and F_{0}	Activation function of the neurons from the hidden layer and output layer
W_{ji}	Connection weights between the neurons of the hidden layer and the inputs
F^{L}_{j}	Fuzzy set associated with the input
$\mu_{Ai}(x)$	Membership function of input in the linguistic variable A_{i}

ABSTRACT

Currently, it is evident that the use of wireless communication systems is growing at an unprecedented rate. Because of this fact, there is need for accelerated studies on these systems to improve the quality of service (QoS) provided to their users. This can be done through various methods including signal propagation modelling. Correspondingly, the current trend in signal prediction modelling is shifting from empirical and deterministic models to computational intelligence models due to their low computation cost and high accuracy. In this study we have developed a universal theoretical signal propagation model using modified Adaptive Neuro-Fuzzy Inference Systems (ANFIS) trained with Particle Swarm Optimization (PSO) algorithm. The aim is to develop a model that is more suitable for signal propagation prediction. This is an improvement to the original ANFIS structure for wireless communication propagation modelling. In the process of its development the original ANFIS was modified and together with its training algorithm PSO formulated. This was followed by the development of equivalent theoretical ANFIS based models for existing empirical and deterministic models using ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS. The theoretical models were subsequently combined into one universal model. The root mean square error (RMSE), mean error (ME) and standard deviation (SD) of the predicted signal were used in the process of training and testing the model. From the results, the LOG10D-PSO-ANFIS model has very low values in the range of 10^{-14} to 10^{-16} for the three-performance metrics compared to 10^{-7} for LOG10D-ANFIS and 10^{-1} to 10^{0} for the original ANFIS. Besides the universal model being accurate, it eliminates the need for many input parameters associated with the individual models. This results in just one input, that is, distance being required. It can also be applied in all environments including indoor, outdoor, urban, suburban and rural setups. For the practical modelling of the behavior of the RSSI, a modified ANFIS based practical model (LOG10D-PSO-R-ANFIS) was also developed, and its results compared to those of other models where its performance was found to be superior.

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

According to (Akyildiz et al., 2020), wireless communication systems have experienced substantial revolutionary progress over the past few years, and this will continue being experienced for some time into the future (Borralho et al., 2021). An important consideration in successful implementation of the personal communication services (PCS) is indoor and outdoor radio communication; transmission of voice, data and video to people on the move inside and outside buildings (Nguyen et al., 2018). Wireless communication covers a wide spectrum of situations that include communication with users walking in office or residential buildings, malls or supermarkets, etc. for indoor and urban, rural, suburban and metropolitan for outdoor, to fixed stations sending messages to robots in motion in assembly lines and current together with future factory environments (Kimoto et al., 2016; Allen et al., 2017; Ullah et al., 2020; Elmezughi et al., 2021; Zhang et al., 2017; Erunkulu et al., 2020; Reardon, 2010).

The few researchers who have looked at wireless propagation prediction models (Nguyen et al., 2018; Kimoto et al., 2016; Allen et al., 2017; Ullah et al., 2020; Elmezughi et al., 2021; Zhang et al., 2017; Ahmad et al., 2020; Popoola et al., 2019; Ma et al., 2021; Popescu et al., 2006; Ahmadien et al., 2020) have not used Adaptive Neuro-Fuzzy Inference Systems (ANFIS) trained with Particle Swarm Optimization (PSO) to develop their models instead they have used the traditional empirical models.

Using any kind of indoor or outdoor solutions, needs one to make a good and accurate prediction which will facilitate the process of determining which solution to apply in a particular situation. This can be seen from a variety of research which has been done by various researchers in (Valcarce et al., 2011; Dagefu & Sarabandi, 2010; Phillips et al., 2012; Al-Saman et al., 2021; Schafer et al., 2005; Gentile et al., 2006; Subrt & Pechac, 2010; Corre & Lostanlen, 2009; Austin et al., 2011; Andrade & Hoefel, 2010; El Khaled et al., 2020; Harinda et al., 2019; Rath et al., 2017; Wu et al., 2020; Wahl et al., 2005; Adonias & Carvalho, 2017; Hosseinzadeh et al., 2011). Although there are many possible directions for future work in radiowave prediction modelling area, (Phillips et al., 2012) believe that measurement- based methods and rigorous (comparative) validation are most needed. Applications that make use of these models require an understanding of their real-world accuracy, and researchers need guidance in choosing amongst the many existing proposals. More work needs be done to resolve the disparity between the quantity of models proposed and the extent to which they have been validated in practice.

As the prevalence and importance of wireless networks continues to grow, so too will the need for better methods of modelling and measuring wireless signal propagation. (Phillips et al., 2012) have given a broad overview of approaches to solving this problem proposed in the last 60 years. Most of this work has been dominated by models that extend on the basic electromagnetic (EM) principles of attenuation with theoretical and empirical corrections. More recently, work has focused on developing complex theoretical deterministic models (Al-Saman et al., 2021).

Network architecture for in-building communications is evolving where lately wireless routers are installed within buildings and use of mobile devices outdoors. In modelling propagation, the following parameters must be considered: construction materials (reinforced concrete, brick, metal, glass and others), types of interiors (rooms with or without windows, hallways with or without door and others), locations within a building (ground floor, *n*-th floor, basement and others), the location of transmitter and receiver antennas (on the same floor, on different floors and others), furniture, people, appliances and other considerations for indoor environments. For outdoor environments, considerations such as, large installations, vehicles, buildings, mountains, hills, foliage and many more need to be taken into account (Elmezughi et al., 2021; Zhang et al., 2017).

Path-loss prediction models have traditionally been based on deterministic and/or empirical methods. Empirical models, such as the one slope (OSM), dual-slope (DSM), multi-wall (MWM), average-wall (AWM), COST231, COST231-Hata (CH) and Hata-Okumura (HO) models are computationally efficient but may not be very accurate since they do not explicitly account for specific propagation phenomena and some of them, like Hata-Okumura, have quite a number of parameters to be considered. On the other hand, deterministic models, such as those based on the geometrical theory of diffraction, integral and parabolic equation can, depending on the topographic database resolution and accuracy, be very accurate but lack in computational efficiency (Al-Saman et al., 2021).

The increase in the popularity of wireless networks has led to increased capacity demand. More and more users prefer wireless technology as compared to wired services. The wireless access broadly consists of two main technologies, the wireless cellular networks, which mainly provide voice and data services to users with high mobility and the Wireless Local Area Networks (WLANs), which provide higher data rates to users with comparatively restricted mobility besides other wireless technologies. To replace the wired services, wireless networks need to provide high data rate services like the wired networks. Nowadays, wireless cellular networks have evolved towards providing high data rate services to their users and thus, striving to replace the WLANs as well or the use of these technologies together as hybrid (Reardon, 2010).

With the passage of time, the demand for higher capacity and data rates is increasing. CISCO predicted a 39-fold increase in the data traffic from 2009 to 2014 (Reardon, 2010). A number of technologies and standards have been developed to cope with this increasing demand. The standards like 3 GPPs High Speed Packet Access (HSPA), Long Term Evolution (LTE) and LTE advanced, 3 GPP2s Evolution-Data Optimized (EVDO) and Ultra-Wide Band (UWB) and Worldwide Interoperability for Microwave Access (WiMAX) have been developed to provide high speed communication to end users (Ahmad et al., 2020). To achieve high data rates, signals with high Signal to Interference plus Noise Ratio (SINR) should be received, keeping in mind that transmitter should not cause significant interference to other users by transmitting high power signals. High data rates also require higher order modulation and coding schemes, which are currently used in the above-mentioned standards. However, higher order modulation and coding schemes are more susceptible to noise in a given environment. On the other hand, capacity is generally increased by providing a larger number of channels per area (cell). This is possible by reducing the area of each cell and thus increasing channel reuse. Classical approaches like Cell Splitting and Cell

Sectoring are widely used in current wireless standards to increase system capacity (Popoola et al., 2019). Besides all these improvements, good radio prediction modelling is also required to increase the quality of service (QoS). (Phillips et al., 2012) believe the next generation of models will be data-centric, deriving insight from directed measurements and possibly using hybridized prediction techniques.

An alternative approach to the field strength prediction in these environments is given by prediction models based on Artificial Intelligence (AI) methods such as artificial neural networks. During the last few years, Artificial Neural Networks (ANN) have experienced a great development whose applications are already very numerous (Popescu et al., 2006). These ANNs based models have been shown to successfully perform path-loss predictions in rural, suburban, urban, and indoor environments (Popescu et al., 2006). A drawback with multilayered feed-forward networks that contain numerous neurons in each layer is the required training time (Popescu et al., 2006). The neuro-fuzzy system, used in the research, is a neural network that learns to classify data using fuzzy rules and fuzzy classifications (fuzzy sets). A neuro-fuzzy system has advantages over fuzzy systems and traditional neural networks: A traditional neural network is often described as being like a "black box," in the sense that once it is trained, it is very hard to see why it gives a particular response to a set of inputs.

Fuzzy systems and neuro-fuzzy systems do not have this disadvantage. Once a fuzzy system has been set up, it is very easy to see which rules fired and, thus, why it gave a particular answer to a set of inputs. Similarly, it is possible with a neuro-fuzzy system to see which rules have been developed by the system, and these rules can be examined by experts to ensure that they correctly address the problem (Popoola et al., 2019). The ordinary ANFIS also has a weakness that it does not include a property on which the wireless communication propagation prediction can be used, a feature that has been incorporated in the proposed modified ANFIS to increase its accuracy.

This work research led to the development of a universal theoretical signal propagation model using modified Adaptive Neuro-Fuzzy Inference Systems (ANFIS) trained with Particle Swarm Optimization (PSO) algorithm. This is an improvement to the original ANFIS structure since it can be used in wireless communication propagation modelling. The modified ANFIS together with its training algorithm, that is, PSO have been formulated and used to develop equivalent empirical and deterministic theoretical models using ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS. From these models, a universal model was developed that eliminates the need for many input parameters associated with the individual models whose application is in all environments i.e., indoor, outdoor, urban, suburban and rural setups.

1.2 Problem statement

Currently, the new multimedia services and high data rate applications intensify the need of good quality indoor as well as outdoor radiowave coverage. Hence, providing good quality voice and data services is of great importance. This would also be beneficial for the cellular and data provision operators in the form of increased revenue and reduced churn. Using any kind of indoor or outdoor solutions, needs one to do a good and accurate prediction, which will facilitate the process of determining which solution to apply in which particular situation. Wireless signal propagation modelling is one of the concepts that is used to facilitate the process of placing the wireless communication systems at locations that result in optimum reception of signals. This is done by using computers that are installed with software that incorporates these models. The optimum locations are determined through simulations before the actual installation of these systems is done. This process has number of advantages that include reduction of cost and time. By using accurate modelling, optimum locations are obtained that lead to high quality service in both indoor and outdoor environments. In this study the development of a modified ANFIS and a universal wireless propagation model to address some of the problems currently facing radiowave propagation modelling outlined above.

1.3 Justification

As the prevalence and importance of wireless networks continues to grow, so is the need for better methods of modelling and measuring wireless signal propagation. Most of the work that has been done is dominated by models that extend on the basic electromagnetic principles of attenuation with theoretical and empirical corrections. More recent, work has also focused on developing complex theoretical deterministic models. Most researchers believe the next generation of models need to be data-centric, deriving insight from directed measurements and possibly using hybridized prediction techniques. Since most of the empirical models that have been used before have low accuracy levels, our research was based on ANFIS that is modified, by adding a layer that incorporates a wireless signal propagation behavior feature, to increase the signal prediction accuracy. Its accuracy was enhanced further by using the PSO learning algorithm to train the modified ANFIS.

1.4 Objectives

1.4.1 Main objective

The main objective is to develop a novel universal signal propagation prediction model based on PSO trained modified ANFIS for wireless communication networks with high accuracy and flexibility.

1.4.2 Specific objectives

i. To modify the original ANFIS making it suitable for wireless signal propagation prediction modelling.

ii. To formulate the modified ANFIS architecture and its PSO training in relation to wireless signal propagation prediction modelling.

iii. To develop equivalent theoretical ANFIS based models (ANFIS, LOG10D-ANFIS and LOG10-PSO-ANFIS) for existing empirical models.

iv. To develop a universal theoretical model based on PSO trained LOG10D-ANFIS for RSSI prediction.

v. To evaluate and validate the performance of the developed LOG10D-PSO-R-ANFIS practical model using both simulated and real field data.

1.5 Research contributions

The following is a list of the contributions made as a result of this research;

i. Modification of the original ANFIS making it suitable for wireless signal propagation prediction modelling.

ii. Formulation of the modified ANFIS architecture and its PSO training for wireless signal propagation modelling.

iii. Development of equivalent theoretical ANFIS, LOG10D-ANFIS and LOG10-PSO-ANFIS based models for one slope, dual-slope, multi-wall, average-wall, COST231, COST231-Hata, Hata-Okumura and Two-ray ground reflection models.

iv. Development of a universal theoretical model based on PSO trained modified ANFIS.

v. Development of LOG10D-PSO-R-ANFIS model to predict the measured RSSI.

1.6 Thesis structure

The thesis is organized as follows. Chapter one deals with the introduction which is basically about the need for undertaking this research. It also gives a brief account of wireless communication systems, the objectives of the research as well as the list of the contributions in relation to this research.

Chapter two handles the literature review which explains the fundamental concepts of wireless communication systems which include propagation, effects on propagation like path losses, received signal propagation modelling. It also discusses the different wireless communication systems that are to be modelled.

Chapter three deals with the methodology applied in modelling propagation that includes ANN, ANFIS and PSO trained ANFIS. It outlines the methodologies involved in developing the novel modified ANFIS wireless communication models in this research.

Chapter four discusses the results obtained for the different models developed in the research. These include LOG10DANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS. Their performance is compared with the original ANFIS performance, and the expected better performance is clearly outlined. This is done using tables with RMSE, MAE, STD and R squared and graphical representations of the variation of RSSI with distance as well as error level variation with distance.

Chapter five is about conclusion and recommendations based on the research findings.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

Current signal propagation modelling is a quite new and still a rapidly developing discipline. It has become essential with the installation of Mobile, WLAN and pico-cell mobile systems installation inside and outside buildings as is evident with (Akyildiz et al., 2020; Borralho et al., 2021; Nguyen et al., 2018; Kimoto et al., 2016; Allen et al., 2017; Ullah et al., 2020; Elmezughi et al., 2021; Zhang et al., 2017; Erunkulu et al., 2020; Reardon, 2010; Ahmad et al., 2020; Popoola et al., 2019; Ma et al., 2021; Popescu et al., 2006; Ahmadien et al., 2020; Valcarce et al., 2011; Dagefu & Sarabandi, 2010; Phillips et al., 2012; Al-Saman et al., 2021; Schafer et al., 2005; Gentile et al., 2006; Subrt & Pechac, 2010; Corre & Lostanlen, 2009; Austin et al., 2011; Andrade & Hoefel, 2010; El Khaled et al., 2020; Harinda et al., 2019; Rath et al., 2017; Wu et al., 2020; Wahl et al., 2005; Adonias & Carvalho, 2017; Hosseinzadeh et al., 2011). Many companies spend a great deal of their resources on automating their wireless system design supported by propagation modelling and others. This has led to a variety of different models for application to different environments. These include One Slope Model (OSM), Dual Slope Model (DSM), Multi-Wall Model (MWM), Average-Wall Model (AWM), COST231, CH, HO, ray-optical models, dominant path model, parflow approach, ray-optical-method of moment hybrid model, ray-optical-multi-wall hybrid and Artificial Intelligence based models (Valcarce et al., 2011; Phillips et al., 2012; Andrade & Hoefel, 2010). The following section deals with the concepts that necessitate extensive research on wireless propagation prediction modelling.

2.2 Wireless communication concepts

Communication is the process of transferring information from one location to another. Telecommunication is communicating over long distances [41]. In terms of media for transferring information, there are two types namely guided and unguided (wireless). Guided media is characterized by physical media applied in the transfer of information which include unshielded twisted pair (UTP), shielded twisted pair (STP), coaxial cable and fiber optic cable. Unguided media involves the use of wireless media to transfer information from one point to another leading to the concept of wireless communication (Rappaport et al., 2015). Different media signals include light, infrared, microwaves, radio waves and sound waves which in this case use free space. Under this section a number of wireless communication theoretical concepts are briefly discussed starting with the fundamentals, propagation methods, effects on wave propagation, propagation losses and wireless communication systems (Saakian, 2020). These concepts are important in relation to this thesis research since the solution is based on them in terms of fundamentals and application.

2.2.1 Wireless communication fundamentals

According to [41] radio wave communication is the science that deals with production, processing, transmission, reception and measuring of radio waves. Radio waves are EM waves which are part of the EM spectrum. The EM spectrum includes the following waves gamma rays, x-rays, ultraviolet (UV), visible light, infrared, microwaves and radio waves as indicated in (Rappaport et al., 2015). In some cases, microwaves are also considered to be part of radio waves. The entire processes above are basically done using a communication system. This is a combination of different components put together to perform different components. In the radio communication system, the input will be unprocessed radio waves and output will be processed radio waves. The different processes include modulation, sampling, quantization, demodulation, encoding and decoding, multiplexing and de-multiplexing, compression and filtering.

Radio wave formation and generation is where a radio wave is formed by passing a varying current through a conductive wire. The current passing through it will be because of varying voltage. This will generate an EM wave which has electric and magnetic energy alternating in nature. The electric and magnetic fields resulting are perpendicular in nature forming a plane wave. These waves are based and analyzed using Maxwell’s equations.

Isotropic radiation is considered as a dot which is a source of radio waves releasing in all directions in space. The radiation pattern is basically spherical in nature. The power at a given distance is determined using the following relation.

$$p_d = \frac{P_t}{A} = \frac{P_t}{4\pi r^2} \qquad 2.1$$

Where $P_d$is the power density, P_t is the transmitted Power and r is the distance.

Most of the analysis done for radiowave propagation is based on equation 2.1.

2.2.2 Propagation methods

These methods are used in the process of transmitting radiowaves through the space medium. According to (Garg, 2010; Rappaport et al., 2015; Saakian, 2020; Winder & Carr 2002; Crane, 2003), they include ground wave propagation, surface propagation, space propagation and sky wave propagation. The universal model developed in this thesis is to be applied in the environments that use these propagation methods.

Ground wave propagation is where propagation is done on the surface of the earth more so in the stratosphere section. Very Low Frequency (VLF) to Medium Frequency (MF) are used in this propagation (transmission). It includes the two-wave propagation i.e., surface and space. The surface wave is close to the ground where some of it is reflected and received at the receiver. The signal propagated can sometimes go beyond the optical horizon. This horizon is called a radio horizon. This is possible because of the signal moving in a curved manner.

Space wave propagation is considered in the cases where the antenna heights are large resulting to the space wave dominating over the surface and therefore this propagation mainly occurs in the troposphere. This is because there are few reflections which can affect the space wave. Also, this kind of propagation takes place at higher frequencies.

Sky wave propagation is a kind of propagation that takes place in the ionosphere. Also referred to as ionospheric propagation. The ionosphere has several layers with different levels of ionization. This will result in different permittivity in these layers. The layers include D layer, E layer, F_1 layer and the F_2 layer. The D layer is close to the earth surface while the F_2 layer is the last. Because of these varying properties, when a wave is sent into space it interacts with the different layers getting into space.

2.2.3 Effects on wave propagation

There are several effects that affect a wave as it moves from the transmitter to the receiver. These effects will depend on the medium properties like smoothness, size of obstacles in space, density, climate and geographical location. The effects include reflection, refraction, absorption, scattering, fading, multiple path propagation, noise and depolarization as analyzed in (Saakian, 2020).

Reflection is the process where a signal bounces back to the source due to the presence of obstacles or boundaries. Since this is an EM wave (signal) then it obeys the laws of reflection. Reflection can occur in the different sections of the atmosphere depending on the properties of the different sections. The effect of reflection to a signal includes destructive and constructive effects.

Refraction is where the signal bends when it transits from one medium to another. Commonly occurs with media of difference densities. This occurs when the signal transverses two media with different densities. It is based on Snell's law.

Absorption is where a signal transfers some of its energy to an obstacle as it is being transmitted. It can affect space or ground waves. In space, depending on the size and type of these particles the wave can be absorbed. In the ground it will occur for horizontally polarized waves which get absorbed by the earth's surface. This is because the earth conducts the signal, hence absorbing it. Can also occur or affect vertically polarized waves and in this case the signal moving through different media densities which change from vertical polarization to horizontal. At this point it will also start getting absorbed.

Scattering is considered as where a signal encounters a rough obstacle forcing it to bounce or move in different directions. Scattering can have negative or positive effects. Negative effect can be destructive since the signal will not be able to reach the receiver hence causing destructive effect on the incident signal. Positive effects can be used for long distance communication where clouds of ionized air can be used to scatter the signal to the receivers.

Diffraction occurs when a signal comes into contact with a sharp edge. It will tend to propagate on the shadow part of the object. This can affect the signal strength since it will not be able to reach the destination. This can also be positive in that, for instance mobile communication, a receiver at the shadow part of the object can be able to receive the signal.

Multiple path propagation is a kind of propagation that results from the different effects of the signals. Some of these effects include reflection, refraction, diffraction and scattering. All the above when they affect the signal it moves in different directions. As the signal moves in different direction the rays can meet at the receiving antenna. When signals are propagated this way then this is called multiple path propagation. Rays will take different directions but then meet at the receiving antenna. For instance, a surface wave reflected can meet the direct wave at the receiving antenna.

Noise is any unwanted EM signal. This can occur in transmitting and receiving systems as well as free space.

Polarization is the process of giving a signal an orientation as it moves from the transmitter to the receiver. There are different types of orientation i.e., linear (horizontal and vertical), circular and elliptical. In most cases it is the way the electric field changes with time. This is very important for a receiver to be able to receive its intended radiowaves. If the antenna polarization is not correctly done to match with that of the signal, then it might not be able to receive this signal.

Depolarization is where the signal changes its orientation as it moves from the transmitter to receiver. This is due to the properties of the medium and presence of obstacles along the signal path. This will result to the antenna not being able to receive the signal correctly.

All these effects are very important in wireless communication signal propagation modelling since they either cause interference or affect the signals in several ways, the main one being the variation of the attenuation constant.

2.2.4 Propagation losses in communication links

These are losses experienced by a signal propagating in space. As explained by (Garg, 2010) there are several combinations to these losses where one of the main losses is free space loss. They can be classified as path loss, diffraction losses, scattering losses and shadow losses.

Path losses are losses due to the signal propagation along a given path in space. These include free spaces losses, atmospheric losses, precipitation and losses due to obstacles.

Free space losses are obtained by using the propagation expression derived as given in the analysis and equation 2.8 below.

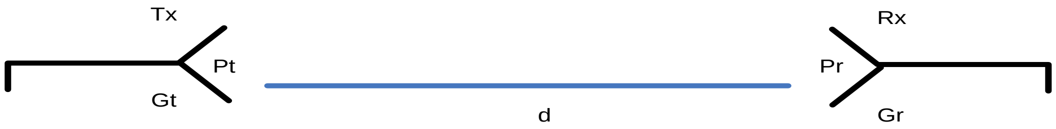


Figure 2.1: Transmitter and Receiver

Where $T_x$is the transmitter, $P_t$is the transmitted power, $G_t$is the Transmitter gain, $G_R$is the Receiver gain, $P_r$is the Received, power, $A_{eff}$is the Effective area or 'aperture' of the antenna, and $log$is the Logarithm to base 10.

The power density at any point from the transmitter is given by equation 2.1

Introducing the transmitter gain.

$$P_{dG} = \frac{P_t G_t}{4\pi d^2} \qquad 2.2$$

at the receiver the received power is given by;

$$P_R = \frac{P_t G_t A_{eff}}{4\pi d^2} \qquad 2.3$$

$$A_{eff} = \frac{G_R \lambda^2}{4\pi d^2 \times 4\pi} = \frac{P_t G_t G_R \lambda^2}{(4\pi d)^2}, \qquad 2.4$$

$$P_R = \frac{P_t G_t G_R}{\left(\frac{4\pi d}{\lambda}\right)^2} \qquad 2.5$$

$$L = \left(\frac{4\pi d}{\lambda}\right)^2 \qquad 2.6$$

$$L = \left(\frac{4\pi f d}{3 \times 10^8}\right)^2$$

$$L = \left(\frac{4\pi f d \times 10^6 \times 10^3}{3 \times 10^8}\right)^2 \; in\, dB,$$

$$L_{dB} = 10 \log_{10} L \qquad 2.7$$

$$10\, log\left(\frac{4\pi \times 10^6 \times 10^3 fd}{3 \times 10^8}\right)^2 = 20 log \frac{4\pi \times 10^6 \times 10^3}{3 \times 10^8} + 20\, log + 20 \log_{10} d$$

$$L_{dB} = 32.44 + 20\log_{10} f + 20\log_{10} d \qquad 2.8$$

$$P_{RdB} = P_{tdB} + G_{tdB} + G_{rdB} - L_{dB} \qquad 2.9$$

All the propagation models developed by different researchers (Ullah et al., 2020; Elmezughi et al., 2021; Zhang et al., 2017; Erunkulu et al., 2020; Reardon, 2010) and (Valcarce et al., 2011; Dagefu & Sarabandi, 2010; Phillips et al., 2012; Al-Saman et al., 2021; Schafer et al., 2005; Gentile et al., 2006; Subrt & Pechac, 2010; Corre & Lostanlen, 2009; Austin et al., 2011; Andrade & Hoefel, 2010; El Khaled et al., 2020; Harinda et al., 2019; Rath et al., 2017; Wu et al., 2020; Wahl et al., 2005; Adonias & Carvalho, 2017; Hosseinzadeh et al., 2011) take equation 2.9 as the foundation to their model equations. When the other losses are included, the path loss exponent increases from the default 2 to a higher value depending on the environment under consideration.

Other losses include multipath fading, attenuation, diffraction, scattering, atmospheric and cosmic losses (Garg, 2010). Diffraction will occur due to the signal coming across sharp edges of objects. Scattering is due to the obstacles of different sizes. When the signal strikes these obstacles, it will be reflected in different directions resulting in losses. Atmospheric losses also include obstacles in space like dust, fog, precipitation, storms etc. Cosmic losses are as a result of galactic noise (radiation from stars and other galactic bodies). They emit radio frequency interference which can affect the signal and can also be used to study the emitting bodies. Other losses can result from antenna misalignment. Improper impedance matching and problems with polarization (polarization mismatch). The other kinds of losses which are very important to study are due to fading.

Fading is where a signal loses strength or undergoes variations as it moves through the channel (medium). Mainly caused by multipath signal transmission and shadowing due to obstacles like mountains. There are two types of fading slow where the signal variations over time and is due to large obstacles like mountains. Fast fading is where there is considerable variation of signal strength over a given period of time. This is due to small obstacles, of which some are not permanent. For instance, foliage, small buildings and particles in the path of propagation.

Fading models are theoretical formula which can be used to predict the signal levels at a given point considering the fading characteristics as those of a given environment. There are three types as discussed in (Garg, 2010) i.e., Rician distribution, Rayleigh distribution and Lognormal distribution. These are responsible for the continuous variation of the received signal strength at a given location.

2.2.5 Wireless communication systems

This section lists some of the wireless communication systems that use the concept of wireless propagation modelling in their design and implementation. There are several types of these wireless communication networks and systems that include the Satellite systems (Roddy, 2006), the Radar, WLANs (Labiod 2007) like Bluetooth, Wi-Fi, WiMAX and Zig-bee, Radio, Television Vision (TV), Pagers and Cellular networks (Mfuko & Omido, 2020; Omae et al., 2022). All these systems have a direct link to wireless propagation modelling discussed in section 2.3.

2.3 Wireless communication propagation modelling

2.3.1 Introduction

A model is a representation of a system which can either be physical or mathematical as indicated in (Ptolemaeus, 2014). Modelling is the process of coming up with a model. Simulation is the process of determining/predicting the behavior of a system by using a model which can be done by varying different input parameters to predict the output of a given system.

2.3.2 Path loss modelling

Modelling in wireless communication as indicated in (Garg, 2010) is the process of developing a given mathematical representation that can be used in predicting or obtaining given results. Path loss models are mathematical expressions used in the prediction of losses depending on a given environment. Models are important since they make planning easier-facilitates the process of determining where to locate BTSs in cellular networks for optimal performance in terms of coverage and quality of signal. This reduces the cost of planning and implementation of communication systems. When engineers need to determine where to locate BTS without modelling they are required to do field measurements which can be very costly to the company. They also reduce the planning time/duration. The use of models makes the process of planning to be faster than without using models. This is done using computers which can be able to predict the propagation characteristics of a given environment. Modification is possible when using models instead of using manual methods. This is because in software environment one can place the BSs at different locations to determine the propagation characteristics. The use of software is easier to record and store information which can help in future if the environments under consideration are similar. Models also have a few limitations that include difficulty in modelling complex environments, for instance indoor environments which have a complex structure. Some environments require the use of more than one kind of model. Initial and maintenance cost of some models can be high.

2.3.3 Types of path loss models

Models can be classified into four main categories i.e., deterministic, empirical, hybrid and Artificial Intelligence based models as indicated in (Andrade & Hoefel, 2010). Deterministic are models which with a set of inputs parameters one can determine what is expected at the output. They are site specific where they take a particular set of environmental characteristics and structure of the obstacles. They are more accurate depending on the obstacles being considered. They consume a lot of time and resources, for instance memory.

Empirical are obtained by performing experiments from which data is obtained then used in their development. They are not site specific in which case they can be used in several similar environments. They are not very accurate when compared with deterministic but are faster and require fewer resources. Hybrid of deterministic and Empirical make use of the advantages of the two models. They combine the accuracy of deterministic and the simplicity of the empirical. These include the ray optical and Method of Moment (MoM) which combines the features of Ray optical models of the accuracy and the MoM analysis to increase the accuracy of the prediction. One can also combine the ray optical and Finite Difference Time Domain (FDTD) methods and the ray optical and multi-wall model (MWM) that combines the properties of accuracy of ray optical and simplicity of MWM.

2.3.4 Deterministic models

These include ray optical with ray tracing and ray launching, finite difference, method of moments and dominant path method. Ray optical use the basics of optics to perform the analysis of radio wave signals. The basics include reflection, diffraction, refraction and scattering among others. There are two methods, ray tracing and ray launching. Ray tracing is where a ray is propagated and followed from the transmitter to the receiver considering the obstacles along its path. Ray launching where a ray is launched at different angles and the effects on it by different obstacles analyzed. Depending on the obstacles the ray can be reflected or diffracted. For uniform boundary it will be reflected and for sharp edged objects the rays will be diffracted. It is a point to multipoint concept where a single transmitter and several receivers are considered.

The Dominant path method is based on the ray tracing method where a ray is followed and then it keeps on passing through a given path several times than the other paths. If the ray keeps on passing through given rooms from the transmitter to a receiver, then this will be taken as the dominant path. This is used in analyzing the effect of the rays as they pass through this path depending on the obstacles it encounters. For instance, a corridor can result in this effect.

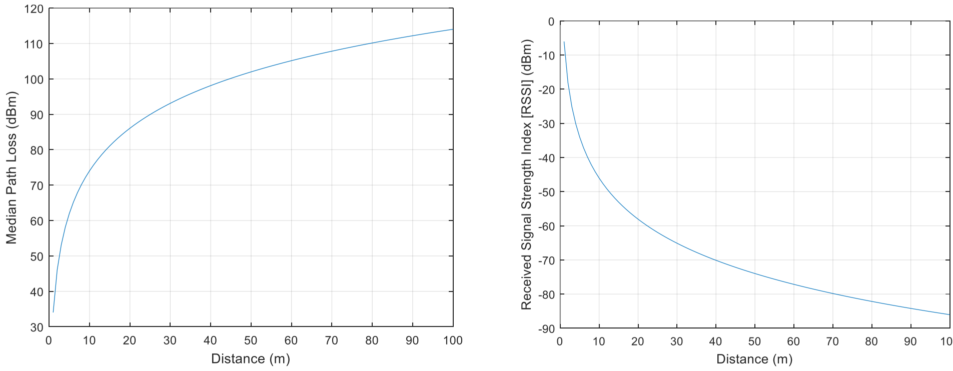
The finite difference are methods that solve Maxwell’s equation in discretized space time grid. They include finite difference time domain which is the most commonly used method in analyzing radiowaves. It is a mature method since it has been used in the design and analysis of antennas such as patch antennas. It solves Maxwell’s equations using small levels of space combined with time to form a discretized space time grid.

The methods of moments is another commonly used method together with FDTD in antenna design and analysis. It is also mature and is based on solving Maxwell’s equations in a discretized wire grid. All these methods have a high level of accuracy but require time and memory resources. They are used in software form (simulations).

According to (Alotaibi et al., 2009) two-ray ground reflection model uses the 4th power law together with mobile antenna and base station antenna heights. Its expression is given by;

$$\mathrm{L} = \mathrm{L_o}(\mathrm{d_0}) + 10\alpha\log_{10}\frac{\mathrm{d}}{\mathrm{d_o}} - 20\log_{10}\mathrm{h_m} - 20\log_{10}\mathrm{h_b} \qquad 2.14$$

Where $\mathrm{L_o}(\mathrm{d_0})$ is the loss at a distance of 1 meter, α is the 4-path loss exponent, d is the distance between Tx and Rx, h_m is the mobile station antenna height and h_b is the base station antenna height.



Loss **RSSI**

Figure 2.2: Two-ray ground reflection Model loss and RSSI versus distance graph

Figure 2.2 represent the plot for the variation of loss and RSSI with respect to distance for the two-ray ground reflection Model based on equation 2.14 assuming Wi-Fi (downloads.linksys.com) parameters that include transmitted power (P_t) of 100mW, transmitter and receiver antenna gains G_t as well as G_r of 4dB (https://fccid.io).

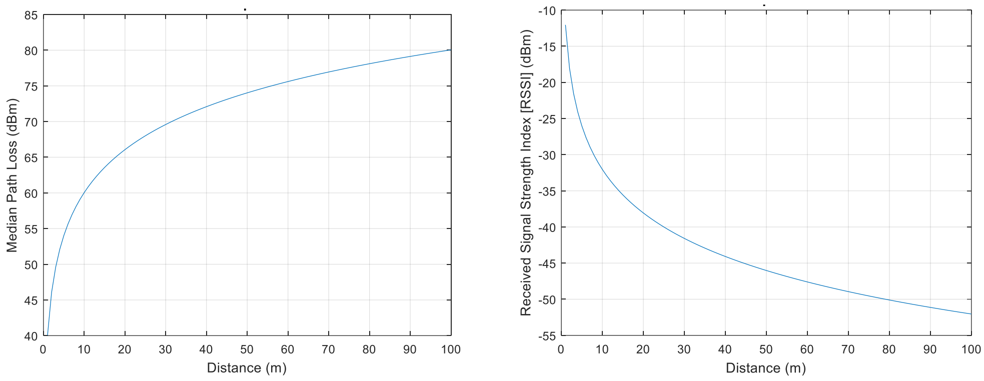
2.3.5 Empirical models

Empirical models describe the signal level loss by formulas with experimental parameters optimized by measurement campaigns in various buildings to make the parameters of the model as universal as possible. From the experiments done in field measurements a mathematical model is developed. These methods are characterized by a mathematical model in a formula form. They are based on the free space path loss model. The free space path loss model is modified to develop these models. These models include OSM, DSM, MWM, AWM, COST231, CH and HO models.

One slope model referred to as COST231 OSM is the simplest approach to signal loss prediction, because it is based only on the distance between the transmitter and the receiver. This simplest prediction model does not take into account the position of obstacles, the influence of which is affected only by the power decay factor (α). Factor α and the signal loss at a distance d_0 from the transmitter increase for an environment with more losses, but they are constant for the whole building (Ahmadien et al., 2020; Valcarce et al., 2011; Dagefu & Sarabandi, 2010). Figure 2.3 represents the plot for the variation of loss and RSSI with distance for the OSM based on equation 2.15 with 2.4 GHz frequency and path loss exponent of 2 together with the other Wi-Fi parameters.

$$\mathrm{L} = \mathrm{L_o}(\mathrm{d_0}) + 10\alpha\log_{10}\frac{\mathrm{d}}{\mathrm{d_0}} \tag{2.15}$$

Where L is the predicted signal loss (dB), $L_0(d_0)$ is the signal loss at distance 1m from transmitter (dB), α is the path loss exponent (-), d is the distance between antennas (m) and d_0 is the reference distance between the antennas (usually 1 m) (m).



Loss **RSSI**

Figure 2.3: One Slope Model loss and RSSI versus distance

The *dual slope model* divides the path into two sections i.e., the LOS section and the obstructed LOS section. The loss is given by;

For $1 < \mathrm{d} < \mathrm{d_{bp}}$

$$L = 10n_1 logd \tag{2.16}$$

For d > d_{bd}

$$L = 10n_1 logd + 10n_2 log\frac{d}{d_{bp}} \quad 2.17$$

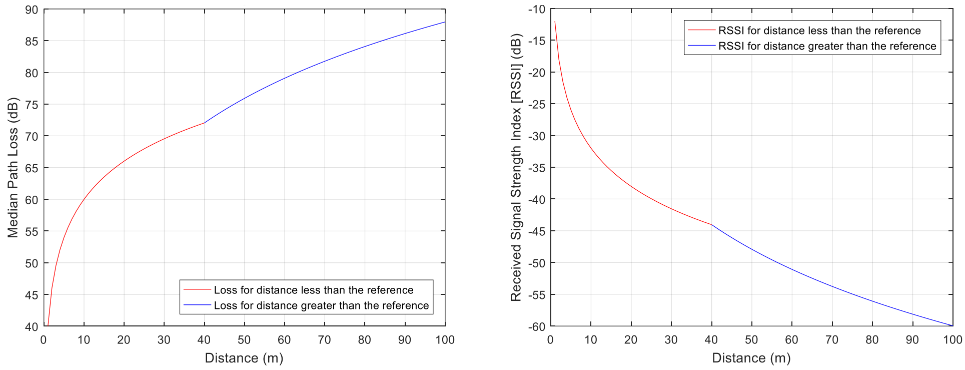
Where d is the distance between Tx and Rx and n_1 and n_2 is the path loss exponents.

The breakpoint distance d_{bp} considers indoor environments, the ellipsoidal Fresnel zone can be obstructed by the ceiling or the walls, anticipating the LOS region:

d_{bp} = Breaking point distance given by

$$d_{bp} = \frac{4h_b h_m}{\lambda} \quad 2.18$$

Where λ is wavelength h_b and h_m denote the shortest distance from the ground or wall of the access point (AP) and station (STA), respectively (Andrade & Hoefel, 2010). Can be used for both indoor and outdoor environments. Figure 2.4 represents the plot for the variation of loss and RSSI with distance for the DSM using a frequency of 2.4 GHz and 40m breakpoint distance.



Loss **RSSI**

Figure 2.4: Dual Slope Model loss and RSSI versus distance

Multiwall model considers several walls. The OSM is insufficiently accurate for most applications, due to the usually inhomogeneous structure of buildings with long waveguiding corridors or large open spaces on one side and small complex rooms with many obstacles on the other side. For such cases, the more accurate, but still partly empirical, Multi Wall Model (MWM) employing a site-specific building structure description can be used (Andrade & Hoefel, 2010). The Multi-Wall model considers wall and floor penetration loss factors in addition to the free space loss as represented in equation 2.19. The transmission loss factors of the walls or floors passed by the straight-line joining the two antennas are cumulated into the total penetration loss L_{Walls} in equation 2.20 or L_{floors} in equation 2.21, respectively. Depending on the model, either homogenous wall or floor transmission loss factors or individual transmission loss factors can be used. The more detailed the description of the walls and floors, the better the prediction accuracy. The penetration losses are optimized as other empirical parameters from measurements, so they are not equal to the real obstacle transmission losses, but only correspond to the appropriate empirical attenuation factors of the obstacles.

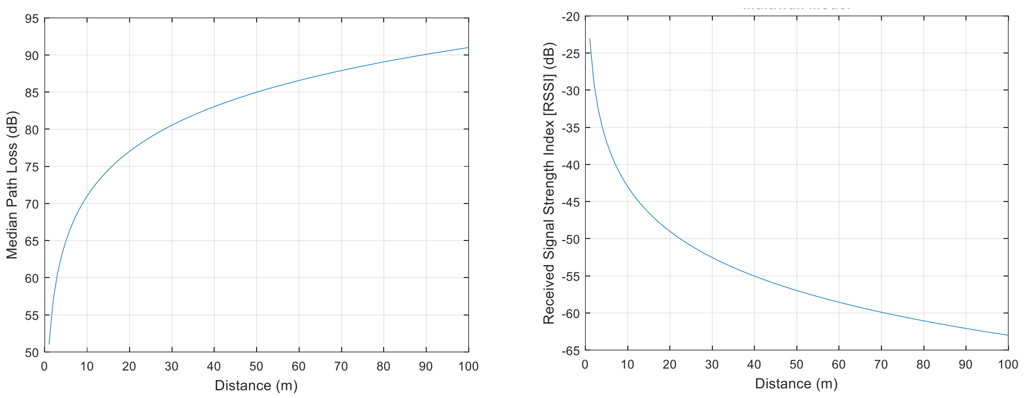
$$\mathrm{L_{MWM} = L_1 + 20log_{10}(d) + L_{Walls} + L_{Floors}} \qquad 2.19$$

$$\mathrm{L_{Walls}} = \textstyle\sum_{\mathrm{i=1}}^{\mathrm{l}} \mathrm{a_{w_i} k_{w_i}} \qquad 2.20$$

$$\mathrm{L_{floors} = a_f k_f} \qquad 2.21$$

Where L_{MWM} is the predicted signal loss (dB), L_1 is the free space loss at a distance of 1m from transmitter (dB), L_{Walls} is the contribution of walls to total signal loss (dB), L_{Floors} is the contribution of floors to total signal loss (dB), a_{wi} is the transmission loss factor of one wall of i-th kind (dB), k_{wi} is the number of walls of i-th kind (-), a_f is the transmission loss factor of one floor (dB) and k_f is the number of floors (-).

Since the MWM considers the positions and specific transmission loss factor of walls, its results are more accurate than those of OSM. However, the shadowing effect of more closely adjacent walls are often overestimated, because their cumulated transmission loss factors lead to very small values of predicted signal level behind these elements. In other words, the real signal may not follow a straight-line between antennas, but it can go round the walls. The computation time of the MWM is also quite short, and the sensitivity of the model to the accuracy of the description of the building is limited due to the simple consideration of only the number of obstacles passed by a straight line (Rath et al., 2017). Figure 2.5 represents the plot for the variation of loss and RSSI with distance for the MWM.



Loss **RSSI**

Figure 2.5: MW Model loss and RSSI versus distance

Average Walls Model is based on the Cost-231 multi-wall except that the loss due to obstructing walls is aggregated in just one parameter L. Therefore, for a single floor environment, the path loss estimated by;

$$L_{dB} = 20\log_{10}d + k_{walls}L_{walls} \qquad 2.22$$

Where k_{walls} denotes the number of penetrated walls. To determine the parameter L_{walls}, each wall obstructing the direct path between the receiver and the transmitter antennas must have its loss measured as follows:

The loss of the first wall in dB is given by:

$$L_1 = L_{t1} - L_{0dB} - 20\log_{10}d \qquad 2.23$$

Where L_{0dB} is the path loss obtained at 1 meter distant from the transmitter; L denotes the measured total loss from 1 meter distant after the obstructing wall. For the second wall the loss of the first wall also must be taken into account. Therefore, the loss in dB of the second obstructing wall can be estimated as

$$L_2 = L_{t2} - L_{0,dB} - 20\log_{10}d - L_1 \qquad 2.24$$

Keeping on the above methodology, the i^{th} wall loss is given by

$$L_i = L_{ti} - L_{0,dB} - 20\log_{10}d - \sum_{j=1}^{i} L_j \qquad 2.25$$

where the sum spans the losses of walls obtained previously. After all wall losses of the environment had been obtained, then the wall losses average value is computed and assigned to the parameter L_{walls} (Andrade & Hoefel, 2010).

Okumura-Hata model is based on Okumura's analysis of path-loss characteristics based on a large amount of experimental data collected around Tokyo, Japan. He selected propagation path conditions and obtained the average path-loss curves under flat urban areas. Then he applied several correction factors for other propagation conditions, such as:

- Antenna height and carrier frequency
- Suburban, quasi-open space, open space, or hilly terrain areas
- Diffraction loss due to mountains
- Sea or lake areas
- Road shape

Hata derived empirical formulas for the median path loss to fit Okumura curves as represented in Figure 2.6. Hata's equations are classified into three models as given in equations 2.26, 2.27 and 2.28 for urban, sub-urban and rural environments respectively.

Urban Area

$$\mathrm{L_{50} = 69.55 + 26.16log_{10}f_c + (44.9 - 6.55log_{10}h_b)log_{10}d - 13.82log_{10}h_b - a(h_m)\ dB} \qquad 2.26$$

Where $\mathrm{a(h_m)}$ is the Correction factor (dB) for mobile antenna height, L_{50} is the medium path loss, h_b is the base station antenna height, Fc is the carrier frequency, h_m is the MS antenna height and d is the distance between antennas.

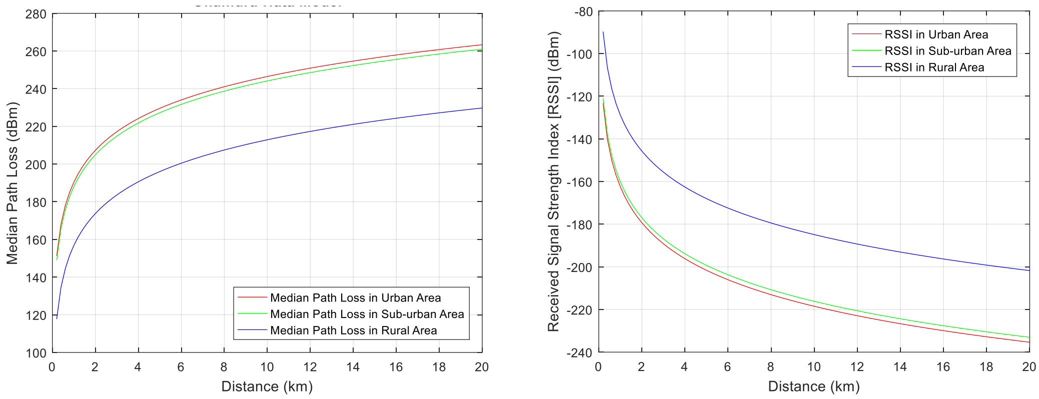
Sub-urban areas

$$\mathrm{L_{50\,sub\,urban} = L_{50\,urban} - 2\left[log_{10}\left(\frac{f_c}{28}\right)^2 - 5.4\right]} \qquad 2.27$$

Rural

$$\mathrm{L_{50\,rural} = L_{50\,urban} - 4.78(log_{10}\,f_c)^2 + 18.33log_{10}f_c - 40.94dB} \qquad 2.28$$

Okumura's model is suitable for 150MHZ to 1920 MHZ frequencies range and a transmitter receiver separation of 1 km to 100 km. It can also find applications for base station antenna height in range of 30 m to 1000 m and a mobile antenna height less than 10 m. Our universal model solves this problem by taking care of all environments as well as frequency and distance range.



Loss **RSSI**

Figure 2.6: Hata-Okumura Model loss and RSSI versus distance

COST-231 Model is a combination of empirical and deterministic models for estimating the path loss in an urban area over the frequency range of 800MHz to 2000MHz. The model is used primarily in Europe for the GSM 1800 system (Martinez, 2009). It has the following working parameters and mathematical representations and figure 2.7.

$$L_0 = 4 - 0.114(\text{phi} - 55) \qquad 2.29$$

$$L_f = 32.4 + 20\log_{10} (d) + 20\log_{10}10(f_c) \qquad 2.30$$

$$L_{rts} = -16.9 - 10\log_{10}10(W) + 10\log_{10}10(f_c) + 20\log_{10} (d_{hm}) + L_0 \qquad 2.31$$

$$L_{bsh} = -18\log_{10}10(11+d_{hb}) \qquad 2.32$$

$$k_d = 18 - 15d_{hb}/d_{hm} \qquad 2.33$$

$$k_a = 54 - 0.8h_b \qquad 2.34$$

$$k_f = 4 + 0.7((f_c/925)-1) \qquad 2.35$$

$$L_{ms} = L_{bsh} + k_a + k_d\log_{10}10(d) + k_f\log_{10}10(f_c) - 9\log_{10}10(b) \qquad 2.36$$

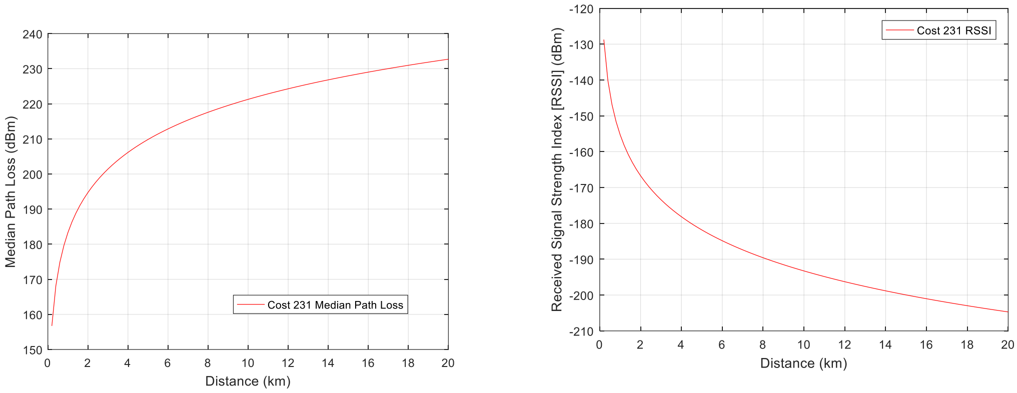
$$d_{hm} = h_r - h_m \qquad 2.37$$

$$d_{hb} = h_b - h_r \qquad 2.38$$

$$L_{50} = L_f + L_{rts} + L_{ms} dB \qquad 2.39$$

Where f_c = carrier frequency, W = street width (m), b = distance between building along radio path (m), d= separation between transmitter and receiver (km), h_r = average building height (m), h_b = base station antenna height, h_m = MS antenna height, phi = incident angle relative to the street, L_f = Free space path loss = $32.44 + 20\log_{10}f_c + 20\log_{10}d$, L_{rts} = roof top to street diffraction and scattering losses and L_{ms} = Multiscreen losses.

Equation 2.39 is the final equation for this model.



Loss **RSSI**

Figure 2.7: COST231 Model loss and RSSI versus distance

COST231-HATA Model is an improvement of the Hata model to extend operation to a frequency of 2000MHz according to (Alotaibi et al., 2009). The model is used primarily in Europe for the GSM 1800 system (Martinez, 2009). Its equations as well as graphical representations are given in equations 2.40 to 2.44 and figure 2.8.

$$ahm1 = (hm((1.1(\log_{10}10(f_c))) - 0.7)) - ((1.56(\log_{10}10(f_c))) - 0.8) \qquad 2.40$$

Metropolitan area

$$L_{cm} = 46.3 + (33.9(\log_{10}10(f_c))) - (h_b13.82) - ahm1 + ((44.9-6.55(\log_{10}10(h_b))))(\log_{10}10(d)))+3 \qquad 2.41$$

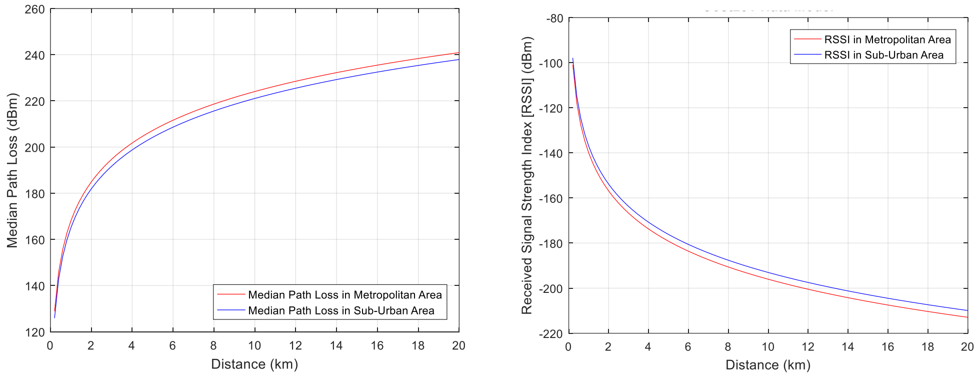
$$rL_{cm} = P_t + G_t + G_r - L_{cm}+30 \qquad 2.42$$

Sub-urban area

$$L_{csu} = 46.3 + (33.9(\log_{10}(f_c))) - (h_b 13.82) - ahm1 + ((44.9-(6.55(\log_{10}(h_b))))(\log_{10}(d))) \quad 2.43$$

$$rL_{csu} = P_t + G_t + G_r - L_{csu}+30 \quad 2.44$$

COST-Hata-Model is usually restricted to large and small macro-cells applications where the base station antenna heights are just above rooftop levels adjacent to the base station. According to (Alotaibi et al., 2009), this model's formula and its modification must not be used for micro-cells. A limitation solved by our developed universal model.



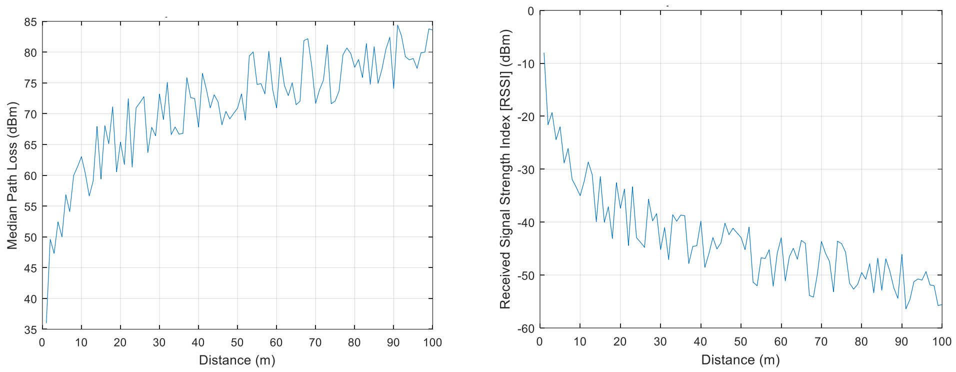
Loss **RSSI**

Figure 2.8: COST231-Hata Model loss and RSSI versus distance

One slope with random input model was developed since the one slope model is not enough to describe the varying characteristics of a practical environment. To take care of this variation a random variable X_σ with a standard deviation σ depending on the environment and distance. Its choice is often determined from measurements. According to (Alotaibi et al., 2009) this value can range between 6 and 10dB. The resulting expression is represented using the equation 2.45 and plots in figure 2.9;

$$L = L_o(d_0) + 10\alpha\log_{10}\frac{d}{d_0} + X_\sigma \quad 2.45$$

Where $L_o(d_0)$ =Loss at a distance of 1 meter, α = path loss exponent which varies according to the environment, d = distance between Tx and Rx and X_σ=random variable.



Loss RSSI

Figure 2.9: One Slope with random input Model loss and RSSI versus distance

2.4 Artificial intelligence modelling

2.4.1 Artificial neural networks

According to (Sumathi & Paneerselvam 2010), these are techniques used to perform different tasks and make decisions. They are in the form of processing elements connected together in parallel and series combinations. For human beings, they are basically cells connected loosely as nodes. They facilitate the process of transmitting signals from one location to the other. They involve the use of mathematical techniques that perform different tasks and make different decisions. This is a branch of Artificial Intelligence.

2.4.2 Biological inspiration

Artificial neural networks are developed based on the concept of operation of the human brain which has several neurons connected in series and parallel. The human brain is the section of the body which does most of the processing and manipulation in the human being. The Brain has several cells of different categories and sizes due to the different functionalities. One of the most important cells is the neuron. The brain has up to 100 billion neurons connected together to perform different functions. The brain has two sections: the right side and the left side. The left side of the brain deals with calculation and rules which make use of the if then rules. Calculations are usually arithmetic and logic operations. It also deals with concepts. Neurons on the left side of the brain are connected in series i.e., sequential. It solves problems and performs different functions using the rule-based concepts and therefore behaves like expert systems. Expert systems borrow their concept of operation from the left side of the brain. Expert systems have a database to store reference information. It also contains an inference section, which looks up for the information in the database. The right side of the brain also deals with images, pictures, functions and controls. Neurons on this side are usually connected in parallel. This is what inspired the invention of neural networks. The form of controls in this case includes feedback concepts which is applied in control systems (Gurney & Kevin, 2014).

2.4.3 Natural neuron

This is the basic building block of the brain. It is basically a cell that can receive signals and transmit them to other cells. It operates in a way that it uses threshold i.e., if several signals are received from other neurons they add up or build up hence causing the neuron to fire up if the signal reaches a given threshold. But if the signal build up is not enough to fire, then the neuron remains dormant. A neuron is a very simple element when considered on its own; but many when they are put together, they form very complex system. It consists of a cell body (soma), dendrites, axon, axon hillock, synapses as represented in figure 2.10.

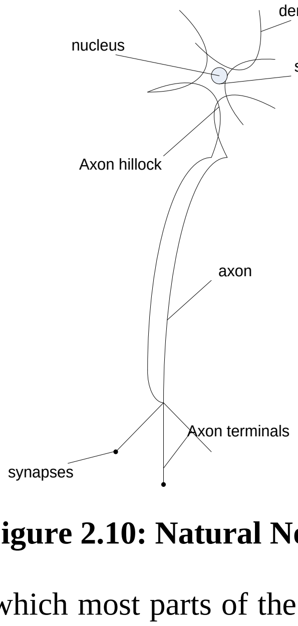


Figure 2.10: Natural Neuron

Soma is the cell body from which most parts of the neuron branch out. It consists of the nucleus which has most of the protein. The dendrites and axon extend out of the cell body.

Dendrites are hair like extensions from the cell body. Their main function is to receive input from other cells. The inputs are basically neuro-signals i.e., electrochemical signals resulting from (ions) which include, Na^+,K^+, Ca^+ ions. They connect to the other neurons.

Axon is a cable like structure extending from the cell body. Its main function is to pass signals from the cell body to the other neurons. They can also pass the signals back to the cell body from other cells. This can be considered as feedback. Towards the axon end, are several branches which enable the neuron to get into contact with several other neurons and are referred to as Axon Terminals. At the end of the Axon terminals are Synapses which will also facilitate the process of connecting to other neurons.

Axon Hillock is where the axon meets the Soma. It contains very high levels of ions like Na and Potassium and thus considered as the most sensitive section of the neuron.

The operation of the biological neuron is such that it receives inputs from other neurons through the dendrites. The inputs are processed by the neuron body and passed to the output through the axon. The axon passes the processed signals to the other neurons through synapses. When the input signals are accumulated, they are compared with a given threshold value. If the accumulated signals are beyond the threshold value, then the neuron fires i.e., passes the input to the output. But if the accumulated signal is less than the threshold value then the neuron remains dormant (does not fire) (Peretto, 1992).

2.4.4 Artificial neuron

This is the mathematical representation (model) of the biological neuron. It borrows some percentage of its structure and operations from the biological neuron. The first model was proposed by Mclluoch and Pitt’s in the 1940’s (Gurney & Kevin, 2014). The structure was as given in figure 2.11;

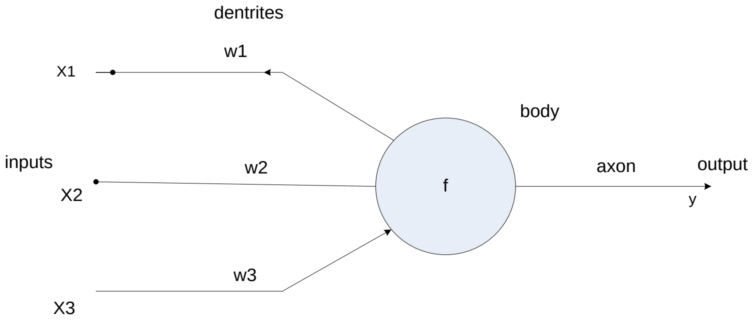


Figure 2.11: Mclluoch’s Artificial Neuron model

This was a very simple model which was operated on the principle of Excitatory and Inhibitory. Its inputs and outputs were in terms of 0 and 1 and represented by $\propto$. The inputs passed through the weights represented by w the weights are multiplied by the inputs before they are summed up. For instance, $\propto$ is multiplied by w_1 and the same for the other weights and inputs which results to

$$x_1w_1 + x_2w_2 + x_3w_3 + \cdots \qquad 2.46$$

The weights are such that they are either excitatory or inhibitory. In the excitatory mode 1 was used and in inhibitory mode -1 was used. After the inputs were multiplied by their respective weights and summed up the result was compared with a given threshold value T. If the result was greater than the threshold value, then output. But if the result was less than the threshold value then the output was 0 and 1. This model was related to the biological neuron in a way that the lines passing the inputs to the node imitate the operation of the dendrites. The nodes act as the body where the signals are accumulated by the lines connecting to the output acted as the axon.

$$y = x_1w_1 + x_2w_2 + x_3w_3 + \cdots \ldots \ldots + x_nw_n \qquad 2.47$$

A single input model has a single input *p* passed through a weight *w* as represented in figure 2.12.

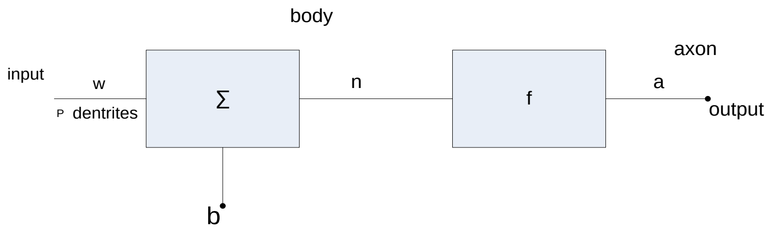


Figure 2.12: Single input MATLAB based model

The input *p* is multiplied with the weight *w* to obtain *wp*. The product will then be added to the bias *b*. Therefore, the results of the multiplication will be scalars. The bias *b* is usually considered as 1. After the summation the result will be;

$$n=wp+b \tag{2.48}$$

which is a scalar as aforementioned.

The *n* result will be passed through a function *f* where the output *a* is obtained by multiplying the *f* with *n* which results to

$$a=fn \tag{2.49}$$

$$a=f(wp+b) \tag{2.50}$$

In some functions the input *p* and the weight *w* are used to give the distance separating them. One such function is the Radial Basis function where also the summation is substituted with the multiplication. When compared with biological neurons, the weight is considered as the dendrite. The summation and the function *f* are considered as the body. The line connecting to the output is considered as the Axon. The function section uses different types of transfer functions. These functions are referred to as activation functions (Peretto, 1992).

2.4.5 Activation functions

These are functions that are used to transfer the input to a particular output according to the problem being solved. They include linear functions, sigmoid functions and hard limit functions among others.

Linear Functions is a function that transfers inputs to the outputs. This is done by multiplying the input with a scalar. Most used in the Input Layers of neural networks.

Sigmoid Function is a function that has a curved shape and transforms inputs to outputs in a way that it changes any value from $-\infty$ to $+\infty$ to between 0 and 1 in curved from.

Hard Limit Functions are functions that transform the inputs in a way that all the negative values are considered as 0 while all the positive values are considered as 1. This is equivalent to a Step Function in signal processing.

The neuron above can be divided into 3 sections in terms of input functions, net function and transfer functions.

Input functions consists of inputs section with input *p* and the weight *w* which are multiplied to form *wp.*

Net function is the resulting function after adding the bias to the product of the input *p* and weight *w* as given in equation 2.48.

Transfer Function has different functions as aforementioned. More functions can be obtained from the artificial neuron network MATLAB toolbox. The output is given in equation 2.50.

The weights and the bias are adaptive parameters. These functions can be changed. The changes depend on the outputs. When the target output and the input are considered, the difference dictates the level of change of the weights and biases. This is done to reduce the difference (error) to its minimal and optimal value. The process of changing the weights and biases to minimize the error or improve the performance of a system is called tuning parameter/training the neural network/learning of the neural.

2.4.6 Types of network architectures

According to (Peretto, 1992), there are three main categories which include single layer feedforward, multilayer feedforward and recurrent (feedback network).

Single Layer Feed Forward is a network with a single layer of neurons. It has several inputs, *p* which goes into a summation node together with the bias (*b*). At the node the inputs multiplied with the weights are added to give an immediate output as given in equation 2.48. The output *n* is passed through a function *f* which depending on the type and application will result to equation 2.49. The main feature of this network is that it does not have feedback. There is no signal propagating from the output to the input but only from input to the output.

Multilayer Feed Forward is characterized by several layers connected together in a series format.

The outputs of the 1st layer are connected inputs to the 2nd layer where outputs become the inputs to the 3rd layer and so on. In most cases the 1st two layers are considered as hidden layers while the 3rd layer is considered as the output layer. Similarly, there is feedback where signals are propagated from the input to the output only.

Recurrent Networks are networks with feedback where the output signal can also propagate backwards i.e., from the output to the input. It is referred to as self-feedback networks. The output is feedback to its input where it is summed up together with other inputs. Another type is where we have several neurons forming a layer where the outputs are feedback to the inputs of all the neurons.

Multilayer Perceptron Neural Network (MLP-NN)

Figure 2.13 shows the configuration of a multilayer perceptron with one hidden layer and one output layer as discussed in (Popescu et al., 2006) and (Wu et al., 2020). The network shown here is fully interconnected.

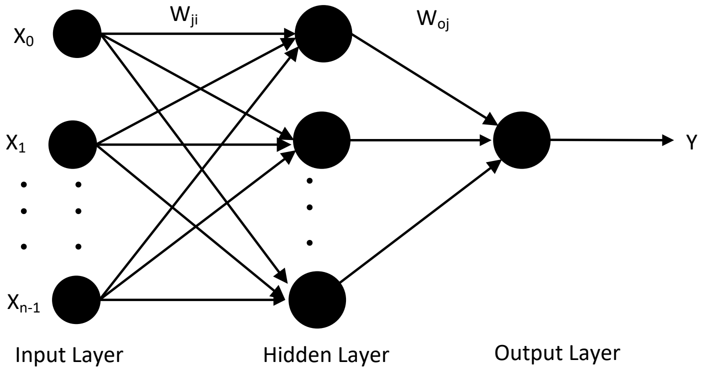


Figure 2.13: Configuration of the MLP-NN

This means that each neuron of a layer is connected to each neuron of the next layer so that only forward transmission through the network is possible, from the input layer to the output layer through the hidden layers. Two kinds of signals are identified in this network:

- The function signals (also called input signals) that come in at the input of the network, propagate forward (neuron by neuron) through the network and reach the output end of the network as output signals;
- The error signals that originate at the output neuron of the network and propagate backward (layer by layer) through the network. The output of the neural network is described by the following equation:

$$y = F_o\left(\sum_{j=0}^{M} W_{oj}\left(F_h\left(\sum_{i=0}^{n} W_{ji}X_i\right)\right)\right) \qquad 2.51$$

Where W_{oj} represents the synaptic weights from neuron j in the hidden layer to the single output neuron, X_i represents the i^{th} element of the input vector, F_h and F_o are the activation function of the neurons from the hidden layer and output layer, respectively and W_{ji} are the connection weights between the neurons of the hidden layer and the inputs.

The learning phase of the network proceeds by adaptively adjusting the free parameters of the system based on the mean squared error E, represented by equation 2.52, between predicted and measured path loss for a set of appropriately selected training examples:

$$E = \frac{1}{2}\sum_{i=1}^{m}(y_i - d_i)^2 \qquad 2.52$$

Where y_i is the output value calculated by the network and d_i represents the expected output.

When the error between network output and the desired output is minimized, the learning process is terminated and the network can be used in a testing phase with test vectors. At this stage, the neural network is described by the optimal weight configuration, which means that theoretically ensures the output error minimization.

Generalized Radial Basis Function Neural Network (RBF-NN)

According to (Popescu et al., 2006), the Generalized Radial Basis Function Neural Network (RBF-NN) is a neural network architecture that can solve any function approximation problem.

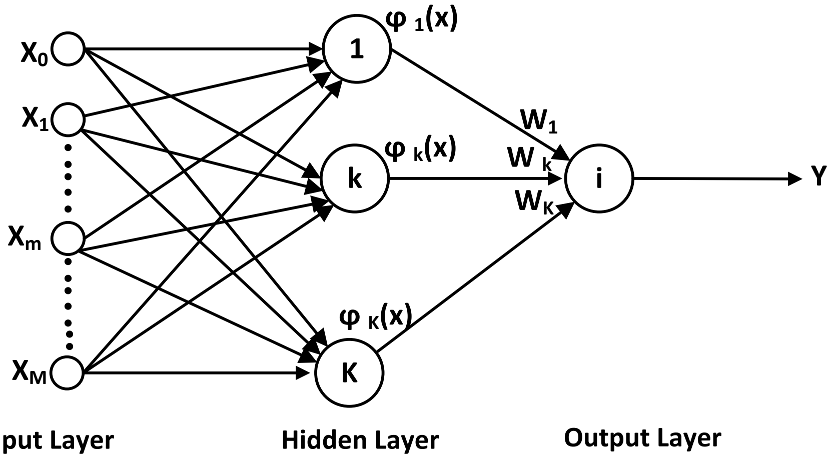


Figure 2.14: RBF-NN architecture

The learning process is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data, with the criterion for the “best fit” being measured in some statistical sense. The generalization is equivalent to the use of this multidimensional surface to interpolate the test data. As it can be seen from Figure 2.14, the Generalized Radial Basis Function Neural Network (RBF–NN) consists of three layers of nodes with entirely different roles:

- The input layer, where the inputs are applied,
- The hidden layer, where a nonlinear transformation is applied on the data from the input space to the hidden space; in most applications the hidden space is of high dimensionality.
- The linear output layer, where the outputs are produced.

The most popular choice for the function φ is multivariate Gaussian function with an appropriate mean and auto covariance matrix.

The outputs of the hidden layer units are of the form

$$\varphi_k[X] = exp[-(X - V_k^x)^T(X - V_k^x)/(2\sigma^2)] \qquad 2.53$$

Where V_k^x are the corresponding clusters for the inputs and V_k^y are the corresponding clusters for the outputs obtained by applying a clustering technique of the input/output data that produces K cluster centres. The parameter σ controls the "width" of the radial basic function and is commonly referred to as the spread parameter.

$V^y{}_k$ is defined as

$$V_k^y = \sum_{y(p)\in\, cluster\ k} y(p) \qquad 2.54$$

The outputs of the hidden layer nodes are multiplied with appropriate interconnection weights to produce the output of the GRNN. The weight for the hidden node k (that is w_k) is equal to

$$W_k = V_k^y / \sum_{k=1}^{k} N_k exp\left[-\frac{d(x, V_k^x)^2}{2\sigma^2}\right] \qquad 2.55$$

$N_k$is the number of input data in the cluster centre k, and

$$d(X, V_k^x) = (X - V_k^x)^T (X - V_k^x) \qquad 2.56$$

With

$$V_k^x = \sum_{x(p)\in\, cluster\ k} x(p) \qquad 2.57$$

When the ANN based models are compared to other popular large-scale prediction models, they demonstrate very good performance for all types of environments such as outdoor as well as indoor giving greater accuracy. The ANN models are not computationally extensive as compared to deterministic models. Some of the limitations that come with ANN modelling include data overfitting due to ANN architectures with complex layers, falling into local minima that results to low performance, as well as increased training time that keeps them from being used in real-time applications. Most of these limitations are solved using ANFIS that we have used in our research.

2.4.7 Training an artificial neural network

As discussed in (Peretto, 1992) training is the process of changing the weights and biases so that the network gives the desired output. This is done in a way obtained that the desired outputs and the actual output are compared by subtraction and an error obtained. These errors should be minimized/reduced as much as possible. This process of changing the weights and biases is iterative (goes through several steps before it can stop). It can also be called the learning process of the network. The accuracy of the output will depend on the training algorithm used. There are two main types of learning/training learning with a teacher and learning without a teacher.

Learning with a Teacher is where a system is given an input/output combination from which it can learn. This is in the form of an example from which the system can pick certain characteristics features. The actual output of the system and the desired or target output are compared and an error obtained this error is propagated back to the input layers where the weights are modified accordingly to reduce this error. This goes through several iterations called epochs. When the error reduces to minimum possible value the system stops the training process.

Learning without a Teacher is where the system learns without using examples or without a target output. It usually uses the inputs where the system picks patterns in the input vector. There are 2 types given by reinforcement learning and unsupervised learning where in reinforcement learning, a target output is not provided for the system to use. In this case, the system will take the input as it passes through it and the gives an output which can be a desired output or not. This can be determined by the critic where if the output is as required then there is a reward but if not, there is a penalty. The search for the best output is done using heuristic methods. The heuristic method is where there is a search space for a solution in which case the system goes through a known route to the solution. Other forms of search methods are blind search i.e., also referred to as Self Organizing Maps (SOM) as discussed by (Kihato, 2013). In unsupervised Learning there is no teacher or target. The system also takes the input and through its own judgment can give a particular output. This process also goes through several iterations to give an output which can be considered correct.

2.4.8 Learning rules

As discussed in (Gurney & Kevin, 2014), there are several rules that can be used in the process of training a network. These are rules based on the earliest form of learning form of learning methods. They include Hebb's rule, Hopfield rule, Delta rule, Kohenen rule, Outstar rule, Memory based rule, Genetic Algorithm (GA) and Particle Swam Optimization (PSO) among many others.

Hebb's Rule can be used in supervised/unsupervised learning. It is based on changing the weights according to the state of the neurons.

Hopfield Rule is like Hebb's rule where the only addition is that the weight is modified by a given magnitude. The modification is based on the learning rate. If both neurons are active, then increase the weight by the learning rate. If one of the neurons is negative, then reduce the weight by the learning rate.

Delta Rule also uses Hebb's concept but is based on the error at output. The target output is compared with desired output where an error is obtained. The weights are then modified to reduce this error. The error is propagated back to the input layers to facilitate the process of reducing it. This method is also called the Least Mean Square (LMS) or Gradient Descent Method (GD).

Kohonen Rule uses the concept of competition to enable the learning process. Neurons compete to learn. The neurons with the best weight output will be allowed to modify its weights. It can also allow a few of its neighbors to modify their weights. The number of neighbors will depend on the pattern to be solved.

Perceptron Learning learns using supervised learning. It takes the target output *t* and the actual output *a*, compares the two to obtain an error. This error is then used to modify the weights and biases.

Outstar Rule uses the relationship between the inputs and outputs to modify the weights. It uses the principle of repetitiveness to learn where an input output combination is passed through the network or neuron. The neuron/network can pick a characteristic trait from the combination. In the future the network is able to use the statistical parameters obtained in classifying other input output combinations. If the system is applied with an input, then it can be able to give a specific output according to the trait picked.

Memory Based is where a system has a memory with which it keeps previously correctly classified example. By using the information stored in memory, it can be able to classify an input correctly.

2.5 Fuzzy inference systems and ANFIS

2.5.1 Fuzzy inference systems

This research used fuzzy inference systems more specifically the ANFIS discussed in the section that follows. As its name suggests, it is a system that uses fuzzy logic to perform different functions. Based on (www.mathworks.com), it deals with different fuzzy concepts which include set theory, if-then rules and reasoning. This system can efficiently perform function approximation, which we have used it for in this research, besides other functions like classification. Its basic structure, given in figure 2.15 below, consists of three components given as: a rule base, a knowledge base and a reasoning mechanism. The rule base contains fuzzy rules, the knowledge base that sets the membership functions used in the fuzzy rules and the reasoning mechanism that executes the inference process on the rules to give a reasonable output (Jang, 1993).

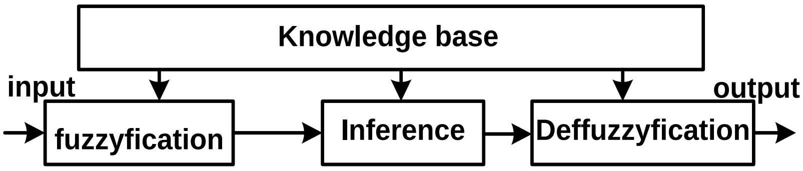


Figure 2.15: Fuzzy Inference System structure

The different types of this system designed for function approximation that include Tsukamoto, Mamdani's and Takagi Sugeno where in our study the Takagi Sugeno (Ghomsheh et al., 2007; Gharghan et al., 2018) system was used due to its advantages that include high computational efficiency, good compatibility with linear, optimization and adaptive techniques and its suitability with mathematical analysis.

Taking an input vector X =(x_1, x_2,...,x_p)^T the system output Y can be given by the Sugeno inference system as;

R^L: If (x_1 is $F^L{}_1$, and x_2 is $F^L{}_2$,..., and x_p is $F^L{}_p$),

Then (Y = Y^L= $c^L{}_0x_0+c^L{}_1x_1+...+c^L{}_px_p$).

Where, $F^L{}_j$ represent the fuzzy set associated with the input x_j in the L_{th} rule and Y^L is output as a result of rule R^L (L=1,...,m.). The parameters that are used to define the membership functions for $F^L{}_j$ are called the premise parameters, and $c^L{}_i$ are called the consequent parameters. In the case of a real-valued input vector $X=(x_1,x_2,...,x_p)^T$, the overall output of the Sugeno fuzzy inference systems is given as a weighted average of the Y^L as indicated below.

$$Y = \frac{\sum_{L=1}^{m} w^L Y^L}{\sum_{L=1}^{m} w^L} \qquad 2.59$$

where the weight w^L is the truth value of the proposition $Y = Y^L$ and is defined as

$$w^L = \prod_{i=1}^{p} \mu\, F_i^L(x_j) \qquad 2.60$$

and where $\mu F^L{}_i(x_i)$ is a membership function defined on the fuzzy set $F^L{}_j$.

2.5.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Adaptive Neuro-Fuzzy Inference System (ANFIS) was first proposed by (Jang, 1993). It is a combination of Fuzzy Logic (FL) and Artificial Neural Network (ANN) which captures the strengths and reduces the limitations of both techniques for building Inference Systems (IS) with better results and intelligence. In this case, fuzzy logic deals with fuzzy set theory that relates to classes of objects with boundaries whose membership is a matter of degree. It can also be seen as a platform that computes with words instead of numbers which is closer to human intuition and makes use of tolerance for imprecision, thus lowering the solution cost (Sumathi & Paneerselvam 2010). As indicated earlier in the previous chapter, Artificial Neural Networks consist of an interconnection of simple processing elements that operate simultaneously in parallel modelling the biological nervous system. Neural Networks are considered to be able to learn from input data by modifying the values of the connections referred to as weights between the elements. These two Artificial Intelligence concepts merged offer the neural networks learning capability and the fuzzy logic knowledge representation that makes inferences from observations resulting to a very powerful tool.

2.5.3 Basic ANFIS architecture

The ANFIS architecture described here is based on type 3 fuzzy inference system (other popular types are the type 1 and type 2). In the type 3 inference system, the Takagi and Sugeno's (TKS) if-then rules are used. The output of each rule is obtained by adding a constant term to the linear combination of the input variables. Final output is then computed by taking the weighted average of each rule's output. This type of architecture, with two inputs (x and y) and one output, z, is shown in figure 2.16 (Ghomsheh et al., 2007). Where A_1 and B_1 are linguistic variables, p, q and r are the ANFIS consequent parameters.

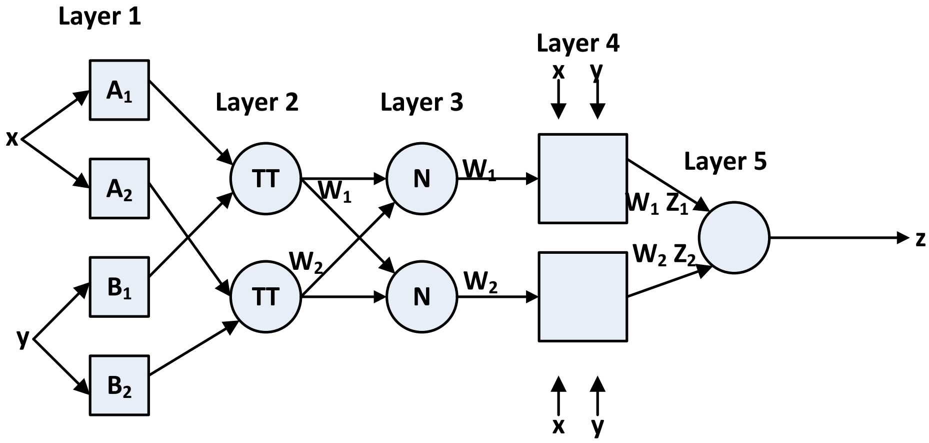


Figure 2.16: Type 3 ANFIS Architecture

Basic ANFIS rules

The basic ANFIS structure is based on the following rules.

$$Rule\ 1: If\ x\ is\ A_1\ and\ y\ is\ B_1, then\ z_1 = p_1x + q_1y + r_1$$

$$Rule\ 2: If\ x\ is\ A_2\ and\ y\ is\ B_2, then\ z_2 = p_2x + q_2y + r_2$$

The ANFIS structure is the functional equivalent of a supervised, feed-forward neural network with one input layer, three hidden layers and one output layer, whose functionality are as described below:

Layer 1 (Fuzzy Layer): Every node in this layer is an adaptive layer that generates the membership grades of the input vectors. Usually, a bell-shaped (Gaussian) function with maximum equal to 1 and minimum equal to 0 is used for implementing the node function:

$$O_i^1 = f(x, a, b, c) = \mu_{Ai}(x) = \frac{1}{1+|(x-c_i)/a_i|^{2b_i}}$$

2.61

$$\mu_{\,Ai}(x) = \exp\{-\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}\}$$

2.62

Where O_i^1 is the output of the i^{th}node in the first layer, $\mu_{Ai}(x)$ is the membership function of the input in the linguistic variable A_i. The parameter set $\{a_i, b_i, c_i\}$ are responsible for defining the shapes of the membership functions. These parameters are called premise parameters.

Layer 2 (Product Layer): Each node in this layer determines the firing strength of a rule by multiplying the membership functions associated with the rules. The nodes in this layer are fixed in nature with the firing strength of a particular rule, the output of a node, given by:

$$w_i = O_i^2 = \mu_{Ai}(x).\mu_{Bi}(y), i = 1, 2 \qquad 2.63$$

Any other T-norm operator that performs fuzzy AND operation can be used in this layer to combine the inputs.

Layer 3 (Normalized Layer): This layer consists of fixed nodes that are used to compute the ratio of the ith rule's firing strength to the total of all firing strengths. Its output is given by:

$$\overline{w} = O_i^3 = \frac{w_i}{w_1+w_2}\ , i = 1, 2, \qquad 2.64$$

The outputs of this layer are known as normalized firing strength for convenience purposes.

Layer 4 (Defuzzify Layer): This is an adaptive layer with node function given by:

$$\overline{w_\imath} z_i = O_i^4 = \overline{w_\imath}(p_i x + q_i y + r_i) \qquad 2.65$$

It essentially computes the contribution of each rule to the overall output. It is referred to as a defuzzification layer and provides output values resulting from the inference of rules. The parameters in this layer given as $\{p_i,\ q_i,\ r_i\}$ are known as consequent parameters appearing at the output section of the network.

Layer 5 (Total Output Layer): There is only one fixed node in this layer that basically computes the overall output as the summation of contribution from each rule whose function is given as:

$$\sum_i \overline{w_i} z_i = O_i^5 = \sum_i \frac{w_i z_i}{\sum_i z_i} \qquad 2.66$$

The ordinary ANFIS does not have a feature that makes it suitable for wireless signal prediction modelling a feature that has been added to our modified ANFIS whose performance increases due to the introduced logarithmic function.

2.6 ANFIS learning algorithms

The development of the ANFIS concept has seen the proposal of several learning methods to facilitate the process of obtaining optimal set of rules. They include a merger between Min-Max and ANFIS proposed by Mascioli et al and nonlinear least square by Lavenberg-Marquardt (Ghomsheh et al., 2007).

Four methods used to update the ANFIS structure parameters introduced by (Jang, 1993) are as given below according to their level of computation complexities:

1. Gradient descent (GD) only- used to update all the parameters.
2. Gradient descent only and one pass of least square estimator (LSE)- the gradient descent takes over to update all parameters after the LSE is first applied only once at the beginning to obtain the initial values of the consequent parameters.
3. Gradient descent only and LSE- this is a hybrid learning.
4. Sequential LSE-updates all the parameters using extended Kalman filter.
5. Genetic algorithm (GA).
6. Particle swarm optimization (PSO).

2.6.1 ANFIS hybrid learning

As indicated in (Gurney & Kevin, 2014), ANFIS networks, as any other neural network, might be trained by backpropagation of the resulting error and adjustment of the adaptive parameters according to this propagation in order to minimize it.

Nevertheless, ANFIS presents some form of linearity with respect to some of its parameters, due to its structure, allowing for the application of the much efficient Least Squares Method. The use of the combination of both LeastSquares and Steepest Descent methods is referred to as Hybrid Learning.

As commented, the adaptive parameters of the ANFIS network divides into premise parameters in layer 1, and consequent parameters in layer 4. The contribution of these last set to the network output is linear:

$$\mathrm{Y} = \sum_{i=1}^{p} \dot{\mathrm{w}}_i (\textstyle\sum_{k=1}^{n} q_{ik} x_{k+r_i}) \qquad 2.67$$

therefore, their computation using Least-Square Method, or its recursive version is advised, for the exact theta (θ) is obtained with no need of several iterations and requires less computational effort and time on what a process named forward pass. On the contrary, premise parameters will be computed using the Steepest Descent Method, by backpropagating the error through the network, during the backward pass. Both training steps constitute the Hybrid Learning methodology and are presented below.

Forward pass characterized by the Recursive Least-Square Method being used to evaluate the consequent parameters. The θ vector will be defined as follows:

$$\theta = \{q_{11}\ \ q_{12}\ \ q_{13}\ \ r_1\ \ q_{21}\ \ q_{22}\ \ \ q_{23}\ \ r_2\ \ q_{31}\ \ q_{32}\ \ q_{33}\}^T \qquad 2.68$$

and y will be the desired output vector (not to be confused with the network results, after Layer 5). Considering that the required number of training pairs to have a definite system will be n(p+1), where n refers to the number of inputs and p to the number of rules, the y vector would be as follows:

$$\theta = \left\{y_1, y_2, \dots\ y_{n(p+1)}\right\}^T \qquad 2.69$$

Meanwhile, matrix A can be defined after Equations 3.7 and 3.8

$$A =$$

$$\begin{bmatrix} \overline{w_1}\, x_1^{(1)} & \overline{w_1}\, x_2^{(1)} & \overline{w_1}\, x_3^{(1)} & \overline{w_1} & \ldots\ldots\ldots..\overline{w_p}\, x_1^{(1)} & \overline{w_p}\, x_2^{(1)} & \overline{w_p}\, x_3^{(1)} & \overline{w_p} \\ \overline{w_1}\, x_1^{(2)} & \overline{w_1}\, x_2^{(2)} & \overline{w_1}\, x_3^{(2)} & \overline{w_1} & \ldots\ldots\ldots..\overline{w_p}\, x_1^{(2)} & \overline{w_p}\, x_2^{(2)} & \overline{w_p}\, x_3^{(2)} & \overline{w_p} \\ & \vdots & \vdots & \vdots \quad \ddots & \vdots & \vdots & \vdots & \\ \overline{w_1}\, x_2^{(m)} & \overline{w_1}\, x_3^{(m)} & \overline{w_1}\, x_3^{(m)} & \overline{w_1} & \ldots\ldots\ldots..\overline{w_p}\, x_1^{(m)} & \overline{w_p}\, x_2^{(m)} & \overline{w_p}\, x_3^{(m)} & \overline{w_p} \end{bmatrix}$$

2.70

where the superscript between parenthesis denotes the training pair number and m has been defined as m=n(p+1) to simplify the notation. Within the recursive part of the method, the vector A^T corresponding to the training pair k+1 is defined as any row of the matrix A:

$$a^T =$$

$$\left\{\overline{w_1}\, x_2^{(k+1)}\, \overline{w_1}\, x_3^{(k+1)} \quad \overline{w_1}\, x_3^{(k+1)} \quad \overline{w_1} \quad \ldots \overline{w_p}\, x_1^{(k+1)} \quad \overline{w_p}\, x_2^{(k+1)} \quad \overline{w_p}\, x_3^{(k+1)} \quad \overline{w_p}\right\}$$

2.71

Backward pass involves the error signal propagating backwards through the network until the dependence of this error to each of the premise parameters is evaluated. Once known this gradient, the parameters may be updated by Steepest Descent:

$$a_{ij}(k+1) = a_{ij}(k) - k_j\, \frac{\nabla E}{\|\nabla E\|} \qquad 2.72$$

$$\frac{\partial E}{\partial a_{ij}} = \sum_{k=1}^{p} \frac{\partial E}{\partial y}\, \frac{\partial y}{\partial \bar{w}_k}\, \frac{\partial \bar{w}_k}{\partial w_i}\, \frac{\partial w_i}{\partial \mu_{ij}}\, \frac{\partial \mu_{ij}}{\partial a_{ij}} = \frac{\partial w_i}{\partial \mu_{ij}}\, \frac{\partial \mu_{ij}}{\partial a_{ij}} \sum_{k=1}^{p} \frac{\partial E}{\partial y}\, \frac{\partial y}{\partial \bar{w}_k}\, \frac{\partial \bar{w}_k}{\partial w_i} \qquad 2.73$$

Chain rule is used to evaluate the partial derivatives:

Where here ai,k stands for any premise parameter ai,k, bi,k or ci,k. The partial derivatives are derived as:

$$\frac{\partial\, E}{\partial y} = \frac{\partial}{\partial y}[(y_d - y)^2] = y - y_d \equiv e \qquad 2.74$$

$$\frac{\partial\, y}{\partial \bar{w}_k} = \frac{\partial}{\partial \bar{w}_k}[(\bar{w}_k \sum_{i=1}^{n} q_{ki} x_i + r_k)] = f_k \equiv e_{3.k} \qquad 2.75$$

$$\frac{\partial\,\bar{w}_k}{\partial \bar{w}_i} = f(x) = \begin{cases} \frac{\partial}{\partial \bar{w}_k}\left(\frac{w_k}{\sum_{i=1}^{p} w_i}\right) = \frac{\bar{w}_k}{w_k}(1-\bar{w}_k)\; i = k \\ \frac{\partial}{\partial \bar{w}_k}\left(\frac{w_k}{\sum_{i=1}^{p} w_i}\right) = -\frac{\bar{w}_k^2}{w_k} \quad i \neq k \end{cases} \qquad 2.76$$

$$\frac{\partial\,\bar{w}_i}{\partial \bar{w}_{\mu ij}} = \frac{\prod_{l=1}^{n} \mu ij}{\mu_{ij}} = \frac{w_i}{\mu_{ij}} \qquad 2.77$$

The $\frac{\partial\,\mu_{ij}}{\partial a_{ij}}$ derivative depends on the membership function used, and will be different for each premise parameter $a_{i,k}$, $b_{i,k}$ or $c_{i,k}$. To simplify the notation, the following function is introduced:

$$\varphi_{ij}(x) = \frac{x_j - c_{ij}}{a_{ij}} \quad \text{i=1… p and j=1…n} \qquad 2.78$$

It is also noticeable, that the absolute value within expression 3.4 may be rewritten in order to ease differentiation:

$$\mu_{ij}(x) = \frac{1}{1+\left|\frac{x_j-c_{ij}}{a_{ij}}\right|^{2b_{ij}=}} = \frac{1}{1+\left[\left(\frac{x-c_{ij}}{a_{ij}}\right)^2\right]^{2b_{ij}=}} = \frac{1}{1+\left(\varphi_{ij}^2\right)^{b_{ij}}} \qquad 2.79$$

The following partial derivatives result, for premise parameters a_{ik}:

$$\frac{\partial\,\mu_{ij}}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}} \frac{1}{1+\varphi_{ij}^{2b_{ij}}} = \frac{-1}{\left(1+\varphi_{ij}^{2b_{ij}}\right)^2} 2b_{ij}\varphi_{ij}^{2b_{ij}-1} \frac{c_{ij}-x_j}{a_{ij}^2} = \frac{2b_{ij}}{a_{ij}} \varphi_{ij}^{2b_{ij}} \mu_{ij}^2 \qquad 2.80$$

For b_{ij}:

$$\frac{\partial\,\mu_i}{\partial b_i} = \frac{\partial}{\partial b_i} \frac{1}{1+\varphi_i^{2b_i}} = \frac{-\varphi_i^{2b_i}\, ln\varphi_i^2}{\left(1+\varphi_i^{2b_i}\right)^2} = -\varphi_i^{2bi}\, ln\varphi_i^2\, \mu_i^2 \qquad 2.81$$

And for c_{ij}

$$\frac{\partial\,\mu_i}{\partial c_i} = \frac{\partial}{\partial c_i} \frac{1}{1+\varphi_i^{2b_i}} = \frac{1}{\left(1+\varphi_i^{2b_i}\right)^2} \frac{2b_i\varphi_i^{2b_i-1}}{a_i} = \frac{2b_i}{x-c_i} \varphi_i^{2b_i} \mu_i^2 \qquad 2.82$$

2.6.2 Genetic Algorithm (GA)

As indicated in (Hassan et al., 2005) and (Kihato, 2013), GA uses the concept of natural existence and survival. The strongest will be able to survive. It uses the concept to optimize solutions to given problems. The algorithm has the following steps.

1. Initialize the chromosome population, mutation or crossover rate.
   Done according to the problem being solved.
2. Define the fitness function that will measure the performance also according to the problem being solved.
3. Randomly pick a number of population chromosomes.
4. Measure their fitness for each of the chromosomes picked.
5. Pick a pair of chromosomes from the population where the probability of one being picked depends on the fitness of the chromosomes. The higher the fitness the higher the probability.
6. Apply the mutation or crossover to obtain children chromosomes.
7. Place the children back to the original population.
8. Starting from step number 3, pick another population and repeat the process till the original population is replaced by a new population.

This algorithm mainly concentrates on the global search

2.6.3 Particle Swarm Optimization (PSO)

Particle Swarm Optimization is a global optimization technique developed by Eberhart and Kennedy in 1995 (Eberhart & Kennedy, 1995), where the underlying motivation of its algorithm was the social behavior observable in nature, such as flocks of birds and schools of fish in order to model swarms of particles moving towards the most promising regions of the search space. It exhibits good performance in finding solutions to static optimization problems where it is better than other algorithms like Genetic Algorithm (Hassan et al., 2005). Apart from this it also exploits a population of individuals to synchronously probe promising regions of the search space. In this case, the population is referred to as a swarm and the individuals (i.e., the search points) to as particles. The consideration here is that each particle in the swarm represents a candidate solution to the optimization problem. For a PSO system, every particle moves with an adjustable velocity through the search space, where it adjusts its position in the search space according to its own experience and that of neighboring particles. After this it retains a memory of the best position it ever encountered. A particle therefore makes use of the best position encountered by itself and the best position of neighbors to position itself towards the global minimum. This therefore results in particles “flying” towards the global minimum, while still searching a wide area around the best solution (Ghomsheh et al., 2007). Each particle’s performance (i.e., the “closeness” of a particle to the global minimum) is measured according to a predefined fitness function which is related to the problem being solved. For (Sumathi & Paneerselvam 2010) case, a particle represents the weight vector of NNs, including biases. The total number of weights and biases give the dimension of the search space (Sumathi & Paneerselvam 2010).

The iterative process of PSO can be described as follows (www.yarpiz.com; Zhang, 2022).

- **Step 1:** Initialize a population size, their positions, velocities of agents and the number of weights and biases.
- **Step 2:** Set the current best fitness achieved by particle *p* as *pbest* in which case the *pbest* with best value is set as *gbest* and this value is stored.
- **Step 3:** The desired optimization fitness function f_p for each particle is evaluated as the Mean Square Error (MSE) over a given data set.
- **Step 4:** The evaluated fitness value f_p of each particle is compared with its *pbest* value. For cases where f_p< pbest, pbest = f_p and best$_{\text{xp}}$= x_p, where x_p are the current coordinates of particle p, and bestxp are the coordinates corresponding to particle *p’s* best fitness so far.
- **Step 5:** For each particle the objective function value is calculated for new positions such that if a better position is achieved by an agent, *pbest* value is replaced by the current value. As is the case with Step 1, *gbest* value is selected among *pbest* values in which case if the new *gbest* value is better than previous *gbest* value, the *gbest* value is replaced by the current *gbest* value and this value is stored. If f_p< gbest then gbest = p, where *gbest* is the particle having the overall best fitness over all particles in the swarm.
- **Step 6:** The velocity and location of the particle is changed according to Equations 2.83 and 2.84, respectively.
- **Step 7:** Each particle p is flown according to Equation 2.83.

- **Step 8:** Stop if the maximum number of predetermined iterations (epochs) is exceeded stop; otherwise, Loop to step 3 until convergence.

$$V_i = wV_{i-1} + c_1 * rand() * \left(best_{xp} - xp\right) + c_2 * rand() * \left(best_{xgbest} - xp\right) \qquad 2.83$$

Where c_1 and c_2 are personal and social acceleration constants respectively that control how the particles fly from one another, and *rand* returns a uniform random number between 0 and 1.

$$xp = xpp + V_i \qquad 2.84$$

V_i is the current velocity, V_{i-1} is the previous velocity, xp is the present location of the particle, xpp is the previous location of the particle, and i is the particle index.

In step 5 the coordinates bestxp and bestxgbest are used to pull the particles towards the global minimum (Ghomsheh et al., 2007).

When the PSO learning algorithm is applied in training the modified ANFIS, its performance tremendously increases as compared with other training algorithms. For instance, according to (Adeyiga et al., 2022) the performance of PSO is found to be better than the Genetic Algorithm, since the PSO algorithm carries out global search and local searches simultaneously, while the Genetic Algorithm concentrates on the global search. The PSO optimization algorithm is also easy to develop and apply.

2.6.4 Evaluation criteria

The performance evaluation of the proposed approach will be done by measuring the estimation accuracy which is defined as the difference between the actual and estimated values. The first typical fitting criterion, mean square error (MSE) is defined in equation 2.85:

$$MSE = \frac{1}{N}\sum_{j=1}^{N}\left(y_j - \hat{y_j}\right)^2 \qquad 2.85$$

where N is the total number of data, y is actual target value, and $\hat{y}$ is estimated target value.

The tests are implemented several times to ensure that MSE converges to a minimum value. The initial weights values will randomly be assigned within the range [-1; 1]. The training accuracy is expressed in terms of the mean absolute error (ME), standard deviation (SD) and root mean squared error (RMSE). The absolute mean error (ME) is expressed as

$$e_i = |P_{measured} - P_{simulated}| \qquad 2.86$$

$$\bar{e} = \frac{1}{N}\sum_{j=1}^{N} e_j \qquad 2.87$$

where the terms *measured* and *simulated* denote received signal strength that are obtained by measurement and simulated by ANFIS, while N is total number of samples. The standard deviation is given by;

$$\sigma = \sqrt{\frac{1}{N-1}(e_i - \bar{e})^2} \qquad 2.88$$

The root mean squared error (RMSE) can also be determined according to the expression.

$$RMSE = \sqrt{\sigma^2 + \bar{e}^2} \qquad 2.89$$

In this training the RMSE was used.

2.7 Research gap

In the process of wireless communication propagation modelling, most researchers have not used artificial intelligence-based models as noted in the models discussed by (Kimoto et al., 2016; Allen et al., 2017; Ullah et al., 2020; Elmezughi et al., 2021; Zhang et al., 2017; Erunkulu et al., 2020), (Ahmadien et al., 2020; Valcarce et al., 2011; Dagefu & Sarabandi, 2010) and section 2.3, where the deterministic and empirical models are individually analyzed. Most of these models have a number of limitations compared to the models developed in our research. For instance, according to (Alotaibi et al., 2009), the Okumura's model is suitable for particular range of frequencies given as 150 to 1920 MHZ, a particular range of transmitter receiver separation that is 1 to 100 km and base station antenna height ranging from 30 to 1000 m and a mobile antenna height less than 10 m. so there is need to develop a universal model that solves this problem by taking care of all environments as well as frequency and distance ranges. Other empirical models like COST-Hata-Model, as presented by (Alotaibi et al., 2009), are usually restricted to large and small macro-cells applications, where according to (Alotaibi et al., 2009) this model's formula and its modification must not be applied in micro-cells. So, there is a need to develop a universal model based on the modified ANFIS. On the other hand, those who have used AI based modelling have worked with ANN and the ANFIS models that show better results than the traditional models. However, these models have also been used in individual form as discussed under section 2.4 and 2.5 supported by (Ahmad et al., 2020; Popoola et al., 2019; Ma et al., 2021; Popescu et al., 2006; Ahmadien et al., 2020; Valcarce et al., 2011; Vilovic & Burum, 2011; Alotaibi et al., 2009). Based on their analysis it is noted that the AI models are more efficient when dealing with prediction of practical RSSI, while the deterministic and empirical models are mostly theoretical. Comparing ANN to other popular large-scale prediction models, the ANN based models demonstrate very good performance for all types of environments such as outdoor as well as indoor giving greater accuracy as well as being less computationally extensive as compared to deterministic models. Besides this ANN modelling has a number of limitations that include data overfitting, low performance and increased training time. So, there is a need to develop a model that can handle these limitations, that is the modified ANFIS that combines the computation with numbers from neural networks and that of words that comes with fuzzy logic. The researchers who have dealt with the ANFIS based models have not shown on how they directly relate to wireless signal propagation modelling which our research has addressed by introducing the logarithm to base 10 operator to the conventional ANFIS as well as using PSO learning algorithm. This increased the accuracy level of the ordinary ANFIS when used for wireless signal prediction modelling as discussed under the results section. The model was named LOG10D-PSO-ANFIS because the modified ANFIS has a log10d operator and it is trained using the PSO optimization algorithm. Training the modified ANFIS, using the PSO learning algorithm, its performance in relation to wireless signal propagation modelling increases tremendously as compared to the results obtained from other training algorithms. Taking an example of PSO and GA, (Adeyiga et al., 2022) have indicated that the PSO performance is better than that of GA. This is because besides the PSO algorithm being easy to develop and apply, it also incorporates both global and local searches simultaneously whereas the GA concentrates on the global search. Through this research, the formulation of the ordinary ANFIS, the modified ANFIS and the PSO learning algorithm were undertaken as opposed to the other works which didn’t have this analysis that is important in understanding the algorithms by researchers interested in these concepts. Besides these additions we were able to generate a universal theoretical model, based on a modified ANFIS, that combines all the models into one, work that has not been done by any researcher before.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This section deals with the process that has been used to obtain the universal theoretical wireless signal propagation model. The first step was to formulate the conventional ANFIS architecture with two inputs and three membership functions where the inputs considered were arbitrary. This is followed by the description of the proposed modified ANFIS. This ANFIS (LOG10D-ANFIS) has an additional layer referred to as log10d where the distance is first passed through a logarithm to base 10 function before it goes to the fuzzification layer that is the first layer of the ordinary ANFIS. This is mainly to facilitate its application in wireless signal propagation modelling and therefore increasing the model’s accuracy. The formulation of the proposed modified ANFIS where the mathematical representations of the different operations, in each layer, on the input data are explained through from the first to the sixth layer is done. After this, the formulation of the modified ANFIS training process based on the PSO learning algorithm is undertaken where the corresponding mathematical relations on this process are also explained. This training process is used to generate the premise and consequent parameters for the proposed modified ANFIS. Based on this modified ANFIS trained with the PSO algorithm, the premise and consequent parameters for different theoretical models are generated. The models considered include the one slope, dual-slope, multi-wall, average-wall, COST231, COST231-Hata, Hata-Okumura and Two-ray ground reflection models considered as the sample models. For each of these models, an ANFIS equivalent model is developed through the premise and consequent parameters generated from the PSO training process. From these parameters a novel universal PSO trained modified ANFIS model (LOG10D-PSO-ANFIS) is developed where for each of the theoretical models required the corresponding parameters are used in the modified ANFIS structure to get its equivalent model. Apart from this universal theoretical model, another model that is to deal with practical RSSI values referred to as LOG10D-PSO-R-ANFIS is also developed starting with undertaking continuous wave measurements in a corridor using a phone and a Wi-Fi router. These measurements are used to train and test the model that is given a random input generated by the PSO algorithm together with the parameters used on the modified ANFIS model. Under this subsection, the process of performing the continuous wave measurements is discussed followed by the experimental setup to obtain the measured values. The next subsection shows how the formulation of the modified ANFIS model with a random input is undertaken and finally a brief description of the formulation of the PSO training process of this model is also described. The MATLAB ANFIS tool was used in the training and testing of the conventional and proposed models. The tool was used to generate the training and testing RMSE, MSE and SD as performance parameters for the models. These parameters were then compared to determine the best performing model as discussed under the results section.

3.2 Formulation of the conventional ANFIS architecture

Taking the original ANFIS with two inputs as represented in figure 2.16, its formulation starts from the generation of its rules that are given by;

$Rule\ 1: If\ x\ is\ A_1\ and\ y\ is\ B_1 then\ z_1 = p_1x + q_1y + r_1,$

$Rule\ 2: If\ x\ is\ A_2\ and\ y\ is\ B_2\ then\ z_2 = p_2x + q_2y + r_2,$

x and *y* are arbitrary inputs while *z* is their corresponding output.

With two inputs and 3 MFs the analysis is as represented below.

Taking arbitrary values of *x*, *y*, a_i, b_i and c_i as;

For A;

$[a_1\ b_1\ c_1]$

$[a_2\ b_2\ c_2]$

$[a_3\ b_3\ c_3]$

For B;

$[a_1\ b_1\ c_1]$

$[a_2\ b_2\ c_2]$

$[a_3\ b_3\ c_3]$

and p_i, qi and r_i as;

$[p_1\ q_1\ r_1]$

$[p_2\ q_2\ r_2]$

$[p_3\ q_3\ r_3]$

$[p_4\ q_4\ r_4]$

$[p_5\ q_5\ r_5]$

$[p_6\ q_6\ r_6]$

$[p_7\ q_7\ r_7]$

$[p_8\ q_8\ r_8]$

$[p_9\ q_9\ r_9]$

$Rule\ 1\!: If\ x\ is\ A_1\ and\ y\ is\ B_1, then\ z_1 =\ p_1x + q_1y + r_1,$

$If\ x\ is\ \exp\left\{-\left[\left(\frac{x-c_1}{a_1}\right)^2\right]^{b_1}\right\} and\ y\ is\ \exp\left\{-\left[\left(\frac{y-c_1}{a_1}\right)^2\right]^{b_1}\right\},\ then\ z_1 =\ p_1x +$

$q_1y + r_1,$

$Rule\ 2\!: Ifx\ is\ A_1\ and\ y\ is\ B_2, then\ z_2 =\ p_2x + q_2y + r_2,$

$Rule\ 3\!: If\ x\ is\ A_1\ and\ y\ is\ B_3, then\ z_3 =\ p_3x + q_3y + r_3,$

$Rule\ 4\!: If\ x\ is\ A_2\ and\ y\ is\ B_1, then\ z_4 =\ p_4x + q_4y + r_4,$

$ule\ 5\!: If\ x\ is\ A_2\ and\ y\ is\ B_2, then\ z_5 =\ p_5x + q_5y + r_5,$

$Rule\ 6\!: If\ x\ is\ A_2\ and\ y\ is\ B_3, then\ z_6 =\ p_6x + q_6y + r_6,$

$Rule\ 7\!: If\ x\ is\ A_3\ and\ y\ is\ B_1, then\ z_7 =\ p_7x + q_7y + r_7,$

$Rule\ 8\!: If\ x\ is\ A_3\ and\ y\ is\ B_2, then\ z_8 =\ p_8x + q_8y + r_8,$

$Rule\ 9\!: If\ x\ is\ A_3\ and\ y\ is\ B_3, then\ z_9 =\ p_9x + q_9y + r_9,$

The equations that follow for each layer are equivalent to equations 2.61 to 2.66 but expounded to show the formulation process.

Layer 1 (Fuzzy Layer):

In this layer the two inputs are transformed into their membership grades A1 to A3 and B1 to B3.

$$O_i^1 = f(x, a, b, c) = \mu_{Ai}(x) = \frac{1}{1+|(x-c_i)/a_i|^{2b_i}}$$

$$\mu_{\ Ai}(x) = \exp\{-\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}\}$$

$$\mu_{\ Bi}(y) = \exp\{-\left[\left(\frac{y-c_i}{a_i}\right)^2\right]^{b_i}\}$$

$$A_{\ 1} = \exp\{-\left[\left(\frac{x-c_1}{a_1}\right)^2\right]^{b_1}\}$$

$$A_{\ 2} = \exp\{-\left[\left(\frac{x-c_2}{a_2}\right)^2\right]^{b_2}\}$$

$$A_{\ 3} = \exp\{-\left[\left(\frac{x-c_3}{a_3}\right)^2\right]^{b_3}\}$$

$$B_{\ 1} = \exp\{-\left[\left(\frac{y-c_1}{a_1}\right)^2\right]^{b_1}\}$$

$$B_{\ 2} = \exp\{-\left[\left(\frac{y-c_2}{a_2}\right)^2\right]^{b_2}\}$$

$$B_{\ 3} = \exp\{-\left[\left(\frac{y-c_3}{a_3}\right)^2\right]^{b_3}\}$$

Layer 2 (Product Layer):

Here a T-norm operator that performs fuzzy AND operation is used to combine the inputs membership grades A and B.

$$w_i = O_i^2 = \mu_{Ai}(x).\mu_{Bi}(y), i = 1, 2$$

$$w_1 = \mu_{A1}(x).\mu_{B1}(y),$$

$$w_1 = A_1.B_1$$

$w_2 = A_1.B_2$

$w_3 = A_1.B_3$

$w_4 = A_2.B_1$

$w_5 = A_2.B_2$

$w_6 = A_2.B_3$

$w_7 = A_3.B_1$

$w_8 = A_3.B_2$

$w_9 = A_3.B_3$

Layer 3 (Normalized Layer):

In this layer the ratio of the ith rule's firing strength to the total of all firing strengths is computed.

$$\overline{w}_i = O_i^3 = \frac{w_i}{w_1+w_2+\cdots+w_9}\ , i = 1, \dots., 9,$$

$\overline{w}_1 = \frac{w_1}{w_1+\cdots+w_9}$

$\overline{w}_2 = \frac{w_2}{w_1+\cdots+w_9}$

$\overline{w}_3 = \frac{w_3}{w_1+\cdots+w_9}$

$\overline{w}_4 = \frac{w_4}{w_1+\cdots+w_9}$

$\overline{w}_5 = \frac{w_5}{w_1+\cdots+w_9}$

$\overline{w}_6 = \frac{w_6}{w_1+\cdots+w_9}$

$\overline{w}_7 = \frac{w_7}{w_1+\cdots+w_9}$

$\overline{w}_8 = \frac{w_8}{w_1+\cdots+w_9}$

$\overline{w}_9 = \frac{w_9}{w_1+\cdots+w_9}$

Layer 4 (Defuzzify Layer):

This layer computes the contribution of each rule to the overall output. Since there are two inputs, all the consequent parameters *pᵢ*, *qᵢ* and *rᵢ* are considered.

$$\bar{w}_i z_i = O_i^4 = \bar{w}_i(p_i x + q_i y + r_i)$$

$$\bar{w}_1 z_1 = \bar{w}_1(p_1 x + q_1 y + r_1),$$

$$\bar{w}_2 z_2 = \bar{w}_2(p_2 x + q_2 y + r_2)$$

$$\bar{w}_3 z_3 = \bar{w}_3(p_3 x + q_3 y + r_3)$$

$$\bar{w}_4 z_4 = \bar{w}_4(p_4 x + q_4 y + r_4)$$

$$\bar{w}_5 z_5 = \bar{w}_5(p_5 x + q_5 y + r_5)$$

$$\bar{w}_6 z_6 = \bar{w}_6(p_6 x + q_6 y + r_6)$$

$$\bar{w}_7 z_7 = \bar{w}_7(p_7 x + q_7 y + r_7)$$

$$\bar{w}_8 z_8 = \bar{w}_8(p_8 x + q_8 y + r_8)$$

$$\bar{w}_9 z_9 = \bar{w}_9(p_9 x + q_9 y + r_9)$$

Layer 5 (Total Output Layer):

This layer determines the overall output as the summation of contribution from each rule.

$$z = \sum_i \bar{w}_i z_i = O_i^5 = \frac{\sum_i w_i z_i}{\sum_i w_i} = \frac{w_1 z_1 + \cdots + w_9 z_9}{w_1 + \cdots + w_9} = \bar{w}_1 z_1 + \cdots + \bar{w}_9 z_9$$

3.3 The proposed modified ANFIS architecture

Our research came up with a modified ANFIS, as represented in figure 3.1, that has an additional layer outside the ordinary ANFIS of five layers. This layer is responsible for modifying the input data by applying the logarithm to base 10 operator which then feeds the ordinary ANFIS to process the modified data resulting to accurate output values. All the other five layers retain their functions as indicated in equations 2.61 to 2.66 with the modifications as represented in equations 3.1 to 3.6. The rules incorporating the logarithm to base 10 operator are given as follows for a single input;

$$Rule\ 1: If\ log10x\ is\ A_1\ then\ z_1 = p_1 log10x + r_1$$

$$Rule\ 2: If\ log10x\ is\ A_2\ then\ z_2 = \ p_2 log10x\ + r_2$$

Where x is the distance and z is the RSSI.

The introduction of the logarithm to base 10 operator enables the ANFIS structure to accept and process wireless signals with high levels of accuracy in relation to the root mean square error (RMSE). This is because the variation of these signals follows a trace represented by logarithm to base 10 output as discussed in section 2.3.

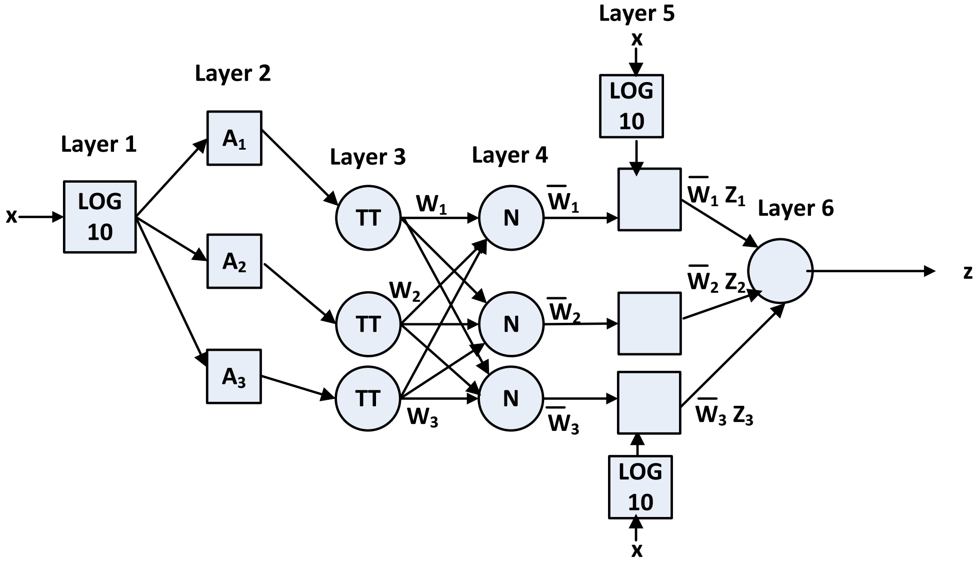


Figure 3.1: Modified ANFIS Structure with a single input

3.4 Formulation of the proposed modified ANFIS algorithm

The following representation is a step-by-step analysis of the modified ANFIS algorithm starting from layer 1 to layer 6. The total number of parameters to be considered is fifteen resulting from 9 by 1 plus 2 by 3 generally represented as *g* by *h* and *u* by *v* where *g* is the total number of premise parameters per input, *h* is the number of inputs, *u* is the number of consequent parameters per rule while *v* is the number of rules. The formulation with substituted values is given in appendix I.

Taking the ANFIS premise parameter values of a_i, b_i and c_i as;

$[a_1\ b_1\ c_1]$

$[a_2\ b_2\ c_2]$

[a_3 b_3 c_3]

and the consequent parameters p_i and r_i as;

[p_1 r_1]

[p_2 r_2]

[p_3 r_3]

The ANFIS structure rules that were used are represented as;

$Rule\ 1\!: If\ log10(x)\ is\ A_1, then\ z_1 =\ p_1 log10(x) + r_1,$

$$If\ log10(x)\ is\ \exp\{-\left[\left(\frac{log10(x)-c_1}{a_1}\right)^2\right]^{b_1}\}\ then\ z_1 =\ p_1 log10(x)\ + r_1,$$

Performing the same for rule 2 and 3

$Rule\ 2\!: If\ log10(x)\ is\ A_2, then\ z_2 =\ p_2 log10(x) + r_2,$

$$If\ log10(x)\ is\ \exp\{-\left[\left(\frac{log10(x)-c_2}{a_2}\right)^2\right]^{b_2}\}\ then\ z_2 =\ p_2 log10(x)\ + r_2,$$

$Rule\ 3\!: If\ log10(x)\ is\ A_3, then\ z_3 =\ p_2 log10(x) + r_3,$

$$If\ log10(x)\ is\ \exp\{-\left[\left(\frac{log10(x)-c_3}{a_3}\right)^2\right]^{b_3}\}\ then\ z_3 =\ p_3 log10(x)\ + r_3,$$

Where *x* is the distance and *z* represents the RSSI.

Layer 1 (Logarithmic layer):

This layer deals with the transformation of the distance values, as obtained from the one slope model, into their logarithmic representation. The resulting values are then fed into layer 2 which performs the fuzzification operation. Each of the distance values represented as *x* are transformed into their logarithmic equivalent using the logarithm to base 10 operator as shown in equation 3.1.

$$O^1 = log10(x) \qquad 3.1$$

This operator acts as the link between the original ANFIS model, and the wireless signal propagation modelling concept based on the models discussed in section 2.3 of this thesis since it introduces the logarithmic operator available in all these models.

Layer 2 (Fuzzy Layer):

Every node in this layer is an adaptive layer that generates the membership grades of the input vectors. Usually, a bell-shaped (Gaussian) function with minimum equal to 0 and maximum equal to 1 is used for implementing the node function as represented in equation 3.2 and 3.3. It receives its input that is distance from layer 1 which is operated on by the logarithm to base 10 operator.

$$O_i^2 = f(log10(x), a, b, c) = \mu_{Ai}(log10(x)) = \frac{1}{1+|(log10(x)-c_i)/a_i|^{2b_i}} \qquad 3.2$$

$$\mu_{\ Ai}(log10(x)) = \exp\{-\left[\left(\frac{log10(x)-c_i}{a_i}\right)^2\right]^{b_i}\} \qquad 3.3$$

$$A_{\ 1} = \exp\{-\left[\left(\frac{log10(x)-c_1}{a_1}\right)^2\right]^{b_1}\}$$

$$A_{\ 2} = \exp\{-\left[\left(\frac{log10(x)-c_2}{a_2}\right)^2\right]^{b_2}\}$$

$$A_{\ 3} = \exp\{-\left[\left(\frac{log10(x)-c_3}{a_3}\right)^2\right]^{b_3}\}$$

Where O_i^1 is the output of the i^{th}node in the first layer, $\mu_{Ai}(log10(x))$ is the membership function of input in the linguistic variable A_i. The parameter set $\{a_i, b_i, c_i\}$, referred to as the premise parameters, are responsible for defining the shapes of the membership functions. *A₁* to *A₃* represent the values from the three rules.

These parameters are tuned in the process of training the modified ANFIS so that it can give a desired output.

Layer 3 (Product Layer)

Each node in this layer determines the firing strength of a rule by multiplying the membership functions associated with the rules. The nodes in this layer are fixed with the firing strength of a particular rule, the output of a node, given by equation 3.4:

$$w_i = O_i^3 = \mu_{Ai}(log10(x)).\mu_{Bi}(y), i = 1, 2, 3 \qquad 3.4$$

Any other T-norm operator that performs fuzzy AND operation can be used in this layer to combine the inputs in the case where the inputs are more than one. This is based on P^N nodes denoting the number of rules in this layer with each node representing the antecedent part of rule (*if* part of an *if-then* rule) where N is the number of input variables and P is the number of membership functions. Since there is one input for this case, the resulting outputs are given by w_1 to w_3. Since there is one input, the following relations are obtained.

$$w_1 = \mu_{A1}(log10(x)),$$

$$w_2 = \mu_{A2}(log10(x)),$$

$$w_3 = \mu_{A3}(log10(x)),$$

Layer 4 (Normalized Layer)

As discussed earlier under the ordinary ANFIS, this layer consists of fixed nodes that are used to compute the ratio of the ith rule's firing strength to the total of all firing strengths. Its output is given by equation 3.5 and is referred to as normalized firing strength.

$$\overline{w}_i = O_i^4 = \frac{w_i}{w_1+w_2+w_3} , i = 1, 2, 3 \qquad 3.5$$

$$\overline{w}_1 = \frac{w_1}{w_1+w_2+w_3}$$

$$\bar{w}_2 = \frac{w_2}{w_1 + w_2 + w_3}$$

$$\bar{w}_3 = \frac{w_3}{w_1 + w_2 + w_3}$$

In the modified ANFIS, the logarithm to base 10 operator affects this layer's output as computed in layer 3. This layer has P^N nodes as well.

Layer 5 (Defuzzify Layer)

Just like the ordinary ANFIS, this layer is an adaptive node whose representation is as given in equation 3.6. It does the function of computing the contribution of each rule to the overall output also known as fuzzification that provide output values resulting from the inference of rules. The parameters in this layer given as $\{p_i,\ q_i,\ r_i\}$ and known as consequent parameters, appearing at the output section of the network, are adaptive just as like the premise parameters in layer 2. They are to be modified using an appropriate training algorithm. Since there is just one input that is distance, only the parameters p and r are considered in case as represented in equation 3.6 with the logarithm to base 10 operator on the distance included. The resulting outputs are given by $\bar{w}_1 z_1$ to $\bar{w}_3 z_3$ since this ANFIS is using three membership functions. These outputs when combined result to the RSSI output being predicted by the modified ANFIS.

$$\bar{w}_i z_i = O_i^5 = \bar{w}_i (p_i log10(x) + r_i) \qquad 3.6$$

$$\bar{w}_1 z_1 = \bar{w}_1 (p_1 log10(x) + r_1),$$

$$\bar{w}_2 z_2 = \bar{w}_2 (p_2 log10(x) + r_2),$$

$$\bar{w}_3 z_3 = \bar{w}_3 (p_3 log10(x) + r_3),$$

Layer 6 (Total Output Layer)

This layer consists of a fixed node that basically computes the overall output as the summation of contribution from each rule whose function is given as in equation 3.7. The computation in this layer results to the modified ANFIS predicted RSSI output represented by *z* in equation 3.7.

$$z = \sum_i \bar{w}_i z_i = O_i^6 = \frac{\sum_i w_i z_i}{\sum_i w_i} = \frac{w_1 z_1 + w_2 z_2 + w_2 z_3}{w_1 + w_2 + w_3} = \bar{w}_1 z_1 + \bar{w}_2 z_2 + \bar{w}_3 z_3 \qquad 3.7$$

3.5 Formulation of the PSO learning algorithm for the modified ANFIS architecture

The training and validation processes are among the important steps used to develop an accurate process model using ANFIS where a set of input-output patterns is repeated to the ANFIS in the training process. The weights of all the interconnections between neurons are adjusted repeatedly until the specified input yields the desired output. From these iterations, the ANFIS learns the right input-output response behavior (Ghomsheh et al., 2007). In our research PSO is employed for updating the ANFIS parameters where ANFIS has two types of parameters which need training that is, the antecedent part parameters and the conclusion part parameters. It is assumed that the membership functions are Gaussian as given in equation 3.3, and their parameters are $\{a_i, b_i, c_i\}$, where a_i is the variance of membership functions, c_i is the center of membership functions (MFs) and b_i is a trainable parameter. The parameters $\{p_i,\ q_i,\ r_i\}$ of conclusion part are also trained in our solution. As indicated above there are 3 sets of trainable parameters in antecedent part $\{a_i, b_i, c_i\}$ where each of these parameters has m particles which represent the number of MFs. The conclusion parameters ($\{p_i,\ q_i,\ r_i\}$) are also trained during optimization process. They are also m particles, where the fitness is defined as root mean square error (RMSE). In the first step the parameters are initialized randomly after which they are updated using PSO algorithm depending on the number of iterations. During each iteration the parameters sets are being updated according to the fitness function RMSE. This algorithm is as represented in the figure 3.2 flowchart.

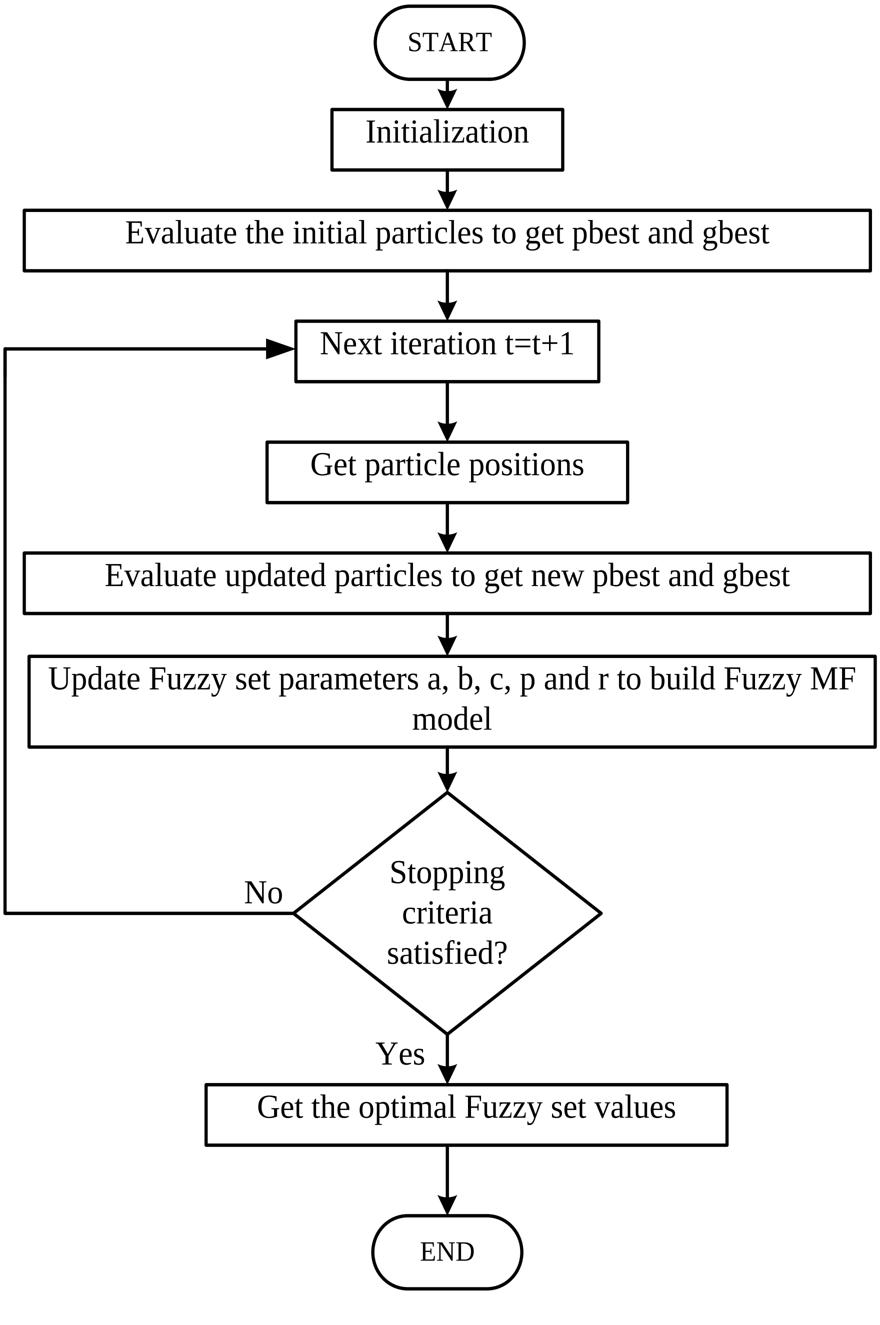

START
Initialization
Evaluate the initial particles to get pbest and gbest
Next iteration t=t+1
Get particle positions
Evaluate updated particles to get new pbest and gbest
Update Fuzzy set parameters a, b, c, p and r to build Fuzzy MF model
Stopping criteria satisfied?
No
Yes
Get the optimal Fuzzy set values
END

Figure 3.2: Modified ANFIS training with PSO flowchart

The minimum of the fitness function $f(x) = ANFIS\text{-}RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \widehat{y_j}\right)^2}$ for the ANFIS structure where N is the total number of data, y is actual target value, and $\hat{y}$ is estimated target value. The membership function parameters are obtained using the PSO algorithm with *m* particles (sets of ANFIS membership function parameters). The ANFIS requires 15 parameters since it has 3 MFs with one input. This results to

15 as the number of parameters that are to be searched for by the PSO. The parameters are represented by *a1*, *b1*, *c1*, *a2*, *b2*, *c2*, *a3*, *b3* and *c3* as the premise parameters while the consequent parameters are given by *p1*, *r1*, *p2*, *r2*, *p3* and *r3*.

The *m* initial positions (membership function parameters) with 15 variables each given as;

$x_{01}^{t_0}$ =[x_{01}^{1} x_{01}^{2}………………x_{01}^{15}];

$x_{02}^{t_0}$ =[x_{02}^{1} x_{02}^{2}………………x_{02}^{15}];

…………

$x_{m}^{t_0}$ =[x_{m}^{1} x_{m}^{2}………………x_{m}^{15}];

and are to be applied as indicated in the description that follows.

Using the particles with initial positions above the detailed computations for iterations 1, 2 and 1000 are analyzed starting with step 1.

Step1: choose the number of particles: x_{01}, x_{02}, ….., x_m

The initial population (i.e., the iteration number t=t_0) can be represented as $x_i^{t_0}. i = 1,2, … . , m$:

$x_{01}^{t_0}$ =[x_{01}^{1} x_{01}^{2}………………x_{01}^{15}];

$x_{02}^{t_0}$ =[x_{02}^{1} x_{02}^{2}………………x_{02}^{15}];

…………

$x_{m}^{t_0}$ =[x_{m}^{1} x_{m}^{2}………………x_{m}^{15}];

Evaluate the objective function values as $f(x) = RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \widehat{y_j}\right)^2}$

For $f_{01}^{t_0}$ substituting the $x_{01}^{t_0}$ = [x_{01}^{1} x_{01}^{2}………………x_{01}^{15}] membership function parameters into the FIS in APPENDIX I the RMSE value is obtained.

$$f\left(x_{01}^{t_0}\right) = RMSE_{x_{01}^{t_0}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \widehat{y_j}\right)^2}$$

Use the same objective function for $f_{02}^{t_0}$ substituting the $x_{02}^{t_0}$ membership function parameters to obtain the RMSE output.

$$f\left(x_{02}^{t_0}\right) = RMSE_{x_{02}^{t_0}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \widehat{y_J}\right)^2},$$

……….

The process is repeated for all the *m* sets of parameters where for $f_m^{t_0}$ substituting the $x_m^{t_0}$ membership function parameters the RMSE for $x_m^{t_0}$ parameters is obtained.

$$f\left(x_m^{t_0}\right) = RMSE_{x_m^{t_0}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \widehat{y_J}\right)^2},$$

Let $c_1 = 2$ and $x_2 = 2$;

Set the initial velocities of each particle to zero:

$v_i^{t_0} = 0, i.e. v_1^{t_0} = v_2^{t_0} = \cdots = v_m^{t_0} = 0$

$v_{01}^{t_0}$=[0 0 … 0]; $v_{02}^{t_0}$=[0 0 … 0]; …. $v_m^{t_0}$=[0 0… 0];

Step 2: Set the iteration number as t_1 and go to step 3

Step 3: Find the personal best for each particle by

$$P_{best,i}^{t_1} = \begin{Bmatrix} P_{best,i}^{t_0} & \text{if } f_i^{t_1} > P_{best,i}^{t_0} \\ x_i^{t_1} & \text{if } f_i^{t_1} \leq P_{best,i}^{t_0} \end{Bmatrix},$$

So,

$P_{best,01}^{t_1}$ =[$x_{01}^{t_1,1}$ $x_{01}^{t_1,2}$……………….$x_{01}^{t_1,15}$];

$P_{best,02}^{t_1}$ =[$x_{02}^{t_1,1}$ $x_{02}^{t_1,2}$……………….$x_{02}^{t_1,15}$];

…………

$P_{best,m}^{t_1}$ =[$x_m^{t_1,1}$ $x_m^{t_1,2}$……………….$x_m^{t_1,15}$];

Step 4: Find the global best by using

$G_{best}^{t_1} = \min\{P_{best,i}^{t_1}\}$ where $i = 1,2,\dots,m.$

Step 5: Considering the random numbers in the range (0,1) as

$r_1^{t_0}$ and $r_2^{t_0}$ and find the velocities of the particles by

$v_i^{t_1} = w * (v_i^{t_0} + c_1 r_1^{t_0}[P_{best,i}^{t_0} - x_i^{t_0}] + c_2 r_2^{t_0}[G_{best,i}^{t_0} - x_i^{t_0}]); i = 1,\dots,m$

Where *w* is the inertia coefficient and taken as 1 while *wdamp* is the damping ratio of the inertia coefficient taken 0.99.

So

$v_{01}^{t_1} = w * (v_{01}^{t_0} + c_1 r_1^{t_0}[P_{best,01}^{t_0} - x_{01}^{t_0}] + c_2 r_2^{t_0}[G_{best,01}^{t_0} - x_{01}^{t_0}]);$

$v_{02}^{t_1} = w * (v_{02}^{t_0} + c_1 r_1^{t_0}[P_{best,02}^{t_0} - x_{02}^{t_0}] + c_2 r_2^{t_0}[G_{best,02}^{t_0} - x_{02}^{t_0}]);$

…..

$v_m^{t_1} = w * (v_m^{t_0} + c_1 r_1^{t_0}[P_{best,m}^{t_0} - x_m^{t_0}] + c_2 r_2^{t_0}[G_{best,m}^{t_0} - x_m^{t_0}]);$

Setting minimum velocity and maximum velocity

Step 6: Find the new values of $x_i^{t_1}, i = 1,\dots,m$ by taking

$x_i^{t_1} = x_i^{t_0} + v_i^{t_1}$

So

$x_{01}^{t_1} = x_{01}^{t_0} + v_{01}^{t_1}$

$x_{02}^{t_1} = x_{02}^{t_0} + v_{02}^{t_1}$

….

$x_m^{t_1} = x_m^{t_0} + v_m^{t_1}$

Set minimum position as the lowest value and maximum position as maximum value.

Step 7: Evaluate the objective function values of $f(x_{01}^{t_1})$ to $f(x_m^{t_1})$ that is obtain the RMSE for the other sets of membership function parameters by applying them into the FIS structure.

Step 8: Stopping criterion:

If the terminal rule is not satisfied, go to step 2,

Otherwise, stop the iteration and output the results.

…….

Step2: Set the iteration number as t_{999}, then go to step 3.

Step3: Find the personal best for each particle.

$$P_{best,i}^{t_{999}} = \left\{\begin{matrix} P_{best,i}^{t_{998}} & \text{if } f_i^{t_{999}} > P_{best,i}^{t_{998}} \\ x_i^{t_{999}} & \text{if } f_i^{t_{999}} \leq P_{best,i}^{t_{998}} \end{matrix}\right\},$$

$P_{best,01}^{t_{999}}$ =[$x_{01}^{t_{999},1}$ $x_{01}^{t_{999},2}$………………$x_{01}^{t_{999},15}$];

$P_{best,02}^{t_{999}}$ =[$x_{02}^{t_{999},1}$ $x_{02}^{t_{999},2}$………………$x_{02}^{t_{999},15}$];

…………

$P_{best,m}^{t_{999}}$ =[$x_{m}^{t_{999},1}$ $x_{m}^{t_{999},2}$………………$x_{m}^{t_{999},15}$];

Step 4: find the global best

$G_{best}^{t_{999}} = \min\{P_{best,i}^{t_{999}}\}$ where $i = 1,2, \dots, m$.

Step 5: By considering the random numbers in the range (0,1) as $r_1^{t_{999}}$ and $r_2^{t_{999}}$ and find the velocities of the particle by

$$v_i^{t_{1000}} = w * (v_i^{t_{999}} + c_1 r_1^{t_{999}}\left[P_{best,i}^{t_{999}} - x_i^{t_{999}}\right] + c_2 r_2^{t_{999}}\left[G_{best,i}^{t_{999}} - x_i^{t_{999}}\right]); i = 1, \dots, m$$

So

$$v_{01}^{t_{1000}} = w * (v_{01}^{999} + c_1 r_1^{999}\left[P_{best,01}^{999} - x_{01}^{999}\right] + c_2 r_2^{t_{999}}\left[G_{best,01}^{999} - x_{01}^{999}\right]);$$

$$v_{02}^{t_{1000}} = w * (v_{02}^{999} + c_1 r_1^{t_{999}}\left[P_{best,02}^{999} - x_{02}^{999}\right] + c_2 r_2^{999}\left[G_{best,02}^{999} - x_{02}^{999}\right]);$$

……..

$$v_{m}^{t_{1000}} = w * (v_{m}^{999} + c_1 r_1^{t_{999}}\left[P_{best,m}^{999} - x_{m}^{999}\right] + c_2 r_2^{999}\left[G_{best,m}^{999} - x_{m}^{999}\right]);$$

Set minimum velocity and maximum velocity

Step 6: Find the new values of $x_i^{t_{1000}}, i = 1, \dots, m$ by

$x_i^{t_{1000}} = x_i^{t_{999}} + v_i^{t_{1000}}$

So

$x_{01}^{t_{1000}} = x_{01}^{t_{999}} + v_{01}^{t_{1000}}$

$x_{02}^{t_{1000}} = x_{02}^{t_{999}} + v_{02}^{t_{1000}}$

….

$x_m^{t_{1000}} = x_m^{t_{999}} + v_m^{t_{1000}}$

Set the minimum position value and maximum position value.

Step 7: Evaluate the objective function $f\left(x_{01}^{t_{1000}}\right)$ up to $f\left(x_m^{t_{1000}}\right)$ for the t_{1000} iteration by substituting the $x_{01}^{t_{1000}}$ to $x_m^{t_{1000}}$ values of the membership function parameters into the FIS structure. This will result to RMSE$_{t_{1000}}$ values from which the minimum RMSE value is picked to obtain the required best RMSE value.

Step 8: Stopping criterion:

If the terminal rule is not satisfied, go to step 2, otherwise stop the iteration, and output the results.

Finally, the values $x_i^{t_{1000}}, i = 1, \dots, m$ did not converge, but the requirement to stop at iteration t_{1000} is reached. If the iterations continue, convergence is reached when the positions of all particles converge to similar values, then the method has converged and the corresponding values of $x_i^{t_t}$ are the optimum solution. Therefore, the iterative process is stopped due to the requirement that it stops at iteration t_{1000}. This gives an optimum solution of training and testing RMSE. The final best particle position, that is, the final global particle position gives the best membership function parameters for the best performing ANFIS in terms of training and testing input distance and output RSSI. The obtained optimum premise and consequent parameters are applied in figure 3.1 to give the predicted wireless signal RSSI.

3.6 Development of equivalent theoretical ANFIS based models

These models were developed using original ANFIS, modified ANFIS that is LOG10D-ANFIS and PSO trained modified ANFIS that is LOG10D-PSO-ANFIS for one slope, dual-slope, multi-wall, average-wall, COST231, COST231-Hata, Hata-Okumura and Two-ray ground reflection models. Starting with the original ANFIS trained with hybrid algorithm of gradient descent and back propagation, the RMSE, ME, and SD are obtained. The same parameters are obtained using the modified ANFIS trained with the hybrid algorithm. Similarly, this was done for the modified ANFIS trained using the PSO algorithm. The parameters were then compared to obtain the most accurate model in relation wireless signal propagation modelling.

3.7 The novel universal theoretical model based on PSO trained LOG10DANFIS with distance as input

This is the modified ANFIS with the logarithm to base 10 operator with a single input. It requires only 15 parameters for an ANFIS with 3 membership functions. The required parameters are obtained by multiplying the 9 premise parameters by 1 input added to 2 consequent parameters multiplied by the 3 rules. This forms the developed novel universal model whose corresponding representation is given in figure 3.3 that is similar to figure 3.1 the difference being that PSO has been used to update the parameters.

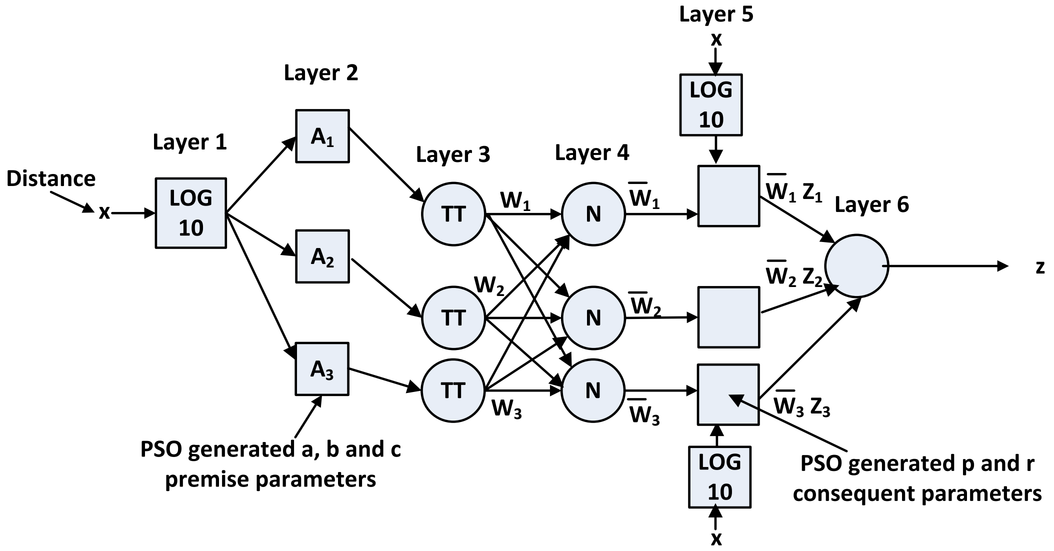


Figure 3.3: PSO trained ANFIS structure with one input distance

Figure 3.3 represents the ANFIS based developed universal model. Using the modified ANFIS PSO trained model, the individual membership parameters for each model are obtained through the training process. Each of the obtained membership parameters can then be applied to a single ANFIS model depending on the application of its application. Figure 3.4 shows the block diagram representation of the universal model. The model can be used for all environments and all frequencies. The membership function parameters for each original model to be used are as given below.

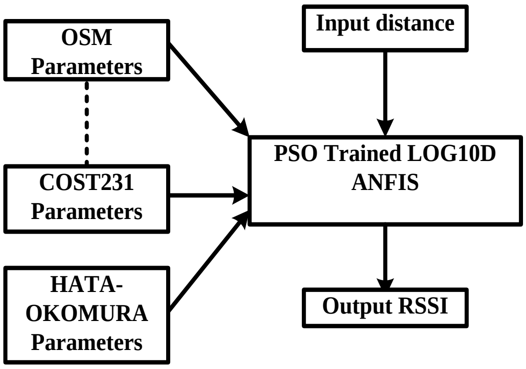


Figure 3.4: PSO trained modified ANFIS universal model block diagram

m= input ('membership functions = ');

One slope model (osm) membership functions

m_{osm}=[osm_{m1} osm_{m2} osm_{m3} osm_{m4} osm_{m5} osm_{m6} osm_{m7} osm_{m8} osm_{m9} osm_{m10} osm_{m11} osm_{m12} osm_{m13} osm_{m14} osm_{m15}];

Dual slope model (dsm) membership functions

m_{dsm}=[dsm_{m1} dsm_{m2} dsm_{m3} dsm_{m4} dsm_{m5} dsm_{m6} dsm_{m7} dsm_{m8} dsm_{m9} dsm_{m10} dsm_{m11} dsm_{m12} dsm_{m13} dsm_{m14} dsm_{m15}];

Multi-wall model (mwm) membership functions

m_{mwm}=[mwm_{m1} mwm_{m2} mwm_{m3} mwm_{m4} mwm_{m5} mwm_{m6} mwm_{m7} mwm_{m8} mwm_{m9} mwm_{m10} mwm_{m11} mwm_{m12} mwm_{m13} mwm_{m14} mwm_{m15}];

Hata-Okomura-Rural model (horm) membership functions

m_{horm}=[$horm_{m1}$ $horm_{m2}$ $horm_{m3}$ $horm_{m4}$ $horm_{m5}$ $horm_{m6}$ $horm_{m7}$ $horm_{m8}$ $horm_{m9}$ $horm_{m10}$ $horm_{m11}$ $horm_{m12}$ $horm_{m13}$ $horm_{m14}$ $horm_{m15}$];

Hata-Okomura-suburban model membership functions

m_{hosm}=[$hosm_{m1}$ $hosm_{m2}$ $hosm_{m3}$ $hosm_{m4}$ $hosm_{m5}$ $hosm_{m6}$ $hosm_{m7}$ $hosm_{m8}$ $hosm_{m9}$ $hosm_{m10}$ $hosm_{m11}$ $hosm_{m12}$ $hosm_{m13}$ $hosm_{m14}$ $hosm_{m15}$];

Hata-Okomura-urban model membership functions

m_{houm}=[$houm_{m1}$ $houm_{m2}$ $houm_{m3}$ $houm_{m4}$ $houm_{m5}$ $houm_{m6}$ $houm_{m7}$ $houm_{m8}$ $houm_{m9}$ $houm_{m10}$ $houm_{m11}$ $houm_{m12}$ $houm_{m13}$ $houm_{m14}$ $houm_{m15}$];

COST231-Hata-metropolitan model membership functions

m_{chmm}=[$chmm_{m1}$ $chmm_{m2}$ $chmm_{m3}$ $chmm_{m4}$ $chmm_{m5}$ $chmm_{m6}$ $chmm_{m7}$ $chmm_{m8}$ $chmm_{m9}$ $chmm_{m10}$ $chmm_{m11}$ $chmm_{m12}$ $chmm_{m13}$ $chmm_{m14}$ $chmm_{m15}$];

COST231-Hata-suburban model membership functions

m_{chsm}=[$chsm_{m1}$ $chsm_{m2}$ $chsm_{m3}$ $chsm_{m4}$ $chsm_{m5}$ $chsm_{m6}$ $chsm_{m7}$ $chsm_{m8}$ $chsm_{m9}$ $chsm_{m10}$ $chsm_{m11}$ $chsm_{m12}$ $chsm_{m13}$ $chsm_{m14}$ $chsm_{m15}$];

COST231 model membership functions

m_{cm}=[cm_{m1} cm_{m2} cm_{m3} cm_{m4} cm_{m5} cm_{m6} cm_{m7} cm_{m8} cm_{m9} cm_{m10} cm_{m11} cm_{m12} cm_{m13} cm_{m14} cm_{m15}];

Two-ray ground reflection model membership functions

m_{trgrm}=[$trgrm_{m1}$ $trgrm_{m2}$ $trgrm_{m3}$ $trgrm_{m4}$ $trgrm_{m5}$ $trgrm_{m6}$ $trgrm_{m7}$ $trgrm_{m8}$ $trgrm_{m9}$ $trgrm_{m10}$ $trgrm_{m11}$ $trgrm_{m12}$ $trgrm_{m13}$ $trgrm_{m14}$ $trgrm_{m15}$];

3.8 Development of an ANFIS based model (LOG10D-PSO-R-ANFIS) to predict measured RSSI.

For the prediction of real field data, the PSO trained modified ANFIS with random input was used. The process started with taking measurements along a corridor where a Wi-Fi router was taken as a transmitter while a phone was used as the receiver as indicated under experimental setup section. The measured RSSI values were recorded and used with the practical based model. These values were then used to give the range of the possible RSSI values to be generated by the model. Formulation of the practical based model was done in terms the modified ANFIS with two inputs and its corresponding PSO training process. The random RSSI values to be used with the modified ANFIS are generated using the PSO algorithm during the training process.

Continuous wave measurements for practical RSSI modelling

For this experimental setup, a Wi-Fi router and a Wi-Fi enabled mobile device were used. The mobile device distance was varied from 1m to 42m along a corridor. The 42m distance was used because it was the size of the corridor where the measurements took place. In this study, the measurement equipment consisted of a transmitter and a receiver. The narrow band continuous wave (CW) transmitter, which was transmitting at a specific frequency, was used together with the receiving mobile device. The obtained data is to be used to come up with a model that simulates the real RSSI in a given environment. In this case the setup was done in a corridor as represented in figure 3.5 to figure 3.7.

Experimental setup for practical measurement of RSSI

Figure 3.5 is a graphic representation of the experimental setup to facilitate the collection of practical RSSI signal data whereas figures 3.6 and 3.7 are picture representations showing the devices and tools used in the setup.

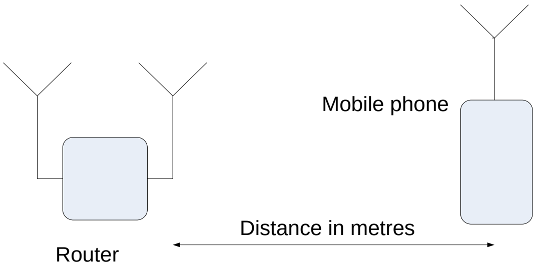


Figure 3.5: Diagram of the experimental Set-up

Figure 3.6: Image for the experimental Set-up

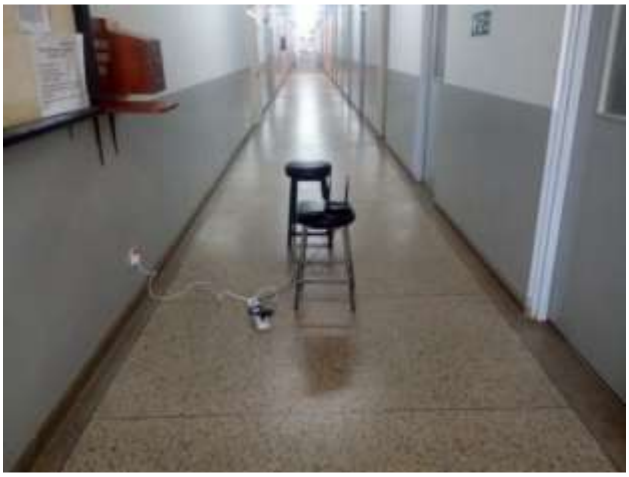


Figure 3.7: Image for the experimental Set-up in a corridor

The steps for carrying out the experiment are as follows;

i. A tape measure was used to measure a distance of 42m that was subdivided into 42 points each 1m apart.

ii. The Wi-Fi enabled mobile device was manually moved metre by metre away from the router and took the readings for every 1m to 42m.

3.8.1 Formulation of the modified ANFIS and random input

In this section the step-by-step formulation of the modified ANFIS with a random input is undertaken from layer 1 to layer 6. In this case the total number of parameters to be considered is forty-five as obtained from 9 by 2 plus 3 by 9 where the first 9 is the number of premise parameters per input, 2 is the number of inputs, 3 is the number of consequent parameters per rule while 9 is the number of rules. Figure 3.8 shows a hybrid trained modified ANFIS structure with two inputs where one of the inputs is the PSO generated random input.

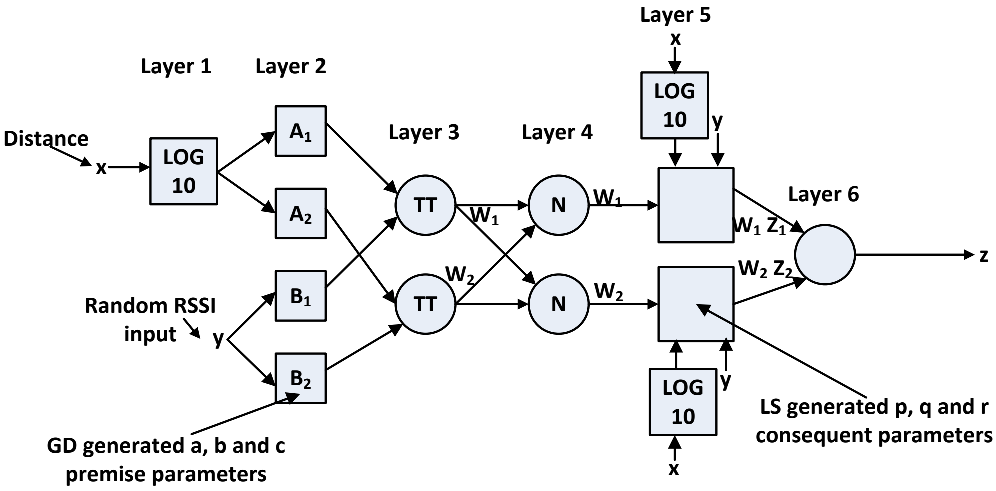


Figure 3.8: Hybrid trained modified ANFIS structure with two inputs

$Rule\ 1{:}\ If\ log10x\ is\ A_1\ and\ y\ is\ B_1 then\ z_1 =\ p_1 log10x + q_1 y + r_1,$

$Rule\ 2{:}\ If\ log10x\ is\ A_2\ and\ y\ is\ B_2\ then\ z_2 =\ p_2 log10x + q_2 y + r_2,$

x=distance, *y*=random input and *z*=RSSI

R is a random input which also shows the adaptability of the system-output modelled to vary according to the *R* random input

Two inputs with 3 MFs based on the analysis below.

Taking arbitrary values of distance *x*, random RSSI *y*, a_i, b_i and c_i as;

For A;

[a_1 b_1 c_1]

[a_2 b_2 c_2]

[a_3 b_3 c_3]

For B;

[a_1 b_1 c_1]

[a_2 b_2 c_2]

[a_3 b_3 c_3]

and p_i, qi and r_i as;

[p_1 q_1 r_1]

[p_2 q_2 r_2]

[p_3 q_3 r_3]

[p_4 q_4 r_4]

[p_5 q_5 r_5]

[p_6 q_6 r_6]

[p_7 q_7 r_7]

[p_8 q_8 r_8]

[p_9 q_9 r_9]

$Rule\ 1\colon If\ log10(x)\ is\ A_1\ and\ y\ is\ B_1, then\ z_1 =\ p_1 log10(x) + q_1 y + r_1,$

$$If\ log10(x) is \exp\left\{-\left[\left(\frac{log10(x)-c_1}{a_1}\right)^2\right]^{b_1}\right\} and\ y\ is \exp\left\{-\left[\left(\frac{y-c_1}{a_1}\right)^2\right]^{b_1}\right\},$$

$then\ z_1 =\ p_1 log10(x) + q_1 y + r_1,$

$Rule\ 2\colon If\ log10(x)\ is\ A_1\ and\ y\ is\ B_2, then\ z_2 =\ p_2 log10(x) + q_2 y + r_2,$

$Rule\ 3\colon If\ log10(x)\ is\ A_1\ and\ y\ is\ B_3, then\ z_3 =\ p_3 log10(x) + q_3 y + r_3,$

$Rule\ 4\colon If\ log10(x)\ is\ A_2\ and\ y\ is\ B_1, then\ z_4 =\ p_4 log10(x) + q_4 y + r_4,$

$Rule\ 5\colon If\ log10(x)\ is\ A_2\ and\ y\ is\ B_2, then\ z_5 =\ p_5 log10(x) + q_5 y + r_5,$

$Rule\ 6\colon If\ log10(x)\ is\ A_2\ and\ y\ is\ B_3, then\ z_6 =\ p_6 log10(x) + q_6 y + r_6,$

$Rule\ 7\colon If\ log10(x)\ is\ A_3\ and\ y\ is\ B_1, then\ z_7 =\ p_7 log10(x) + q_7 y + r_7,$

$Rule\ 8\colon If\ log10(x)\ is\ A_3\ and\ y\ is\ B_2, then\ z_8 =\ p_8 log10(x) + q_8 y + r_8,$

$Rule\ 9\colon If\ log10(x)\ is\ A_3\ and\ y\ is\ B_3, then\ z_9 =\ p_9 log10(x) + q_9 y + r_9,$

Layer 1 (Fuzzy Layer):

$$O_i^1 = f(log10(x), a, b, c) = \mu_{Ai}(log10(x)) = \frac{1}{1+|(log10(x)-c_i)/a_i|^{2b_i}}$$

$$\mu_{\ Ai}(log10(x)) = \exp\{-\left[\left(\frac{log10(x)-c_i}{a_i}\right)^2\right]^{b_i}\}$$

$$\mu_{\ Bi}(y) = \exp\{-\left[\left(\frac{y-c_i}{a_i}\right)^2\right]^{b_i}\}$$

$$A_{\ 1} = \exp\{-\left[\left(\frac{log10(x)-c_1}{a_1}\right)^2\right]^{b_1}\}$$

$$A_{\ 2} = \exp\{-\left[\left(\frac{log10(x)-c_2}{a_2}\right)^2\right]^{b_2}\}$$

$$A_{\ 3} = \exp\{-\left[\left(\frac{log10(x)-c_3}{a_3}\right)^2\right]^{b_3}\}$$

$$B_{\ 1} = \exp\{-\left[\left(\frac{y-c_1}{a_1}\right)^2\right]^{b_1}\}$$

$$B_{\ 2} = \exp\{-\left[\left(\frac{y-c_2}{a_2}\right)^2\right]^{b_2}\}$$

$$B_{\ 3} = \exp\{-\left[\left(\frac{y-c_3}{a_3}\right)^2\right]^{b_3}\}$$

Layer 2 (Product Layer):

$$w_i = O_i^2 = \mu_{Ai}(log10(x)).\mu_{Bi}(y), i = 1, 2$$

$$w_1 = \mu_{A1}(log10(x)).\mu_{B1}(y),$$

$$w_1 = A_1.B_1$$

$$w_2 = A_1.B_2$$

$$w_3 = A_1.B_3$$

$$w_4 = A_2.B_1$$

$$w_5 = A_2.B_2$$

$$w_6 = A_2.B_3$$

$$w_7 = A_3.B_1$$

$$w_8 = A_3.B_2$$

$$w_9 = A_3.B_3$$

Layer 3 (Normalized Layer):

$$\overline{w}_i = O_i^3 = \frac{w_i}{w_1+w_2+\cdots+w_9} \text{ }, i = 1, \ldots., 9,$$

$$\overline{w}_1 = \frac{w_1}{w_1+\cdots+w_9}$$

$$\overline{w}_2 = \frac{w_2}{w_1+\cdots+w_9}$$

$$\overline{w}_3 = \frac{w_3}{w_1+\cdots+w_9}$$

$$\overline{w}_4 = \frac{w_4}{w_1+\cdots+w_9}$$

$$\overline{w}_5 = \frac{w_5}{w_1+\cdots+w_9}$$

$$\overline{w}_6 = \frac{w_6}{w_1+\cdots+w_9}$$

$$\overline{w}_7 = \frac{w_7}{w_1+\cdots+w_9}$$

$$\overline{w}_8 = \frac{w_8}{w_1+\cdots+w_9}$$

$$\overline{w}_9 = \frac{w_9}{w_1+\cdots+w_9}$$

Layer 4 (Defuzzify Layer):

$$\overline{w}_i z_i = O_i^4 = \overline{w}_i(p_i log10(x) + q_i y + r_i)$$

$$\overline{w}_1 z_1 = \overline{w}_1(p_1 log10(x) + q_1 y + r_1),$$

$$\overline{w}_2 z_2 = \overline{w}_2(p_2 log10(x) + q_2 y + r_2)$$

$$\overline{w}_3 z_3 = \overline{w}_3(p_3 log10(x) + q_3 y + r_3)$$

$$\overline{w}_4 z_4 = \overline{w}_4(p_4 log10(x) + q_4 y + r_4)$$

$$\overline{w}_5 z_5 = \overline{w}_5(p_5 log10(x) + q_5 y + r_5)$$

$$\overline{w}_6 z_6 = \overline{w}_6(p_6 log10(x) + q_6 y + r_6)$$

$$\overline{w}_7 z_7 = \overline{w}_7(p_7 log10(x) + q_7 y + r_7)$$

$$\bar{w}_8 z_8 = \bar{w}_8(p_8 log10(x) + q_8 y + r_8)$$

$$\bar{w}_9 z_9 = \bar{w}_9(p_9 log10(x) + q_9 y + r_9)$$

Layer 5 (Total Output Layer):

$$z = \sum_i \bar{w}_i z_i = O_i^5 = \frac{\sum_i w_i z_i}{\sum_i w_i} = \frac{w_1 z_1 + \cdots + w_9 z_9}{w_1 + \cdots + w_9} = \bar{w}_1 z_1 + \cdots + \bar{w}_9 z_9$$

3.8.2 Formulation of PSO trained LOG10DANFIS with distance and PSO generated random RSSI as inputs

Using PSO to obtain the random input together with ANFIS membership function parameters that best approximate the measured values. The minimum of the function $f(x) = ANFIS\text{-}RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \hat{y}_j\right)^2}$ for the following ANFIS with membership function parameters *x* within the range of $min \leq x \leq max$ using the PSO algorithm with *m* particles (*m* sets of random inputs and *m* sets of ANFIS membership function parameters) are obtained. The given ANFIS requires 45 parameters since it has 3 MFs with two inputs. *R* random inputs are also required since there are *R* distance points. The two result to *45+R* as the number of variables that are to be searched for by PSO. where N is the total number of data, y is actual target value, and $\hat{y}$ is estimated target value. The following represent the formulation for the same. The parameters are represented by a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3 and c_3 as the premise parameters for both inputs distance and RSSI while the consequent parameters are given by p_1 to p_9, q_1 to q_9 and r_1 to r_9 together with the *R* random inputs representing the random RSSI inputs. The other steps are like those discussed in section 3.5 where the parameter 15 is replaced by *45+R* as the number of parameters to be searched for by ANFIS.

CHAPTER FOUR

RESULTS, ANALYSIS AND DISCUSSION

4.1 Introduction

This section gives tabular and graphical representations comparing the performance of ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS for the one slope, dual-slope, multi-wall, average-wall, COST231, COST231-Hata, Hata-Okumura, one slope with random input and Two-ray ground reflection models. Their performance regarding the measured RSSI is also done as well as the performance of LOG10D-PSO-R-ANFIS practical model where a random variable is introduced at the input. The comparison is done based on 100 and 1000 data-points where considerable error difference is noted for the cases. The RMSE, ME and SD are the main selected parameters used to show the differences where the RMSE values show clear differences. The parameters which define the ANFIS based models are also tabulated in different tables for each model. For 100 data points the LOG10D-ANFIS generally performs slightly better than LOG10D-PSO-ANFIS whereas for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors which are both in the range of exponent -14 as compared to exponent -7 for 100 data points. This is because a higher number of data points results to a smoother curve effectively leading to lower errors based on Nyquist sampling theorem. ANFIS error generally increases with increase in number of data points since ANFIS is based on linear modelling where the output is represented using a straight-line equation y=mx+c that is z=px+r translating to RSSI=p*distance+r for a single input. The separation between any two points is linear for lower data points which results to lower errors. This applies to all the analyzed models except the dual slope model which has breakpoint that interferes with the smoothness of the curve. The ANFIS parameters generated for each model are used to obtain the universal model based on the modified ANFIS trained using the PSO algorithm (LOG10D-PSO-ANFIS). The performance of LOG10D-PSO-ANFIS universal theoretical model is compared with that of radial basis function (RBF) neural network model trained with particle swarm optimization (PSO) algorithm, MLP-NN and RBF-NN models where it is found to outperform these other models.

4.2 Performance comparison for one slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

This section gives the results generated for the theoretical one slope model in relation to the ordinary ANFIS, LOG10DANFIS being the modified ANFIS and LOG10D-PSO-ANFIS also based on the modified ANFIS with 6 layers.

Table 4.1: Training performance comparison between One Slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.1952	0.1672	0.1014
LOG10D-ANFIS	2.6987e-05	2.2827e-05	1.4503e-05
LOG10D-PSO-ANFIS	2.7207e-05	2.6931e-05	1.9740e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	0.3180	0.2315	0.2183
LOG10D-ANFIS	1.1652e-07	7.6010e-08	8.8374e-08
LOG10D-PSO-ANFIS	4.7379e-15	2.9535e-15	3.7074e-15

Table 4.2: Testing performance comparison between One Slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.2933	0.1970	0.2207
LOG10D-ANFIS	2.5055e-05	2.0111e-05	1.5175e-05
LOG10D-PSO-ANFIS	2.2893e-05	2.7863e-05	1.7432e-05

b) 1000 data points

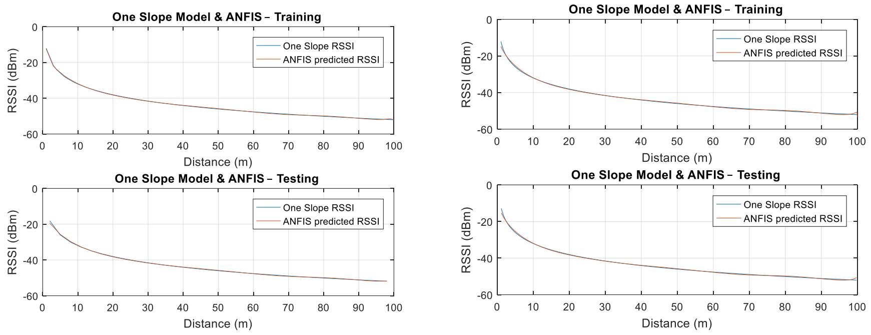
	RMSE	**ME**	**SD**
ANFIS	0.3146	0.2302	0.2148
LOG10D-ANFIS	1.1637e-07	7.5947e-08	8.8309e-08
LOG10D-PSO-ANFIS	4.7169e-15	2.9126e-15	3.7158e-15

Table 4.3: One Slope LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-30.4042	-44.4841	-17.1511	-39.8081	-19.3858
	-21.2635	-16.4371	-23.2478	-20.0000	-12.0442
	-26.6433	-33.4410	-12.9429	-38.4691	-22.6328

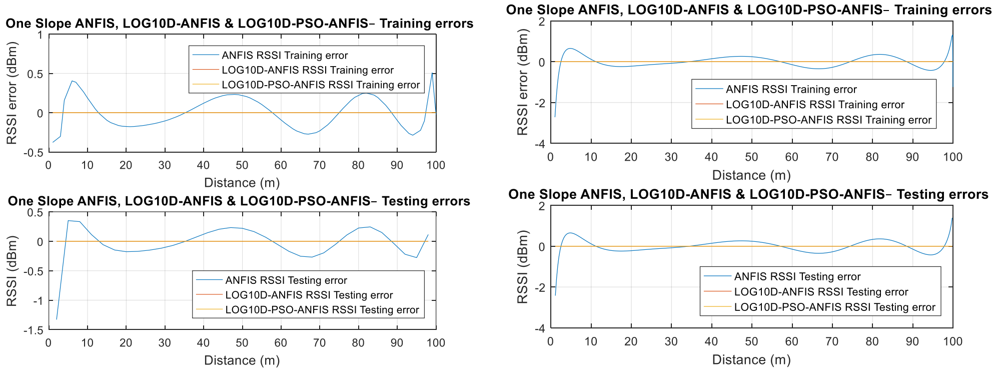
For 100 data points the LOG10D-ANFIS performs slightly better for training and lower for testing than LOG10D-PSO-ANFIS with training RMSE of 2.6987e-05 and

2.7207e-05 and testing RMSE of 2.5055e-05 and 2.2893e-05 respectively whereas for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 1.1652e-07 and 4.7379e-15 and testing RMSE as 1.1637e-07 and 4.7169e-15 respectively. Both LOG10D-ANFIS and LOG10D-PSO-ANFIS perform far better than ordinary ANFIS which has training RMSE of 0.1952 and testing RMSE of 0.2933 for 100 data points and 0.3180 and 0.3146 for 1000 data points as shown in tables 4.1 and 4.2. Table 4.3 shows the One Slope D LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figures 4.1 to 4.3.



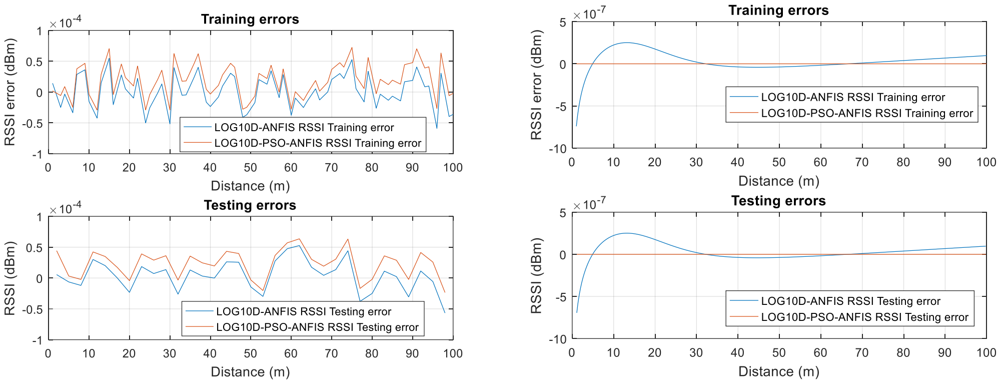
(a) 100 data points **(b) 1000 data points**

Figure 4.1: OSM and ANFIS training and testing



(a) 100 data points **(b) 1000 data points**

Figure 4.2: OSM ANFIS, LOG10D-ANFIS & LOG10D-PSO-ANFIS training and testing errors



(a) 100 data points **(b) 1000 data points**

Figure 4.3: OSM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.1 shows the training and testing comparison between the one slope model and the ordinary ANFIS where it can be noted that the two have a difference in terms of their RSSI being an indication that the ordinary ANFIS does not perform well in approximating the one slope model. Figure 4.2 is about the graphical comparison between the training and testing errors of the one slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS both of which are based on the modified ANFIS. It is noted that the two models that are based on LOG10D outperform the ordinary ANFIS since the two appear to be at a constant value zero while the ones for ANFIS vary from -4 to 2 dBm. Figure 4.3 shows the relation between the one slope LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where the latter performs much better compared to the former since its errors are much lower for both 100 and 1000 data points with the ones for 1000 data points tending to zero as indicated in figure 4.3 (b).

4.3 Performance comparison for dual Slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

Table 4.4: Training performance comparison between Dual Slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.1855	0.1577	0.0984
LOG10D-ANFIS	3.1707e-05	2.6958e-05	1.6816e-05
LOG10D-PSO-ANFIS	0.0397	0.0306	0.0256

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	0.1070	0.0809	0.0702
LOG10D-ANFIS	9.0216e-06	6.1048e-06	6.6473e-06
LOG10D-PSO-ANFIS	0.0578	0.0470	0.0337

Table 4.5: Testing performance comparison between Dual Slope ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.2569	0.1794	0.1867
LOG10D-ANFIS	3.0562e-05	2.6341e-05	1.5739e-05
LOG10D-PSO-ANFIS	0.0407	0.0310	0.0267

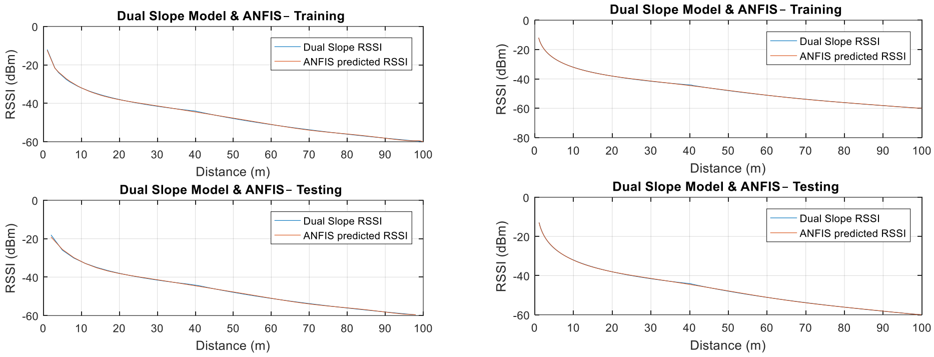
b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	0.1062	0.0804	0.0695
LOG10D-ANFIS	9.1051e-06	6.1125e-06	6.7585e-06
LOG10D-PSO-ANFIS	0.0577	0.0469	0.0336

Table 4.6: Dual Slope LOG10D-ANFIS RSSI prediction models premise and consequent parameters after training

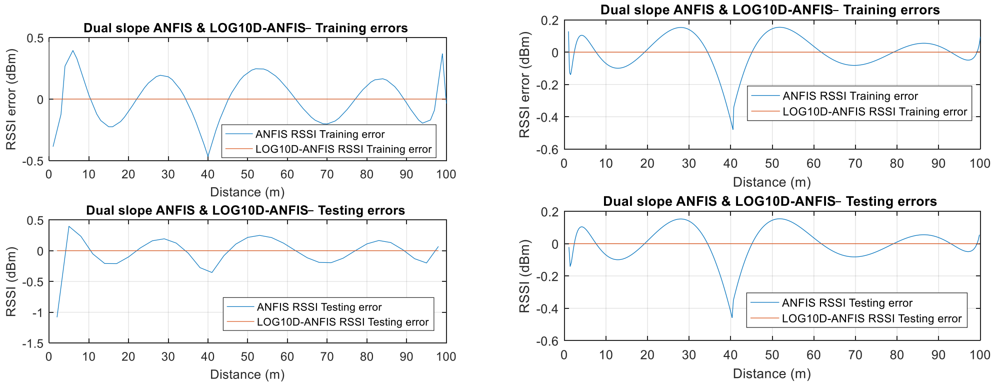
	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-ANFIS	0.1319	3.029	0.652	-20	-12.04
	0.1214	3.242	1.391	-20	-12.04
	0.2256	4.11	3.572	-20	20

In the case of DSM, 100 data points, the LOG10D-ANFIS performs quite well for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 3.1707e-05 and 0.0397 and testing RMSE of 3.0562e-05 and 0.0407 respectively whereas for 1000 data points LOG10D- ANFIS still performs better than LOG10D-PSO-ANFIS in both training and testing errors with training RMSE given as 9.0216e-06 and 0.0578 and testing RMSE as 9.1051e-06 and 0.0577 respectively. Both LOG10D-ANFIS and LOG10D-PSO-ANFIS perform far better than ordinary ANFIS which has training RMSE of 0.1855 and testing RMSE of 0.2569 for 100 data points and 0.1070 and 0.1062 for 1000 data points as shown in tables 4.4 and 4.5. Table 4.6 shows the Dual Slope LOG10D-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are also as shown in figures 4.4 to 4.8.



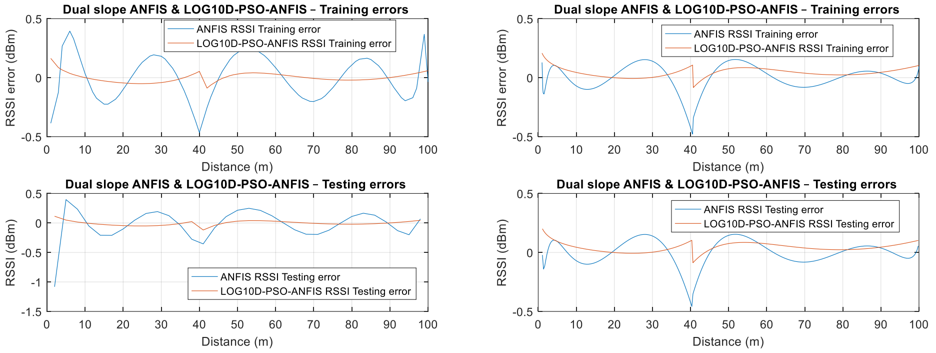
(a) 100 data points **(b) 1000 data points**

Figure 4.4: DSM and ANFIS training and testing



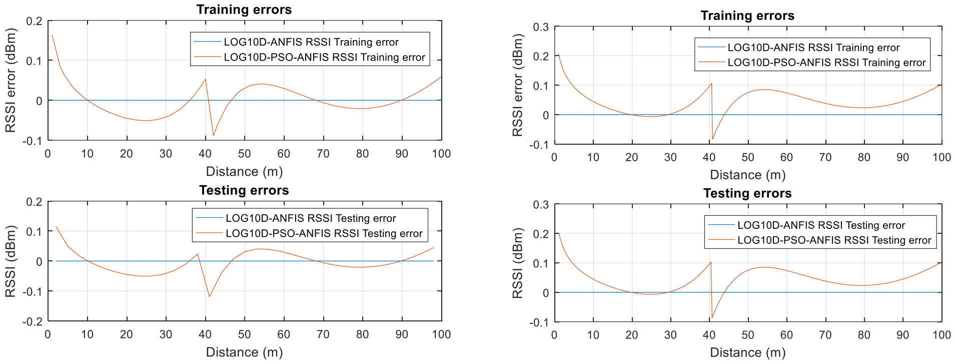
(a) 100 data points **(b) 1000 data points**

Figure 4.5: DSM ANFIS and LOG10D-ANFIS training and testing errors



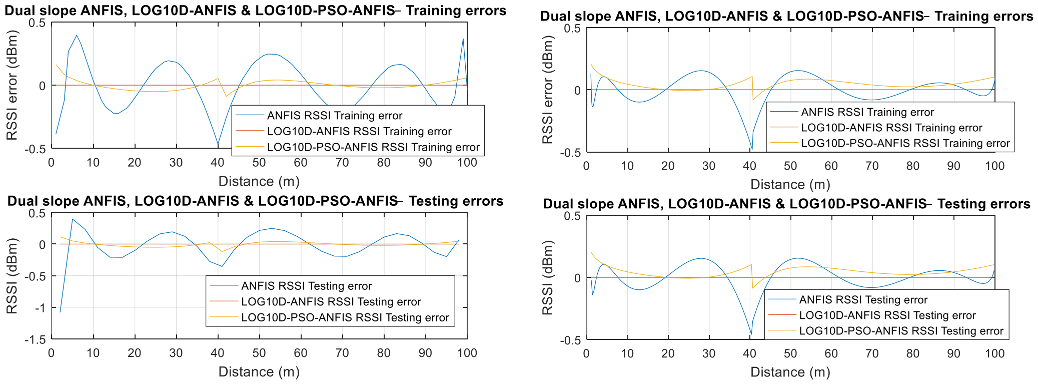
(a) 100 data points **(b) 1000 data points**

Figure 4.6: DSM ANFIS and LOG10D-PSO-ANFIS training and testing errors



(a) 100 data points **(b) 1000 data points**

Figure 4.7: DSM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors



(a) 100 data points **(b) 1000 data points**

Figure 4.8: DSM ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

For the dual slope model, figure 4.4 shows the relation between the training and testing performance the dual slope and ordinary ANFIS where it can be noted that the two also have a difference in terms of their RSSI being an indication that the ordinary ANFIS does not perform well in approximating the dual slope model. Figure 4.5 shows the graphical relation between the training and testing errors of the dual slope ANFIS and LOG10D-ANFIS while figure 4.6 shows the relation between the one slope ANFIS and LOG10D-PSO-ANFIS training and testing errors. Figure 4.7 shows the relation between the one slope LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors whereas figure 4.8 shows the comparison performance of all the three models. From these figures it can be noted that both the LOG10D based models perform much better compared to the ordinary ANFIS since their errors are much lower for both 100 and 1000 data points as indicated in figure 4.8 where the three are compared though the LOG10D-ANFIS performs better than the others because of the breakpoint that separates the two slopes.

4.4 Performance comparison for multi-wall ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

Table 4.7: Training performance comparison between Multi-Wall ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	ME	SD
ANFIS	0.1947	0.1669	0.1012
LOG10D-ANFIS	2.7580e-05	2.3322e-05	1.4834e-05
LOG10D-PSO-ANFIS	9.9441e-05	9.4695e-05	3.0583e-05

b) 1000 data points

	RMSE	ME	SD
ANFIS	0.3180	0.2315	0.2183
LOG10D-ANFIS	7.2909e-08	4.8753e-08	5.4252e-08
LOG10D-PSO-ANFIS	4.8225e-15	3.2757e-15	3.5419e-15

Table 4.8: Testing performance comparison between Multi-Wall ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.2940	0.1969	0.2218
LOG10D-ANFIS	2.6610e-05	2.3239e-05	1.3164e-05
LOG10D-PSO-ANFIS	1.0427e-04	1.0087e-04	2.6804e-05

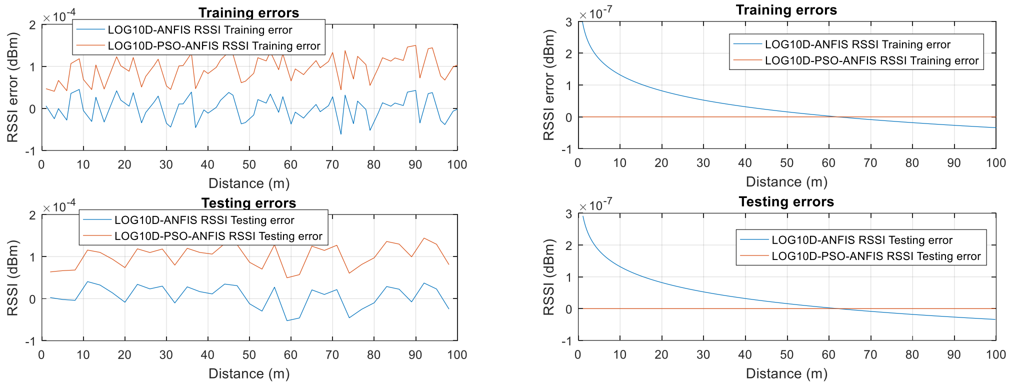
b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	0.3146	0.2302	0.2147
LOG10D-ANFIS	7.2936e-08	4.8771e-08	5.4314e-08
LOG10D-PSO-ANFIS	5.0731e-15	3.5954e-15	3.5844e-15

Table 4.9: Multi-Wall LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-40.3386	-38.6915	-22.2768	-50.2905	-32.6188
	-15.7458	-34.2936	-5.2165	-27.1827	-28.2698
	-42.9084	-53.0827	-44.0994	-20.0000	-23.0442

Based on the MWM, for 100 data points, it is noted that the LOG10D-ANFIS performs slightly better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 2.7580e-05 and 9.9441e-05 and testing RMSE of 2.6610e-05 and 1.0427e-04 respectively while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 7.2909e-08 and 4.8225e-15 and testing RMSE as 7.2936e-08 and 5.0731e-15 respectively as shown in tables 4.7 and 4.8. Table 4.9 shows the Multi-Wall LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figure 4.9.



(a) 100 data points **(b) 1000 data points**

Figure 4.9: MWM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.9 shows the relation between the multiwall LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where the former performs much better compared to the latter for the 100 data points since its errors are much lower. For 1000 data points it is the opposite where the PSO based model does extremely well compared to LOG10D-ANFIS since the errors are tending to zero as indicated in figure 4.9 (b). This is because there is a smooth relation between the RSSI and distance for this model. All the other relations are similar to the ones for the one slope-based models in figures 4.1-4.3.

4.5 Performance comparison for Okomura Hata ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

4.5.1 Rural

Table 4.10: Training performance comparison between Hata-Okumura Rural ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.6725	0.4532	0.5005
LOG10D-ANFIS	2.5953e-05	2.1947e-05	1.3957e-05
LOG10D-PSO-ANFIS	3.3272e-05	2.7576e-05	1.8757e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.8957	0.8877	1.6762
LOG10D-ANFIS	7.7492e-07	5.6225e-07	5.3368e-07
LOG10D-PSO-ANFIS	1.0939e-14	4.3357e-15	1.0051e-14

Table 4.11: Testing performance comparison between Hata-Okumura Rural ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.6998	0.4489	0.5452
LOG10D-ANFIS	2.7269e-05	2.3418e-05	1.4187e-05
LOG10D-PSO-ANFIS	3.4549e-05	2.7603e-05	2.1100e-05

b) 1000 data points

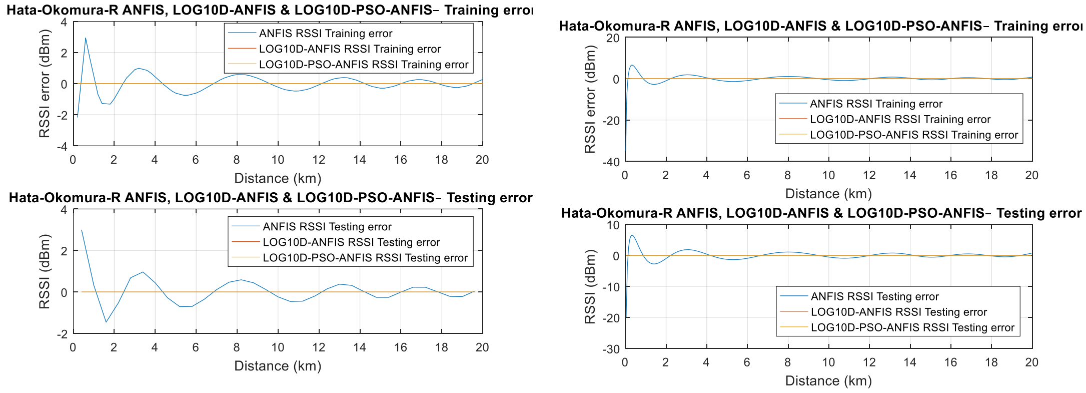
	RMSE	ME	SD
ANFIS	1.6473	0.8767	1.3967
LOG10D-ANFIS	7.7502e-07	5.6237e-07	5.3409e-07
LOG10D-PSO-ANFIS	1.0124e-14	3.7127e-15	9.4326e-15

Table 4.12: Hata-Okumura Rural LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-163.788	-60.7698	-35.9215	-201.0000	-100.424
	-136.453	-140.107	-140.251	-56.0283	-128.388
	-142.263	-69.0993	-55.5749	-46.0372	-200.771

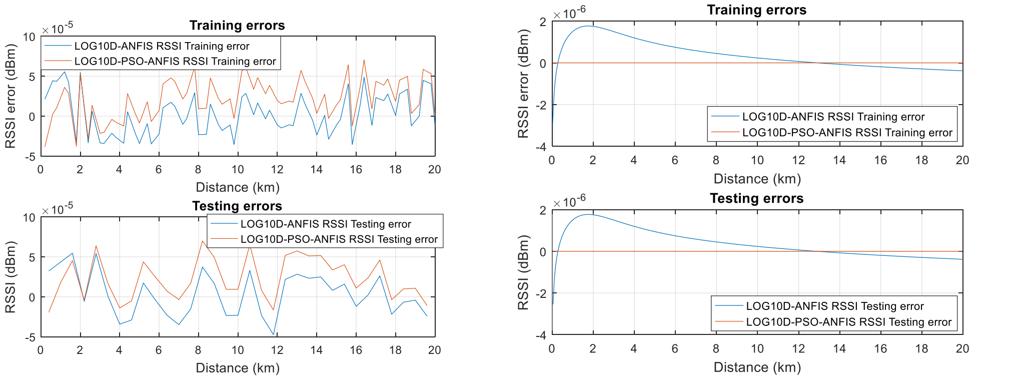
With the HORM, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of

2.5953e-05 and 3.3272e-05 and testing RMSE of 2.7269e-05 and 3.4549e-05 respectively than the plain ANFIS while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 7.7492e-07 and 1.0939e-14 and testing RMSE as 7.7502e-07 and 1.0124e-14 respectively as shown in tables 4.10 and 4.11. Table 4.12 shows the Hata-Okumura rural LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figures 4.10 and 4.11.



(a) 100 data points **(b) 1000 data points**

Figure 1 Figure 4.10: HORM ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors



(a) 100 data points **(b) 1000 data points**

Figure 4.11: HORM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.10 shows the graphical contrast between the training and testing errors of the Hata-Okomura rural ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS with the last two based on the modified ANFIS. From the relations it is noted that the two models that are based on the LOG10D outperform the ordinary ANFIS since the two appear to be at a constant value zero while the one for ANFIS vary from -3 to 2 dBm. Figure 4.11 shows the comparison between the Hata-Okomura rural LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where the PSO based one outperforms the former with 1000 data points since its errors are much lower tending to zero as indicated in figure 4.11 (b). This is not case with 100 data points since PSO seems to perform better for smoother relations with higher sampling frequency.

4.5.2 Suburban

Table 4.13: Training performance comparison between Hata-Okumura Sub-Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	ME	SD
ANFIS	0.6725	0.4534	0.5005
LOG10D-ANFIS	2.6685e-05	2.0875e-05	1.6747e-05
LOG10D-PSO-ANFIS	2.9536e-05	2.3413e-05	1.8142e-05

b) 1000 data points

	RMSE	ME	SD
ANFIS	1.8957	0.8877	1.6762
LOG10D-ANFIS	3.7399e-07	2.4694e-07	2.8109e-07
LOG10D-PSO-ANFIS	1.3866e-14	6.6261e-15	1.2189e-14

Table 4.14: Testing performance comparison between Hata-Okumura Sub-Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.7000	0.4491	0.5453
LOG10D-ANFIS	3.2954e-05	2.6878e-05	1.9361e-05
LOG10D-PSO-ANFIS	3.3516e-05	2.7206e-05	1.9878e-05

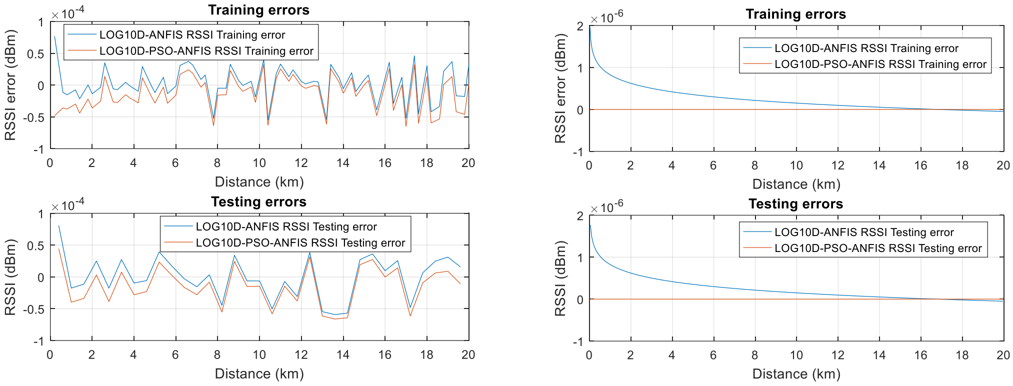
b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.6473	0.8767	1.3967
LOG10D-ANFIS	3.7339e-07	2.4707e-07	2.8037e-07
LOG10D-PSO-ANFIS	1.4631e-14	7.2121e-15	1.2750e-14

Table 4.15: Hata-Okumura Sub-Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-153.995	-142.231	-82.0268	-176.1464	-89.6252
	-93.7246	-69.1132	-169.692	-32.9347	-144.297
	-105.704	-122.405	-181.440	-79.1218	-174.266

For the HOSUM, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 2.6685e-05 and 2.9536e-05 and testing RMSE of 3.2954e-05 and 3.3516e-05 respectively than the plain ANFIS while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 3.7399e-07 and 1.3866e-14 and testing RMSE as 3.7339e-07 and 1.4631e-14 respectively as shown in tables 4.13 and 4.14. Table 4.15 shows the Hata-Okumura sub-urban LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figure 4.12.



(a) 100 data points **(b) 1000 data points**

Figure 4.12: HOSUM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.12 shows the comparison between the Hata-Okomura suburban LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where the LOG10D-ANFIS outperforms the one that is based on PSO for 100 data points. This is quite different when it comes to 1000 data points since its errors are much lower where they tend to zero as indicated in figure 4.12 (b) where the LOG10D-PSO-ANFIS excels in performance due to the smooth relations.

4.5.3 Urban

Table 4.16: Training performance comparison between Hata-Okumura Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.6725	0.4534	0.5005
LOG10D-ANFIS	2.6128e-05	2.1238e-05	1.5334e-05
LOG10D-PSO-ANFIS	2.7672e-05	2.2531e-05	1.6186e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.8957	0.8877	1.6762
LOG10D-ANFIS	3.8101e-07	2.5164e-07	2.8629e-07
LOG10D-PSO-ANFIS	1.4359e-14	6.9243e-15	1.2589e-14

Table 4.17: Testing performance comparison between Hata-Okumura Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	ME	SD
ANFIS	0.7000	0.4491	0.5453
LOG10D-ANFIS	2.6990e-05	2.3077e-05	1.4214e-05
LOG10D-PSO-ANFIS	2.6146e-05	2.2886e-05	1.2840e-05

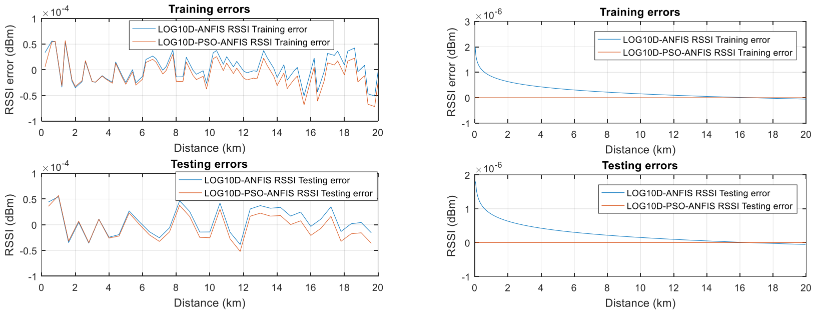
b) 1000 data points

	RMSE	ME	SD
ANFIS	1.6473	0.8767	1.3967
LOG10D-ANFIS	3.8039e-07	2.5177e-07	2.8558e-07
LOG10D-PSO-ANFIS	1.4296e-14	6.7000e-15	1.2648e-14

Table 4.18: Hata-Okumura Urban LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-106.865	-233.777	-41.7293	-234.0000	-69.3173
	1.3000	-224.589	-234.000	-48.2855	-145.836
	1.3000	-234.000	-234.000	-63.7711	-177.09

For the HOUM, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 2.6128e-05 and 2.7672e-05 and testing RMSE of 2.6990e-05 and 2.6146e-05 respectively than the plain ANFIS with training RMSE of 0.6725 and testing RMSE of 0.7000 while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 3.8101e-07 and 1.4359e-14 and testing RMSE as 3.8039e-07 and 1.4296e-14 respectively as compared to the plain ANFIS with training RMSE of 1.8957 and testing RMSE of 1.6473 as shown in tables 4.16 and 4.17. Table 18 shows the Hata-Okumura urban LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figures 4.13.



(a) 100 data points **(b) 1000 data points**

Figure 4.13: HOUM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.13 shows the comparison between the Hata-Okomura urban LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where both have a similar performance for 100 data points. This changes when it comes to 1000 data points since its errors are much closer to zero as indicated in figure 4.13 (b) where the LOG10D-PSO-ANFIS performs very well with up to 10^{-14} RMSE values due to the smooth relations.

4.6 Performance comparison for COST231-Hata ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

4.6.1 Metropolitan

Table 4.19: Training performance comparison between COST231-HATA Metropolitan ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.6725	0.4536	0.5002
LOG10D-ANFIS	3.0203e-05	2.6627e-05	1.4362e-05
LOG10D-PSO-ANFIS	2.9445e-05	2.5468e-05	1.4890e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.8957	0.8877	1.6762
LOG10D-ANFIS	8.2930e-07	6.5268e-07	5.1200e-07
LOG10D-PSO-ANFIS	1.0726e-14	4.0907e-15	9.9231e-15

Table 4.20: Testing performance comparison between COST231-HATA Metropolitan ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.7002	0.4491	0.5455
LOG10D-ANFIS	2.9550e-05	2.6440e-05	1.3399e-05
LOG10D-PSO-ANFIS	2.9513e-05	2.5601e-05	1.4910e-05

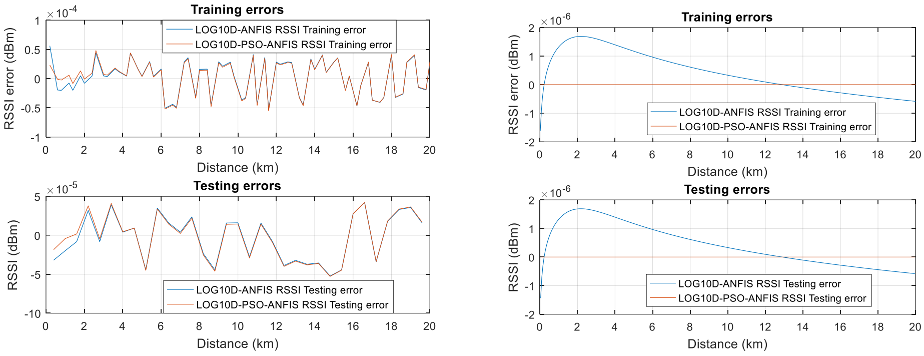
b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.6473	0.8768	1.3967
LOG10D-ANFIS	8.2970e-07	6.5274e-07	5.1295e-07
LOG10D-PSO-ANFIS	1.2270e-14	5.3984e-15	1.1035e-14

Table 4.21: COST231-HATA Metropolitan ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-01.9583	-43.1031	-62.3113	-83.4139	-89.9228
	-91.8924	-95.8520	-40.8291	-64.1855	-23.7094
	-81.9529	-45.8042	-64.2723	-47.8710	-53.6633

For the COST231-Hata Metropolitan model, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 3.0203e-05 and 2.9445e-05 and testing RMSE of 2.9550e-05 and 2.9513e-05 respectively than the plain ANFIS with training RMSE of 0.6725 and testing RMSE of 0.7002 while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 8.2930e-07 and 1.0726e-14 and testing RMSE as 8.2970e-07 and 1.2270e-14 respectively as compared to the plain ANFIS with training RMSE of 1.8957 and testing RMSE of 1.6473 as shown in tables 4.19 and 4.20. Table 4.21 shows the COST231-Hata Metropolitan LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figure 4.14.



(a) 100 data points **(b) 1000 data points**

Figure 4.14: CHMM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.14 shows the comparison between the COST-231-Hata metropolitan LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where both also have largely similar performance for 100 data points. With 1000 data points the LOG10D-PSO-ANFIS performs very well with up to 10^{-14} RMSE values due to the smooth relations since the errors are much closer to zero as indicated in figure 4.13 (b).

4.6.2 Sub-Urban

Table 4.22: Training performance comparison between COST231-HATA Sub-Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.6725	0.4532	0.5005
LOG10D-ANFIS	3.0192e-05	2.6615e-05	1.4363e-05
LOG10D-PSO-ANFIS	6.8745e-05	6.0724e-05	3.2469e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.8957	0.8877	1.6762
LOG10D-ANFIS	8.1030e-07	6.2711e-07	5.1354e-07
LOG10D-PSO-ANFIS	1.0684e-14	4.0694e-15	9.8859e-15

Table 4.23: Testing performance comparison between COST231-HATA Sub-Urban ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) **100 data points**

	RMSE	ME	SD
ANFIS	0.6997	0.4489	0.5451
LOG10D-ANFIS	2.9560e-05	2.6454e-05	1.3394e-05
LOG10D-PSO-ANFIS	7.2501e-05	6.5123e-05	3.2358e-05

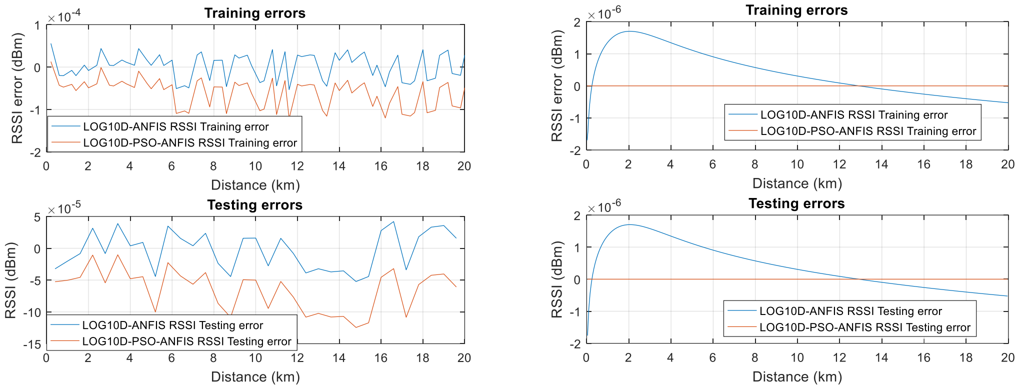
b) **1000 data points**

	RMSE	ME	SD
ANFIS	1.6473	0.8768	1.3967
LOG10D-ANFIS	8.1165e-07	6.2781e-07	5.1520e-07
LOG10D-PSO-ANFIS	1.2338e-14	5.4198e-15	1.1100e-14

Table 4.24: COST231-HATA Sub-Urban LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	-64.1418	-82.3528	-83.4471	-10.0000	-52.3375
	-57.3453	-77.0805	-60.0873	-71.3520	-29.1993
	0.9054	-12.2142	-61.3340	-40.7046	-44.8518

In the case of the COST231-Hata Urban model, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 3.0192e-05 and 6.8745e-05 and testing RMSE of 2.9560e-05 and 7.2501e-05 respectively than the plain ANFIS with training RMSE of 0.6725 and testing RMSE of 0.6997 while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 8.1030e-07 and 1.0684e-14 and testing RMSE as 8.1165e-07 and 1.2338e-14 respectively as compared to the plain ANFIS with training RMSE of 1.8957 and testing RMSE of 1.6473 as shown in tables 4.22 and 4.23. Table 4.24 shows the COST231-Hata Urban LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figure 4.15.



(a) 100 data points **(b) 1000 data points**

Figure 4.15: CHSUM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.15 shows the comparison between the COST-231-Hata urban LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where both also have similar performance as those of COST-231-Hata metropolitan with slight difference in the graphical relation values which are more pronounced for the 100 data points as shown in the figure.

4.7 Performance comparison for COST231 ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

Table 4.25: Training performance comparison between COST231 ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.4562	0.3078	0.3392
LOG10D-ANFIS	3.1176e-05	2.7649e-05	1.4512e-05
LOG10D-PSO-ANFIS	3.1405e-05	2.8400e-05	1.3507e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	1.2857	0.6022	1.1368
LOG10D-ANFIS	4.1078e-07	2.6253e-07	3.1617e-07
LOG10D-PSO-ANFIS	2.1424e-14	1.4999e-14	1.5309e-14

Table 4.26: Testing performance comparison between COST231 ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	ME	SD
ANFIS	0.4751	0.3048	0.3700
LOG10D-ANFIS	3.0955e-05	2.5939e-05	1.7155e-05
LOG10D-PSO-ANFIS	2.9206e-05	2.4961e-05	1.5399e-05

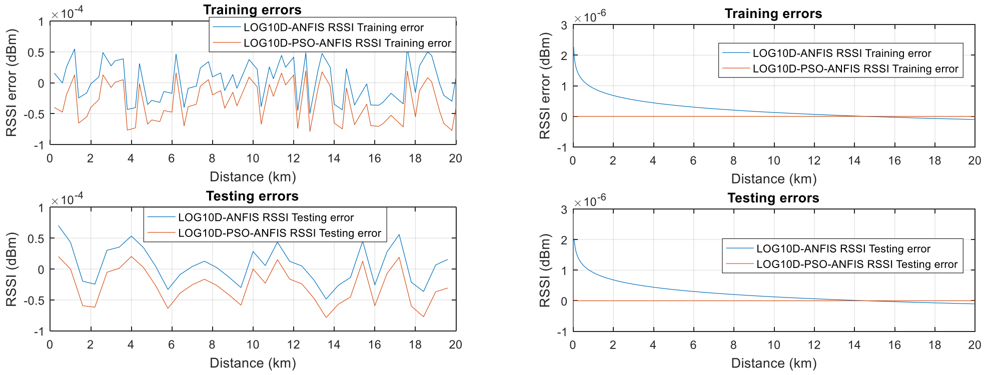
b) 1000 data points

	RMSE	ME	SD
ANFIS	1.1173	0.5948	0.9472
LOG10D-ANFIS	4.0997e-07	2.6258e-07	3.1533e-07
LOG10D-PSO-ANFIS	2.1874e-14	1.5064e-14	1.5884e-14

Table 4.27: COST231 LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			Consequent	
	a	b	c	p	r
LOG10D-PSO-ANFIS	-40.9631	-44.9048	-20.3709	-11.9749	-68.9166
	74.5600	16.4457	-04.7000	-47.2513	-33.6719
	-11.8412	-43.7937	-38.2236	-38.0000	-55.2745

For the COST231 model, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 3.1176e-05 and 3.1405e-05 and testing RMSE of 3.0955e-05 and 2.9206e-05 respectively than the plain ANFIS with training RMSE of 0.4562 and testing RMSE of 0.4751 while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 4.1078e-07 and 2.1424e-14 and testing RMSE as 4.0997e-07 and 2.1874e-14 respectively as compared to the plain ANFIS with training RMSE of 1.2857 and testing RMSE of 1.1173 as shown in tables 4.25 and 4.26. Table 4.27 shows the COST231 LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figure 4.16.



(a) 100 data points **(b) 1000 data points**

Figure 4.16: CM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.16 is a graphical comparison between the COST-231 LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where the LOG10D-ANFIS performs better than LOG10D-PSO-ANFIS for 100 data points while the opposite is true when considering 1000 data points with the LOG10D-PSO-ANFIS errors tending to zero as shown in figure 4.16 (b).

4.9 Performance comparison for two-ray ground reflection ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS models

Table 4.28: Training performance comparison between Two-Ray Ground Reflection ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.3906	0.3347	0.2030
LOG10D-ANFIS	3.1211e-05	2.6005e-05	1.7390e-05
LOG10D-PSO-ANFIS	2.9790e-05	2.4808e-05	1.6616e-05

b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	0.6360	0.4629	0.4365
LOG10D-ANFIS	4.7600e-06	2.7336e-06	3.8997e-06
LOG10D-PSO-ANFIS	2.2234e-15	7.3238e-16	2.1009e-15

Table 4.29: Testing performance comparison between Two-Ray Ground Reflection ANFIS, LOG10D-ANFIS and LOG10D-PSO-ANFIS RSSI prediction models

a) 100 data points

	RMSE	**ME**	**SD**
ANFIS	0.5864	0.3940	0.4409
LOG10D-ANFIS	3.1912e-05	2.7003e-05	1.7271e-05
LOG10D-PSO-ANFIS	3.0309e-05	2.6605e-05	1.4746e-05

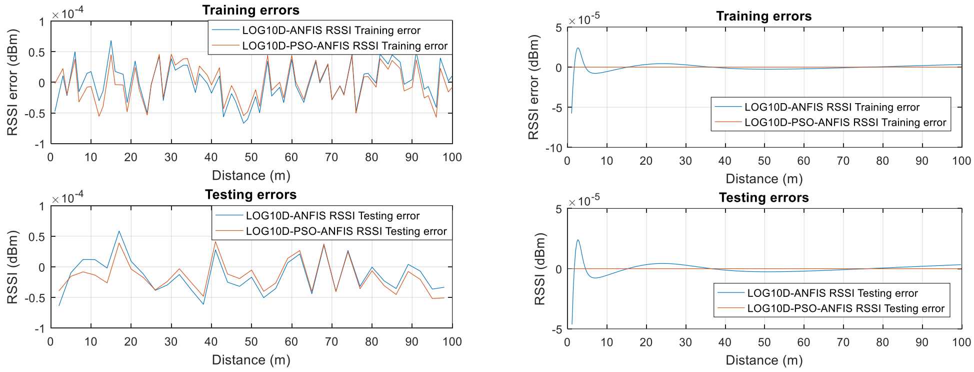
b) 1000 data points

	RMSE	**ME**	**SD**
ANFIS	0.6292	0.4604	0.4295
LOG10D-ANFIS	4.6777e-06	2.7183e-06	3.8125e-06
LOG10D-PSO-ANFIS	2.0471e-15	6.1079e-16	1.9568e-15

Table 4.30: Two-Ray Ground Reflection LOG10D-PSO-ANFIS RSSI prediction model premise and consequent parameters after training

	Premise			**Consequent**	
	a	**b**	**c**	**p**	**r**
LOG10D-PSO-ANFIS	6.278	-15.85	50.32	-40	-6.023
	2.22	26.36	-68.85	-6.172	-16.83
	47.52	55.66	68.56	-36.82	35.28

For the Two-Ray Ground Reflection model, 100 data points, it is noted that the LOG10D-ANFIS performs better for both training and testing than LOG10D-PSO-ANFIS with training RMSE of 3.1211e-05 and 2.9790e-05 and testing RMSE of 3.1912e-05 and 3.0309e-05 respectively than the plain ANFIS with training RMSE of 0.3906 and testing RMSE of 0.5864 while for 1000 data points LOG10D-PSO-ANFIS performs so well than LOG10D-ANFIS in both training and testing errors with training RMSE given as 4.7600e-06 and 2.2234e-15 and testing RMSE as 4.6777e-06 and 2.0471e-15 respectively as compared to the plain ANFIS with training RMSE of 0.6360 and testing RMSE of 0.6292 as shown in tables 4.28 and 4.29. Table 4.30 shows the Two-Ray Ground Reflection LOG10D-PSO-ANFIS RSSI prediction model premise (a, b and c) and consequent (p and r) parameters after training. The corresponding plots are as shown in figure 4.17.



(a) 100 data points **(b) 1000 data points**

Figure 4.17: TRGRM LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors

Figure 4.17 is a graphical comparison between the two-ray ground reflection LOG10D-ANFIS and LOG10D-PSO-ANFIS training and testing errors where the two have same performance for 100 data points while when considering 1000 data points the LOG10D-PSO-ANFIS errors are very low as shown in figure 4.17 (b).

4.8 Performance comparison for one slope with random input ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS models

Table 4.31: Training performance comparison between one slope with random input ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS RSSI prediction models

	RMSE	**ME**	**SD**
ANFIS	3.3022	2.8065	1.7533
LOG10D-ANFIS	3.2607	2.7060	1.8329
LOG10D-PSO-ANFIS	3.3694	2.9212	1.6918
LOG10D-PSO-R-ANFIS	0.3868	0.1827	0.3435

Table 4.32: Testing performance comparison between one slope with random input ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS RSSI prediction models

	RMSE	ME	SD
ANFIS	3.5366	2.9467	1.9860
PSO-ANFIS	3.5257	2.9733	1.9240
LOG10D-PSO-ANFIS	3.4694	2.8880	1.9525
LOG10D-PSO-R-ANFIS	3.9057	3.1629	2.3268

Table 4.33: One slope with random input LOG10D-PSO-R-ANFIS RSSI prediction models premise and consequent parameters after training

	Premise			**Consequent**		
	a	**b**	**c**	**p**	**q**	**r**
LOG10D-PSO-R-ANFIS	-8.8713	61.4673	-4.2742	-7.8577	6.9061	-1.247
	41.0087	0.9617	45.8134	48.8460	-2.4151	24.040
	-8.3527	32.5640	56.8797	67.2852	70.7261	0.4422
	47.3384	38.7837	-1.7818	16.5779	-7.3111	54.099
	3.5615	76.3629	-0.0189	5.9613	64.6412	9.5661
	71.9284	28.0197	88.4115	-3.6880	12.9965	-2.875
				38.2997	37.2093	41.260
				-6.0000	50.6034	41.699
				-5.6349	28.1616	-1.943

For the One Slope with Random Input model, LOG10D-PSO-R-ANFIS performs better for training than the other representations with training RMSE of 0.3868 and testing RMSE of 3.9057 as shown in tables 4.31 and 4.32. Table 4.33 shows the One Slope with Random Input ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS RSSI prediction models premise (a, b and c) and consequent (p, q and r) parameters after training. The corresponding plots are as shown in figures 4.18 to 4.22.

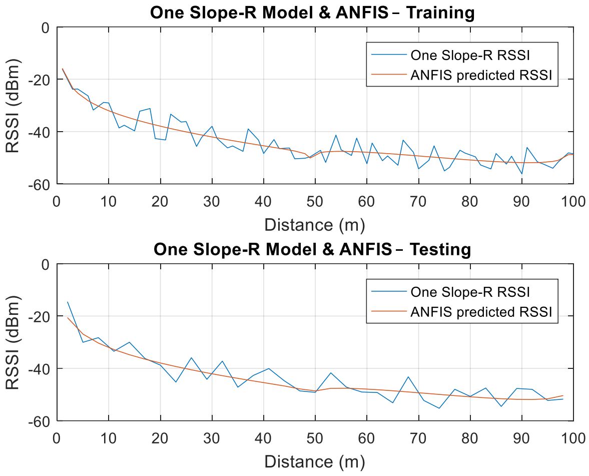


Figure 4.18: OSRM and ANFIS training and testing

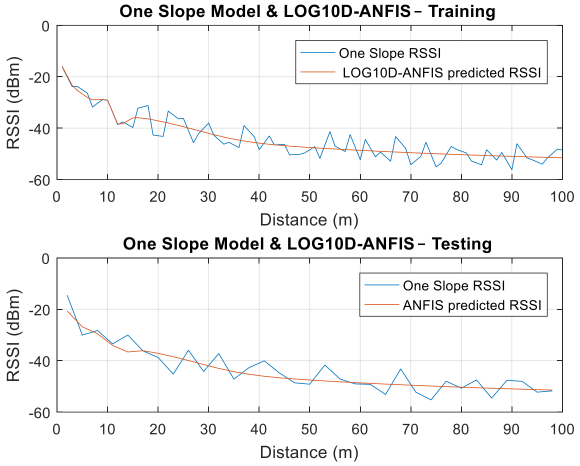


Figure 4.19: OSRM and LOG10D-ANFIS training and testing

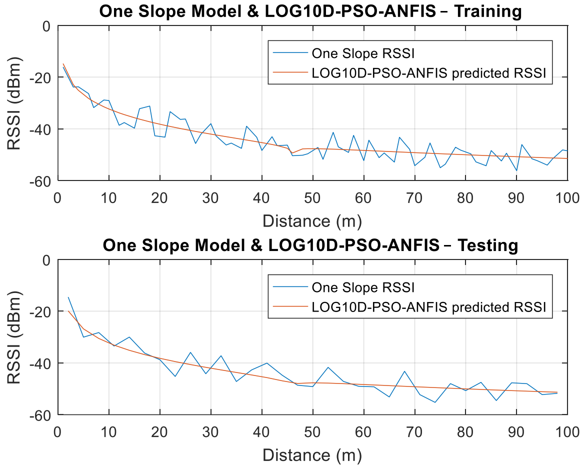


Figure 4.20: OSRM and LOG10D-PSO-ANFIS training and testing

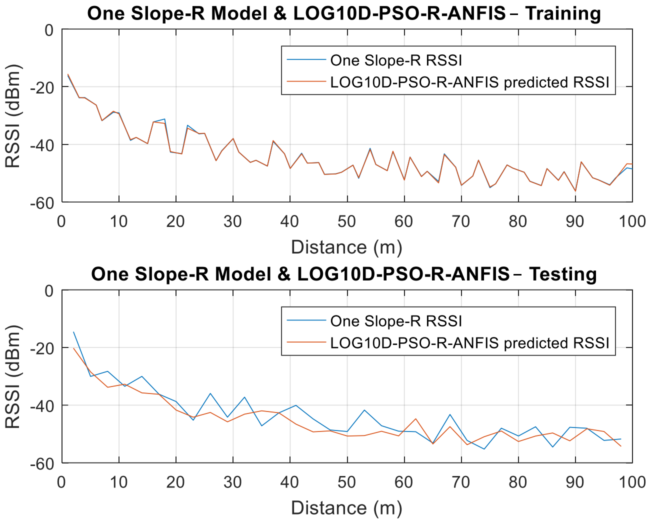


Figure 4.21: OSRM and LOG10D-PSO-R-ANFIS training and testing

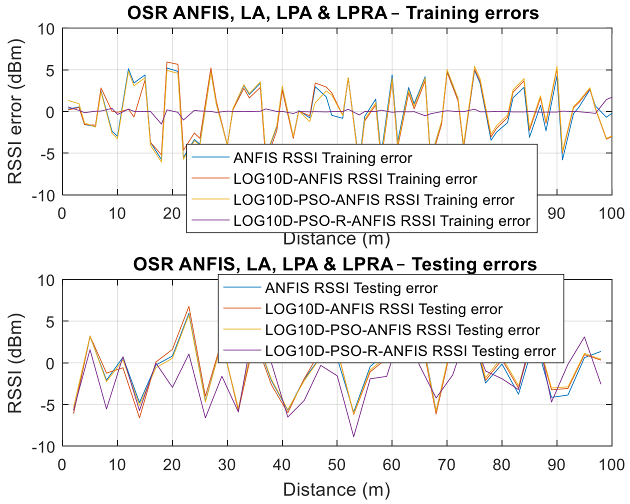


Figure 4.22: OSRM ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS training and testing errors

Figures 4.18 to 4.21 show the graphical comparison between the one slope with random input model with the other models i.e., the ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS while figure 4.22 is shows the performance relation among the one slope with random input model, ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS training and testing errors. It is noted that the LOG10D-PSO-R-ANFIS performs better for training than the other representations with training RMSE closer to zero though its testing RMSE is higher with similar levels as represented figure 4.22.

4.9 Performance of the different ANFIS based models in predicting the measured RSSI

Table 4.34: Training performance comparison between measured ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS RSSI prediction models

	RMSE	ME	SD
ANFIS	2.2825	1.7659	1.4621
LOG10D-ANFIS	2.3027	1.7910	1.4632
LOG10D-PSO-ANFIS	2.4071	1.9286	1.4563
LOG10D-PSO-R-ANFIS	0.0026	0.0016	0.0021

Table 4.35: Testing performance comparison between MEASURED ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS RSSI prediction models

	RMSE	ME	SD
ANFIS	2.9123	2.1874	1.9660
LOG10D-ANFIS	3.0325	2.3645	1.9414
LOG10D-PSO-ANFIS	3.0788	2.3341	2.0528
LOG10D-PSO-R-ANFIS	2.9310	2.4028	1.7162

Table 4.36: Measured LOG10D-PSO-R-ANFIS RSSI prediction models premise and consequent parameters after training

	Premise			**Consequent**		
	a	**b**	**c**	**p**	**q**	**r**
LOG10D-PSO-R-ANFIS	-9.514	-11.21	-45.77	-16.49	-20.6	-2.508
	4.786	-13.86	8.736	-56.94	-51.96	-34.38
	1.402	-53.72	-22.14	-10.59	-6.394	-46.22
	41.2	-10.89	-13.94	17.64	-14.71	-12.99
	-67.04	-40.82	17.28	41.2	-8.213	-27.5
	-14.02	29.88	-10.69	-42.1	-2.818	-41.41
				-34.98	-34.96	-33
				-56.42	17.44	-35.06
				15.44	6.44	-39.8

For the Measured RSSI, LOG10D-PSO-R-ANFIS performs better for both training and testing than the other representations with training RMSE of 0.0026 and testing RMSE of 2.931 as shown in tables 4.34 and 4.35. Table 4.36 shows the Measured ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS RSSI prediction models premise (a, b and c) and consequent (p, q and r) parameters after training. The corresponding plots are as shown in figures 4.23 to 4.27. Figure 4.28

represents the variation of RSSI with distance based on the PSO obtained random input RSSI limits while figure 4.29 represents the same but predicted to 150 metres. The representation shows the upper and lower limits to the predicted RSSI at any distance from 0.4 to 150m.

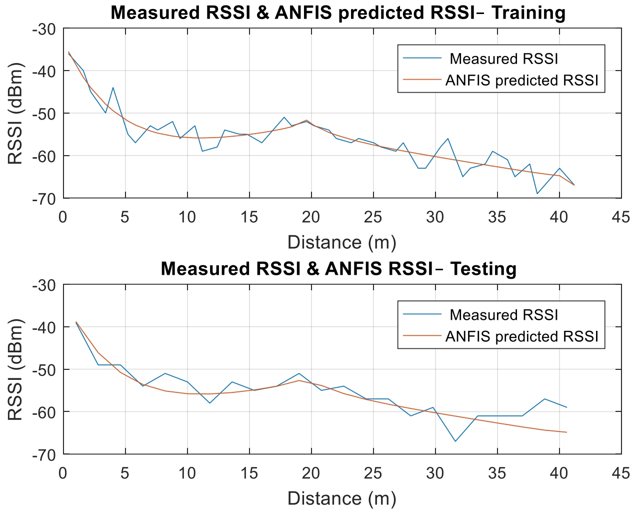


Figure 4.23: Measured and ANFIS training and testing

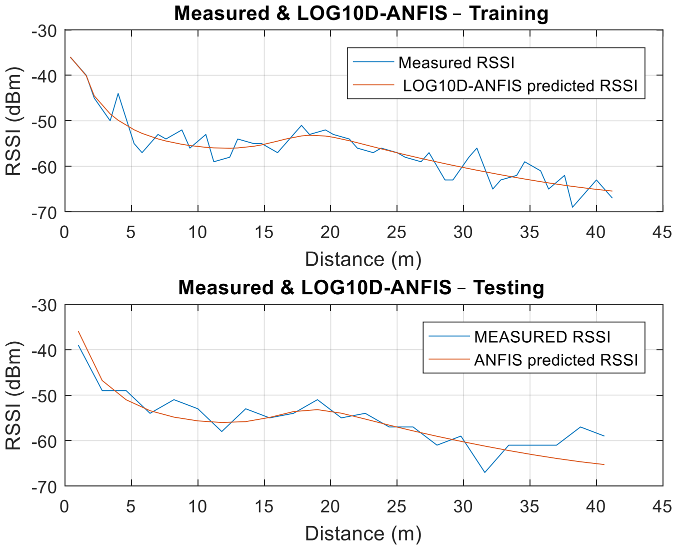


Figure 4.24: Measured and LOG10D-ANFIS training and testing

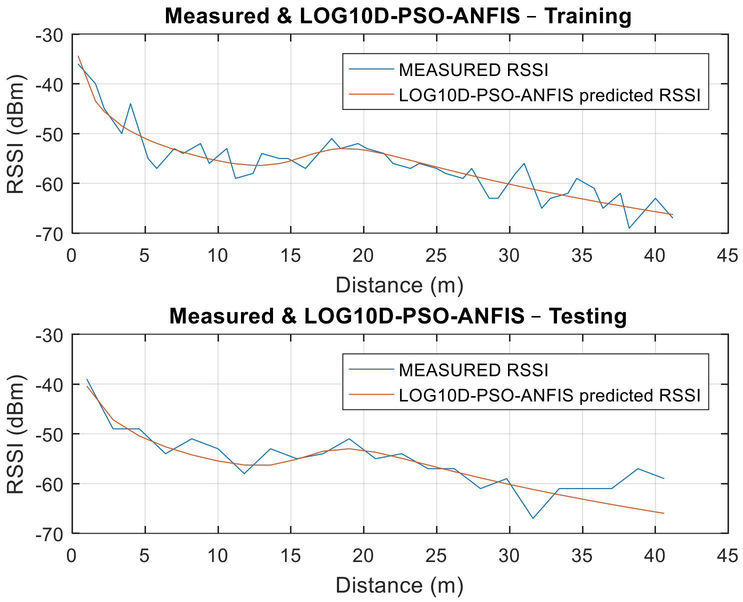


Figure 4.25: Measured and LOG10D-PSO-ANFIS training and testing

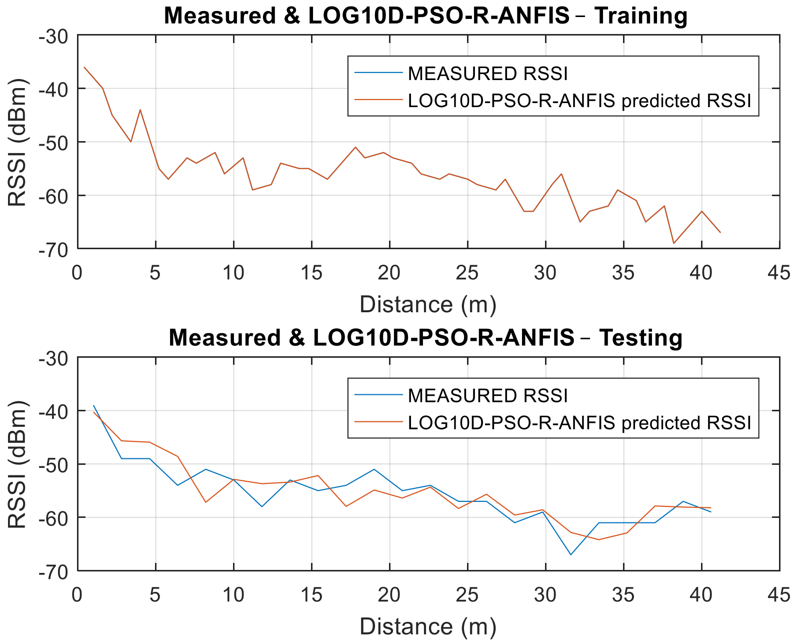


Figure 4.26: Measured and LOG10D-PSO-R-ANFIS training and testing

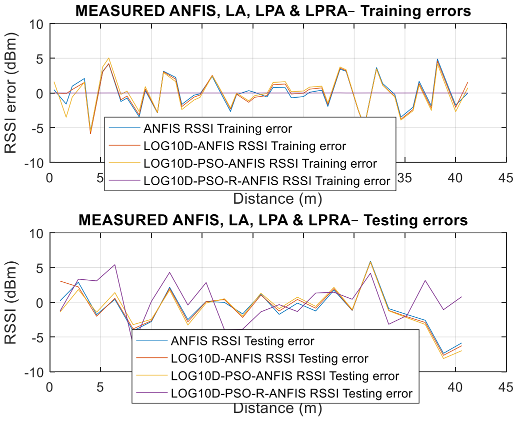


Figure 4.27: Measured ANFIS, LOG10D-ANFIS, LOG10D-PSO-ANFIS and LOG10D-PSO-R-ANFIS training and testing errors

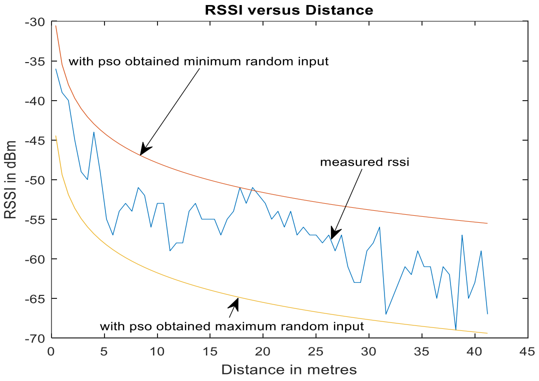


Figure 4.28: RSSI versus distance with PSO obtained random input RSSI limits

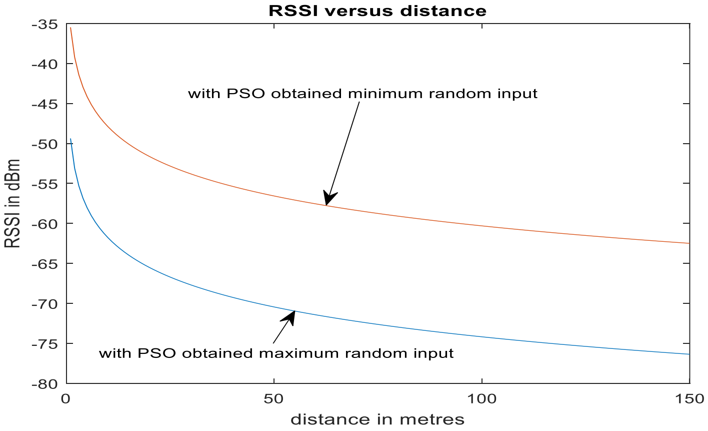


Figure 4.29: RSSI versus distance with PSO obtained random input RSSI limits to 150 metres

4.10 Performance comparison for the universal model LOG10D-PSO-ANFIS with other AI models

The performance of LOG10D-PSO-ANFIS is compared with that of radial basis function (RBF) neural network model trained with particle swarm optimization (PSO) algorithm, MLP-NN and RBF-NN models whose results are given in (Vilovic & Burum, 2011; Alotaibi et al., 2009) and (downloads.linksys.com). It is noted that the developed novel model outperforms all the other models with training and testing RMSE of 0.0026 and 2.931 against 2.245 of RBF-PSO, 2.27 and 4.23 of RBF-NN and 3.61 and 4.38 of MLP-NN as shown in tables 4.37 and 4.38 respectively.

Table 4.37: Training results for LOG10D-PSO-ANFIS, RBF-PSO trained, MLP-NN and RBF-NN models

	RMSE	ME	SD
LOG10D-PSO-ANFIS	0.0026	0.0021	0.0016
RBF-PSO	2.245	1.847	1.270
RBF-NN	2.27	1.49	1.71
MLP-NN	3.61	2.77	2.31

Table 4.38: Testing results for LOG10D-PSO-ANFIS, RBF-PSO trained, MLP-NN and RBF-NN models

	RMSE	ME	SD
LOG10D-PSO-ANFIS	2.9310	2.4028	1.7162
RBF-PSO	-	-	-
RBF-NN	4.23	3.09	2.88
MLP-NN	4.38	3.05	3.15
LOG10D-PSO-ANFIS	2.9310	2.4028	1.7162

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

The objectives of this research were achieved where the main objective to develop a novel universal wireless communication propagation model using PSO trained modified ANFIS, with high accuracy and flexibility, was realized through the following.

- The original ANFIS was modified by introducing a logarithm to base 10 operator making it suitable for wireless signal propagation prediction modelling. This increased its accuracy levels up to 10^{14} times compared with the ordinary ANFIS in relation to radiowave signal prediction.
- The formulation of the modified ANFIS and its PSO training concepts, as used in wireless communication, was undertaken. This helped in understanding the two main concepts used in this research, that is, ANFIS and PSO. This is also important to the other researchers to enable them to concentrate on the application of these methods in relation to wireless communication systems modelling as the formulation has been well covered.
- After formulation of the ANFIS and PSO, the development of equivalent theoretical ANFIS based models for existing empirical models based on ANFIS, LOG10D-ANFIS and LOG10-PSO-ANFIS. The results were that for 100 data points the LOG10D-ANFIS generally performs slightly better than LOG10D-PSO-ANFIS whereas for 1000 data points LOG10D-PSO-ANFIS performs far much better than LOG10D-ANFIS in both training and testing errors which are to the tune of 10^{-14} as compared to 10^{-7} for 100 data points. Both these models were found to perform quite well when compared to the ordinary ANFIS model being improved on which had errors to the tune of 10^{-1}.
- From the above theoretical equivalent models, a novel universal theoretical model (LOG10D-PSO-ANFIS model) was developed for all the single slope models. This model, with a single structure, can receive membership functions

based on the single slope models and predict their RSSI accurately when compared with the ordinary ANFIS.

- A practical modelling of the behavior of the RSSI, a LOG10D-PSO-R-ANFIS practical model was developed. Its results were found to be superior.

5.2 Recommendations

- To develop a single AI based model for all the models including higher slopes using same structure and same membership functions with high accuracy since the current work didn’t manage to cover the higher slopes accurately. This model should be able to use the varying characteristics captured by the different other models in a single representation.
- A Graphical User Interface implementation of this model be done by either using MATLAB or any other programming language. This can make it directly usable by different wireless service providers in the field of wireless communication.

REFERENCES

Adeyiga, J. A., Sotonwa, K. A., & Adenibuyan, M. T. (2022). Comparison of Genetic Algorithm and Particle Swarm Optimization Techniques in Intelligent Parking System. *J Adv Mater Sci Eng, 2*(1), 1-10.

Adonias, G. L., & Carvalho, J. N. (2017, September). Dominant path model fitted to signal measurements of digital tv in non-line of sight propagation. *In 2017 IEEE Radio and Antenna Days of the Indian Ocean* (RADIO) (pp. 1-2). IEEE.

Ahmad, R., Sundararajan, E. A., & Khalifeh, A. (2020). A survey on femtocell handover management in dense heterogeneous 5G networks. *Telecommunication Systems, 75*, 481-507.

Ahmadien, O., Ates, H. F., Baykas, T., & Gunturk, B. K. (2020). Predicting path loss distribution of an area from satellite images using deep learning. *IEEE Access, 8*, 64982-64991.

Akyildiz, I. F., Kak, A., & Nie, S. (2020). 6G and beyond: The future of wireless communications systems. *IEEE access, 8*, 133995-134030.

Allen, B., Mahato, S., Gao, Y., & Salous, S. (2017). Indoor-to-outdoor empirical path loss modelling for femtocell networks at 0.9, 2, 2.5 and 3.5 GHz using singular value decomposition. *IET Microwaves, Antennas & Propagation, 11*(9), 1203-1211.

Alotaibi, F. D., Abdennour, A., & Ali, A. A. (2009). A real-time intelligent wireless mobile station location estimator with application to TETRA network. *IEEE transactions on mobile computing, 8*(11), 1495-1509.

Al-Saman, A., Cheffena, M., Elijah, O., Al-Gumaei, Y. A., Abdul Rahim, S. K., & Al-Hadhrami, T. (2021). Survey of millimeter-wave propagation measurements and models in indoor environments. *Electronics, 10*(14), 1653.

Andrade, C. B., & Hoefel, R. P. F. (2010, May). IEEE 802.11 WLANs: A comparison on indoor coverage models. In *CCECE 2010* (pp. 1-6). IEEE.

Austin, A. C., Neve, M. J., & Rowe, G. B. (2011). Modelling propagation in multifloor buildings using the FDTD method. *IEEE Transactions on Antennas and Propagation, 59*(11), 4239-4246.

Borralho, R., Mohamed, A., Quddus, A. U., Vieira, P., & Tafazolli, R. (2021). A survey on coverage enhancement in cellular networks: Challenges and solutions for future deployments. *IEEE Communications Surveys & Tutorials*, 23(2), 1302-1341.

Corre, Y., & Lostanlen, Y. (2009). Three-dimensional urban EM wave propagation model for radio network planning and optimization over large areas. *IEEE Transactions on Vehicular Technology, 58*(7), 3112-3123.

Crane, R. K. (2003). *Propagation handbook for wireless communication system design*. CRC press.

Dagefu, F. T., & Sarabandi, K. (2010, July). Simulation and measurement of near-ground wave propagation for indoor scenarios. *In 2010 IEEE Antennas and Propagation Society International Symposium* (pp. 1-4). IEEE.

de la Roche, G., Valcarce, A., & Zhang, J. (2011, May). Hybrid model for indoor-to-outdoor femtocell radio coverage prediction. In *2011 IEEE 73rd Vehicular Technology Conference* (VTC Spring) (pp. 1-5). IEEE.

Eberhart, R., & Kennedy, J. (1995, November). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942-1948).

El Khaled, Z., Ajib, W., & Mcheick, H. (2020). An accurate empirical path loss model for heterogeneous fixed wireless networks below 5.8 GHz frequencies. *IEEE Access*, 8, 182755-182775.

Elmezughi, M. K., Afullo, T. J., & Oyie, N. O. (2021). Performance study of path loss models at 14, 18, and 22 GHz in an indoor corridor environment for wireless communications. *SAIEE Africa Research Journal*, *112*(1), 32-45.

Erunkulu, O. O., Zungeru, A. M., Lebekwe, C. K., & Chuma, J. M. (2020). Cellular communications coverage prediction techniques: A survey and comparison. *IEEE Access, 8*, 113052-113077.

FCC, "*W5 Mobile Phone Test Report Measurement Report* TECNO MOBILE LIMITED", (https://fccid.io).

Garg, V. (2010). *Wireless communications & networking*. London: Elsevier.

Gentile, M., Monorchio, A., & Manara, G. (2006, July). Electromagnetic Propagation into Not Deterministic Sceneries by Using an Efficient Ray Tracing Simulator. In 2006 *IEEE Antennas and Propagation Society International Symposium* (pp. 4743-4746). IEEE.

Gharghan, S. K., Nordin, R., Jawad, A. M., Jawad, H. M., & Ismail, M. (2018). Adaptive neural fuzzy inference system for accurate localization of wireless sensor network in outdoor and indoor cycling applications. *IEEE Access*, *6*, 38475-38489.

Ghomsheh, V. S., Shoorehdeli, M. A., & Teshnehlab, M. (2007, June). Training ANFIS structure with modified PSO algorithm. In 2007 *Mediterranean Conference on Control & Automation* (pp. 1-6). IEEE.

Gurney, Kevin, (2014). *An introduction to neural networks.* York: Press.

Harinda, E., Hosseinzadeh, S., Larijani, H., & Gibson, R. M. (2019, April). Comparative performance analysis of empirical propagation models for lorawan 868mhz in an urban scenario. In *2019 IEEE 5th World Forum on Internet of Things* (WF-IoT) (pp. 154-159). IEEE.

Hassan, R., Cohanim, B., De Weck, O., & Venter, G. (2005, April). A comparison of particle swarm optimization and the genetic algorithm. In 46th *AIAA/ASME/ASCE/AHS/ASC* structures, structural dynamics and materials conference (p. 1897).

Hosseinzadeh, S., Larijani, H., & Curtis, K. (2017, June). An enhanced modified multi wall propagation model. In *2017 Global Internet of Things Summit* (GIoTS) (pp. 1-4). IEEE.

Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, *23*(3), 665-685.

Kihato, P. K. (2013). Analysis and visualization of metabolic syndrome using self organizing maps (SOM) (Doctoral dissertation).

Kimoto, H., Nishimori, K., Omaki, N., Kitao, K., & Imai, T. (2016). Experimental evaluation on outdoor to indoor propagation characteristics for multiple microwave bands. *IEICE Communications Express, 5*(3), 85-89.

Labiod, H., Afifi, H., & De Santis, C. (2007). *Wi-Fi, bluetooth, zigbee and wimax* (Vol. 1). Berlin, Germany:: Springer.

Linksys, "Linksys E900 I Wireless-N300 Router", (downloads.linksys.com).

Ma, S., Cheng, H., & Lee, H. (2021). A Practical Approach to Indoor Path Loss Modelling Based on Deep Learning. *J. Comput. Sci. Eng., 15*(2), 84-95.

Mathworks, Fuzzy Logic Toolbox User's Guide R2018b. https://www.mathworks.com.

Mfuko, M., & Omido, K. (2020). Mobile subscription, penetration and coverage trends in Kenya's telecommunication sector: A critique. International Journal of Advanced Research in Management and Social Sciences, 9(1), 127-133.

Nguyen, S. L., Jarvelainen, J., Karttunen, A., Haneda, K., & Putkonen, J. (2018). Comparing radio propagation channels between 28 and 140 GHz bands in a shopping mall, 515-5.

Omae, M. O., Ndungu, E. N., & Kibet, P. L. (2022, April). Indoor LOS Wi-Fi Coverage Prediction using ANN. In *Proceedings of the Sustainable Research and Innovation Conference* (pp. 289-294).

Peretto, P. (1992). *An introduction to the modelling of neural networks* (Vol. 2). Cambridge University Press.

Phillips, C., Sicker, D., & Grunwald, D. (2012). A survey of wireless path loss prediction and coverage mapping methods. *IEEE Communications Surveys & Tutorials, 15*(1), 255-270.

Popescu, I., Nikitopoulos, D., Constantinou, P., & Nafornita, I. (2006, September). Comparison of ANN based models for path loss prediction in indoor environment. *In IEEE Vehicular Technology Conference* (pp. 1-5). IEEE.

Popoola, S. I., Jefia, A., Atayero, A. A., Kingsley, O., Faruk, N., Oseni, O. F., & Abolade, R. O. (2019). Determination of neural network parameters for path loss prediction in very high frequency wireless channel. *IEEE access*, *7,* 150462-150483.

Ptolemaeus, C. (Ed.). (2014). System design, modelling, and simulation: using *Ptolemy* II (Vol. 1). Berkeley: Ptolemy. org.

Rappaport, T. S., Heath Jr, R. W., Daniels, R. C., & Murdock, J. N. (2015). *Millimeter wave wireless communications.* Pearson Education.

Rath, H. K., Timmadasari, S., Panigrahi, B., & Simha, A. (2017, March). Realistic indoor path loss modelling for regular WiFi operations in India. In *2017 Twenty-third National Conference on Communications (NCC)* (pp. 1-6). IEEE.

Reardon, M. (2010). Cisco predicts wireless data explosion. Press release, 9th Feb 2010, online available.

Roddy, D. (2006). *Satellite communications*. McGraw-Hill Education.

Saakian, A. (2020). *Radio wave propagation fundamentals*. Artech House.

Schafer, T. M., & Wiesbeck, W. (2005). Simulation of radiowave propagation in hospitals based on FDTD and ray-optical methods. *IEEE transactions on antennas and propagation*, *53*(8), 2381-2388.

Subrt, L., & Pechac, P. (2010). Semi-deterministic propagation model for subterranean galleries and tunnels. IEEE Transactions on Antennas and Propagation, 58(11), 3701-3706.

Sumathi, S., & Paneerselvam, S. (2010). *Computational intelligence paradigms: theory & applications using MATLAB.* CRC Press.

Ullah, U., Kamboh, U. R., Hossain, F., & Danish, M. (2020). Outdoor-to-indoor and indoor-to-indoor propagation path loss modelling using smart 3D ray tracing algorithm at 28 GHz mmWave. *Arabian Journal for Science and Engineering, 45*, 10223-10232.

Valcarce, A., De La Roche, G., Nagy, L., Wagen, J. F., & Gorce, J. M. (2011). A new trend in propagation prediction. *IEEE Vehicular Technology Magazine, 6*(2), 73-81.

Vilovic, I., & Burum, N. (2011, April). A comparison of MLP and RBF neural networks architectures for electromagnetic field prediction in indoor environments. In *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP*) (pp. 1719-1723). IEEE.

Wahl, R., Wölfle, G., Wertz, P., Wildbolz, P., & Landstorfer, F. (2005). Dominant path prediction model for urban scenarios. *14th IST Mobile and Wireless Communications Summit*, Dresden (Germany).

Winder, S., & Carr, J. (2002). *Newnes radio and RF engineering pocket book*. Newnes.

Wu, L., He, D., Ai, B., Wang, J., Qi, H., Guan, K., & Zhong, Z. (2020). Artificial neural network-based path loss prediction for wireless communication network. *IEEE access, 8*, 199523-199538.

Wu, L., He, D., Ai, B., Wang, J., Qi, H., Guan, K., & Zhong, Z. (2020). Artificial neural network-based path loss prediction for wireless communication network. *IEEE access, 8*, 199523-199538.

Yarpiz, "Particle Swam Optimization in MATLAB", http://www.yarpiz.com.

Zhang, L., Rodríguez-Piñeiro, J., Fernández, J. R., García-Naya, J. A., Matolak, D. W., Briso, C., & Castedo, L. (2017). Propagation modelling for outdoor-to-indoor and indoor-to-indoor wireless links in high-speed train. *Measurement, 110*, 43-52.

Zhang, W. (2022). Particle swarm optimization: A Matlab algorithm. *Selforganizology*, *9*(3-4), 35-41.

APPENDICES

Appendix I: Program codes

Code for PSO obtained random input and ANFIS membership functions

```
CostFunction =@(x) Sphere18rnanfisp1trnts15(x);  % Cost Function
   nVar = 114;        % Number of Unknown (Decision) Variables
    VarSize = [1 nVar];         % Matrix Size of Decision Variables
    VarMin = -69;   % Lower Bound of Decision Variables
   VarMax = 41.2;    % Upper Bound of Decision Variables
    %% Parameters of PSO
    MaxIt = 1000;   % Maximum Number of Iterations
    nPop = 50;     % Population Size (Swarm Size)
    w = 1;           % Intertia Coefficient
   wdamp = 0.99;   % Damping Ratio of Inertia Coefficient
   c1 = 2;         % Personal Acceleration Coefficient
   c2 = 2;         % Social Acceleration Coefficient
    % The Flag for Showing Iteration Information
   %ShowIterInfo = params.ShowIterInfo;
    MaxVelocity = 0.08*(VarMax-VarMin);
   MinVelocity = -MaxVelocity;
      %% Initialization
    % The Particle Template
   empty_particle.Position = [];
   empty_particle.Velocity = [];
   empty_particle.Cost = [];
   empty_particle.Best.Position = [];
   empty_particle.Best.Cost = [];
    % Create Population Array
   particle = repmat(empty_particle, nPop, 1);
    % Initialize Global Best
   GlobalBest.Cost = inf;
    % Initialize Population Members
   for i=1:nPop
        % Generate Random Solution
      particle(i).Position = VarMin+rand(1,nVar)*(VarMax-VarMin);
        % Initialize Velocity
      particle(i).Velocity = zeros(VarSize);
        % Evaluation
      particle(i).Cost = CostFunction(particle(i).Position);
        % Update the Personal Best
      particle(i).Best.Position = particle(i).Position;
      particle(i).Best.Cost = particle(i).Cost;
        % Update Global Best
      if particle(i).Best.Cost < GlobalBest.Cost
      GlobalBest = particle(i).Best;
      end
    end
    % Array to Hold Best Cost Value on Each Iteration
   BestCosts = zeros(MaxIt, 1);
    %% Main Loop of PSO
    for it=1:MaxIt
   for i=1:nPop
      % Update Velocity
     particle(i).Velocity = w*particle(i).Velocity ...
         + c1*rand(VarSize).*(particle(i).Best.Position - particle(i).Position) ...
         + c2*rand(VarSize).*(GlobalBest.Position - particle(i).Position);
      % Apply Velocity Limits
     particle(i).Velocity = max(particle(i).Velocity, MinVelocity);
     particle(i).Velocity = min(particle(i).Velocity, MaxVelocity);
    % Update Position
     particle(i).Position = particle(i).Position + particle(i).Velocity;

     % Apply Lower and Upper Bound Limits
     particle(i).Position = max(particle(i).Position, VarMin);
     particle(i).Position = min(particle(i).Position, VarMax);
      % Evaluation
     particle(i).Cost = CostFunction(particle(i).Position);
      % Update Personal Best
     if particle(i).Cost < particle(i).Best.Cost
          particle(i).Best.Position = particle(i).Position;
         particle(i).Best.Cost = particle(i).Cost;
          % Update Global Best
         if particle(i).Best.Cost < GlobalBest.Cost
             GlobalBest.Position = particle(i).Best.Position; %added Update on Global Best position
             GlobalBest = particle(i).Best;
```

```
                end
             end
         end
         % Store the Best Cost Value
        BestCosts(it)=GlobalBest.Cost;
         % Display Iteration Information
        %if ShowIterInfo
            disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCosts(it))]);
       % end
         % Damping Inertia Coefficient
        w = w * wdamp;
     end
    out.pop = particle;
    out.BestSol = GlobalBest;
    out.BestCosts = BestCosts;
```

Code structure courtesy of (www.yarpiz.com)

```
function [ztrn, ztst] = Sphere18rnanfisp1trnts15(x)
distance=[0.4:0.6:41.2];
averageDL =[-36   -39   -40   -45   -49   -50   -44   -49   -55   -57   -54   -53   -54   -51
-52   -56   -53   -53   -59   -58   -58   -54   -53   -55   -55   -55   -57  -55   -54   -51   -53
-51   -52   -53   -55   -54   -56   -54   -57   -56   -57   -57   -58   -57   -59   -57   -61   -
63   -63   -59   -58   -56   -67   -65  -63   -61   -62   -59   -61   -61   -65   -61   -62   -69
-57   -65   -63   -59   -67];
data=[log10(distance)', 0.15*(x(1:69))', averageDL'];
tstdata=data((2:3:69), :);
data((2:3:69), :)=[];
```

```
trndata=data;

numMFs=3;

mfType='gbellmf';

outfis=genfis1(trndata,numMFs,mfType);

outfis.input(1).mf(1).params= x(70:72);

outfis.input(1).mf(2).params= x(73:75);

outfis.input(1).mf(3).params= x(76:78);

outfis.input(2).mf(1).params= x(79:81);

outfis.input(2).mf(2).params= x(82:84);

outfis.input(2).mf(3).params= x(85:87);

outfis.output(1).mf(1).params= x(88:90);

outfis.output(1).mf(2).params= x(91:93);

outfis.output(1).mf(3).params= x(94:96);

outfis.output(1).mf(4).params= x(97:99);

outfis.output(1).mf(5).params= x(100:102);

outfis.output(1).mf(6).params= x(103:105);

outfis.output(1).mf(7).params= x(106:108);

outfis.output(1).mf(8).params= x(109:111);

outfis.output(1).mf(9).params= x(112:114);

%trainingdata

intdl=[trndata(:,1), trndata(:,2)];

outtdl=evalfis(intdl,outfis);

ME=mean(abs(trndata(:,3)'-outtdl'));

%plot(distance, averageDL, trndata(:,1),outtdl)

RMSEtrn=sqrt(mean((trndata(:,3)'-outtdl').^2));
```

```
Sdeviation=std(abs(trndata(:,3)'-outtdl'));
%testing data
intdlt=[tstdata(:,1), tstdata(:,2)];
outtdlt=evalfis(intdlt,outfis);
ME=mean(abs(tstdata(:,3)'-outtdlt'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtst=sqrt(mean((tstdata(:,3)'-outtdlt').^2));
Sdeviation=std(abs(tstdata(:,3)'-outtdlt'));
   ztrn=RMSEtrn;
   ztst=RMSEtst;
  %z=abs(ztrn-ztst);
   %z=[ztrn ztst];
   warning('off','all')
end (www.yarpiz.com)
```

LOG10D-PSO-ANFIS code

```
%% Problem Definiton
CostFunction =@(x) Sphere8tnt(x);  % Cost Function
   nVar = 15;        % Number of Unknown (Decision) Variables
    VarSize = [1 nVar];         % Matrix Size of Decision Variables
    VarMin = -52;   % Lower Bound of Decision Variables
   VarMax = 100;    % Upper Bound of Decision Variables
    %% Parameters of PSO
    MaxIt = 1000;   % Maximum Number of Iterations
    nPop = 50;     % Population Size (Swarm Size)
    w = 1;          % Intertia Coefficient
```

```
wdamp = 0.99;   % Damping Ratio of Inertia Coefficient
c1 = 2;         % Personal Acceleration Coefficient
c2 = 2;         % Social Acceleration Coefficient
 % The Flag for Showing Iteration Information
%ShowIterInfo = params.ShowIterInfo;
 MaxVelocity = 0.01*(VarMax-VarMin);
MinVelocity = -MaxVelocity;
   %% Initialization
 % The Particle Template
empty_particle.Position = [];
empty_particle.Velocity = [];
empty_particle.Cost = [];
empty_particle.Best.Position = [];
empty_particle.Best.Cost = [];
 % Create Population Array
particle = repmat(empty_particle, nPop, 1);
 % Initialize Global Best
GlobalBest.Cost = inf;
 % Initialize Population Members
for i=1:nPop
     % Generate Random Solution
    particle(i).Position = VarMin+rand(1,nVar)*(VarMax-VarMin);
     % Initialize Velocity
    particle(i).Velocity = zeros(VarSize);
     % Evaluation
    particle(i).Cost = CostFunction(particle(i).Position);
     % Update the Personal Best
    particle(i).Best.Position = particle(i).Position;
    particle(i).Best.Cost = particle(i).Cost;
     % Update Global Best
    if particle(i).Best.Cost < GlobalBest.Cost
        GlobalBest = particle(i).Best;
    end
 end
 % Array to Hold Best Cost Value on Each Iteration
BestCosts = zeros(MaxIt, 1);
 %% Main Loop of PSO
 for it=1:MaxIt
     for i=1:nPop
         % Update Velocity
        particle(i).Velocity = w*particle(i).Velocity ...
            + c1*rand(VarSize).*(particle(i).Best.Position - particle(i).Position) ...
            + c2*rand(VarSize).*(GlobalBest.Position - particle(i).Position);
         % Apply Velocity Limits
        particle(i).Velocity = max(particle(i).Velocity, MinVelocity);
        particle(i).Velocity = min(particle(i).Velocity, MaxVelocity);
        % Update Position
        particle(i).Position = particle(i).Position + particle(i).Velocity;

        % Apply Lower and Upper Bound Limits
```

```
          particle(i).Position = max(particle(i).Position, VarMin);
          particle(i).Position = min(particle(i).Position, VarMax);
           % Evaluation
          particle(i).Cost = CostFunction(particle(i).Position);
           % Update Personal Best
          if particle(i).Cost < particle(i).Best.Cost
               particle(i).Best.Position = particle(i).Position;
              particle(i).Best.Cost = particle(i).Cost;
               % Update Global Best
              if particle(i).Best.Cost < GlobalBest.Cost
                 GlobalBest.Position = particle(i).Best.Position; %added Update on Global
Best position
                  GlobalBest = particle(i).Best;
              end
           end
       end
       % Store the Best Cost Value
      BestCosts(it)=GlobalBest.Cost;
       % Display Iteration Information
      %if ShowIterInfo
          disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCosts(it))]);
     % end
       % Damping Inertia Coefficient
      w = w * wdamp;
   end
```

```
    out.pop = particle;

    out.BestSol = GlobalBest;

    out.BestCosts = BestCosts;

function [ztrn, ztst] = Sphere8tnt(x)

distance = linspace(1,100,1000);

Pt = -10; Gt = 4;Gr = 4;

L= 32.44+20*log10(2400)+20*log10(distance/1000);

averageDL=Pt + Gt + Gr-L+30;

%data division*

data=[log10(distance)', averageDL'];

tstdata=data((2:3:1000), :);

data((2:3:1000), :)=[];

trndata=data;

numMFs=3;

mfType='gbellmf';

outfis=genfis1(trndata,numMFs,mfType);

outfis.input(1).mf(1).params= x(1:3);

outfis.input(1).mf(2).params= x(4:6);

outfis.input(1).mf(3).params= x(7:9);

outfis.output(1).mf(1).params= x(10:11);

outfis.output(1).mf(2).params= x(12:13);

outfis.output(1).mf(3).params= x(14:15);

   %trainingdata

intdl=trndata(:,1);

outtdl=evalfis(intdl,outfis);
```

```
ME= mean(abs(trndata(:,2)'-outtdl'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtrn=sqrt(mean((trndata(:,2)'-outtdl').^2));
Sdeviation=std(abs(trndata(:,2)'-outtdl'));
%testing data
intdlt=tstdata(:,1);
outtdlt=evalfis(intdlt,outfis);
ME= mean(abs(tstdata(:,2)'-outtdlt'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtst=sqrt(mean((tstdata(:,2)'-outtdlt').^2));
Sdeviation=std(abs(tstdata(:,2)'-outtdlt'));
   ztrn=RMSEtrn
  ztst=RMSEtst
    warning('off','all')
end
x=[-30.4042  -44.4841  -17.1511  -21.2635  -16.4371  -23.2478  -26.6433  -33.4410  -12.9429  -39.8081  -19.3858  -20.0000 -12.0442  -38.4691  -22.6328];
distance = linspace(1,100,1000);
Pt = -10; Gt = 4;Gr = 4;
L= 32.44+20*log10(2400)+20*log10(distance/1000);
averageDL=Pt + Gt + Gr-L+30;
 %data division*
data=[log10(distance)', averageDL'];
tstdata=data((2:3:1000), :);
data((2:3:1000), :)=[];
```

```
trndata=data;
numMFs=3;
mfType='gbellmf';
outfis=genfis1(trndata,numMFs,mfType);
   outfis.input(1).mf(1).params= x(1:3);
outfis.input(1).mf(2).params= x(4:6);
outfis.input(1).mf(3).params= x(7:9);
outfis.output(1).mf(1).params= x(10:11);
outfis.output(1).mf(2).params= x(12:13);
outfis.output(1).mf(3).params= x(14:15);
 %trainingdata
intdl=trndata(:,1);
outtdl=evalfis(intdl,outfis);
ME= mean(abs(trndata(:,2)'-outtdl'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtrn=sqrt(mean((trndata(:,2)'-outtdl').^2));
Sdeviation=std(abs(trndata(:,2)'-outtdl'));
%testing data
intdlt=tstdata(:,1);
outtdlt=evalfis(intdlt,outfis);
ME= mean(abs(tstdata(:,2)'-outtdlt'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtst=sqrt(mean((tstdata(:,2)'-outtdlt').^2));
Sdeviation=std(abs(tstdata(:,2)'-outtdlt'));
   ztrn=RMSEtrn
```

```
  ztst=RMSEtst
Dtrndata= trndata(:,1);

Dtstdata= tstdata(:,1);
%errors
trneLA=[outtdl-trndata(:,2)];
tsteLA=[outtdlt-tstdata(:,2)];
%Plots
figure(1)
subplot(2,1,1);
plot(Dtrndata, trndata(:,2),  Dtrndata, outtdl)
title('One Slope Model & LOG10D-PSO-ANFIS – Training')
xlabel('Distance (m)')
ylabel('RSSI (dBm)')
legend('One Slope RSSI','LOG10D-PSO-ANFIS predicted RSSI','Location','Northeast')
grid
subplot(2,1,2);
plot(Dtstdata, tstdata(:,2),  Dtstdata, outtdlt)
title('One Slope Model & LOG10D-PSO-ANFIS – Testing')
xlabel('Distance (m)')
ylabel('RSSI (dBm)')
legend('One Slope RSSI','LOG10D-PSO-ANFIS predicted RSSI','Location','Northeast')
grid
figure(2)
```

```
subplot(2,1,1);
plot(Dtrndata, trneA, Dtrndata, trneLA, Dtrndata, trneLPA)
title('One Slope ANFIS, LOG10D-ANFIS & LOG10D-PSO-ANFIS – Training errors')
xlabel('Distance (m)')
ylabel('RSSI error (dBm)')
legend('ANFIS RSSI Training error', 'LOG10D-ANFIS RSSI Training error', 'LOG10D-PSO-ANFIS RSSI Training error','Location','Northeast')
grid
subplot(2,1,2);
plot(Dtstdata, tsteA, Dtstdata, tsteLA, Dtstdata, tsteLPA)
title('One Slope ANFIS, LOG10D-ANFIS & LOG10D-PSO-ANFIS – Testing errors')
xlabel('Distance (m)')
ylabel('RSSI (dBm)')
legend('ANFIS RSSI Testing error', 'LOG10D-ANFIS RSSI Testing error','LOG10D-PSO-ANFIS RSSI Testing error','Location','Southeast')
grid
figure(3)
subplot(2,1,1);
plot(Dtrndata, trneLA, Dtrndata, trneLPA)
title('Training errors')
xlabel('Distance (m)')
ylabel('RSSI error (dBm)')
legend('LOG10D-ANFIS RSSI Training error', 'LOG10D-PSO-ANFIS RSSI Training error','Location','Northeast')
grid
```

```
subplot(2,1,2);
plot(Dtstdata, tsteLA, Dtstdata, tsteLPA)
title('Testing errors')
xlabel('Distance (m)')
ylabel('RSSI (dBm)')
legend('LOG10D-ANFIS RSSI Testing error','LOG10D-PSO-ANFIS RSSI Testing
error','Location','Southeast')
grid
sse=sum(trneLPA.^2); %
tss=sum((mean(trndata(:,2)) - trndata(:,2)).^ 2);
r2trn=1-sse/tss
sse=sum(tsteLPA.^2);
tss=sum((mean(tstdata(:,2)) - tstdata(:,2)).^ 2);
r2tst=1-sse/tss
```

ANFIS for PSO formulation

```
x =[-11.1315  -15.9759  -36.1553  -12.4641  -17.0160   6.6645  -42.9664  -34.5415
26.2031   36.1733  -28.1564   0.2303   4.2159   24.1794  -15.9234   10.3787  -14.5437
-15.9473   25.2186  -15.1005   14.0191  -16.6717  -22.7503  -12.8930  -14.4956  -
36.5014  -3.1021  -18.9643   0.8962  -50.5380  -37.4063  -25.1245  -47.0880  -41.0215
-17.7794  -36.1201  -22.7065  -35.7279  -17.5655  -25.7620   -9.8859  -20.4709  -
14.1760  -31.8734  -8.7446  -24.0368  -6.3774  17.5440  16.7831  -15.9582  -20.8494
-36.0586  12.3764  27.3785  12.2454  20.0235   3.6471  -18.6739   8.8446  -5.5870
22.5857  -29.5050  -0.2746  41.2000  -29.8534  19.5592   4.5425  -30.6846  32.2932
-9.5142  -11.2116  -45.7657   4.7857  -13.8573   8.7356   1.4020  -53.7227  -22.1354
41.1981  -10.8878  -13.9449  -67.0429  -40.8190   17.2752  -14.0202   29.8814  -
10.6895  -16.4866  -20.6013   -2.5082  -56.9382  -51.9578  -34.3824  -10.5858   -
6.3941  -46.2224  17.6400  -14.7111  -12.9891  41.2000  -8.2135  -27.5009  -42.1031
-2.8180   -41.4139   -34.9833   -34.9610   -33.0009   -56.4224    17.4381   -35.0589
15.4361    6.4404  -39.7980];

distance=[0.4:0.6:41.2];

averageDL =[-36  -39  -40  -45  -49  -50  -44  -49  -55  -57  -54  -53  -54  -51
-52  -56  -53  -53  -59  -58  -58  -54  -53  -55  -55  -55  -57 -55  -54  -51  -53
-51  -52  -53  -55  -54  -56  -54  -57  -56  -57  -57  -58  -57  -59  -57  -61  -
63  -63  -59  -58  -56  -67  -65 -63  -61  -62  -59  -61  -61  -65  -61  -62  -69
-57  -65  -63  -59  -67];

%data=[Inputs', 0.15*(x(1:69))', Targets'];

data=[log10(distance)', 0.15*(x(1:69))', averageDL'];

tstdata=data((2:3:69), :);

data((2:3:69), :)=[];

trndata=data;

numMFs=3;

mfType='gbellmf';

outfis=genfis1(trndata,numMFs,mfType);

outfis.input(1).mf(1).params= x(70:72);

outfis.input(1).mf(2).params= x(73:75);

outfis.input(1).mf(3).params= x(76:78);

outfis.input(2).mf(1).params= x(79:81);

outfis.input(2).mf(2).params= x(82:84);

outfis.input(2).mf(3).params= x(85:87);

outfis.output(1).mf(1).params= x(88:90);

outfis.output(1).mf(2).params= x(91:93);

outfis.output(1).mf(3).params= x(94:96);

outfis.output(1).mf(4).params= x(97:99);
outfis.output(1).mf(5).params= x(100:102);
outfis.output(1).mf(6).params= x(103:105);
outfis.output(1).mf(7).params= x(106:108);
outfis.output(1).mf(8).params= x(109:111);
outfis.output(1).mf(9).params= x(112:114);
%trainingdata
intdl=[trndata(:,1), trndata(:,2)];
outtdl=evalfis(intdl,outfis);
ME=mean(abs(trndata(:,3)'-outtdl'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtrn=sqrt(mean((trndata(:,3)'-outtdl').^2));
Sdeviation=std(abs(trndata(:,3)'-outtdl'));
%testing data
intdlt=[tstdata(:,1), tstdata(:,2)];
outtdlt=evalfis(intdlt,outfis);
ME=mean(abs(tstdata(:,3)'-outtdlt'));
%plot(distance, averageDL, trndata(:,1),outtdl)
RMSEtst=sqrt(mean((tstdata(:,3)'-outtdlt').^2));
Sdeviation=std(abs(tstdata(:,3)'-outtdlt'));
```

Novel universal model ANFIS code

```
distance = linspace(1,100,1000);
%Pt = -10; Gt = 4;Gr = 4;
%L= 32.44+20*log10(2400)+20*log10(distance/1000);
%averageDL=Pt + Gt + Gr-L+30;
averageDL=[zeros(1,1000)];
```

```
%data division*

data=[log10(distance)', averageDL'];

tstdata=data((2:3:1000), :);

data((2:3:1000), :)=[];

trndata=data;

numMFs=3;

mfType='gbellmf';

outfis=genfis1(trndata,numMFs,mfType);

outfis.input(1).mf(1).params= m(1:3);

outfis.input(1).mf(2).params= m(4:6);

outfis.input(1).mf(3).params= m(7:9);

outfis.output(1).mf(1).params= m(10:11);

outfis.output(1).mf(2).params= m(12:13);

outfis.output(1).mf(3).params= m(14:15);

%trainingdata

intdl=trndata(:,1);

outtdl=evalfis(intdl,outfis);

%ME= mean(abs(trndata(:,2)'-outtdl'));

plot(10.^trndata(:,1),outtdl)

%RMSEtrn=sqrt(mean((trndata(:,2)'-outtdl').^2));

%Sdeviation=std(abs(trndata(:,2)'-outtdl'));

%testing data

intdlt=tstdata(:,1);

outtdlt=evalfis(intdlt,outfis);

%ME= mean(abs(tstdata(:,2)'-outtdlt'));
```

```
plot(10.^ trndata(:,1),outtdl)

%RMSEtst=sqrt(mean((tstdata(:,2)'-outtdlt').^2));

%Sdeviation=std(abs(tstdata(:,2)'-outtdlt'));

  %ztrn=RMSEtrn

  %ztst=RMSEtst
```

Appendix II: ANFIS and PSO problem formulations and the universal model parameters

Problem formulation of ANFIS with LOG10D algorithm

The following representation is a step-by-step analysis of the modified ANFIS algorithm starting from layer one to layer 6.

Taking values of a_i, b_i and c_i as;

[0.4133 2 0]

[0.4133 2 0.8266]

[0.4133 2 1.653]

and p_i and r_i as;

[-20 -17.04]

[-20 -17.04]

[-20 -17.04]

$Rule\ 1\!: If\ log10(x)\ is\ A_1, then\ z_1 = \ p_1 log10(x) + r_1,$

$$If\ log10(x)\ is\ \exp\{-\left[\left(\frac{log10(x)-c_1}{a_1}\right)^2\right]^{b_1}\}\ then\ z_1 = \ p_1 log10(x)\ + r_1,$$

$$If\ log10(10) is\ \exp\left\{-\left[\left(\frac{log10(10)-0}{0.4133}\right)^2\right]^2\right\} then\ z_1 = \ -20 \times log10(10) -$$

$17.04,$

$$If\ 1\ is\ 1.3059\text{e} - 15, then\ z_1 = -37.0400,$$

Performing the same for rule 2 and 3

$Rule\ 2{:}\ If\ log10(x)\ is\ A_2, then\ z_2 = \ p_2 log10(x) + r_2,$

$If\ 1\ is\ 0.9695, then\ z_2 = \ -37.0400,$

$Rule\ 3{:}\ If\ log10(x)\ is\ A_3, then\ z_3 = \ p_2 log10(x) + r_3,$

$If\ 1\ is\ 0.0020, then\ z_3 = -37.0400,$

x=distance and z=rssi

Layer 1 (Logarithmic layer):

This layer deals with the transformation of the distance values, as obtained from the one slope model, into their logarithmic representation. The resulting values are then fed into layer 2 which performs the fuzzification operation.

Layer 2 (Fuzzy Layer):

$$O_i^2 = f(log10(x), a, b, c) = \mu_{Ai}(log10(x)) = \frac{1}{1+|(log10(x)-c_i)/a_i|^{2b_i}}$$

$$\mu\ _{Ai}(log10(x)) = \exp\{-\left[\left(\frac{log10(x)-c_i}{a_i}\right)^2\right]^{b_i}\}$$

A1=exp(-(((log10(x)-c1)*a1^-1)^2)^b1)

A1=1.3059e-15

A2=exp(-(((log10(x)-c2)*a2^-1)^2)^b2)

A2= 0.9695

A3=exp(-(((log10(x)-c3)*a3^-1)^2)^b3)

A3= 0.0020

Layer 3 (Product Layer)

$w_i = O_i^3 = \mu_{Ai}(log10(x)).\mu_{Bi}(y), i = 1, 2$

$w_1 = O_i^3 = \mu_{A1}(log10(x)),$

w1=1.3059e-15

w2=0.9695

w3=0.0020

Layer 4 (Normalized Layer)

$\overline{w}_i = O_i^4 = \frac{w_i}{w_1+w_2+w_3}, i = 1, 2, 3$

$\overline{w}_1=\frac{w_1}{w_1+w_2+w_3}=\frac{1.3059e-15}{1.3059e-15+0.9695+0.0020}$=1.3442e-15

$\overline{w}_2=\frac{w_2}{w_1+w_2+w_3} = \frac{0.9695}{1.3059e-15+0.9695+0.0020}$=0.9979

$\overline{w}_3=\frac{w_3}{w_1+w_2+w_3} = \frac{0.0020}{1.3059e-15+0.9695+0.0020}$=0.0021

Layer 5 (Defuzzify Layer)

$\overline{w}_i z_i = O_i^5 = \overline{w}_i(p_i log10(x) + q_i y + r_i)$ $\overline{w}_1 z_1 = \overline{w}_1(p_1 log10(x) + r_1),$

$\overline{w}_1 z_1 = 1.3442\mathrm{e} - 15 * (-20 * \log10(10) - 17.04) = -4.9789\mathrm{e} - 14,$

$\overline{w}_2 z_2 = \overline{w}_2(p_2 log10(x) + r_2),$

$\overline{w}_2 z_2 = 0.9979 * (-20 * \log10(10) - 17.04) = -36.9622,$

$\overline{w}_3 z_3 = \overline{w}_3(p_3 log10(x) + r_3),$

$\bar{w}_3 z_3$ =0.0021*(-20*log10(10)-17.04)= -0.0778,

Layer 6 (Total Output Layer)

$$z = \sum_i \bar{w}_i z_i = O_i^6 = \frac{\sum_i w_i z_i}{\sum_i w_i} = \frac{w_1 z_1 + w_2 z_2 + w_2 z_3}{w_1 + w_2 + w_3} = \bar{w}_1 z_1 + \bar{w}_2 z_2 + \bar{w}_3 z_3$$

$$z = -4.9789\mathrm{e} - 14 - 36.9622 - 0.0778 = -37.0400$$

Problem formulation of ANFIS with LOG10D and random input

$Rule\ 1: If\ log10x\ is\ A_1\ and\ y\ is\ B_1 then\ z_1 = \ p_1 log10x + q_1 y + r_1,$

$Rule\ 2: If\ log10x\ is\ A_2\ and\ y\ is\ B_2\ then\ z_2 = \ p_2 log10x + q_2 y + r_2,$

x=distance, y=random input and z=RSSI

r is a random input which also shows the adaptability of the system-output modelled to vary according to the r random input

Two inputs with 3 MFs based on the analysis below

Taking arbitrary values of distance (x)=10, random RSSI (y)=1, a_i, b_i and c_i as;

For A;

a1=-9.514, b1=-11.21 and c1=-45.77

a2=4.786, b2=-13.86 and c2=8.736

a3=1.402, b3=-53.72 and c3=-22.14

For B;

a1=41.2, b1=-10.89 and c1=-13.94

a2=-67.04, b2=-40.82 and c2=17.28

a3=-14.02, b3=29.88 and c3=-10.69

and p_i, *qi* and r_i as;

p1=-16.49, q1=-20.6 and r1=-2.508

p2=-56.94, q2=-51.96 and r2=-34.38

p3=-10.59, q3=-6.394 and r3=-46.22

p4=17.64, q4=-14.71 and r4=-12.99

p5=41.2, q5=-8.213 and r5=-27.5

p6=-42.1, q6=-2.818 and r6=-41.41

p7=-34.98, q7=-34.96 and r7=-33

p8=-56.42, q8=17.44 and r8=-35.06

p9=15.44, q9=6.44 and r9=-39.8

$Rule\ 1{:}\, If\ log10(x)\ is\ A_1\ and\ y\ is\ B_1, then\ z_1 =\ p_1 log10(x) + q_1 y + r_1,$

$$If\ log10(10)\ is \exp\left\{-\left[\left(\frac{log10(10) - -45.77}{-9.514}\right)^2\right]^{-11.21}\right\} and\ 2\ is,$$

$$\exp\left\{-\left[\left(\frac{1 - -13.94}{41.2}\right)^2\right]^{-10.89}\right\}$$

$$then\ z_1 =\ -16.49 \times log10(10) + (-20.6\ \times 1) + (-2.508),$$

$If\ 1\ is\ 1\ and\ 1\ is\ 0, then\ z_1 =\ -39.5980,$

$Rule\ 2{:}\, If\ log10(x)\ is\ A_1\ and\ y\ is\ B_2, then\ z_2 =\ p_2 log10(x) + q_2 y + r_2,$

$If\ log10(10)\ is\ 1\ and\ 1\ is\ 0, then\ z_2 =\ -56.94 \times log10(10) + (-51.96 \times 1) + (-34.38),$

$If\ 1\ is\ 1\ and\ 1\ is\ 0, then\ z_2 = -143.2800,$

$Rule\ 3{:}\, If\ log10(x)\ is\ A_1\ and\ y\ is\ B_3, then\ z_3 =\ p_3 log10(x) + q_3 y + r_3,$

$If\ 1\ is\ 1\ and\ 1\ is\ 1, then\ z_3 = -63.2040,$

$Rule\ 4{:}\, If\ log10(x)\ is\ A_2\ and\ y\ is\ B_1, then\ z_4 =\ p_4 log10(x) + q_4 y + r_4,$

$If\ 1\ is\ 1\ and\ 1\ is\ \ 0, then\ z_4 = -10.0600,$

$Rule\ 5{:}\, If\ log10(x)\ is\ A_2\ and\ y\ is\ B_2, then\ z_5 =\ p_5 log10(x) + q_5 y + r_5,$

$If\ 1\ is\ 1\ and\ 1\ is\ 0, then\ z_5 =\ 5.4870,$

$Rule\ 6: If\ log10(x)\ is\ A_2\ and\ y\ is\ B_3, then\ z_6 =\ p_6 log10(x) + q_6 y + r_6,$

$If\ 1\ is\ 1\ and\ 1\ is\ \ 1, then\ z_6 = -86.3280,$

$Rule\ 7: If\ log10(x)\ is\ A_3\ and\ y\ is\ B_1, then\ z_7 =\ p_7 log10(x) + q_7 y + r_7,$

$If\ 1\ is\ \ 1\ and\ 1\ is\ 0, then\ z_7 = -102.9400,$

$Rule\ 8: If\ log10(x)\ is\ A_3\ and\ y\ is\ B_2, then\ z_8 =\ p_8 log10(x) + q_8 y + r_8,$

$If\ 1\ is\ \ 1\ and\ 1\ is\ 0, then\ z_8 = -74.0400,$

$Rule\ 9: If\ log10(x)\ is\ A_3\ and\ y\ is\ B_3, then\ z_9 =\ p_9 log10(x) + q_9 y + r_9,$

$If\ 1\ is\ \ 1\ and\ 1\ is\ \ 1, then\ z_9 = -17.9200,$

Layer 1 (Fuzzy Layer):

$$O_i^1 = f(log10(x), a, b, c) = \mu_{Ai}(log10(x)) = \frac{1}{1+|(log10(x)-c_i)/a_i|^{2b_i}}$$

$$\mu_{\ Ai}(log10(x)) = \exp\{-\left[\left(\frac{log10(x)-c_i}{a_i}\right)^2\right]^{b_i}\}$$

$$\mu_{\ Bi}(y) = \exp\{-\left[\left(\frac{y-c_i}{a_i}\right)^2\right]^{b_i}\}$$

A1=exp(-(((log10(x)-c1)a1^-1)^2)^b1)

A1=1

A2=exp(-(((log10(x)-c2)a2^-1)^2)^b2)

A2= 1

A3=exp(-(((log10(x)-c3)a3^-1)^2)^b3)

A3= 1

B1=exp(-(((y-c1)a1^-1)^2)^b1)

B1=0

B2=exp(-(((y-c2)a2^-1)^2)^b2)

B2= 0

B3=exp(-(((y-c3)a3^-1)^2)^b3)

B3= 1

Layer 2 (Product Layer):

$$w_i = O_i^2 = \mu_{Ai}(log10(x)).\mu_{Bi}(y), i = 1, 2$$

$$w_1 = \mu_{A1}(log10(x)).\mu_{B1}(y),$$

w1=A1*B1

w1=1*0= 0

w2=A1*B2

w2=1*0= 0

w3=A1*B3

w3=1*1= 1

w4=A2*B1

w4=1*0= 0

w5=A2*B2

w5=1*0= 0

w6=A2*B3

w6=1*1= 1

w7=A3*B1

w7=1*0= 0

w8=A3*B2

w8=1*0= 0

w9=A3*B3

w9=1*1= 1

Layer 3 (Normalized Layer):

$$\overline{w}_i = O_i^3 = \frac{w_i}{w_1+w_2+\cdots+w_9}\ , i = 1, \ldots., 9,$$

$$\overline{w}_1 = \frac{w_1}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{0}{3} = 0$$

$$\overline{w}_2 = \frac{w_2}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{0}{3} = 0$$

$$\overline{w}_3 = \frac{w_3}{w_1+\cdots+w_9} = \frac{1}{0+0+1+\cdots+1} = \frac{1}{3} = 0.333$$

$$\overline{w}_4 = \frac{w_4}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{0}{3} = 0$$

$$\overline{w}_5 = \frac{w_5}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{0}{3} = 0$$

$$\overline{w}_6 = \frac{w_6}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{1}{3} = 0.333$$

$$\overline{w}_7 = \frac{w_7}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{0}{3} = 0$$

$$\overline{w}_8 = \frac{w_8}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{0}{3} = 0$$

$$\overline{w}_9 = \frac{w_9}{w_1+\cdots+w_9} = \frac{0}{0+0+1+\cdots+1} = \frac{1}{3} = 0.333$$

Layer 4 (Defuzzify Layer):

$$\overline{w}_i z_i = O_i^4 = \overline{w}_i(p_i log10(x) + q_i y + r_i)$$

$$\overline{w}_1 z_1 = \overline{w}_1(p_1 log10(x) + q_1 y + r_1),$$

$$\overline{w}_1 z_1 = 0\big(-16.49 \times log10(10) + (-20.6 \ \times 1) + (-2.508)\big) = 0,$$

$$\overline{w}_2 z_2 = \overline{w}_2(p_2 log10(x) + q_2 y + r_2) = 0$$

$\overline{w}_3 z_3 = \overline{w}_3(p_3 log10(x) + q_3 y + r_3)$=$0.333 * (-10.59 * \log 10(10) +$ $(-6.394) * 1 + \quad (-46.22))$=-21.0469

$$\overline{w}_4 z_4 = \overline{w}_4(p_4 log10(x) + q_4 y + r_4) = 0$$

$$\overline{w}_5 z_5 = \overline{w}_5(p_5 log10(x) + q_5 y + r_5) = 0$$

$\overline{w}_6 z_6 = \overline{w}_6(p_6 log10(x) + q_6 y + r_6)$=$0.333 *$

$(-42.1 * \log 10(10) + -2.818 * 1 + -41.41)$=-28.7472

$\bar{w}_7 z_7 = \bar{w}_7(p_7 log10(x) + q_7 y + r_7)$=0

$\bar{w}_8 z_8 = \bar{w}_8(p_8 log10(x) + q_8 y + r_8)$=0

$\bar{w}_9 z_9 = \bar{w}_9(p_9 log10(x) + q_9 y + r_9)$=$0.333 *$

$(15.44 * \log 10(10) + 6.44 * 1 + -39.8)$ =-5.9674

Layer 5 (Total Output Layer):

$$z = \sum_i \bar{w}_i z_i = O_i^5 = \frac{\sum_i w_i z_i}{\sum_i w_i} = \frac{w_1 z_1 + \cdots + w_9 z_9}{w_1 + \cdots + w_9} = \bar{w}_1 z_1 + \cdots + \bar{w}_9 z_9$$

$$z = 0 + 0 + (-21.0469) + 0 + 0 + (-28.7472) + 0 + 0 + (-5.9674)$$
$$= -55.7615$$

Problem formulation of PSO trained LOG10DANFIS with distance and PSO generated random RSSI as inputs

Using PSO to obtain the random input together with ANFIS membership function parameters that best approximate the measured values. The minimum of the function $f(x) = ANFIS\text{-}RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2}$ for the following ANFIS with membership function parameters x within the range of $-69 \le x \le 41.2$ using the PSO algorithm with 50 particles (50 sets of random inputs and 50 sets of ANFIS membership function parameters) are obtained. The given ANFIS requires 45 parameters since it has 3 MFs with two inputs. 69 random inputs are also required since there are 69 distance points. The two result to 114 as the number of variables that are to be searched for by PSO. Where t is the actual value and y is the ANFIS predicted value. The following represent the formulation for the same.

The 50 initial positions (random inputs and membership function parameters) with 114 variables each given as;

x01 =[11.9666 22.0022………………22.8338];

x02 =[-45.0223 13.4031……………... -47.9387];

…………

x50 =[-49.9161 -8.6438………………. -29.0260];

and are to be applied as below.

Using the particles with initial positions above the detailed computations for iterations 1, 2 and 1000 are shown below.

Step1: choose the number of particles: x01, x02, ….., x50

The initial Population (i.e. the iteration number *t=0*) can be represented as $x_i^0 . i = 1,2, \ldots . ,50$:

x01 =[11.9666 22.0022………………22.8338];

x02 =[-45.0223 13.4031……………... -47.9387];

…………

x50 =[-49.9161 -8.6438………………. -29.0260];

Evaluate the objective function values as $f(x) = RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2}, i = 1,2, \ldots ,50$:

For f01 substituting the x01

x01 =[11.9666 22.0022………………22.8338];

random RSSI input and membership function parameters into the FIS in the APPENDIX II to obtain its RMSE value.

f01= 86.0455

For f02

% substituting the x02 random RSSI input and membership function parameters

f02=RMSEtrn=sqrt(mean((trndata(:,3)'-outtdl').^2));

f02= 86.0050

……….

For f50

% substituting the x50 random RSSI input and membership function parameters

f50= 57.9675

Let c1=2 and c2=2;

Set the initial velocities of each particle to zero:

$v_i^0 = 0, i.e. v_1^0 = v_2^0 = \cdots = v_0^5 = 0$

v01=[0 0 … 0]; v02=[0 0 … 0]; …. v50=[0 0… 0];

Step 2: set the iteration number as $t = 0 + 1 = 1$ and go to step 3

$$P_{best,i}^{t+1} = \begin{Bmatrix} P_{best,i}^{t} & \text{if } f_i^{t+1} > f_i^t \\ x_i^{t+1} & \text{if } f_i^{t+1} \le f_i^t \end{Bmatrix},$$

So,

pbp01 =[11.9666 22.0022………………22.8338];

pbp02 =[-45.0223 13.4031……………... -47.9387];

…………

pbp50 =[-49.9161 -8.6438………………. -29.0260];

Step 4: Find the global best by

$G_{best,1}^t = \min\{P_{best,i}^t\}$ where $i = 1,2,\dots,50$.

Since, the minimum personal best is $P_{best,35}^1$=[22.0309 -51.6608 …..   -64.5351]

gbp01= pbp035 =[22.0309 -51.6608 …..   -64.5351];

Step 5: considering the random numbers in the range (0,1) as

r11=rand(1,114)

r_1^1 =and

r21 = rand(1,114)

r_1^2 =and find the velocities of the particle by

$v_i^{t+1} = w * (v_i^t + c_1 r_i^t [P_{best,i}^t - x_i^t] + c_2 r_2^t [G_{best,i}^t - x_i^t]); i = 1, \dots, 50$

Where w=1 is the inertia coefficient and wdamp=0.99 damping ratio of the inertia coefficient

So

v11 =w* (v01 + c1* r11.*(pbp01-x01)+c2* r21.*(gbp01-x01));

v_1^1 =[-8.8160   -8.8160 … -8.8160];

$v_2^1 = [7.1858 - 8.6019\ \ldots\ 8.8160]$,

…..

$v_{50}^1 = [8.8160 - 8.8160 \ldots - 8.8160]$,

Setting minimum velocity as -8.8160 and maximum velocity as -8.8160 the above velocities are obtained

Step 6: Find the new values of $x_i^1, i = 1, \ldots$,50 by

$x_i^{t+1} = x_i^t + v_i^{t+1}$

So

x11=x01+v11=[8.0341 13.1863… 14.0178]

x12=x02+v12=[-36.2062 4.5871…-56.7547]

….

x150=x05+v150=[-41.1001 -3.7505… -37.8420]

Set minimum position as -69 and maximum position as 41.2

Step 7: Evaluate the objective function values as $f(x) = RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2}, i = 1,2, \ldots$,50:

Obtain f11 to f150 by substituting the x11 to x150 random RSSI input and membership function parameters into the FIS to obtain its RMSE value

f11= 66.9717, f12= 91.2839 … f150= 27.6944

Step 8: Stopping criterion:

If the terminal rule is not satisfied, go to step 2,

Otherwise, stop the iteration and output the results.

…….

Step2: Set the iteration number as $t = 998 + 1 = 999$, them go to step 3.

Step3: Find the personal best for each particle.

$$P_{best,1}^{999} = \begin{Bmatrix} P_{best,1}^{998} & \text{if } f_1^{999} > f_1^{998} \\ x_1^{999} & \text{if } f_1^{999} \leq f_1^{998} \end{Bmatrix},$$

$P_{best,1}^{999} = [-11.1310\ -15.9107, \ldots - 39.7965];$

$P_{best,2}^{999} = [-11.1316 - 15.9416 \ldots\ - 39.7963];$

……

$P_{best,50}^{999} = [-11.132 - 16.3522 \ldots\ - 39.8007];$

Step 4: find the global best

$G_{best,1}^{t} = \min\{P_{best,i}^{t}\}$ where $i = 1,2, \ldots ,50.$

$G_{best,1}^{999}$=[-11.1315 -15.9759 … -39.7980];

gbp999,1 =[-11.1315 -15.9759 … -39.7980];

Step 5: By considering the random numbers in the range (0.1) as

r11=rand[1,114]

r_1^1 =and

r21=rand[1,114]

r_1^2 =and find the velocities of the particle by

$v_i^{t+1} = w * (v_i^t + c_1 r_i^t [P_{best,i}^t - x_i^t] + c_2 r_2^t [G_{best,i}^t - x_i^t]); i = 1, \ldots ,50$

So

v1000,1 = w*(v999,1 + c1* r11.*(pbp999,1-x999,1)+c2* r21.*(gbp999,1-x999,1));

v_1^3 = [-0.0009 0.5682 … -0.0021];

v_2^3=[-1.461e-05 3.2212 … 0.0514];

……..

v_{50}^{1000}=[0.0002 0.3594 … 0.0078];

Set minimum velocity as -8.816 and maximum velocity as 8.816

Step 6: Find the new values of x_i^{1000} $, i = 1, \ldots , 50$ by

$x_i^{t+1} = x_i^t + v_i^{t+1}$

So $x_1^{1000} = x_1^{999} + v_1^{1000}$

x1000,1=x999,1+v1000,1

x_1^{1000}=[-11.1315 -15.6305 … -39.7985];

x1000,2=x999,2+v1000,2

x_2^{1000}= [-11.1316 -14.3149 … -39.7620];

…..

x1000,50=x999,50+v1000,50

x_{50}^{1000}=[-11.1320 -15.9550 … -39.7958];

Set the minimum position as -69 and maximum position as 41.2

Step 7: Evaluate the objective function values as $f(x) = RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2}\, , i = 1,2, \ldots ,50$:

For f1000,1 substituting the x1000,1 random RSSI input and membership function parameters into the FIS below to obtain its RMSE value

For x1000,1

f1000,1 = 0.0377, f1000,2= 0.1124, …, f1000,37=0.0026 … f1000,50 = 0.0764

Step 8: Stopping criterion:

If the terminal rule is not satisfied, go to step 2, otherwise stop the iteration and output the results.

Finally, the values $x_i^{1000}\, , i = 1, \ldots, 50$ did not converge, but the requirement to stop at iteration 1000 is reached. If the iterations continue, convergence is reached if the positions of all particles converge to similar values, then the method has converged and the corresponding value of x_i^t is the optimum solution. Therefore, the iterative process is stopped due to the requirement that it stops at iteration 1000. This gives an optimum solution of training RMSE as 0.0026 with testing RMSE of 2.931. The final best particle position i.e. the final global particle position gives the best random input RSSI values and the best membership function parameters for the best performing ANFIS.

Appendix III: The universal model parameters

```
m= input ('membership functions = ');
```

One slope model membership functions

```
m=[-30.4042  -44.4841  -17.1511  -39.8081  -19.3858  -21.2635  -16.4371  -23.2478  -
20.0000          -12.0442 -26.6433 -33.4410 -12.9429 -38.4691 -22.6328];
```

Dual slope model membership functions

```
m=[0.1319 3.029 0.652 -20 -12.04 0.1214 3.242 1.391 -20 -12.04 0.2256 4.11 3.572 -
20 20];
```

Multi-wall model membership functions

```
m=[-40.3386  -38.6915  -22.2768  -15.7458  -34.2936   -5.2165  -42.9084  -53.0827  -
44.0994  -50.2905  -32.6188  -27.1827  -28.2698  -20.0000  -23.0442];
```

Hata-Okomura-Rural model membership functions

```
m=[-163.7880    -60.7698    -35.9215  -136.4529  -140.1066  -140.2505  -142.2632    -
55.5749  -69.0993 -201.0000 -100.4240  -56.0283 -128.3882  -46.0372 -200.7711];
```

Hata-Okomura-suburban model membership functions

```
m=[-153.9954  -142.2307    -82.0268    -93.7246    -69.1132  -169.6921  -105.7039  -
122.4051 -181.4401 -176.1464  -89.6252  -32.9347 -144.2973  -79.1218 -174.2660];
```

Hata-Okomura-urban model membership functions

```
m=[-106.8653 -233.7770  -41.7293    1.3000 -224.5889 -234.0000    1.3000 -234.0000
-234.0000 -234.0000  -69.3173  -48.2855 -145.8358  -63.7711 -177.0896];
```

COST231-Hata-metropolitan model membership functions

```
m=[-140.9631 -144.9048  -20.3709   74.5600   16.4457 -204.7000  -11.8412  -43.7937
-138.2236 -111.9749  -68.9166  -47.2513 -133.6719  -38.0000 -155.2745];
```

COST231-Hata-suburban model membership functions

```
m=[-164.1418 -182.3528  -83.4471  -57.3453 -177.0805  -60.0873    0.9054  -12.2142
-61.3340 -210.0000 -152.3375  -71.3520 -129.1993  -40.7046 -144.8518];
```

COST231 model membership functions

m=[-140.9631 -144.9048 -20.3709   74.5600   16.4457 -204.7000 -11.8412 -43.7937 -138.2236 -111.9749 -68.9166 -47.2513 -133.6719 -38.0000 -155.2745];

Two-ray ground reflection model membership functions

m=[6.2783   -15.8521   50.3174   2.2200   26.3593   -68.8450   47.5228   55.6608 68.5609 -40.0000   -6.0234   -6.1718 -16.8286 -36.8161   35.2805];

Appendix IV: List of Publications Arising from the Research

The list below represents publications and presentations arising from the research undertaken in this thesis.

1. Omae M. Oteri, Ndungu E. N and Kibet P. L., “A Universal Wireless Signal Propagation Prediction Model Based on PSO Trained Modified ANFIS”, IJSER vol. 10, Issue 11, November 2019.
2. Omae M. Oteri, Ndungu E. N and Kibet P. L., “Indoor LOS Wi-Fi Signal Coverage Modelling Using PSO Trained LOG10D-ANFIS with Random Input”, IJSER vol. 10, Issue 11, November 2019.
3. Omae M. Oteri, Ndungu E N and Kibet P. L., “Mobile Internet Subscription Trends in Kenya to 2014”, IJSER, vol. 6, No. 2, February 2015.
4. Omae M. O, Ndungu E N and Kibet P. L., “Mobile Subscription, Penetration and Coverage Trends in Kenya's Telecommunication Sector”, IJARAI, vol. 4, No. 1, January 2015.
5. Omae M. O, Ndungu E N and Kibet P. L., “The Application of ANFIS-PSO trained in Signal Propagation Modelling for Indoor Wireless Communication Networks; A Review”, IJSER, vol. 5, No. 12, December 2014.
6. Omae M. O, Ndungu E N and Kibet P. L., “Internal and External Challenges Facing Telecommunications Industry in Kenya”, IJSER, vol. 4, No. 6, June 2013.
7. Omae M. O, Kibet P. L. and Kihato P. K., “Formulation of Improved ANFIS for Function Approximation in Theoretical Wireless Signal Propagation Modelling”, IEEE Delhi Section International Conference on Electrical, Electronics and Computer Engineering (IEEE DELCON-2022), IEEE Delhi section, Netaji Subhas University of Technology, Dwarka, New Delhi, India February 2022.
8. Omae M. O, Kibet P. L. and Kihato P. K., “Comparing the performance of ANFIS, LOG10-ANFIS and LOG10-PSO-ANFIS for universal theoretical wireless signal propagation prediction modelling”, Springer-International Conference on Optical and Wireless Technologies 2021 (OWT-2021), Jaipur India October 2021.

9. Omae M. O, Ndungu E N and Kibet P. L., “Comparing the performance of ANFIS, LOG10-ANFIS, LOG10-PSO-ANFIS and LOG10-PSO-ANFIS with random input in indoor Wi-Fi Signal propagation prediction”, SRI proceedings, May 2019.
10. Omae M. O, Ndungu E N and Kibet P. L., “Wi-Fi Signal Indoor LOS Coverage modelling using PSO-ANFIS”, SRI proceedings, May 2018.
11. Omae M. O, Ndungu E N and Kibet P. L., “Comparing the performance of ANFIS, PSO-ANFIS and PSO-ANFIS with random input in indoor Wi-Fi Signal propagation prediction”, SRI proceedings, May 2018.
12. Omae M. O, Ndungu E N and Kibet P. L., “Wi-Fi Signal Indoor LOS Coverage modelling using ANFIS”, SRI proceedings, May 2016.
13. Omae M. O, Ndungu E N and Kibet P. L., “Indoor LOS Wi-Fi Coverage Prediction using ANN”, SRI proceedings, May 2016.
14. Omae M. O, Ndungu E N and Kibet P. L., “Comparison of Wi-Fi Signal Indoor LOS Coverage and Propagation Through Walls”, SRI proceedings, May 2015.
15. Omae M. O, Ndungu E N and Kibet P. L., “Comparing the Performance of Linksys and D-link in LOS Conditions”, SRI proceedings, May 2015.
16. Omae M. O, Ndungu E N and Kibet P. L., “Mobile Internet Subscription Trends in Kenya”, SRI proceedings, May 2014.
17. Omae M. O, Ndungu E N and Kibet P. L., “Challenges Facing Telecommunications Industry in Kenya”, SRI proceedings, April 2013.
18. Omae M. O, Ndungu E N and Kibet P. L., “Mobile Subscription and Penetration Trends in Kenya's Telecommunication Sector”, SRI proceedings, April 2013.