EVALUATING THE POTENTIAL OF MULTI-PURPOSE USE OF IKPOBA DAM USING ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM AND ARTIFICIAL NEURAL NETWORK

BY

EGBIKI SUNDAY

PG/ENG/1411341

DEPARTMENT OF CIVIL ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN

NIGERIA

DECEMBER, 2019

EVALUATING THE POTENTIAL OF MULTI-PURPOSE USE OF IKPOBA DAM USING ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM AND ARTIFICIAL NEURAL NETWORK

BY

EGBIKI SUNDAY

PG/ENG/1411341

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DOCTORATE DEGREE (PhD) IN WATER RESOURCES AND ENVIRONMENTAL HEALTH ENGINEERING

IN

DEPARTMENT OF CIVIL ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN

NIGERIA

DECEMBER, 2019

CERTIFICATION

This research was carried out by Egbiki Sunday in partial fulfillment of the requirement for the award of Doctor of Philosophy degree in Civil Engineering (Water Resources and Environmental Health Engineering option) faculty of Engineering, University of Benin, Benin City, Edo State.

………………………………………….. ………………………………………………

Egbiki Sunday **Date**

Student

………………………………………….. ………………………………………………

Prof. J.O. Ehiorobo **Date**

Supervisor

……………………………………….. ………………………………………………

Prof. O. Orie **Date**

Head of Department

CERTIFICATION OF THESIS ON PLAGIARISM

We the undersigned attest and declare that the thesis of Egbiki Sunday, titled, **EVALUATING THE POTENTIAL OF MULTIPURPOSE USE OF IKPOBA DAM USING ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM AND ARTIFICIAL NEURAL NETWORK** has successfully passed the anti-plagiarism test and does not violate copyright.

Prof. J.O. Ehiorobo **Signature and Date**

Supervisor

………………………………

Prof. O. Orie **Signature and Date**

Head of Department

DEDICATION

If it has not been the LORD who was on my side, this research work would have been a failure. I give HIM all the glory especially for being the Chief Water Engineer whose wells, rivers and reservoirs never go dry.

ACKNOWLEDGEMENTS

If I can but see the world today, it is on someone's shoulder I stood to see. I give my profound gratitude to Prof. J.O. Ehiorobo (My Supervisor) and Prof. O.C. Izinyon for giving me the ample chance to carry out this research with them and they are the shoulders that visualize me to the world. They are my mentors and academic fathers in whom I am well pleased. May Almighty God prolong their lives to enjoy the fruits of their labours. I will not fail also to thank Prof. O. Orie, the current Head of Department Civil Engineering Department, University of Benin, Engr. (Dr.) J.O Okovido the immediate past Head of Department, and the entire staff of the Department Of Civil Engineering. I also thank other members such as Prof. O.E. Alutu, Prof. B.U. Anyata (late), Prof. E.O. Eze, Dr. S.O. Osuji, Dr. H.A.P. Audu, Dr. Nwankwo Ebuka, Dr. D.O.E. Osula (late), Dr. Lulu Bobor, Dr. Ngozi Ihimekpen, Dr. S.O. Iyeke, Engr. Solomon Okonofua, Dr. Prince Umasabor and the rest I could not mentioned here. I also wish to express my profound gratitude to Engr. (Dr.) Eguas Atikpo for his assistance and encouragement.

My appreciation also goes to my late parents, Chief Jacob and Mrs. Regina Egbiki for the educational foundation they laid for me. Rest in the Lord's bosom till we meet again to part no more! I am grateful to my amiable daughters, Flourish and Ornament Egbiki, my precious wife Mrs. Mercy Egbiki, and my younger brother Mr. Cusmas Egbiki for their continuous encouragement. I love you all. To the Group managing Director, Engr. Femi Akintunde, Alpha mead Group, Executive Director, Real Estate, Mr. Damola Akindorile, Alpha mead Development Company Limited and the entire staff and also Mr. Bonny Idiogbe (CEO A.Bonny & Sons) for the career foundation they laid for me. I appreciate you all for the goodwill and patience throughout this academic pursuit. God bless you all.

CONTENTS **PAGE**

Certification	ii
Certification of Thesis on Plagiarism	iii
Dedication	iv
Acknowledgements	v
Table of Contents	xi
List of Figures	xii
List of Tables	xiii
List of Plates	xv
Abstract	xvi
1.0. INTRODUCTION	**1-6**
1.1. Background of Study	1
1.2. Statement of the Problem	3
1.3. Aim and Objectives	4
1.4. Scope of Work	5
1.5. Justification for the Study	6
2.0. LITERATURE REVIEW	**7-73**
2.1. River discharge	7
2.1.1. Measurement of river discharge	8
2.1.2. Benefits and uses of river discharge data	11
2.2. Availability of river discharge data in Nigeria	11
2.3. Infilling of missing river discharge data	13
2.4. Data training, validation and testing	15
2.4.1. Data training	15

2.4.2. Data testing 15

2.4. 3.Data validation 16

2.5. Methods for modeling and forecasting river discharge 17

2.5.1. Application of time series conventional methods in water resources 20

2.6. Artificial neural networks (ANN) 23

2.6.1. Development of Artificial neural networks 25

2.6.2. ANN Topologies and Transfer Functions 29

2.6.2. 1. ANN Weights and Biases 30

2.6.2. 2. Epochs Generations and Transfer Functions 31

2.6.2. 3. Transfer Functions 32

2.6.3. Types of ANN 33

2.6.3. 1. Single Layer Perceptron 33

2.6.3. 2. Multi-Layer Perceptron 36

2.6.3. 3. Feed-Forward Neural Networks 38

2.6.3. 4. Feed-Backward Neural Networks 40

2.6.3. 5. Radial Basis Function Networks 41

2.6.3. 6. Back Propagation Neural Networks 42

2.6.4. Data training, validation and testing using ANN 44

2.6.4.1. Data training 45

2.6.4.2. Data testing 47

2.6.4.3. Data validation 49

2.6.5. Merits and Demerits of ANN 50

2.6.6. Application of ANN in water resources and hydrology 54

2.7. Adaptative neuro-fuzzy inference system (ANFIS) 55

2.7.1. ANFIS Architecture 57

2.7.2. ANFIS universal function approximator 60

2.7.3. ANFIS optimization algorithms 63

2.7.4. Data training, validation and testing using ANFIS 64

2.7.5. ANFIS initialization 66

2.7.6. Merits and Demerits of ANFIS 67

2.7.7. Application of ANFIS in water resources and hydrology 69

2.8. Performance evaluation criteria 69

2.8.1. Coefficient of correction 70

2.8.2. Coefficient of determination 71

2.8.3. Mean square error 72

2.8.4. Nash-Sutcliffe efficiency 73

2.8.5. Index of agreement 73

3.0. RESEARCH METHODOLOGY 75-92

3.1. The Study Area 75

3.2. Collection and Pre-processing of Discharge Data 77

3.3. Models Development 80

3.4. Modeling Procedures 81

3.4.1. Discharge Modeling Using Adaptive Neuro-Fuzzy Inference System 82

3.4.2. Discharge Modeling Using Artificial Neural Networks 89

3.5. Performance Evaluation Criteria 92

4.0. RESULTS AND DISCUSSIONS 93-144

4.1. Results and Discussion of ANFIS models 93

4.2. Results and Discussion of ANN models 108

4.2.1. Results and Discussion of ANN_LM models 108

4.2.2. Results and Discussion of ANN_SCG models 117

4.2.3. Results and Discussion of ANN_BR models 126

4.3. Assessment of the possibilities of converting Ikpoba reservoir to a multipurpose one 139

5.0. FINDINGS, CONCLUSIONS AND RECOMMENDATIONS 145-147

5.1. Findings 145

5.2. Conclusions 146

5.3. Recommendations 146

5.4. Contribution to Knowledge 147

5.5. Recommendations for future researches 147

References 148

LIST OF FIGURES **PAGE**

2. 1. Rating curve depth versus flow 8

2.2. McCulloch-Pitts networks 24

2.3. Parts of human neuron 26

2.4. Layers of neurons 27

2.5a. ANN topologies 28

2.5b. Layers of neuron with error propagation 29

2.6. Assigning weights and biases to ANN 30

2.7. Global and local minima of error function 31

2.8. Topology of a transfer function 32

2.9. Topology of a single layer perceptron 34

2.10. XOR problem with single layer perceptron 35

2.11. Feedforeward phase of neural networks 36

2.12. Feedforeward phase of with back error propagation 37

2.13. Multi layered feedforeward network with 3 inputs layers 39

2.14. Multi layered feedforeward network with 4 inputs layers 39

2.15. Multi layered feedbackward network with 4 inputs layers 41

2.16. Radial basic function network with 3 inputs layers 42

2.17. Back propagation neural network 4 hidden layers 44

2.18. Under-fitting of ANN data 52

2.19. Over-fitting of ANN data 53

2.20. Good-fitting of ANN data 54

2.21. ANFIS Architecture model 58

2.22. First order Sugeno fuzzy model 63

2.23. A bell-shaped function 65

2.24. Example of 2-D cluster data 67

3.1. Location Map of Ikpoba dam along Ikpoba River 76

3.2. Mean daily discharge of Ikopba river 79

3.3. Mean daily precipitation of Ikopba river 79

3.4. Mean daily temperature of Ikopba river 80

3.3. ANFIS model network for model-1 84

3.4. ANFIS deffuzification process for model-1 84
3.5. ANFIS model network for model-2 85
3.6. ANFIS deffuzification process for model-2 85
3.7. ANFIS model network for model-3 86
3.8. ANFIS deffuzification process for model-3 86
3.9. ANFIS model network for model-4 87
3.10. ANFIS deffuzification process for model-4 87
3.11. ANFIS model network for model-5 88
3.12. ANFIS deffuzification process for model-5 88
3.13. ANN network model for model-1 90
3.14. ANN network model for model-2 90
3.15. ANN network model for model-3 91
3.16. ANN network model for model-4 91
3.19. ANN network model for model-5 91
4.1. Performance in model-1 using R in ANFIS 94
4.2. Performance in model-1 using MSE in ANFIS 95
4.3. Performance in model-2 using R in ANFIS 95
4.4. Performance in model-2 using MSE in ANFIS 96
4.5. Performance in model-3 using R in ANFIS 96
4.6. Performance in model-3 using MSE in ANFIS 97
4.7. Performance in model-4 using R in ANFIS 97
4.8. Performance in model-4 using MSE in ANFIS 98
4.9. Performance in all the models using training data sets in ANFIS 99
4.10. Performance in all the models using testing data sets in ANFIS 99
4.11. Performance in all the models using all data sets in ANFI 100
4.12. Comparing predicted discharge and targeted discharge using test, train and all data sets 100
4.13. Performance in model-4 using R in ANFIS (1999 data sets) 101
4.14. Performance in model-4 using MSE in ANFIS (1999 data sets) 101
4.15. Performance in model-4 using R in ANFIS (2000 data sets) 102
4.16. Performance in model-4 using MSE in ANFIS (2000 data sets) 102
4.17. Performance in model-5 using R in ANFIS 103
4.18. Performance in model-5 using MSE in ANFIS 104

4.19. Performance evaluation in model-5 using ANFIS 105
4.20. Performance evaluation in model-5 using ANFIS (training phase) 106
4.21. Performance evaluation in model-5 using ANFIS (testing phase) 106
4.22. Performance evaluation in model-5 using ANFIS (all phase) 107
4.23. Performance in model-1 using R in ANN (LM) 110
4.24. Performance in model-1 using MSE in ANN (LM) 110
4.25. Performance in model-2 using R in ANN (LM) 111
4.26. Performance in model-2 using MSE in ANN (LM) 111
4.27. Performance in model-3 using R in ANN (LM) 112
4.28. Performance in model-3 using MSE in ANN (LM) 112
4.29. Performance in model-4 using R in ANN (LM) 113
4.30. Performance in model-4 using MSE in ANN (LM) 113
4.31. Performance in all the models using training data sets in ANN (LM) 114
4.32. Performance in all the models using testing data sets in ANN (LM) 115
4.33. Performance in all the models using all data sets in ANN (LM) 115
4.34. Comparing predicted discharge and target discharge using test, train and all data sets (LM) 116
4.35. Performance in model-1 using R in ANN (SCG) 118
4.36. Performance in model-1 using MSE in ANN (SCG) 118
4.37. Performance in model-2 using R in ANN (SCG) 119
4.38. Performance in model-2 using MSE in ANN (SCG) 119
4.39. Performance in model-3 using R in ANN (SCG) 120
4.40. Performance in model-3 using MSE in ANN (SCG) 120
4.41. Performance in model-4 using R in ANN (SCG) 121
4.42. Performance in model-4 using MSE in ANN (SCG) 121
4.43. Performance in all the models using training data sets in ANN (SCG) 122
4.44. Performance in all the models using testing data sets in ANN (SCG) 123
4.45. Performance in all the models using all data sets in ANN (SCG) 123
4.46. Comparing predicted discharge and target discharge using test, train and all data sets (SCG) 124
4.47. Performance in model-1 using R in ANN (BR) 125
4.48. Performance in model-1 using MSE in ANN (BR) 126
4.49. Performance in model-2 using R in ANN (BR) 126
4.50. Performance in model-2 using MSE in ANN (BR) 127

4.51. Performance in model-3 using R in ANN (BR) 127
4.52. Performance in model-3 using MSE in ANN (BR) 128
4.53. Performance in model-4 using R in ANN (BR) 128
4.54. Performance in model-4 using MSE in ANN ((BR) 129
4.55. Performance in all the models using training data sets in ANN (BR) 130
4.56. Performance in all the models using testing data sets in ANN (BR) 130
4.57. Performance in all the models using all data sets in ANN (BR) 131
4.58. Comparing predicted discharge and target discharge using test, train and all data sets (BR) 131
4.59. Performance evaluation of ANFIS and ANN using training data sets in model-4 135
4.60. Performance evaluation of ANFIS and ANN using testing data sets in model-4 135
4.61. Performance evaluation of ANFIS and ANN using all data sets in model-4 136
4.62. Performance in model-5 using R in ANN (LM) 137
4.63. Performance in model-5 using MSE in ANN (LM) 138
4.64. Performance in model-5 using R in ANN (SCG) 138
4.65. Performance in model-5 using MSE in ANN (SCG) 139
4.66. Performance in model-5 using R in ANN (BR) 139
4.67. Performance in model-5 using MSE in ANN ((BR) 140
4.68. Performance in model-5 using ANFIS, ANN_LM, ANN_SCG, ANN_BR (training) 143
4.69. Performance in model-5 using ANFIS, ANN_LM, ANN_SCG, ANN_BR (testing) 143
4.70. Performance in model-5 using ANFIS, ANN_LM, ANN_SCG, ANN_BR (combined) 144
4.71. % over estimation of the discharge in model-4 using ANFIS and ANN 145
4.72. % over estimation of the discharge in model-5 using ANFIS and ANN 146
4.73. Mean monthly discharge of Ikpoba river (1991-1995) 146
4.74. Run-off volume of Ikpoba river (1991-1995) 149

LIST OF TABLES **PAGE**

3.1. Ikpoba River Hydrological Gauging Station Parameters 77

3.2. Structure of the forecasting models 81

4. 1. Comparison among the four models using performance evaluation criteria in ANFIS 98

4. 2. Performance of model-4 with 1999 and 2000 data sets 103

4. 3. Comparison of model-5 using performance evaluation criteria in ANFIS 104

4. 4.Comparison among the five models using performance evaluation criteria in ANFIS 105

4. 5. Performance of ANN_LM models 114

4. 6. Performance of ANN_SCD models 127

4.7. Performance of ANN_BR models 129

4.8.Comparison of ANN model-4 with 1999 and 2000 data sets using performance evaluation 133

4.9. Comparison of ANFIS model-4 and ANN model-4 using performance evaluation 134

4.10. Comparison of ANN model-5 using performance evaluation for LM, SCG and BR 141

4.11. Comparison of ANFIS model-5 and ANN model-5 using performance evaluation 142

4.12. Mean daily discharge of Ikpoba river (1991-1995) 148

4.13. Run-off volume of Ikpoba river (1991-1995) 148

4.14. Run-off volume of Ikpoba river (1999-2000) 149

LIST OF PLATES   PAGE

2. 1. Measuring river width 9

2. 2. Measuring river depth 9

2. 3. Using a flow meter to measure the velocity of a river 10

Abstract

In this study, the potential of Ikpoba river dam being used as a multipurpose dam was evaluated. Before the evaluation, the flow regime behaviour of the river was modelled and predicted using adaptive neuro-fuzzy inference system (ANFIS) and artificial neural network (ANN) in MATLAB software. The river daily discharge, temperature and precipitation data sets from 1991 to 1995 were used for the prediction. In applying ANFIS using hybrid algorithm, five different models: model-1, model-2, model-3, model-4 and model-5 were created using 1995 data sets as the target outputs in all the five models. Only discharge data sets for 1994; 1994 and 1993; 1994, 1993 and 1992; 1994, 1993, 1992 and 19991 were used as the input data sets for model-1 to model-4 respectively. Model-5 was created by indexing monthly temperature and precipitation into model-4 to see the effect of climate change on the models. ANN was also applied to the same models as created with ANFIS. In ANN, three training algorithms; Levenberg-Marquardt (LM), Scaled Conjugate Gradient (SCG) and Bayesian Regularization (BR) were used. Five performance evaluation criteria namely coefficient of correlation (R), coefficient of determination (R^2), mean square error (MSE), modelling efficiency (E) and index of agreement (IOA) were used for comparative analysis.

The results of both ANFIS and ANN using the five performance evaluation criteria (R, R^2, MSE, E and IOA) showed that model-5 (when the effect of climate change was incorporated) performed better than the other four models. The training phase in model-5 of ANFIS showed an over-estimation of 0.043% of the observed target output sets while an over-estimation of 0.044% was observed in the testing phase. The training phase in model-5 of ANN (LM) showed an over-estimation of 0.11% of the observed target output sets while an over-estimation of 0.14% was observed in the testing phase. The training phase in model-5 of ANN (SCG) showed an over-estimation of 0.21% of the observed target output sets while an over-estimation of 0.31% was observed in the testing phase. The training phase in model-5 of ANN (BR) showed an over-estimation of 0.17% of the observed target output sets while an over-estimation of 0.19% was observed in the testing phase.

It was therefore concluded that ANFIS performed better than ANN in all the five models and that ANN (LM) performed best followed by ANN (BR) and ANN (SCG) in the ANN models. When the potential of Ikpoba dam being used as a multipurpose dam was evaluated, it was discovered that the dam with ultimate water pumping capacity of 160 x 10^6 liters/day could also be utilized to produce 5.26MW of power monthly (with discharge of 31.9m^3/s) using a hydropower plant. The annual volume of water in the reservoir available for this hydropower scheme is 0.523 x $10^6 m^3$.

CHAPTER 1

1.0. INTRODUCTION

1.1. Background of Study

From time immemorial, dams were built as single purpose dams either for water supply or for irrigation. Dams are the backbones upon which the development and management of water resources of river basins are built. As human beings become civilized and with increased population which exerts much pressure on the environment, the need for water demand, irrigation schemes, flood measures/control, tourism, navigation processes and hydropower schemes for energy production have increased correspondingly. This situation brings about the concept of multi-purpose dams which are dams that can be used for more than one purpose. Multipurpose dams are important projects for developing countries, because multiple benefits can be derived from single investment. However, the efficient use of river runoff for such activities have been greatly impaired because of flow irregularities due to environmental seasonal variations and climatic change (WCD, 2000). Also, when dams are used for more than one purpose, optimum operation of each function cannot be guaranteed as conflicting demands are placed on one another. For such a case, the flow regime behaviour within the river basin must be well understood and one sure way of understanding this is modeling and forecasting of river discharge.

Modeling and forecasting of river discharge is of vital importance in water resources management and hydrology. It helps in planning, operational analysis and efficient management of reservoirs, flood control measures, hydraulic design of structures such as dams, weirs, bridge crossings, sluice gates, barrages, stilling basins and spillways. It also plays vital roles in the management of hydropower and hydro peaking, modeling of river sediment transports, aggradations and armouring, modeling of river ecohydraulic behaviour and aquatic lives and modeling of runoff and precipitation (Firat, 2007).

Traditionally, many mathematical and conventional time series methods have been used in forecasting river discharges. Such methods include time series models such as linear regression (LR) (Maier and Dandi, 1996), multiple linear regressions (MLR) (Maier and Dandi, 1996), multivariate regression (MR) (Cogger, 2010), auto regression (AR) (Maity *et.al.* 2010) and auto regressive moving average (ARMA) (Wong, *et al.*, 2010). Also, Auto regressive integrated moving average (ARIMA) (Maity *et al*., 2010), autoregressive moving average with exogenous inputs (ARMAX) (Wong, *et al.,* 2010), nearest neighbour method (NNM) (Emiroglu, *et al*., 2011), support vector machine (SVM) and Monte Carlo simulation (Wang *et al.,* 2009) have been extensively used. All these methods have one or the other inherent problems in their effective forecasting because they assume linearity and stationarity n their modeling. These assumptions make the modeling of a non-linear and dynamic hydrological phenomenon produced inaccurate and ineffective forecasts (Firat, 2007).

In the search for effective methods for forecasting, soft computing methods have become handy. Recently, Artificial Intelligence (AI) models such as Artificial Neural Networks (ANN), Fuzzy Logic Inference System (FLIS) and Pattern Recognition (PR) which mimic the behavior of the human brains have been increasingly and extensively used in the context of hydrological forecasting. The main advantages of these soft computing models are that they are data driven models and they don't need prior knowledge of the models under consideration (Chang and Chang, 2006).

Artificial Neural Networks (ANNs) or simply Neural Networks (NNs), which have been extensively used in water resources management, mimic the behaviour of the human brain. They are helpful and efficient in coping with systems that are deterministically or stochastically difficult to be described. The Fuzzy logic Inference (FLI) approach is also one of the approaches that is gaining popularity in the field of water resources managements these days. It is inherently based on the linguistic uncertainty expression rather than numerical uncertainty.

Although both ANN and FLI have a lot of usefulness, yet there are still little problems with them (Owen, 2001). The main drawback of fuzzy logic modeling is that there is no systematic procedure when designing a fuzzy controller (Firat and Gungor, 2007). This is an advantage of ANN. The disadvantages in using ANNs include its "black box" nature, greater computational burden (an advantage of Fuzzy logic), proneness to over-fitting (an advantage of Fuzzy logic), and the empirical nature of model development. Since most of the advantages of ANN are the disadvantages of FLI and most of the advantages of FLI are the disadvantages of ANN; to overcome these problems and increase efficiency, both ANN and FL need to be coupled together. This hybrid combination is called **Adaptive Neuro-Fuzzy Inference System (ANFIS) or simply Neuro Fuzzy System (NFS)** and is the subject matter of this research.

1.2. Statement of the Problem

The 7th goal of the Millennium Development Goals (MDGs) of the United Nation relates to the enhancement of environmental sustainability by integrating the concepts or principles of sustainable development into the country's policies and programmes so that loss of environmental resources can be avoided. In March, 2015, the Federal Government of Nigeria tried to imbibe this policy by initiating the plan to convert all the single-purpose dams/reservoirs in Nigeria to multipurpose ones. This will enable all the dams/reservoirs to be used as multi-purpose ones that can function in any of the following areas; hydropower generation, irrigation, recreation activities, water supply, flood control, navigation, sanitation, sediment control, groundwater recharge and tourism.

Ikpoba river dam/reservoir is one of such single purpose dam/reservoir in Nigeria used for only water supply which can potentially benefit from such plan. However, to convert single purpose dam/reservoir to multipurpose one, the discharge or flow system behaviour of the dammed river needs to be properly known and understood and especially with due regards to seasonal flow variations and climatic irregularities. Unfortunately, most river basins in developing countries including Nigeria are poorly gauged and as a result, they lack discharge data needed for hydrological modeling and watershed management. This is the case of the dammed river (Ikpoba River dam) under study. Hence, modeling and forecasting of river discharge is a method to address the problem at hand. This will help to augment insufficient data through filling missing discharge data and prediction of future discharge data needed for hydrological modeling and watershed management. Ikpoba River as the case study for this study is poorly gauged and lack sufficient and continuous discharge data. Forecasting models of river flow/discharge using ANFIS are yet to be developed for Ikpoba river basin/catchment to the best of my knowledge. This study will therefore evaluate and develop models needed to forecast the discharge or flow system behaviour of Ikpoba River to evaluate the potential of Ikpoba dam being used as multipurpose dam.

1.3. Aim and Objectives of Study

The main aim of this study is to evaluate the potential of multi-purpose use of Ikpoba river dam using Adaptive Neuro-Fuzzy Inference System (ANFIS) and Artificial Neural Network (ANN).

The objectives of the study are to:

i. develop ANFIS and ANN models for predicting Ikpoba river discharge incorporating climate change effect indexed by precipitation and temperature.
ii. carry out training, testing and validation of the formulated ANFIS and ANN models.
iii. carry out evaluation performance of the formulated models using statistical approaches such as coefficient of correlation (R), coefficient of determination (R^2), mean square error (MSE), Nash-Sutcliffe or modeling efficiency (E) and index of agreement (IOA).

iv. carry out comparison between the formulated ANFIS and ANN models in order to select the best models and approach.
v. carry out an evaluation of the potential of Ikpoba river dam/reservoir being used as a multipurpose one based on the selected model and forecasts.

1.4. Scope of Work

This dissertation work is limited to the application of ANFIS and ANN for Ikpoba river discharge forecast using MATLAB including the evaluation of the possibility of converting Ikpoba river dam to a multipurpose using statistical criteria.

The scope of works includes the following:

i. Collection of available daily and monthly temperature, precipitation, discharge and run-off data for Ikpoba river.
ii. Screening and examination of the available data to remove outliers.
iii. Development of networks models for ANFIS and ANN.
iv. Training, testing and cross validation of the discharge data. Three backpropagation training algorithms such as Levenberg-Marquardt (LM), Scaled Conjugate Gradient (SCG) and Bayesian Regularization (BR) were used to train the various ANN network models created and hybrid algorithms was used for ANFIS network models created.
v. Statistical performance evaluation criteria such as coefficient of correlation (R), coefficient of determination (R^2), mean square error (MSE), Nash-Sutcliffe or modeling efficiency (E) and index of agreement (IOA) were used to make comparisons among the three training algorithms.
vi. The performance criteria were used to predict the best model which was used to evaluate the potential of Ikpoba dam/reservoir being used as a multipurpose one needed in the Water-Energy-Food Nexus.

1.5. Justification for the Study

To help reduce the risk in any decision taken at any given point of interest and to enhance environmental sustainability and disaster mitigation, reliable forecast are needed. Generally, this research study is expected to provide useful data to all the stakeholders involved in water resources management. It will provide useful information for water resources managers and hydraulic engineers in planning efficient operation of multi-purpose reservoirs and, navigation and water pollution control. It will also help in planning effective flood control measure and management of the hydropower plants. It will also provide useful information to the hydraulic engineers in designing hydraulic structures such as dams, weirs, bridge crossings, sluice gates, barrages, stilling basins and spillways. It can also play a great role in flood frequency analysis and control measures. During the analysis and evaluation of river sediment transport, aggradations and armouring, river discharge forecasting is of great importance to the hydraulic engineers. River discharge forecast also plays important roles in evaluating the ecohydraulic behaviour and aquatic lives of a river. On some rivers bearing heavy pollution loads, information from discharge forecasting can benefit the regulation agencies in regulating the discharge of wastes into rivers. Therefore, for successful water resource management tools to be developed and implemented, the analysis and forecast of river discharge data is often require as it also plays a paramount role in the general management of watersheds.

CHAPTER 2

2.0. LITERATURE REVIEW

Built Dams coupled with reservoirs usually serve at least one of the following purposes: hydropower generation, irrigation for agriculture, water supply, flood control, inland navigation, recreation, sediment control, groundwater recharge, fisheries, and tourism (Strandhagen *et al.,* 2006). Multi-purpose dams are commonly used for water supply, hydropower generation and water storage for irrigation. Sometimes, it can be used in controlling flood in areas prone to flooding. Nevertheless, about 70% of large dams are still single-purpose when globally scaled (SADC, 2011). Using dams for multi-purpose activities makes strategic planning system, funding investment and implementation strategies a major problem for the actual realization of the projects. However, multiple purposes dams are still gaining extensive popularity, especially in developing countries due to the fact that several economic and social benefits from a single investment can be derived. Well planned and managed multi-purpose dams can help in meeting and ameliorating some of the major development challenges we face today. They provide renewable energy through hydropower scheme, storage reservoirs for drinking water supply, agricultural food production through irrigation and flood control measure which are rooted in the energy, water and food nexus.

2.1. River Discharge

The total volume of water flowing through a river channel at any given point in time is called river discharge and it is usually measured in cubic meters per second (m^3/s) (Strandhagen *et al.*, 2006). River stage (or stream stage, or gauge height or simply stage) is the elevation of water surface in the channel above a referenced datum usually called mean sea level or above an established datum where the stage or elevation is zero and is usually measured in meter. This zero stage level is arbitrarily established, but is often very close to the streambed level and is measured in meters (m). River discharge depends on precipitation, evapotranspiration and storage factors (Strandhagen *et al.*, 2006). The relationship between river depth and flow (depth-flow) as shown in Figure 2.1 is called rating curve.

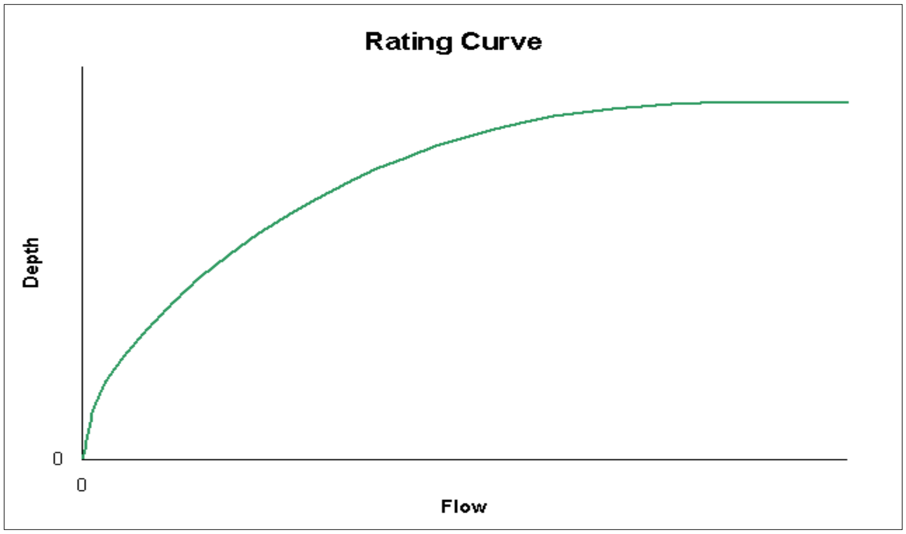


Figure 2.1: Rating curve depth versus flow *(http://water.usgs.gov/edu/streamflow3.html)*

2.1.1. Measurement of River Discharge

Generally, river discharge is the water area in a given channel cross section multiplied by the average velocity of the water in that cross section. This area occupied by the water is the depth of the channel multiplied by the width of the channel.

Area (m^2) = depth (m) x width (m) 2.1

Discharge (m^3/s) = Area (m^2) x Velocity (m/s) 2.2

A measuring tape is usually used to measure the width of the river channel as shown in Plate 2.1.

Plate 2.1: Measuring river width. *(http://water.usgs.gov/edu/streamflow3.html*)

For the river channel depth, a meter ruler or ranging pole is normally used and measurements are taken at regular 30cm to 50cm intervals (depending on the channel size) as shown in Plate 2.2 (http://water.usgs.gov/edu/measureflow.html).

Plate 2.2: Measuring river depth. (http://water.usgs.gov/edu/measureflow.html)

For the velocity, the simplest traditional method is to use a floating object and noting the time and distance the floating object will arrive at a particular location. To compute the velocity, the time is multiplied by the distance. The second method is the use of a flow meter or vane. This helps to record the number of revolutions as water passes over the flow meter or vane as shown in Plate 2.3.

Plate 2.3: Measuring velocity of a river with a flow meter
(http://water.usgs.gov/edu/measureflow.html)

In the United State of America, The United State Geological Survey (USGS) uses several methods and different types of equipment to measure velocity and cross-sectional area of a river. The ones most commonly used are current meter and Acoustic Doppler Current Profiler (ADCP) (http://water.usgs.gov/edu/measureflow.html). The main benefit derived from using ADCP is that it helps to reduce the time taken to make a discharge measurement.

2.1. 2. Benefits/Uses of River Discharge Data

The ultimate goal of data collection in hydrology and water resources, be it precipitation measurements, water-level recordings, discharge gauging, groundwater monitoring and water quality sampling is to provide adequate good quality data useful for decision-making in water resources management as well as in the wide range of operational applications and in research. Stream gauges helps to provide discharge data/information needed for effective decision making in terms of flood prediction, integrated watershed management, engineering design, research and development, operation of hydraulic structures, etc. for sustainable development.

Generally, river discharge data can be utilized in the following areas (Strandhagen *et al.,* 2006):

i. Integrated watershed management including hydropower management
ii. Design of hydraulic structures such as bridges, dams, weirs, stilling basin, etc.
iii. Flood frequency analysis and flood control measures
iv. Sediment transport analysis and river armouring
v. River basin management and ecohydraulic simulation
vi. Regional flood frequency analysis for gauged and ungauged stations

2.2. Availability of River Discharge Data in Nigeria

Following the 1972-74 drought in Nigeria which many described as the worst ever experienced in West, it was not a surprising that the Supreme Military Council promulgated decree 25 of 1976, as a swift move towards the development of Nigeria’s water resources as reported by Vanguard Newspaper on April 3, 2013. Accordingly, that gave birth to 12 River Basin Development Authorities (RBDAs), to harness the nation’s water resources and optimize its agricultural resources for food sufficiency.

The RBDAs include the following as reported by Vanguard Newspaper on April 3, 2013:

1. Anambra - Imo River Basin Development Authority
2. Benin - Owena River Basin Development Authority
3. Chad Basin Development Authority
4. Cross River Basin Development Authority
5. Hadejia - Jamaare River Basin Development Authority
6. Lower Benue River Basin Development Authority
7. Lower Niger River Basin Development Authority
8. Niger Delta Basin Development Authority
9. Ogun/Osun River Basin Development Authority
10. Sokoto Rima River Basin Development Authority
11. Upper Benue River Basin Development Authority
12. Upper Niger River Basin Development Authority

This development, reportedly raised hope among the populace, because it was assumed that the RBDs, would apart from agricultural needs, provide other basic needs such as river data collection and management associated with water resources. Instructively, the RBDAs were primarily established to provide water for irrigation and domestic water supply, improvement of navigation, hydro-electric power generation, and recreation facilities and fisheries projects. The basins were also expected to engender big plantation farming and encourage the establishment of industrial complexes that could bring the private and public sectors in joint business partnership. Additionally, RBDAs were expected to bridge the gap between the rural and urban centers by taking development to the grass roots and discouraging migration from the rural areas to the urban centers as reported by Vanguard Newspaper on April 3, 2013.

But over the years, these river basins and other rivers in Nigeria have been poorly gauged (World Bank, 2003). Discharge measurements were neglected, and where possibly carried out; data were poorly managed and stored as reported by Vanguard Newspaper on April 3, 2013. Most of the discharge data are not updated and most also contain missing data from measurement periods (*http://www.nigeria.gov.ng/fed_min_water_resources.aspx).* That is why it is important to carry out river discharge forecasting for effective future management of the river basins and the infilling of the various missing discharge data in most of the rivers.

2.3. Infilling of Missing River Discharge Data

Recording and archiving of river discharge data are very important since they are valuable assets needed for the sustainable management of water resources worldwide. It also serves as both indicators of past hydrological variability and fundamental contributors to future hydrological models for prediction of data behaviour (Marsh, 2002). The accuracy or completeness of such data records plays a vital role for their effective utilization. Any lapse or missing data can affect the accuracy and calculation of important summary hydrological statistics such as monthly and daily inflow. This can also inhibit the analysis and interpretation of flow variability (Marsh, 2002).

There are different ways to handle missing data. Most of the methods commonly used assume that missing data are missing completely at random (Rubin, 1976). The first method is the deletion method which is the one used in common statistical software like Minitab, Stata, SPSS and R (Enders, 2010). The main drawback in this method is that the sample size are inconsistent which can lead to problems in computing standard errors (Baraldi and Enders, 2010). Mean imputation is another method in which the missing value on a certain variable is replaced by the mean of the available cases. This method maintains the sample size and is easy to use, but the variability in the data is reduced, so the standard deviations and the variance estimates tend to be underestimated. The magnitude of the covariances and correlation also decreases by restricting the variability and this method often causes biased estimates, irrespective of the underlying missing data mechanism (Enders, 2010; Eekhout *et al*, 2013).

Single regression imputation method is another method in which the imputed value is predicted from a regression equation. In this method, the information in the complete observations is used to predict the values of the missing observations and it assumes that the imputed values fall directly on a regression line with a nonzero slope. This implies a correlation of 1 between the predictors and the missing outcome variable. Unlike to the mean imputation method, regression imputation will overestimate the correlations, however, the variances and covariances are under-estimated (Enders, 2010).

Also, multiple imputation method is another method in which the imputation process is repeated multiple times resulting in multiple imputed datasets. In this method, the imputation uncertainty is accounted for by creating these multiple datasets. The multiple imputation process contains three phases: the imputation phase, the analysis phase and the pooling phase (Rubin, 1987; Shafer, 1997; Van Buuren, 2012). This method works well when missing data are missing at random (Eekhout *et al.*, 2013). In the imputation method, the variables that are related to the missing data can be included and this helps to reduce bias and estimates variables more precisely (Van Buuren, 2012).

2.4. Data Training, Testing and Validation

Before data can be used in any field, the data need to be pre-processed, managed and stored in a useable form. Most hydrological data such as precipitation, runoff, discharge, etc contain missing data (Van Buuren, 2012). Such data need to be pre-processed by filling in the missing data before used. Various methods for filling of missing data have been discussed in the preceding section. In soft computing methods for forecasting, data set are trained, tested and validated in order to obtain reliable information about them. This is called post processing of data.

2.4.1. Data Training

To adjust the weights on neural networks such as in ANN and ANFIS, training data set is generally used (Geisser, 1993). It helps to adjust the weights and the bias of the network until a good performance is obtained. Training data set is a percentage (usually the highest) of the overall data set that is used for training processes. For example, it can be 70% training, 20% testing and 10% validation or 60% training, 20% testing and 20% validation.

2.4.2. Data Testing

Unlike data training and validation, testing data set is only normally used for testing the final solution. This will help to actually confirm the accuracy of the predictive power of the network (Geisser, 1993). Testing data set helps you to check and confirm how generalizing the resulting data set is. Testing is normally done with independent data set.

2.4.3. Data Validation

Data Validation is different from data testing. It is usually used for parameter selection and to avoid over-fitting. Validation can be regarded as a part of training data set, because it is used to build models, neural networks or others (Geisser, 1993). Validation data set is used to verify and minimize over-fitting of data. In validation data set, data adjustments are not done on the weights of the network data but they are continually verified in such a way that if there is any increase in accuracy over the training data set, there will also be a corresponding increase in accuracy over a data set that has not been presented to the network before, or at least training has not been done of the network (i.e. validation data set) (Geisser, 1993). Data over-fitting therefore occurs if an increase in the accuracy over the training data gives a corresponding decrease over the validation data set or even remain unchanged (Kohavi, 1995). In this regard, the training process needs to be stopped. If the model is non-linear (like ANN and ANFIS), the training is done on a training set only (Grossman *et al*., 2010). It is therefore possible to get 100% accuracy of data validation and over-fit, leading to poor performance on data testing. In such situations, data sets independent from the training data set, is used for parameter selection for which reason some professionals used the word "Cross Validation’’ on the contrary, a situation where the test is only used to test the performance of a trained model.

Cross-validation, also called **rotation estimation** (Geisser, 1993) is a validation technique for assessing how the results of a statistical analysis will generalize or respond to an independent data set. It is mainly used in a situation where prediction is the main objective, and one wants to estimate how accurate a predictive model will perform in real situation. In performing a prediction model, the model is usually given a known data sets on which training will be run (data training), and a dataset of unknown data against which the model will be tested (data testing) (Dubitzky *et al.,* 2007). The overall goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the validation phase). This will help to limit problems like over-fitting and give an insight or clue on how the model will generalize to an independent dataset (Efron and Tibshirani, 1997).

Also, one of the main advantages for using cross-validation instead of the conventional one (e.g. partitioning the data set into two sets of 75% for training and 25% for test) is that, the error (e.g. Root

Mean Square Error) on the training set in the conventional validation is not a true estimator of model performance. This is because the error on the test data set does not properly represent the actual assessment of model performance (Grossman *et al*., 2010). The reason for this is that there is not enough data available or there is not a good distribution and spread of data to partition it into separate training and test sets in the conventional validation method. In such situation, cross-validation will remove the error performance (Grossman *et al*., 2010). Cross-validation has one main limitation for its use. It only yields meaningful results if the validation set and training set are drawn from the same population and only if human biases are controlled (Varma and Simon, 2006).

2.5. Methods for Modeling and Forecasting River Discharges

Many of activities associated with planning and operation of the components of a water resource system require forecast of future events. For the hydrologic and water resources components, there is the need for both short term and long term forecasts of streamflow events and discharges in order to optimize the system or to plan for future expansion or reduction. Many of these forecast systems are large in spatial extent and have a hydrometric data collection network that is very sparse (Deka and Chandramouli, 2003). Furthermore, the inherently non-linear relationships between input and output variables complicate attempts to forecast streamflow events. These conditions can result in considerable uncertainty in the hydrologic information that is available. These forecasts of river discharge are of vital importance in water resources management and there is thus a need for improvement in forecasting techniques.

Many of the techniques currently used in modeling hydrological time series and generating synthetic streamflow assume linear relationships amongst the variables (Deka and Chandramouli, 2003). The two main groups of techniques include physically based conceptual models and time-series models. Techniques in the first group are specifically designed to mathematically simulate the sub-processes and physical mechanisms that govern the hydrological cycle (Haykin, 2009). These models usually incorporate simplified forms of physical laws and are generally nonlinear, time-invariant, and deterministic, with parameters that are representative of watershed characteristics but ignore the spatially distributed, time-varying, and stochastic properties of the rainfall runoff (R-R) process (Hsu *et al*., 1995). The problem with the conceptual models is that empirical regularities or periodicities are not always evident and can often be masked by noise (Duan *et al*., 1992). It also addresses the physical problem by solving a highly- coupled, non- linear, partial differential equation set which demands huge computing cost and time (Sorooshian *et al*., 1993).

In time-series analysis, stochastic or time-series model are fitted to one or more of the time-series describing the system for purpose which include forecasting, generating synthetic sequences for use in simulation studies, and investigating and modeling the underlying characteristics of the system under study (Sunil Kumar, 1995). Most of the time-series modeling procedures fall within the framework of multivariate autoregressive moving average (ARMA) models (Raman and Sunil Kumar, 1995). Traditionally, the class of ARMA models has been the statistical method most widely used for modeling water resource time series (Maier and Dandy, 1996). In streamflow forecasting, time-series models are used to describe the stochastic structure of the time sequence of streamflows and precipitation values measured over time (Tokar and Johnson (1999). Maier and Dandy (1997) stated that auto-regressive moving average (ARMA) models have been used conventionally for stochastic modeling of time series data of water resources. Time-series models are more practical than conceptual models because one is not required to understand the internal structure of the physical processes that are taking place in the system being modeled (Maier and Dandy, 1996). The limitation of most time-series methods in streamflow forecasting is that the only information they incorporate is that which is present in past flows (Maier and Dandy, 1996). Tokar and Johnson (1999) stated that time series models fails to represent the nonlinearity inherent in the hydrologic processes, and may therefore not always perform well as expected. Many of the available techniques are deficient in that they do not attempt to represent nonlinear dynamics inherent in the transformation of rainfall to runoff (Tokar and Johnson (1999).

The following are the most conventional time series methods used for forecasting in water resources management and their details are well presented in Maier and Dandy (1996):

i. Simple linear regression
ii. Multiple linear regression
iii. Multiple nonlinear regression (MNR)
iv. Multivariate linear regression (MLR)
v. Auto regressive moving average (ARMA)
vi. Auto regressive integrated moving average (ARIMA)
vii. Velvelet transform (VT)
viii. Support vector machine (SVM)

2.5.1. Application of Times Series Convectional Methods in Water Resources

Regression analysis were extensively used to forecast short-term water demand by the following renowned people: Cassuto and Ryan (1979), Hughes (1980), Anderson *et al*. (1980), and Maidment and Parzen (1984). Maidment *et al.* (1985) used regression and time series analysis to developed daily municipal water consumption models with rainfall and air temperature as input variables. Maidment and Miaou (1986) used regression analysis to determine water consumption of nine cities in the United States whereas Smith (1988) developed and applied a time series regression model to forecast daily municipal water use in Washington. Miaou (1990) developed a monthly time series regression model to forecast urban water demand. Zhou *et al.* (2000) developed and applied a regression and a time series forecasting model for an urban sector in Australia. Recently, autoregressive moving series models were developed to analyze the structure of daily urban water consumption in Hong Kong (Wong *et al*., 2010). They concluded that with an increase in the rainfall amount there is a corresponding reduction in seasonal water use which is higher on weekdays than weekends.

Multiple nonlinear regression (MNLR) has provided extensive accurate forecasting results in various fields such as engineering, economics, finance, medicine, and marketing. MNLR entails very high order multiples which approximate complex multivariate functions (Cogger, 2010). MNLR was first applied in environmental engineering to forecast temperature (Cogger, 2010). Miyagashi *et al*. (1999) inferred that when compared with neuro-fuzzy, MNLR models performed better than traditional radial basis functions for weather predictions. Also, Saraycheva (2003) developed and applied a modified version of MNLR for ecological & socio-economic forecasting in Ukraine. In the field of hydrology, Emiroglu *et al*. (2011) compared non-linear regression (NLR) and multiple linear regression (MLR), for forecasting discharge coefficient of triangular labyrinth wires. The result of the study showed that NLR performed better than MLR. Hitherto, no studies have explored MNLR in forecasting urban water demand Emiroglu *et al*. (2011).

In the field of environmental engineering, Simple Vector Machine (SVM) was first used for forecasting air pollutant (Lu *et al*., 2002). Wang *et al*. (2009) concluded that when applied to forecasting monthly river flow discharges, SVMs out-perform ANN models. Khan and Coulibaly (2006) discovered that when applied to a 3–12 month predictions of lake water levels, SVM performed better than MLR and ANNs. Yu *et al*. (2006) used SVMs to predict flood stages level and it was found to be very successful. Han *et al.* (2007) found that SVMs performed better than other traditional models for flood forecasting. Rajasekaran *et al.* (2008) showed that when applied to storm surge predictions, SVMs was very successful while Cimen and Kisi (2009) showed successful result when also applied to daily evaporation estimation. Hourly stream flow forecast was also successfully carried using SVMs (Asefa *et al*., 2006), and the results were found to perform better than ANN (Wang *et al*., 2009) and ARIMA (Maity *et al*., 2010) models for monthly stream flow prediction. Msiza *et al.* (2008) compared the use of ANNs and SVMs in water demand forecasting in South Africa, and the result showed that ANNs performed significantly better than SVMs.

Velvelet transform models have also be extensively used in the water resources engineering and hydrology literature. Wang and Ding (2003) developed a velvelet model used to forecast groundwater levels in China and it was found to be very successful. Cannas *et al.* (2006) developed a velvelet model for monthly rainfall-runoff forecasting in Italy and it was also found to be very successful. Adamowski (2007, 2008a, 2008c) developed a new method of velvelet transform and a cross velvelet based for river flood forecasting and the results appeared to be very successful. Kisi (2008) and Partal (2009) developed a velvelet model for monthly flow forecasting in Turkey and the result showed to be significantly successful. . Kisi (2009) also used velvelet models for forecasting daily flow of intermittent rivers, while Adamowski and Sun (2010) used similar velvelet model for forecasting flow of non-perennial rivers in Cyprus. Shiri and Kisi (2010) combined velvelet with ANN and fuzzy interference systems to accurately forecast short-term and long-term streamflow and they were found to be very successful in forecasting the flood discharge. Of a general note, findings from the literature showed that velvelet ANN models generally outperformed other methods such as MLR, ARIMA, and ANN in hydrological forecasting applications.

In recent years, soft computing methods such as artificial neural networks (ANNs) and fuzzy logic inference system (FLIS) have been introduced and extensively used in hydrological forecasting. These methods do not need a prior knowledge of the process, and they are effective with linear and nonlinear data. Solomatine and Ostfeld (2008) and Maier *et al*. (2010) highlight many critical issues that need to be addressed in greater detail with traditional models. These include, the development and evaluation of hybrid model architectures that attempt to draw on the strengths of different modeling methods and the development of robust modeling procedures that are able to work with ‘‘noisy’’ data. It was in this regard that ANFIS (which combined the capabilities of ANN and FLIS) was developed. ANN and ANFIS the major subjects of this research work are discussed in the next sections in details but before then, it is necessary to describe briefly the application of Fuzzy logic in water resources management.

Zadeh in 1965 developed fuzzy set theory with relative membership concept and proposed fuzzy optimum theory (Tayfur *et al*., 2003), which has a good practical application in engineering field especially in water resources and hydrology (Chen *et al*., 2006; Chang *et al*., 2001; Liong *et al.,* 2000; Mahabir *et al.,* 2000; Nayak *et al.,* 2004a; Firat, 2007; Nayak *et al.,* 2004b; Sen, 2001). Based on the fuzzy optimum theory, a new fuzzy neural network for streamflow forecast was introduced. Yeshewatesfa *et al* (2001) applied FL model for rainfall streamflow modeling and it was found to work well. Nayak *et al* (2005) used Mamdani approach which has been used in some hydrological applications for rainfall streamflow modeling. Gowda and Mayya (2014) applied fuzzy logic model for predicting streamflow for Nethravathi River basin located in Dakshina Kannad, India. The results from the different membership functions applied showed that, fuzzy inference system using triangular membership function show a good performance compared to other models developed.

The fuzzy logic approach has also been extensively applied to flood forecasting (Feng and Hong, 2007; Yu and Chen, 2005; Chang *et al*., 2005), precipitation modeling (Maskey *et al.*, 2004), sediment transport modeling (Tayfur *et al.,* 2003), reservoir operation modeling (Dubrovin *et al.,* 2002), reservoir inflow forecasting (Nayak and Sudheer, 2008), seasonal runoff modeling (Mahabir *et al*., 2003), river stage discharge forecasting (Deka and Chandramouli, 2003), Infiltration modeling, (Abebe *et al.*, 2000), monthly ground water recharge forecasting (Coppla Jr., 2002), regional drought prediction (Pongracz *et al*., 1999), reservoir operating rule based (Russel and Campbell, 1996), groundwater vulnerability evaluation (Shouyu and Guangtao, 2003), irradiation from sunshine duration (Şen, 1998), soil erosion prediction (Mitra *et al.*, 1998) and storm water infiltration modeling (Hong *et al*., 2002).

2.6. Artificial Neural Networks (ANNs)

The development of McCulloch-Pitts network in the 1940's, brought the birth of neural computing (McCulloch and Pitts, 1943; Luger and Stubblefield, 1993). Figure 2.5 are networks with stand-alone "decision machines" that take a set of input data, multiply these input data by their associated weights (bias), and then bring out an output which is the sum of these products. The input values or input activations are therefore related to the output values or output activations by simple mathematical operations involving weights associated with network links. The McCulloch-Pitts networks are totally binary in nature. Their input values are 0's while the output values are 1's. The output node of the network (or the output result) returns a value of 1 if the sum of the products of the inputs and their corresponding weights is greater than or equal to 0. But if otherwise, a 0 value is returned. If the output of the system is to be actually 1, the value of 0 must therefore be used as a threshold that must be exceeded or equalled to.

The above mapping rules which governs the way or manner in which an output node maps input values to output values, is referred to as an activation function. This activation function is used to determine the activation of the output node (Luger and Stubblefield, 1993). The main advantage of the

McCulloch-Pitts networks is that it can be constructed to compute logical functions as in the "X AND Y" case, where no combination of inputs can produce a sum of products that is greater than or equal to 0, except the combination X = Y =1) (Figure 2.2). The McCulloch-Pitts networks do not learn as in the modern ANN or ANFIS, and as such the weight values must be determined in advance using other mathematical or heuristic means (Luger and Stubblefield, 1993). This demerit, propelled some connectionist researchers during the 1950's to look for a better way out (Luger and Stubblefield, 1993). It was in the process of these further researches that artificial neural network was discovered (Luger and Stubblefield, 1993).

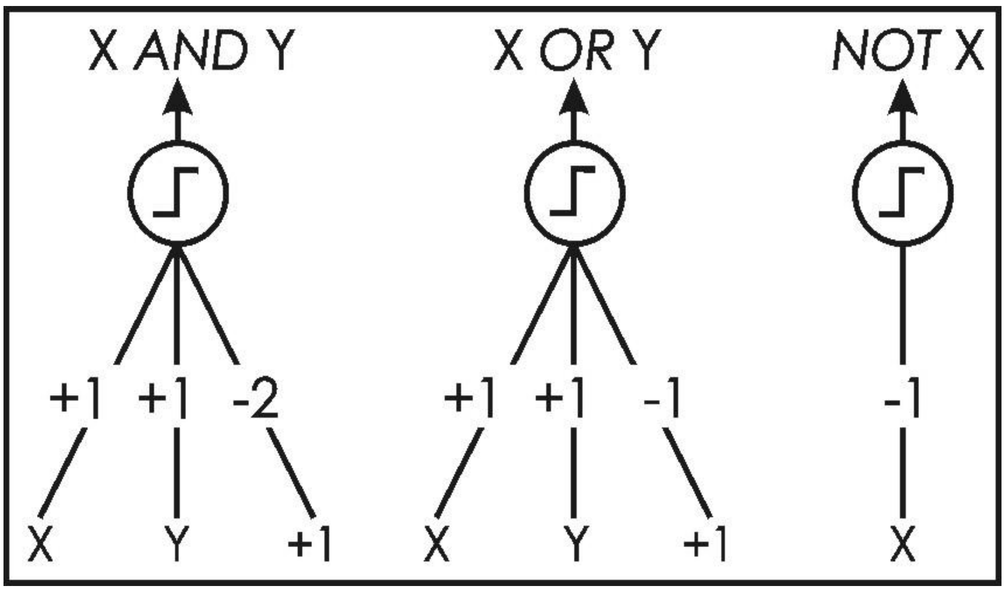


Figure 2.2: McCulloch-Pitts networks (Luger and Stubblefield, 1993)

2.6.1. Development of Artificial Neural Networks (ANNs)

The human brain habours billions of neurons that are interconnected. Due to the pattern or the structure in which these neurons are arranged and operated, humans are able to promptly recognize patterns and process data. Artificial neural network (ANN) is a type of Artificial Intelligence (AI) technique that mimics the behaviour of the human brain (Haykin, 2009). It has the capability to learn, recognize a pattern in a data, adapt solutions over time and process information quickly (Kisi, 2005). ANNs have the ability to model both linear and non-linear systems without any assumptions which is common with most traditional statistical approaches (Haykin, 2009). They have been extensively applied in various aspects of science and engineering especially in water resources engineering and hydrology to solve both linear and nonlinear problems (Rivard and Zmeureanu, 2005; Chantasut *et al*., 2005).

Figure 2.6 shows parts of a typical neuron. Each of the cells comprises a cell body (Soma) that contains the cell nucleus. Dendrites are then connected to the cell body to receive the incoming information. It also has a single long axon with dendrites for receiving outgoing information that is passed to connected neurons. The information (in the form of an electrical potential) is actually transported between neurons along the dendrites. If these potentials reach a certain threshold, the neuron will be activated (fires) and the information is delivered along the neuron's axon to the dendrites, where it is passed on to other neurons (Wei Lu, 2000). An artificial neural network is capable of processing information in a way of connecting together relatively simple information processing units of links that allow each unit to communicate with each other by simple signals (Wei Lu, 2000). Each link then has a numeric weight assigned to it and that is the primary means of long-term storage in the neural network. Weights are also updated during the learning process (Wei Lu, 2000) (Figure 2.3).

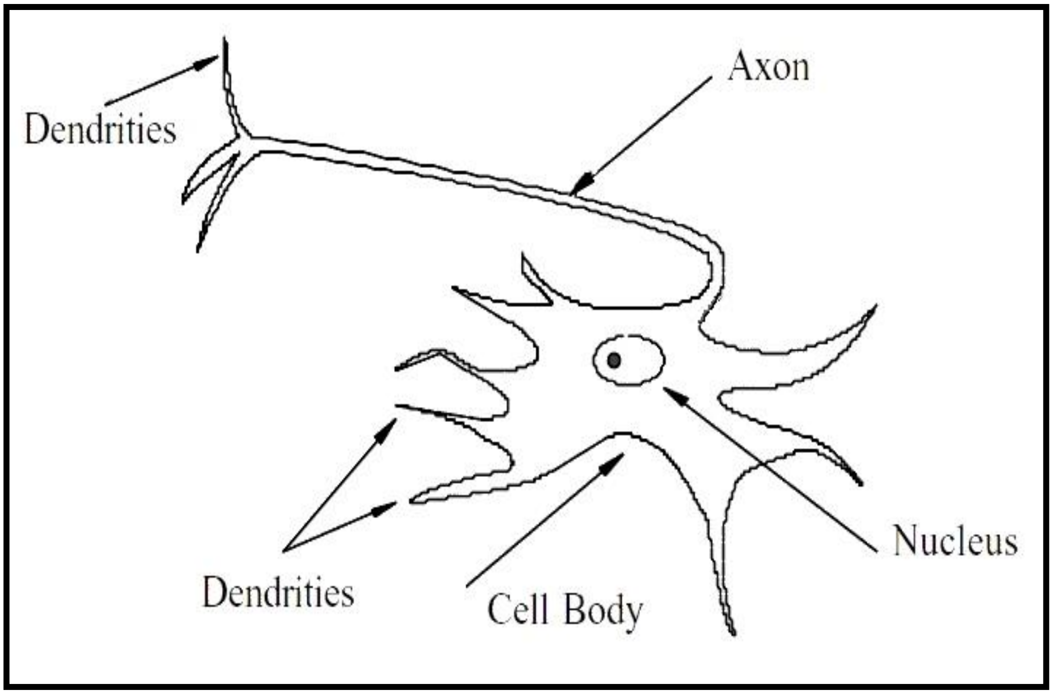


Figure 2.3: Parts of the human neuron (Wei Lu, 2000)

2.6.2. ANN Topologies and Transfer Functions

Topology of a neural network refers to the way the neurons are connected, and it is a vital factor in network functioning and learning (Wei Lu, 2000). An ANN comprises a network of artificial neurons or nodes. These nodes or neurons are connected to each other, and the strength of their connections to one another is assigned a value based on their strength: inhibition (maximum being -1.0) or excitation (maximum being +1.0). If the value of the connection is high, it is an indication of a strong connection. A transfer function is then built within each node's design. There are three types of neurons in an ANN. They are the input **layer nodes**, the **hidden layer nodes**, and the **output layer nodes** as shown in Figure 2.4.

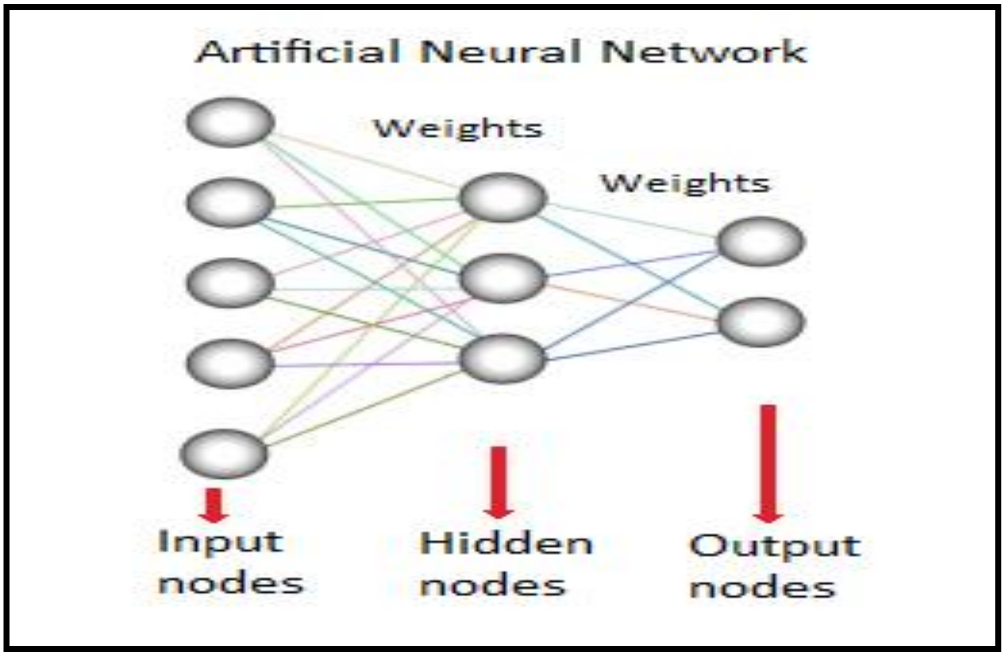


Figure 2.4: Layers of neuron (Wei Lu, 2000)

The input layer neurons in the network receive the input vector and transmit the values to the next layer (hidden layer) of processing elements across connections. This process continues until the output layer is reached which produces the final output result. The input values to the network are all joined or linked to all the neurons in the hidden layer (hidden because they are not visible in the input or the output), the outputs of the hidden neurons are then connected to all the neurons in the output layer. The activations of the output neurons constitute the output of the whole network.

After the whole networks are linked together, the input nodes then take in information in numerical form which are then presented to the network as activation values where each node is given a number. The higher the number, the greater the activation and vice versa. This information is then processed and passed throughout the network. Based on the connection strengths or weights, inhibition or excitation, and transfer functions, the activation value is transferred from node to node. Each of the nodes sums up the received activation values and modifies the value based on its transfer function. The activation then flows through the network, via hidden layers, until it reaches the output nodes. The output nodes then reflect the input in a reasonable way to the outside world. The difference between predicted value and actual value (error) is then backward propagated by assigning them to each node's weights according to the amount of this error the node is presenting (e.g., gradient descent algorithm) as shown in Figures 2.5a and 2.5b.

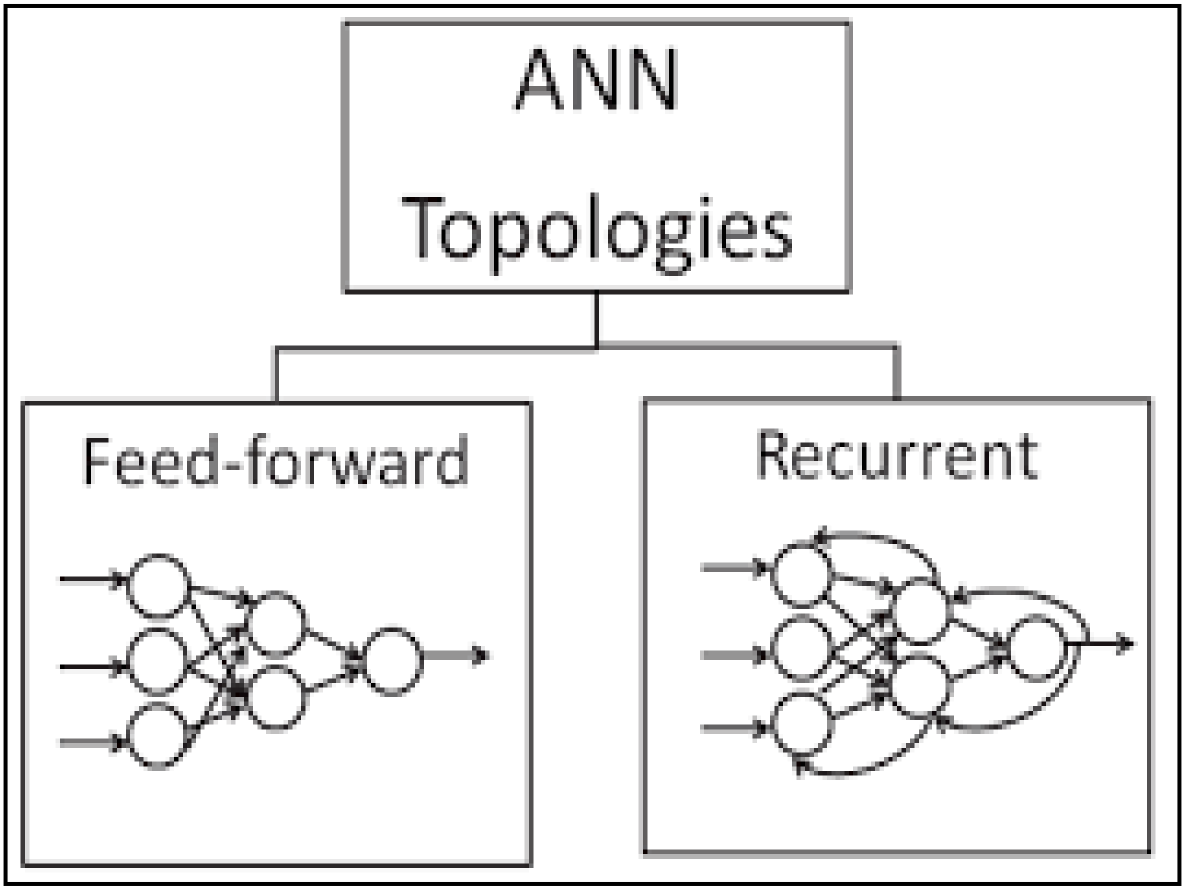


Figure 2.5a: ANN topology (Wei Lu, 2000)

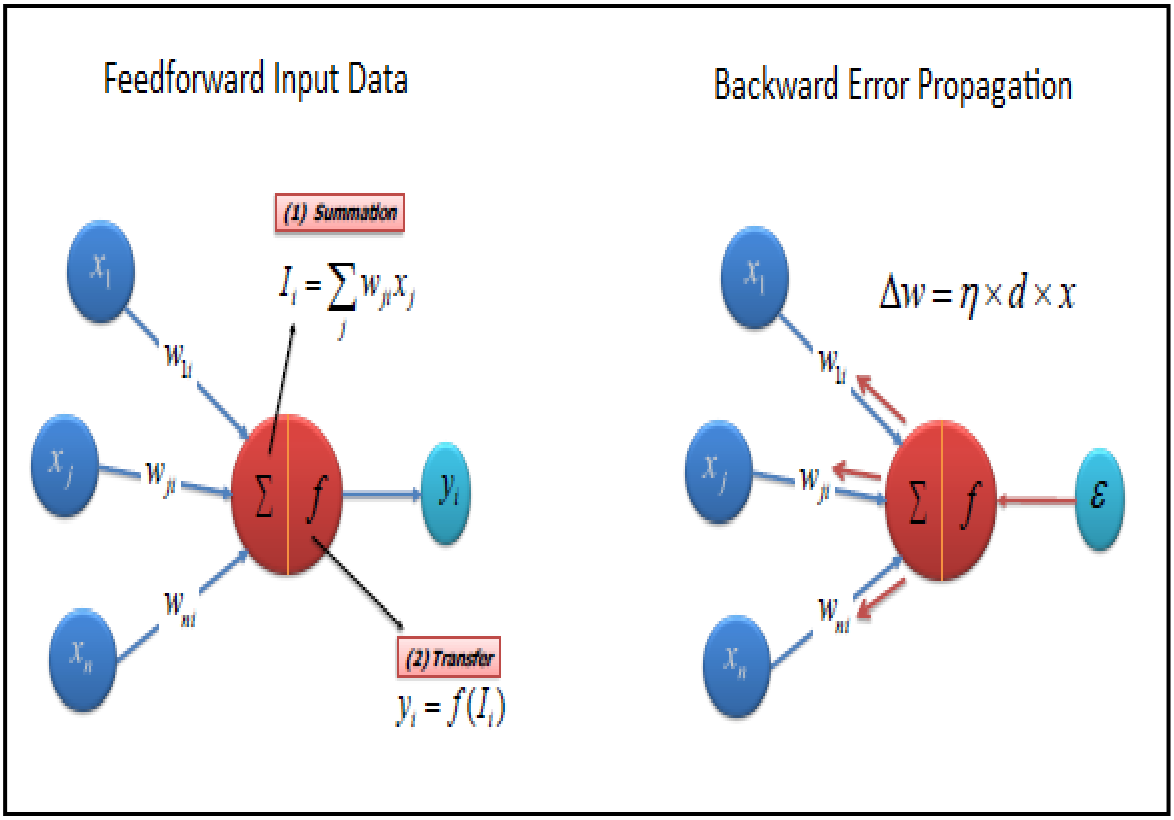


Figure 2.5b: Layers of neuron with error propagation (Wei Lu, 2000)

2.6.2.1. ANN Weights and Biases

Hitherto, the true initial values to assign for weights that will maximize the effectiveness and speed with which a neural network can learns has not been theoretically determined (Gallant, 1993). However in general practice, randomly-generated positive and negative values are assigned as the initial weight values. Such randomly generated values helps minimize the chances of the network becoming trapped in the local minima (Gallant, 1993). Usually, a range of values [-w, + w] where 0.1 < w < 2 is used (Reed and Marks, 1999). The main reason for randomly using initial weights is to break symmetry, while the main reason for using small initial weights is to avoid immediate saturation of the activation function (Reed and Marks, 1999).

These weights are normally assigned to the neurons so that the backpropagation algorithm for multi-layer, feedforward neural networks is effectively implemented. For effective implementations of this algorithm, an additional class of weights known as biases is usually employed. These Biases are values added to the sums computed at each of the node (except the input nodes) during the feedforward propagation phase (Bishop, 1995a). So, the bias assigned to a particular node is added up before the activation function at that same node is used. A negative bias is called a threshold (Bishop, 1995a).

Simply put, biases are values assigned to each node in the intermediate (hidden layer) and output layers of a network. But in practice, these biases are treated in exactly the same way or manner as other weights. These biases are assigned to vectors that lead from a single node having its location outside of the main network with an activation value of 1 always (Figure 2.6).

The main advantage of using biases in a neural network is to increases the capacity and the efficiency of the network. This helps to solve problems by allowing the hyperplanes that separate individual classes to be offset for superior positioning (Gallant, 1993). Details about the uses of biases in neural networks are available in the literature such Gallant (1993), Bishop (1995a) and Reed and Marks (1999).

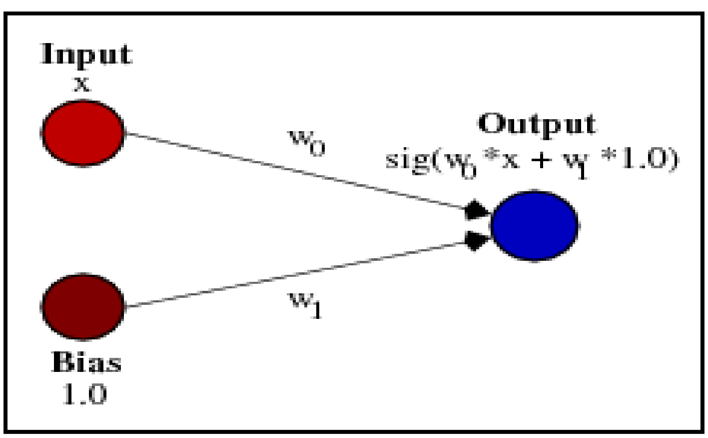


Figure 2.6: Assigning Weights and Biases to ANN (Reed and Marks, 1999)

2.6.2.2. Epochs Generation and Error Functions

Most neural network training algorithms involve making several presentations of the entire data set to the neural network. An "epoch" is a single presentation of the entire data set (Gallant, 1993). Contrarily, some algorithms present data to the neural network a single case at a time to converge to a local minimum as shown in (Figure 2.7). A local minimum or "relative minimum" is the smallest value that locates within a set of points which may or may not be a global minimum. This point is not the lowest value in the entire set (Reed and Marks, 1999). This smallest overall value of a set or function over its entire range is called the global minimum or an absolute minimum, it is entirely impossible to construct an algorithm that will find a global minimum for an arbitrary function (Gallant, 1993).

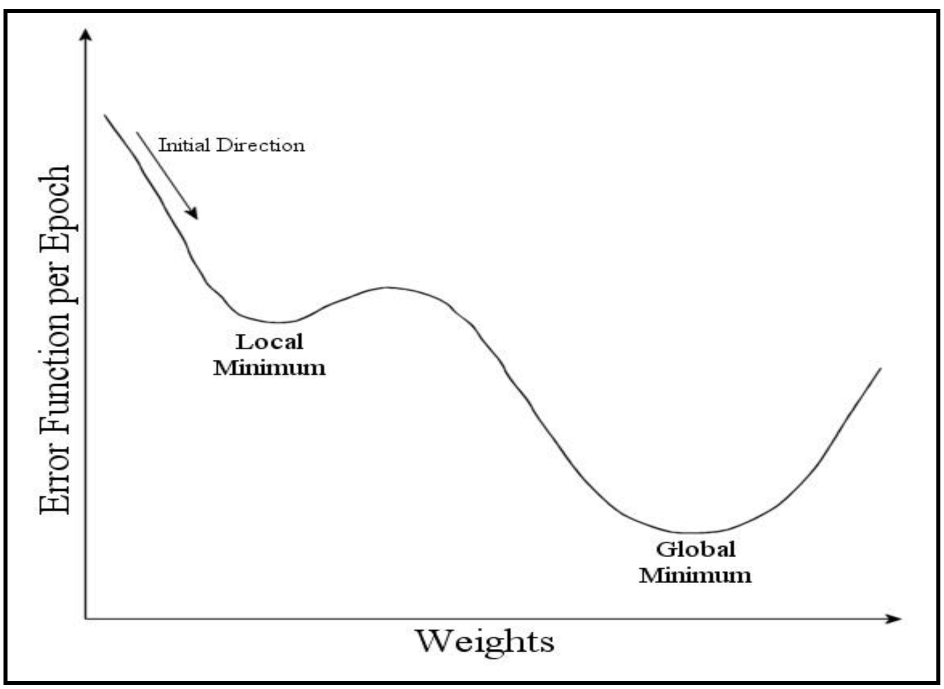


Figure 2.7: Global and Local Minima of Error Function (Gallant, 1993)

Eliminating validation sets from training samples, a fixed number of epochs are used to train the data. An epoch is one of the steps of the training when all the available training samples are shown to the neural networks (NN). It can be done in a progressive way one by one or in one bulk as a batch learning as in MATLAB. The main merit of the batch learning is efficiency; the merit of incremental learning is that it is proved with that training; the NN can estimate any nonlinear function in the limit case while this is not guaranteed in the batch learning case (Gallant, 1993).

2.6.2.3. Transfer (Activation) Functions

The transfer or activation function helps to translate input signals to output signals (Gallant, 1993). There are four types of transfer functions commonly used with ANN. They are the unit step (threshold), sigmoid, piecewise linear, and Gaussian transfer functions (Wei Lu, 2000). The transfer function of a neuron has a number of properties which either enhances the network containing the neuron or simplify it (Wei Lu, 2000). For instance, any multilayer perceptron using a linear transfer function must have an equivalent single-layer network. In this regard, a non-linear function is therefore necessary to gain the merits of a multi-layer network as shown in Figure 2.8.

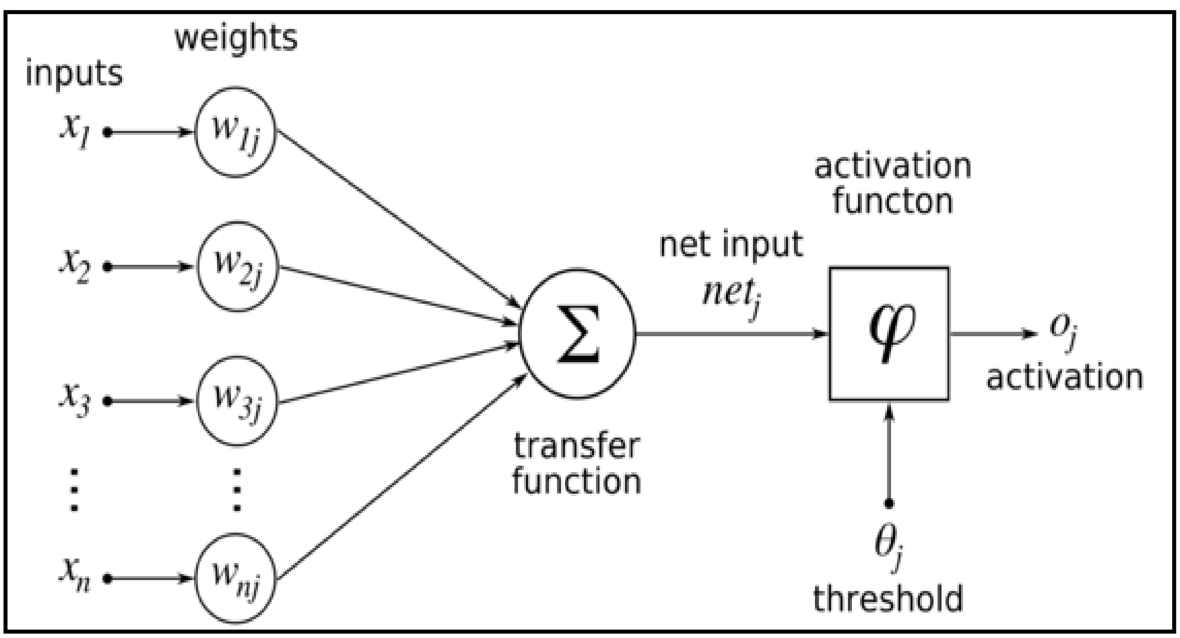


Figure 2.8: Topology of a transfer function (Wei Lu, 2000)

2.6.3. Types of ANN

An arrangement of one input layer of McCulloch-Pitts neurons feeding forward to one output layer of McCulloch-Pitts neurons is known as a Perceptron and it can be connected together in any way we like (Rezaeianzadeh, 2010). ANNs can be categorized into two types according to the number of layers as single layer and multilayer networks. It can also be categorized according to data flow as feedforward and feedbackward (recurrent) networks (Rezaeianzadeh, 2010).

2.6.3.1. Single Layer Perceptron (SLP)

The single layer perceptron (SLP) is the simplest type of artificial neural networks and is normally a feed-forward network based on a threshold transfer function (Wei Lu, 2000). It is normally classified with a binary target (1, 0) in a linear form. It comprises a single layer of output node. The inputs are fed directly to the outputs via a series of weights and that is the main reason it is considered the simplest kind of feed-forward network (Wei Lu, 2000). At each node, the summation of the products of the weights and the inputs is usually computed. If the computed value is above some threshold (typically 0), the neuron fires and takes the activated value (typically 1). If otherwise, it takes the deactivated value (typically -1). Neurons having this kind of activation function are normally called artificial neurons or linear threshold units. In some literature, the term perceptron often refers to networks consisting of just one of these units. Single-unit perceptrons are only capable of learning linearly separable patterns and are rarely used in the field of hydrology (Rezaeianzadeh, 2010). It is as shown in Figure 2.9.

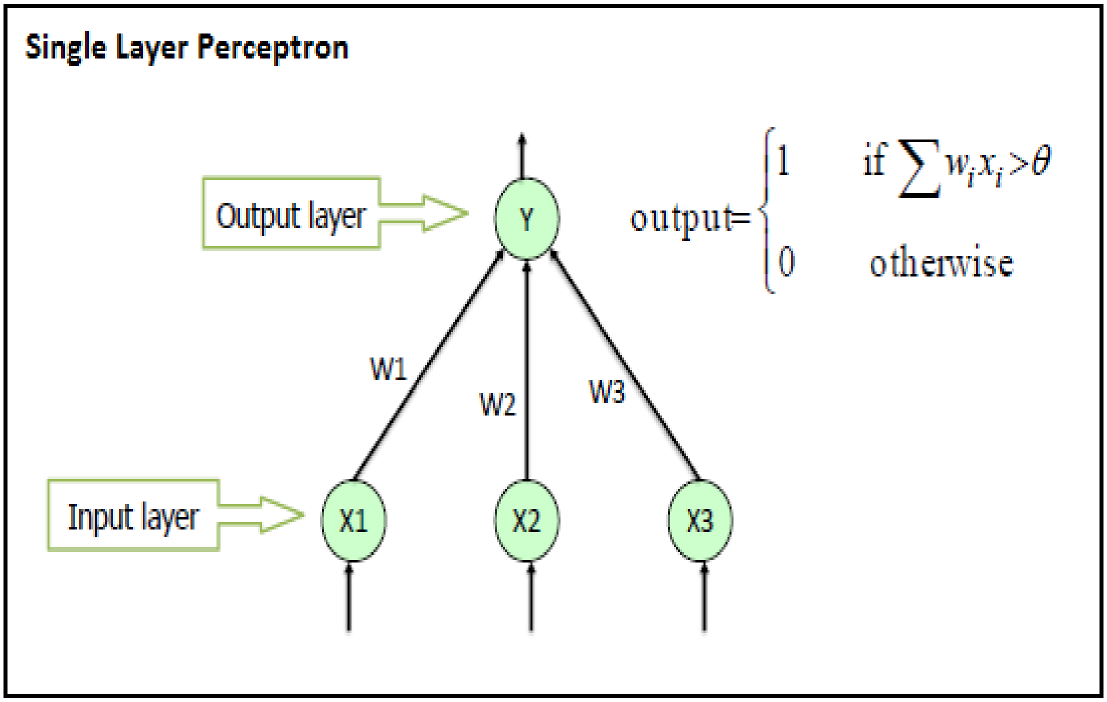


Figure 2.9: Topology of a single layer perceptron (Wei Lu, 2000)

The algorithm in a single layer perceptron (Figure 2.9) does not have a priori knowledge, so the initial weights (w) are assigned randomly. SLP sums all the weighted inputs and if the sum is above the threshold (some predetermined value), SLP is said to be activated (output = 1).

Output

$$w_1x_1 + w_2x_2 + \ldots + w_nx_n > \theta \Longrightarrow 1 \qquad 2.3.$$

$$w_1x_1 + w_2x_2 + \ldots + w_nx_n \leq \theta \Longrightarrow 0 \qquad 2.4$$

The input values (x) are presented to the perceptron, and if the predicted output is the same as the desired output (equations above), then the performance is considered satisfactory and no changes to the weights (w) are made. However, if the output does not match the desired target, then the weights need to be changed to reduce the error as shown in Equation 2.5.

Perceptron Weights Adjustment

$$\Delta w = \eta \times d \times x \quad (2.5)$$

Where:

Δw = perceptron weights adjustment

d = Predicted output-Desired output

η = Learning rate, usually less than 1

x = input data

Because SLP is a linear classifier and if the cases are not linearly separable, the learning process will never reach a point where all the cases are classified properly (Barry, 2000). The most famous example of the inability of perceptron to solve problems with linearly non-separable cases is the XOR problem figure 2.13. However, a multi-layer perceptron using the backpropagation algorithm can successfully classify the XOR data (Barry, 2000) as shown in Figure 2.10.

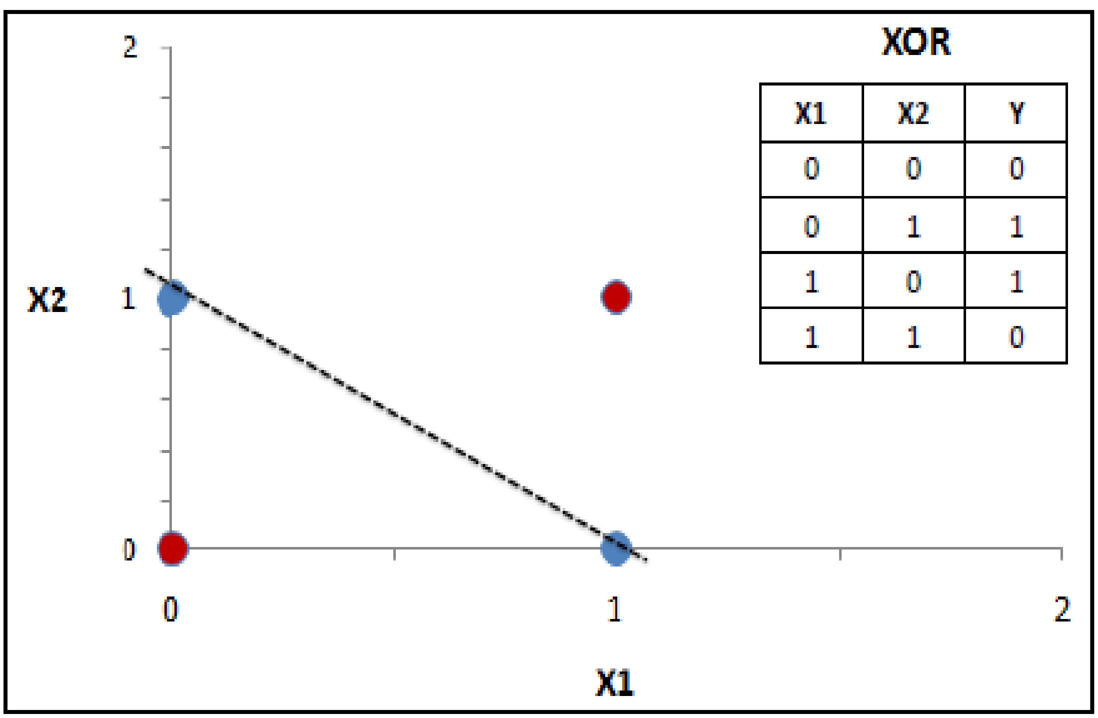


Figure 2.10: XOR Problem with Single Layer Perceptron (Barry, 2000)

2.6.3.2. Multi-Layer Perceptron (MLP)

A multi-layer perceptron (MLP) has the same structure of a single layer perceptron but with one or more hidden layers (Masters, 1993). It is usually a backpropagation algorithm with two phases: the feedforward phase and the feedbackard phase. In the feedforward phase, the activations are propagated from the input to the output layer while in the feedbackward phase, the error difference between the observed actual and the requested nominal value in the output layer are propagated backwards. This helps to modify the weights and bias values (Masters, 1993).

i. Forward propagation (FP)

This helps to propagate the input data sets by adding all the weighted inputs and then computing outputs using sigmoid threshold (Figure 2.11).

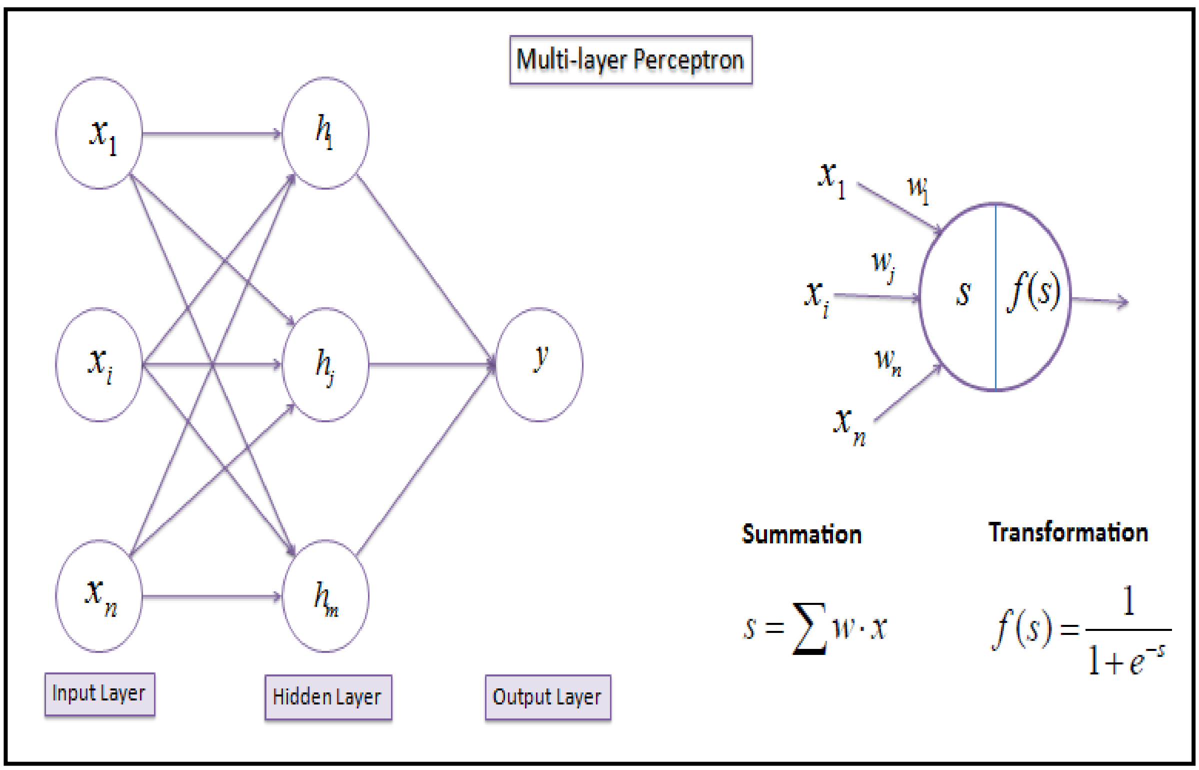


Figure 2.11: Feedforward phase (Masters, 1993)

ii. Backward propagation (BP)

This helps to propagate the errors backward by assigning them to each unit according to the amount of this error the unit represents (Figure 2.12).

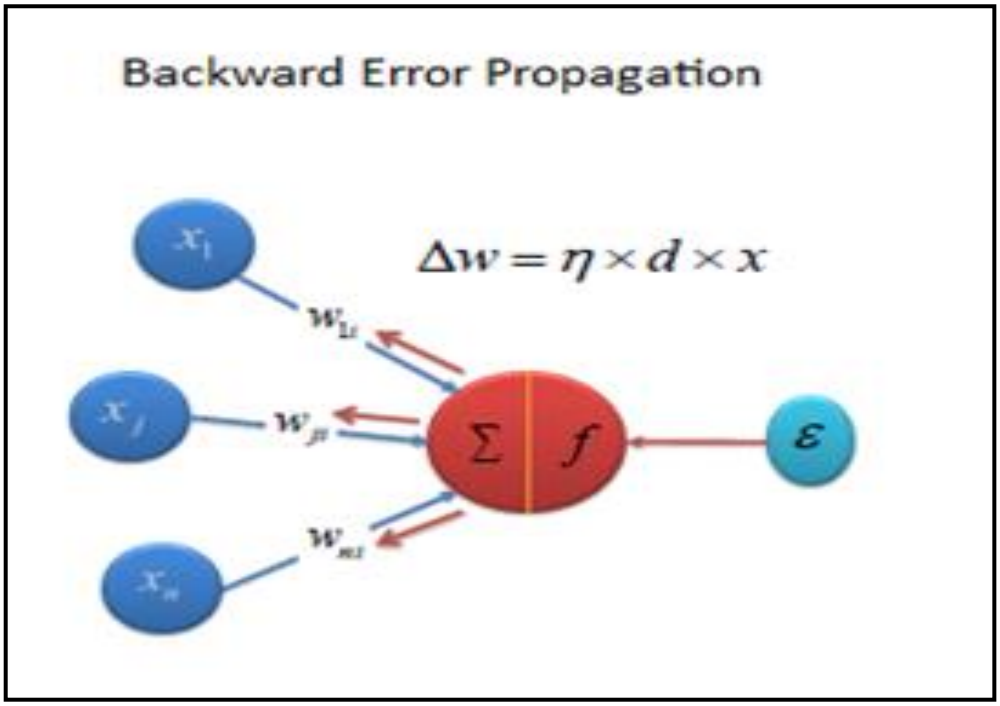


Figure 2.15: Feedforward phase with back error propagation (Masters, 1993)

The forms of the equations are shown below:

1. Error in any output neuron

$$d_o = y \text{ x } (1 - y) \text{ x } (t - y) \qquad 2.6$$

2. Error in any hidden neuron

$$d_i = y_i \text{ x } (1 - y_i) \text{ x } (w_i \text{ x } d_o) \qquad 2.7$$

3. Change in the weights

$$\Delta w = \eta \text{ x } d \text{ x } x \qquad 2.8$$

Where *y* is the original output data, *t* is the time for the model, *w* is the weight, *d* is the predicted output, *x* is the input data and η is the learning rate usually less than 1.

The ANN models and the output updating procedure are normally based on the structure of the multi-layer perceptron (MLP) principle and are most commonly used in hydrology and water resources (Kisi, 2005). The feedforward multi-layer perceptron is popularly used in the field of hydrological modeling (Maier and Dandy, 2000; Dawson and Wilby, 2001).

2.6.3.3. Feed-forward Neural Networks

As already discussed, neural networks can be classified as feedforward or feedbackward networks due to the direction of the information and processing (Haddad *et al.*, 2005). A feed-forward neural network comprises a large number of simple neuron-like processing units which are arranged in layers. Every unit in the layer is connected with all the units in the previous layer and these connections are not all equal as each may has a different strength or weight which encode the knowledge of the network. The process is a linked one in which data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. This form of network only acts as a classifier as there is no feedback between layers and no cycles or loops in the network are formed (Haddad *et al.*, 2005). This is the major reason they are called feed-forward neural networks indicating one directional.

In Figure 2.13, an example of multi-layered networks moving from right to left is shown. It has an input layer of 3 units, a hidden layer with 3 units and an output layer with 2 units.

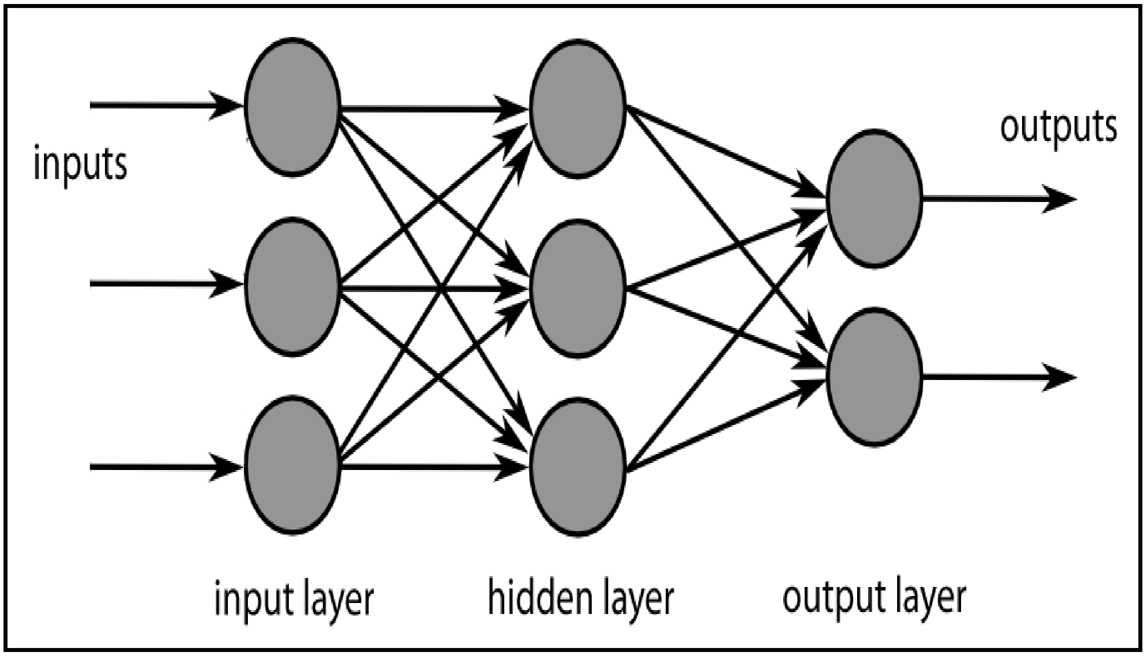


Figure 2.13: Multi-layered feed-forward network with 3 input layers (Haddad *et al.*, 2005)

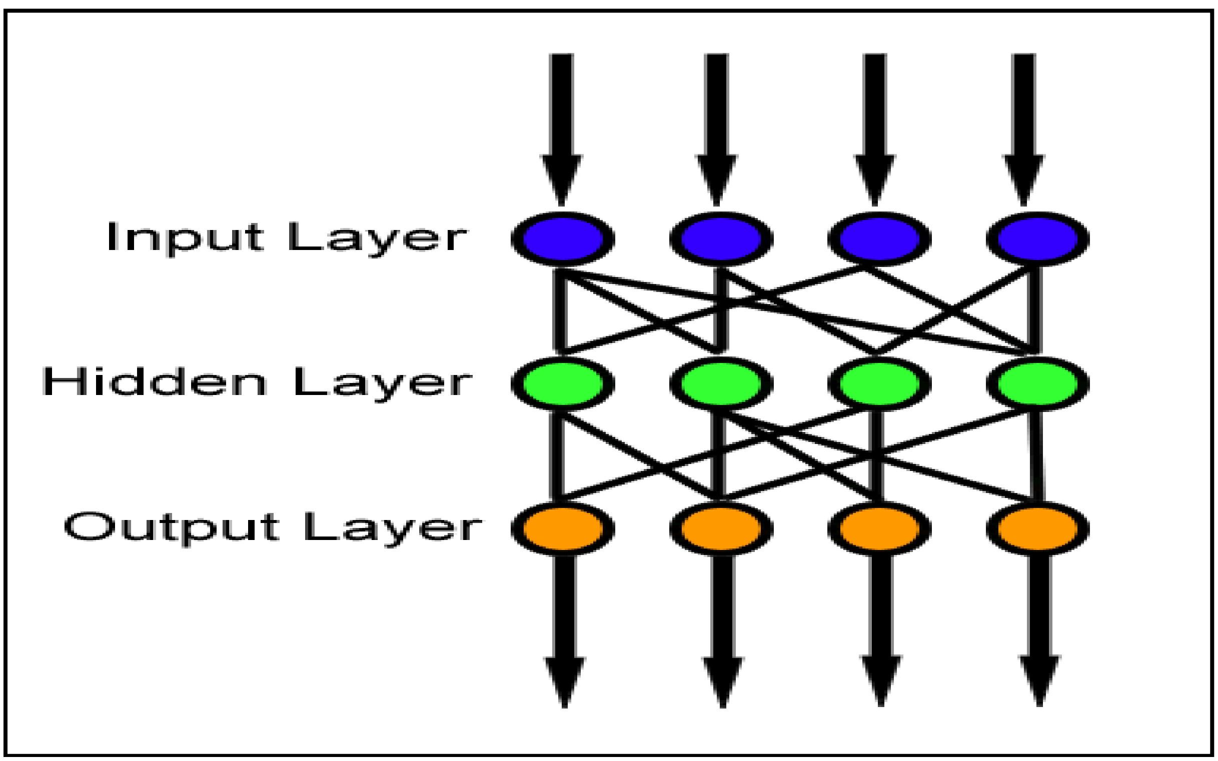


Figure 2.14: Multi-layered feed-forward network with 4 input layers (Haddad *et al.*, 2005)

Also, in Figure 2.14, an example of multi-layered networks from top to bottom is shown. It has an input layer 4 units, a hidden layer with 4 units and an output layer with 4 units. The inputs are marked as circles and they do not belong to any layer of the network. However, such inputs are sometimes considered as virtual layers with layer number 0). Any layer that is not an output layer is considered as a hidden layer. This network under consideration thus has 1 hidden layer and 1 output layer. The figure also shows all the connections between the units in different layers. A layer helps to connect one layer to the previous layer. The feed-forward neural network was the first and simplest type of artificial neural network formed (Haddad *et al.*, 2005).

2.6.3.4. Feed-backward (Recurrent) Neural Networks

In the feed-backward or recurrent neural network (RNN), the connections between units form a directed cycle with a dynamic temporal behaviour (Abbott, 1997). RNNs can use their internal memory to process arbitrary sequences of inputs unlike the feedforward neural networks. Because of this, they are popularly used in handwriting or pattern recognition where they have achieved the best known results (Abbott, 1997).

Multilayer perceptron (MLP) only allow feedforward connections between each neuron and the neurons in the proceeding layer (Abbott, 1997). Unlike multi-layer perceptron, Recurrent neural networks allow arbitrary connections between neurons, both forward and recurrent (feedback). A nonlinear mapping obtained by a recurrent neural network is not only dependent on the current input, but also is dependent on the previous inputs through the feedback connections to the input (Abbott, 1997) as shown in Figure 2.15.

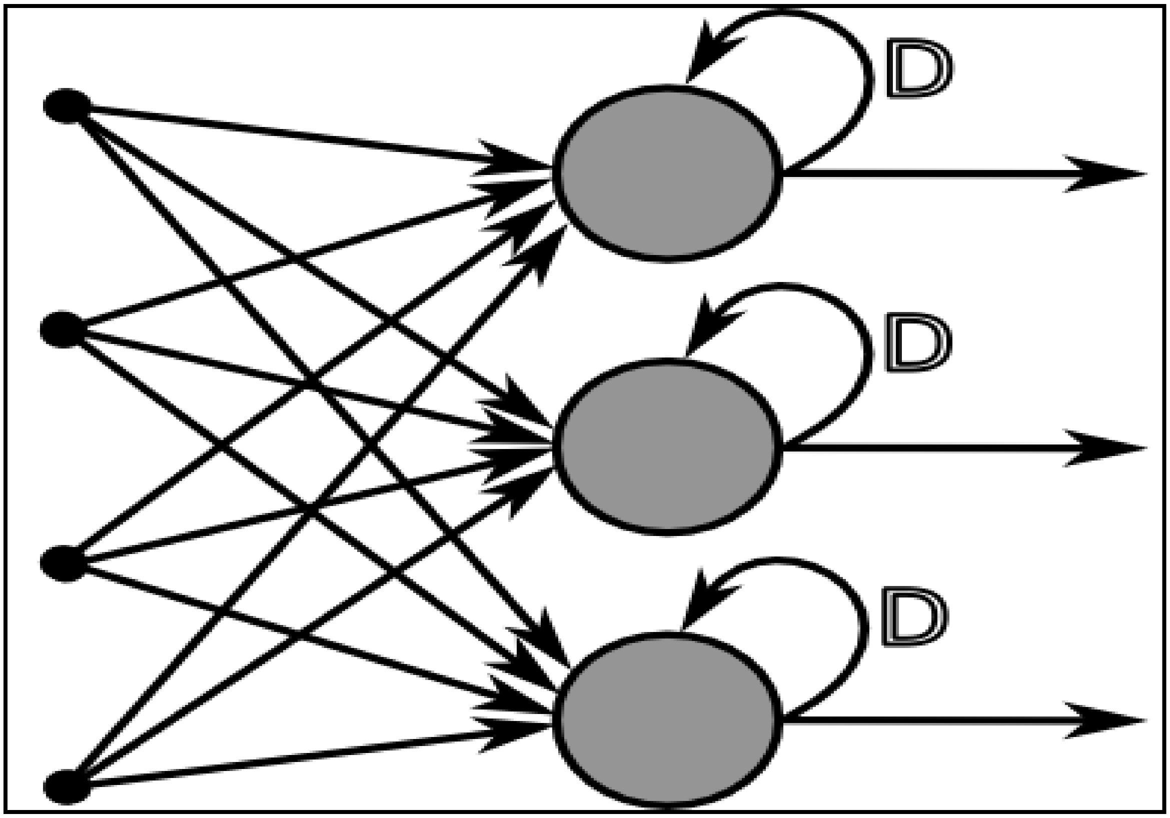


Figure 2.15: Multi-layered feed-backward network with 4 input layers (Abbott, 1997)

2.6.3.5. Radial Basis Function Networks (RBF)

RBF networks like the other networks, have three layers as the input layer, the hidden layer and the output layer. In the input layer, each of the neuron corresponds to each of the predictor variables. In categorizing the predictive variables, n-1 neurons are used where n is the number of categories (Abbott, 1997). In the Hidden layer, there are variable number of neurons and each of the neurons comprises a radial basis function centered on a point with the same dimensions as the predictor variables. The output layer has a weighted sum of outputs from the hidden layer. This help to form the network outputs layer as shown in Figure 2.16.

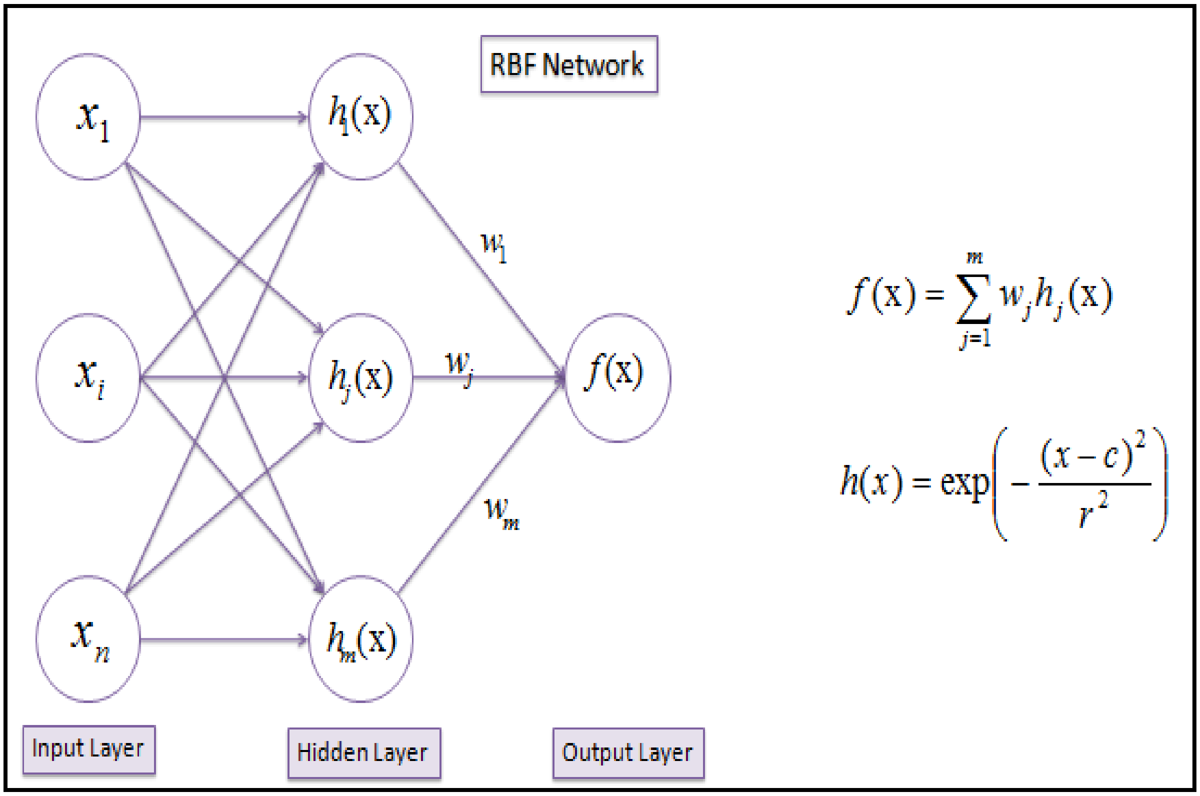


Figure 2.19: Radial Basis Function Network with 3 input layers (Abbott, 1997)

2.6.3.6. Backpropagation Neural Networks

The Backpropagation neural network is an example of a multilayered, feedforward neural network. It is mostly and commonly used in building neural network. It is also considered as one of the simplest and most general methods used for supervised training of multilayered neural networks (Lu *et al.*, 2002). It helps to approximate the non-linear relationship between the input and the output by internally adjusting the weight values. It can also be further generalized for the input that is not included in the training patterns, that is, testing data set (predictive abilities) (Lu *et al.*, 2002).

Backpropagation neural networks have two steps of operation as feedforward and Backpropagation. In the feedforward process, the input data is applied to the input layer and its effect propagates, layer by layer, throughout the network until an output result is produced. The actual output value of the network will then be compared to the expected output. The difference which is the error signal is then computed for each of the output nodes. Because all the hidden nodes have, to some extent, contributed to the errors shown in the output layer, the output error signals are back propagated from the output layer to each node in the hidden layer which immediately contributed to the output layer. This process is then repeated, layer by layer, until each node in the network has received an error signal that describes its relative contribution to the overall error.

Once the error signal for each node has been estimated, the errors are then used by the nodes to update the values for each connection weights until the network converges to a state that allows all the training patterns to be encoded. The Backpropagation algorithm then used the delta rule or **gradient descent technique to** seek for the minimum value of the **error function** in weight space (Rojas, 1996). The weights that minimize the error function are then considered to be a solution to the learning problem. On a general note, the Backpropagation network has two stages, training and testing. During the training phase, the network is given sample inputs and the correct classifications while in the testing phase an independent data set is used.

Figure 2.17 shows the topology of the Backpropagation neural network that includes four input layers, four hidden layers and four output layers. It should be noted that Backpropagation neural networks can have one or more than one hidden layer (Rojas, 1996).

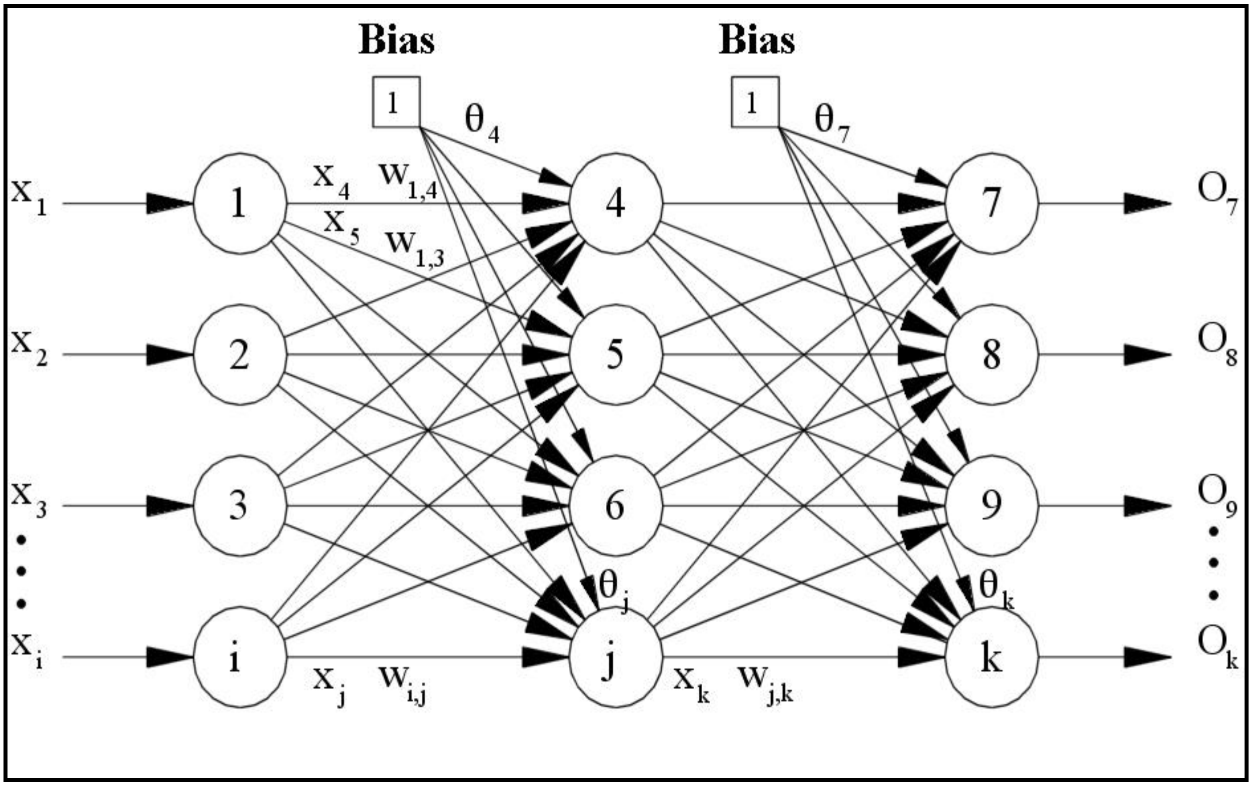


Figure 2.17: Backpropagation Neural Network with four hidden layer (Lu *et al.*, 2002)

2.6.4 Data Training, Validation and Testing Using ANNs

Data training, testing and cross validation are very common with ANN. The process to determine ANN weights and biases is called training and it is similar to the calibration of a mathematical model (Haykin, 2009). The reason of training ANNs dataset is to optimize the error function in the input data so as to determine the best set of weights and biases for output data set. ANN are usually trained with a set of input data and known output data. As the training begins, the weights are initialized, either with a set of random values or based on some previous experience. Next, the weights are systematically changed by the learning algorithm such that for a given input, the difference between the ANN output and actual output is small. Many learning examples are repeatedly presented to the network, and the process is terminated when this difference is less than a specified value and at this stage, the ANN is considered trained (Haykin, 2009).

After the networks have been trained with the training data set, the networks are then tested with testing data set. Testing of network is done in order to determine a good balance between memorization (accuracy) and generalization (Haykin, 2009). Finally the networks are validated in order to compare predicted values against observed values. Depending on the final outcome, either the ANN has to be retrained (if good result is not obtained) or it can be implemented for its intended use (if good result is obtained). Neural network trained better when more input data are used. The number of input, output, and hidden layer nodes depend upon the problem being considered. If the number of nodes in the hidden layer is small, the network may not have sufficient degrees of freedom to learn the process correctly and hence accurate results might not be obtained. But when the number is too high, the training will take a long time and the network may sometimes over fit the data (Karunanithi *et al*., 1994).

There are three major algorithms for training ANNs. The Levenberg-Marguardt algorithm, Scaled Conjugate Gradient algorithm and Bayesian Regularization algorithm.

2.6.4.1. Levenberg-Marguardt Backpropagation Algorithm

The Levenberg-Marguardt algorithm (LM) is one of the best training algorithms used extensively in hydrology and water resources management (Haykin, 2009). It is a basic method used extensively to solve nonlinear least squares problems which occurred when a parameterized function is fitted to a set of measured data points. This process helps to minimize the sum of the squares of the errors between the data points and the function (Vitkovsky´ *et al.,* 2007; Marquardt, 1963). This method is an iterative one which helps to continually improve the fitting of parameter values in order to reduce the sum of the squares of the errors between the function and the measured data points (the error function) (Vitkovsky´ *et al.,* 2007; Marquardt, 1963).

LM is actually a modified version of the classic Newton algorithm whose aim is to search for the minimum point of a nonlinear function and then perform a curve fitting on them (Karul *et al*., 2000). It is represented by the following equation (Karul *et al*., 2000):

$$X_{k+1} = X_k - (J^T J + uI)^{-1} J^T e \qquad 2.9$$

Where:

X is the weight of the neural network, J is the Jacobian matrix of the performance criteria to be minimized, u is the learning rate that controls the learning process, k is the unit vector, T is the time and e is the vector of the case error.

The LM algorithm is actually developed to approach second-order training speed and accuracy without having to compute the Hessian matrix. This is because, the Second-order nonlinear optimization techniques are usually faster and more reliable when compared to most other optimization algorithm methods such as the conjugate gradient (CG) and gradient descent with momentum (GD) algorithms while neglecting the effect of white noise (Pramanik and Panda, 2009; Adamowski and Karapataki, 2008; Adamowski and Chan, 2011). For more detailed description and application of the LM algorithm to ANN training, see Hagan and Menhaj (1994).

Levenberg-Marquardt algorithm is a combination of the **gradient descent method** and the **Gauss-Newton method**. In the former method, the sum of the squared errors is reduced by updating the parameters in the direction of the greatest reduction of the least squares objective (Vitkovsky´ *et al*. 2007; Marquardt, 1963). In the latter method, the sum of the squared errors is reduced by assuming that the least squares function is locally quadratic (Vitkovsky´ *et al.* 2007; Marquardt, 1963). The minimum of this quadratic must be determined. The Levenberg-Marquardt method acts more like a gradient-descent method when the parameters are far from their optimal value and when the parameters are close to their optimal value it acts more like the Gauss-Newton method (Vitkovsky´ *et al.* 2007; Marquardt, 1963).

At times, a general minimization method (The steepest descent method) is used to update parameter values in the direction opposite to the gradient of the objective function (Vitkovsky´ *et al.* 2007). It is a good convergent algorithm for finding the minimum of simple objective functions. But for problems with many parameter values, the gradient descent method is recommended to be used (Vitkovsky´ *et al.* 2007; Marquardt, 1963). For problems of moderately-size, the Gauss-Newton method is popularly used as it converges much faster than gradient-descent methods. The main advantage of this method is that the system is very fast but, it fails to converge to global minimum (Vitkovsky´ *et al.* 2007, Marquardt, 1963). Of a general note, the Levenberg-Marquardt method is very fast to converge to local minimal. One of the big challenges of LM is that it depends on the initial values of the decision variables. It also becomes trapped in the local minimum of the search space.

2.6.4.2. Scaled Conjugate Gradient Backpropagation Algorithm

In the standard backpropagation algorithm such as Levenberg-Marquardt algorithm, the weights in the steepest descent direction are adjusted. The performance function in such direction rapidly decreases along the negative of the gradient though it does not necessarily produce the fastest convergence (Hagan, *et.al*., 1996). In the case of conjugate gradient (CG) algorithms, the reverse is the case. It performed along a direction which produces a faster convergence than the steepest descent direction. This process helps to preserve the error minimization from all previous steps in the algorithm (Kisi and Uncuoglu, 2005). This faster convergence direction is called the conjugate direction.

In using the CG algorithms, the step-size is adjusted at each iteration which enables a minimization search to be conducted along the conjugate gradient direction. This helps to determine the step size and therefore minimizes the performance function along that line. At first iteration, all of the CG algorithms start by searching in the direction of the steepest descent as shown in equation 2.10. Frequently, CG algorithms uses a line search technique to approximate the step size. This process helps to avoid the estimation of the Hessian matrix to determine the optimal distance to move along the current search direction as shown in equation 2.11. After that, the next search direction which is the conjugate to the previous search direction is determined as shown in equation 2.12. To determine this new search direction, the new steepest descent direction with the previous search direction needs to be combined (Hagan, *et al.*, 1996).

$$P_o = -g_o \qquad 2.10$$

$$X_{k+1} = x_k + \alpha_k g_k \qquad 2.11$$

$$P_k = g_k + \beta_k p_{k-1} \qquad 2.12$$

The manner or way in which the factor β_k is computed in the equation above, helps to distinguish the various types CG algorithms (Kisi and Uncuoglu, 2005). The step size can also be estimated by other methods apart from the line search technique. But the whole goal is to combine the model best region approach obtained from both the LM algorithm and the CG algorithm. This combined approach is called Scaled Conjugate Gradient (SCG) and it was introduced to literature by Møller (1993). This process is described in equation 2.13.

Where: s is the Hessian matrix approximation, E is the total error function and $E`$ is the gradient of E, scaling factors λ_k and σ_k are introduced to approximate the Hessian matrix and initialized by user at the beginning of the algorithm such that $0< \lambda_k< 10^{-6}$ and $0<\sigma_k<10^{-4}$. For SCG, β_k factor calculation and direction of the new search can be shown as in (2.14) and (2.15) (Møller, 1993):

$$s_k = \frac{E`(w_k+\sigma_k p_k)-E`(w_k)}{\sigma_k} + \lambda_k p_k \qquad 2.13$$

$$\beta_k = \frac{(|g_{k+1}|^2 - g_{k+1}{}^T g_k)}{g_k{}^T g_k} \qquad 2.14$$

$$p_{k+1} = -g_{k+1} + \beta_k p_k \qquad 2.15$$

2.6.4.3. Bayesian Regularization Backpropagation Algorithm

Bayesian Regularization (BR) is a type of training algorithm that helps to update the weight and bias values of a neural network according to LM optimization principle (Foresee and Hagan, 1997; MacKay, 1992). It helps to reduce or minimize a combination of squared errors and weights which enables the network to determine the correct combination that generalizes well (Pan, *et al*., 2013). BR introduces weights into the training objective function of the network which is denoted as $F(\omega)$ (equation 2.16) (Yue, *et al*., 2011).

$$F(\omega) = \alpha E_{\omega} + \beta E_D \qquad 2.16$$

Where: E_{ω} is the sum of the squared network weights and E_D is the sum of network errors. Both α and β are the objective function parameters.

In using the network of BR, the weights of the network are considered as random variables while the distribution of the network weights and training set are considered as Gaussian distribution. The objective function parameter factors α and β are defined using the Bayes' theorem. The Bayes' theorem is used to compare two variables (or events) such as A and B. It principle is based on their prior (or marginal) probabilities and posterior (or conditional) probabilities as in (equation 2.17) (Li and Shi, 2012):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad 2.17$$

Where: P(A|B) is the posterior probability of A conditional on B, P(B|A) the prior of B conditional on A, and P(B) the non-zero prior probability of event B, which functions as a normalizing constant. The objective function (in equation 2.16) needs to be minimized in order to find the optimal weight space. This minimization is the equivalent of maximizing the posterior probability function as given in (equation 2.18):

$$P(\alpha, \beta | D, M) = \frac{P(D|\alpha, \beta, M) P(\alpha, \beta | M)}{P(D|M)} \qquad 2.18$$

Where: α and β are the factors needed be to optimized, D is the weight distribution, M is the particular neural network architecture, $P(D|M)$ is the normalization factor, $P(\alpha, \beta|M)$ is the uniform prior density for the regularization parameters and $P(D|\alpha, \beta, M)$ is the likelihood function of D for a given α, β,M. Maximizing the posterior function (α, β/D,) is equivalent to maximizing the likelihood function ($D|\alpha$, β,). This helps to determine the optimum values for α and β for a given weight space. After all these stages, the algorithm moves into the LM phase where the Hessian matrix calculations take place. This helps to update the weight space in order to minimize the objective function. If the convergence in the process is not met, the algorithm estimates new values for α and β. In such situation, the whole procedure repeats itself until convergence is finally reached (Yue, *et al.*, 2011).

2.6.5 Merits and Demerits of ANN

The ANN models have many advantages which make them attractive for use especially in developing countries. These advantages are as follow:

a) ANN models are easy to develop, as they do not require very detailed knowledge about the physical functioning of the catchment or extensive data pre-processing (Dastorani, *et al.*, 2010; Chang and Chang, 2006). The key to their success is the determination of the appropriate external inputs to the model.

b) ANN models once calibrated are fast to run. This requires very little execution time on a modest personal computer (PC).

c) In many of the developing countries, the hydrological data is very sparse and scanty yet ANN can work with it.

d) ANNs can handle incomplete, noisy and ambiguous data (Singh and Deo, 2007).

e) It can also solve complex nonlinear problems as nonlinearity is distributed throughout the neural networks (Singh and Deo, 2007).

f) Another advantage of ANN is that it can map input-output relations and conveniently adapt to change in the surrounding environment (Kisi, 2007).

g) Open source codes either free or at a very cheap rate are easily available. For example, the source code of the Stuttgart Neural Network Simulator is freely available on the Web (see *http://www-ra.informatik.uni-tuebingen.de/SNNS/).* The availability of open source code helps the in-house development of river forecasting software and hence reducing the development cost. The in-house development would also help in empowering local institutions and strengthening their technical capacity.

The disadvantages in using ANNs include its "black box" nature, greater computational burden (an advantage of Fuzzy logic), proneness to over-fitting (an advantage of Fuzzy logic), and the empirical nature of model development (Singh and Deo, 2007). ANN, as a black box learning approach cannot interpret relationship between input and output (though it can map it) and cannot deal with uncertainties (Singh and Deo, 2007). Meanwhile, Fuzzy logic is quite good in handling uncertainties and can interpret relationship between input/output by producing rules (though it cannot map it).

Some issues usually arise when a neural network is generalized. The issues are problems associated with under-training and over-training of data (Karystinos and Pados, 2000). Under-training usually occurs when a neural network is not strong enough to detect a pattern in a complicated data set. This happens in a network with very few hidden nodes that it cannot accurately represent the solution, thereby under-fitting the data (Karystinos and Pados, 2000) (Figure 2.18).

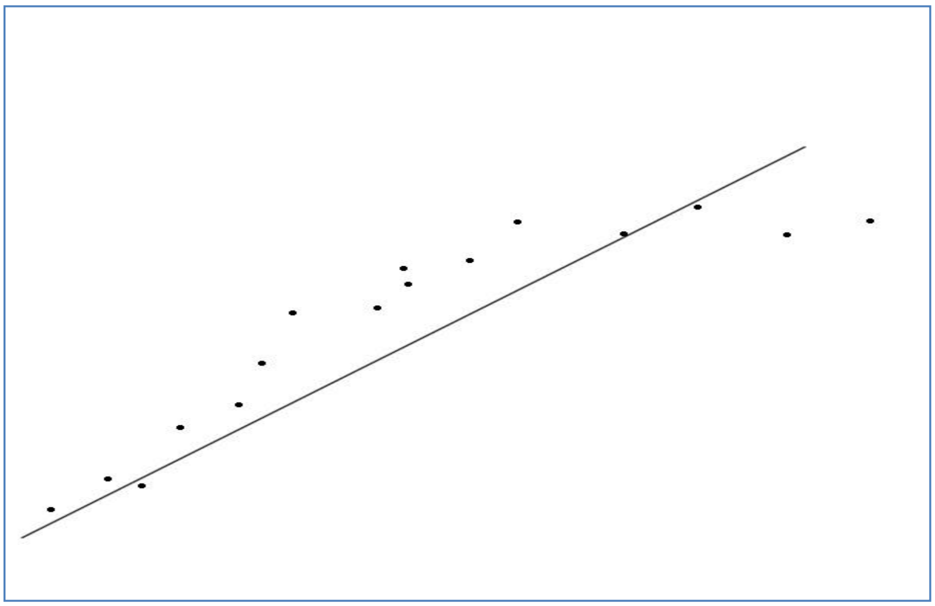


Figure 2.18: Under-fitting of ANN data (Reed and Marks, 1999)

On the other hand, over training involves the adaptation of weight values that are so close to training data that they diminished the effective utilization of the network in processing new data (Bishop, 1995a; Reed and Marks, 1999; Karystinos and Pados, 2000). In over-training or generalization, the network unable to relate general range of values to classes. Instead, the network relates input values to classes in a manner that is contrary to the more general relation that is desired. This results in a network that is too complex hence resulting in predictions that are far beyond the range of the training data (Karystinos and Pados, 2000). Networks with too many hidden nodes will tend to over-fit the solution (Figure 2.19)

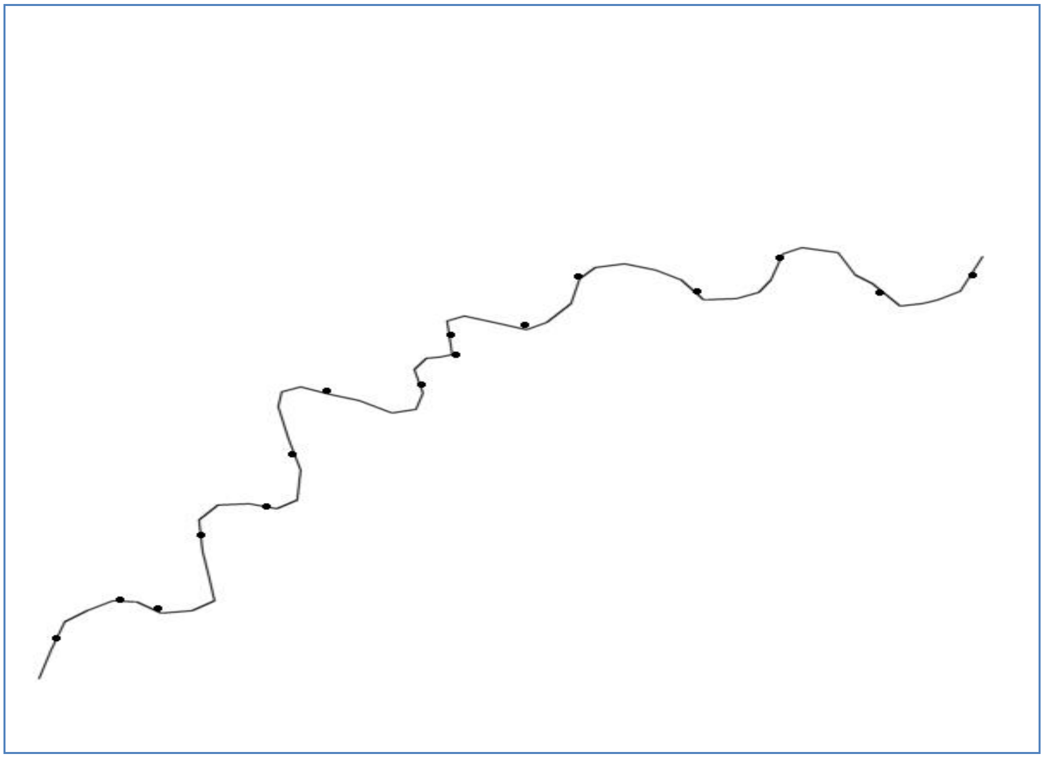


Figure 2.19: Over-fitting of ANN data (Reed and Marks, 1999)

Over-training or generalization can be prevented by terminating training before over training occurs (Reed and Marks, 1999). This means, for any given network with training data and learning algorithm, there will always be an optimal amount of training that produces the best result or generalization. Also, another method to help prevent over-generalization is the use of jitter (i.e., the addition of a small amount of artificial noise to training data while a network is being trained) (Bishop, 1995a; Bishop, 1995b; Reed and Marks, 1999). This involves the addition of a random vector to each training pattern each time it is submitted to the network. The addition of noise to training data allows values that are proximal to true training values to be taken into account during training (Reed and Marks, 1999). A neural network with the "right" number of hidden nodes thus leading to a good solution to the problem is called a good data fitting (Hansen and Salamon, 1990) (Figure 2.20).

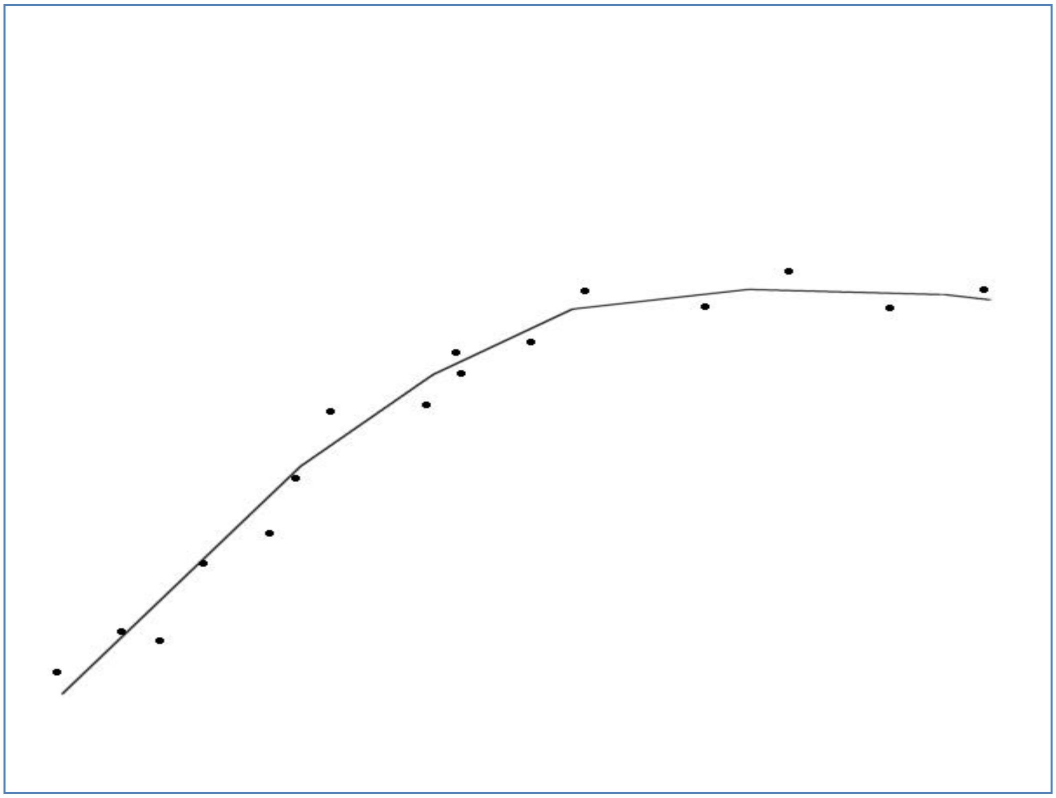


Figure 2.20: Good Fit of ANN data (Reed and Marks, 1999)

2.6.6 Application of ANN in Water Resources and Hydrology

The application of ANNs to water resources problems is on the increase due to their great power and potential to map both linear and nonlinear systems data. This is because most water resource problems are nonlinear and have complex interrelationships which can easily be solved with ANN (Chantasut *et al.,* 2005). The processes that involve several parameters are easily amenable to neuro-computing (Haykin, 2009). Among the many ANN structures that have been studied, the most widely used network structure in the area of hydrology and water resources is the multilayer, feed-forward network (Kisi, 2004, 2007 and 2008).

Previous studies such as Solomatine and Dibike, (2001) applied ANNs to replicate a hydrodynamic/ hydrological model of Apure river basin in Venezuela and it modelled the river accurately. Many studies have demonstrated that the ANN models can be successfully utilized in simulating river flows (Actil and Rat, 2005; Shamseldin, 2006; Coulibaly *et al.,* 2000, 2001, 2009; Chang and Chen, 2001; Shamseldin *et al.,* 2002, 2007; Rajurkar *et al.,* 2004; Goswami *et al.,* 2005; Dawson *et al.,* 2006, 2007, 2009; Abrahart *et al.,* 2007; Boucher *et al.,* 2009; Fernando and Shamseldin, 2009; Pramanik and Panda, 2009). ANN has also been widely used for modeling of hydrologic process, namely rainfall stream flow forecasting (Govindaraju and Rao, 2000). Fernando *et al.* (2002) studied on stream flow forecasting using radial basis function (RBF) networks with orthogonal least square (OLS) algorithm. ANN has also been used to predict daily stream flow (Pudilo-Calvo and Portela, 2007); daily rainfall (Ramirez *et al*., 2005; Kumar *et al.,* 2005), evaporation (Moghaddamnia *et al*., 2009), temperature (Tabari *et al.,* 2010; Bilgili, 2010), evapotranspiration (Sabziparvar and Tabari, 2010; Tabari *et al.,* 2010; Wang *et al.,* 2011) and snowmelt and runoff for watershed (Wu *et al.,* 2005; Tabari *et al.*, 2010).

ANN has been applied with success in stream flow forecasting (Fernando *et al*., 2002; Sinha *et al*., 2013), reservoir inflow forecasting (Jain and Srivastva, 2005), Sediment yield modeling (Senthil Kumar *et al*., 2012; Wang *et al.,* 2009 and Raghuvanshi *et al*., 2006). Senthil Kumar *et al*., (2012), Raghuvanshi *et al.*, (2006) applied the ANN model for prediction of suspended sediment yield on weekly and monthly basis for eastern coast river. ANN was also applied to stream flow generation (Ahmed, 2007), river level prediction (Leahy *et al.,* 2008), daily river flow forecasting (Pulido and Portela, 2007), river flow prediction (Kisi, 2004 and 2007), intermittent stream flow forecasting (Kisi *et al.*, 2012), rainfall-runoff prediction modeling (Dawson, 1998), flood routing prediction (Shrestha, 2003), prediction of sediment load concentration in rivers (Nagy *et al.*, 2002), pesticide prediction in groundwater (Sahooa *et al.,* 2005), daily outflow forecasting (Rezaeianzadeh, 2013), hydrologic predictions at multiple gauging stations (Mutlu *et al.,* 2008) and rainfall-runoff transformation modeling (Lorrai and Sechi, 1995).

2.7. Adaptive Neuro-Fuzzy Inference System (ANFIS)

Adaptive Neuro-Fuzzy Inference System or simply Neuro-fuzzy was first proposed by Prof. J.S. Roger Jang in 1993 (Ren, *et al*., 2006). Adaptive Neuro-Fuzzy Inference Systems (ANFIS) are a class of adaptive neural networks that are functionally equivalent to fuzzy inference systems. It offers the combination of learning, adaptability and nonlinear, time-variant problem solving characteristics of Artificial Neural Networks (ANN) plus the important concepts of approximate reasoning and treatment of information provided by the fuzzy set theory (Ren, *et al*., 2006). It represent a hybrid platform for solving actual complex problems that require the use of intelligent systems and are better alternative to the conventional model-based control schemes (Kosko, 1999). They effectively deal with the common issues of uncertainty and unknown variations in parameters and structure. Hence, it improves the robustness of the control system (Khan, 1999).

Adaptive Neuro-Fuzzy Inference System is based on the first-order Sugeno fuzzy (Ren, *et al*., 2006). It is an approach where the fusion of neural networks and fuzzy logic find their strengths. These two techniques complement each other because most of the disadvantages of ANN are the advantages of Fuzzy logic and most of the disadvantages of Fuzzy logic are the advantages of ANN (Dixon, 2005). A marriage between artificial neural networks and fuzzy logic techniques should help overcome the short comings of both techniques as given by (Kosko, 1999). The neuro-fuzzy approaches employ heuristic learning strategies derived from the domain of neural networks theory to support the development of a fuzzy system (Dixon, 2005). It is possible to completely map neural networks knowledge to fuzzy logic in ANFIS (Khan, 1999). The ability of neural network to learn fuzzy structure from the input-output data sets in an interactive manner has encouraged many researchers to combine the ANN and the fuzzy logic effectively to organize network structure itself and to adapt the parameters of fuzzy system. Also, several well-known neuro-fuzzy modeling algorithms are available in the literature, such as fuzzy inference networks, fuzzy aggregation networks, neural network-driven fuzzy reasoning, fuzzy modeling networks, and fuzzy associative memory systems, etc that can help map ANN and fuzzy logic together (Lin and Namin, 2005; Dawson and Wilby, 1998; Kosko, 1999). ANFIS uses the learning ability of ANN to define the input-output relationship (mapping) and construct the fuzzy rules by determining the input structure (interpretation). In other words, it combines the verbal power of a FL system and the numerical power of the ANN. Interpreted system results of ANFIS are obtained by the thinking and reasoning capability of the fuzzy logic. It provides the possibility to interpret the extracted results from Neuro-Fuzzy models, which is not possible with pure black box models such as ANNs. In this perspective, an expert can modify the rules or even add some rules based on his knowledge to expand the validity of the model (Babuška and Verbruggen, 2003) thereby making ANFIS preferable in forecasting than either ANN or FLI used individually.

2.7.1 ANFIS Architectures

There are no hard and fixed rules for developing ANFIS architectures (Lin and Namin, 2005). Sometimes, a general framework can be followed based on previous successful applications or experience. Ideally, an ANFIS is an ANN that is functionally equivalent to a first-order Sugeno-style FIS. Practically, there are six layers in an ANFIS model (Lin and Namin, 2005). One input layer, four hidden layers and one output layer. Each layer performs a particular task to forward the signals. Such an ANFIS model is shown in Figure 2.21.

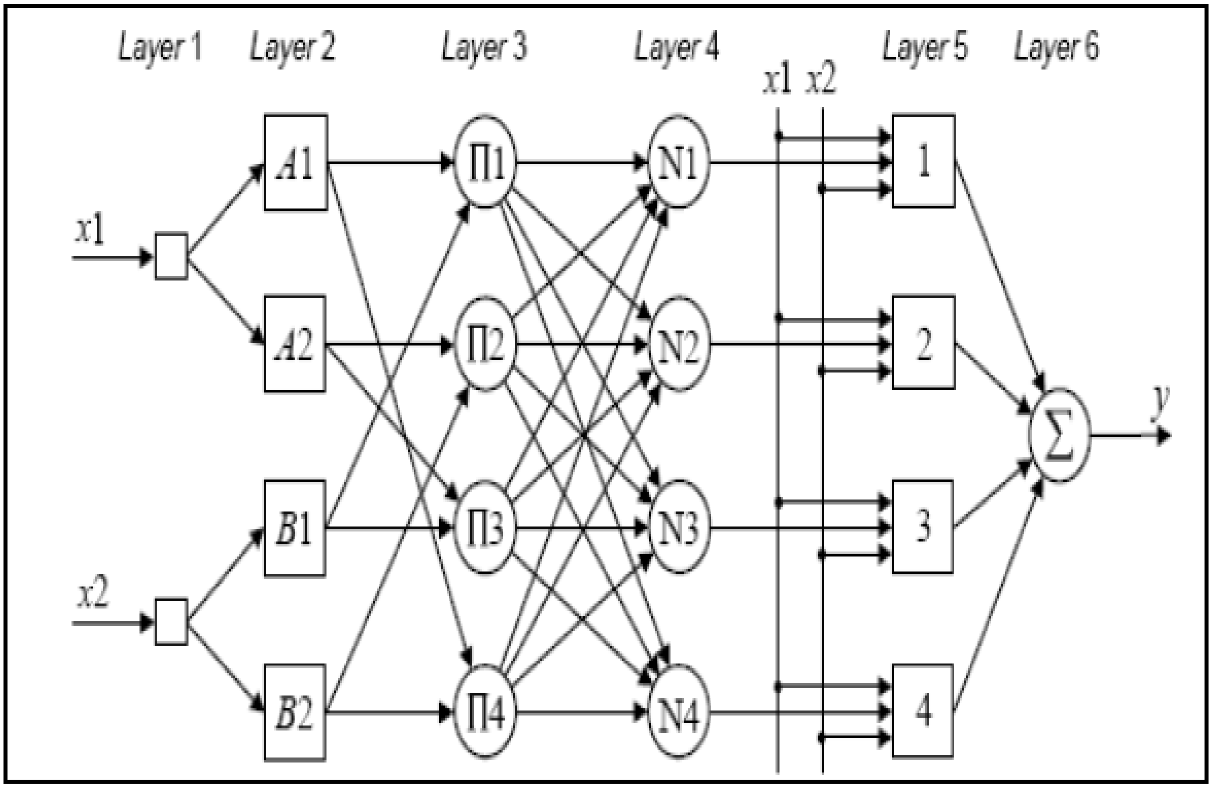


Figure 2.21: ANFIS Architecture Model (Dixon, 2005)

The first layer of the ANFIS model is the input layer. The Neurons in this layer helps to transmit the external input (crisp) signals to the next layer as shown in Equation 2.19 (Dixon, 2005).

$$x_i^1 + y_i^1 \qquad 2.19$$

Where x_i^1 is the input signal and y_i^1 is the output signal of neuron i in the first layer.

The second layer which is the first hidden layer of the ANFIS model is the fuzzification layer. Neurons in this layer represent antecedent fuzzy sets of fuzzy rules. A fuzzification neuron receives an input signal and then determines the degree to which this signal belongs to the neuron's fuzzy set. If we let x_i^2 be the input and y_i^2 be the output signal of neuron i in the second layer, then we have:

$$y_i^2 = f(x_i^2) \qquad 2.20$$

Where *f* represents the activation function of neuron i, and is set to a certain membership function.

The third layer which is the second hidden layer is the fuzzy rule layer. Each neuron in this layer corresponds to a single first-order Sugeno fuzzy rule which receives signals only from the fuzzification neurons that are involved in the antecedents of the fuzzy rule it represents. It helps to compute the truth value of the rule. In an ANFIS, the 'product' operator is used to evaluate the conjunction of the antecedents (Dixon, 2005). Therefore, we have:

$$y_i^3 = \prod_c^m x^3 c_i \qquad 2.21$$

Where $x^3 c_i$ is the signal from fuzzification neuron c in the second layer to neuron i in the third layer. y_i^3 is the output signal of neuron i in this layer; and m is the number of antecedents of the fuzzy rule neuron i represents.

The fourth layer which is the third hidden layer is the normalization layer. Each neuron in this layer receives signals from all rule neurons in the third layer and then calculates the so-called normalized firing strength of a given rule. This strength value represents the contribution of a given rule to the final result and is obtained as (Dixon, 2005):

$$y_i{}^4 = \frac{x_d^4}{\sum x_d^4} \qquad 2.22$$

Where x_d^4 is the signal from rule neuron d in the third layer to neuron i in the fourth layer; $y_i{}^4$ is the output signal of neuron i in this layer; and n is the number of rule neurons in the third layer.

The fifth layer which is the fourth hidden layer is the defuzzification layer. Each neuron in this layer is connected to the respective normalization neuron in the fourth layer and also receives initial input signals. A defuzzification neuron computes the 'weighted consequent value' of a given rule as:

$$y_i^5 = x_i^5 (k_{io} + k_{i1} x_i + k_{i2} x_2 + \ldots\ldots\ldots.. + k_{in} x_n) \qquad 2.23$$

Where x_i^5 is the input and y_i^5 is the output signal of neuron i in the fifth layer; and $k_{io} + k_{i1} x_i + k_{i2} x_2 + \ldots\ldots\ldots.. + k_{in} x_n$ is a set of consequent parameters of rule i.

The sixth layer which is the output layer is the summation layer. There is only one neuron in this layer, which calculates the sum of outputs of all defuzzification neurons in the fifth layer and consequently produces the overall ANFIS output y as follows (Dixon, 2005):

$$y = \sum_{i=1}^{n} x_i \qquad 2.24$$

Where x_i is the signal from defuzzification neuron i in the fifth layer to this summation neuron; and n is the number of defuzzification neurons, namely the number of fuzzy rules in the ANFIS model.

2.7.2 ANFIS Universal Function Approximators

Artificial Neural Networks (ANN), Fuzzy Logic (FL), and Adaptive Neuro Fuzzy Inference System (ANFIS) are known as universal function approximators which have been used in various applications. In general, a function approximator is a method that closely captures the relationship between input–output variables. Both the Mamdani and Sugeno systems are known as universal approximators, as they can approximate any continuous functions to any degree of accuracy (Tütmez and Tercan, 2007). The smaller the error tolerance, the more the fuzzy rules are needed. Fuzzy models can always produce nonlinear modeling solutions; in practice, when the required number of fuzzy sets and rules are provided (Tütmez and Tercan, 2007).

The Sugeno Fuzzy model was proposed by Takagi, Sugeno and Kang in an effort to develop a systematic approach to generating fuzzy rules from a given input-output dataset. It was introduced in 1985 and it is similar to the Mamdani's method in many respects. In this method, the first two parts of the fuzzy inference process which are fuzzifying the inputs and applying the fuzzy operator, are the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant (first order or zero order) respectively. Among many FIS models, the Sugeno fuzzy model is the most widely applied one for its high interpretability, computational efficiency, and built-in optimal and adaptive techniques (Ren, *et al*., 2006). Sugeno Inference System is commonly used with ANFIS while the Mamdani is commonly used with Fuzzy logic (Ren, *et al*., 2006). Sugeno Inference System can be first order or zero order. But the sugeno first order inference system is commonly used with ANFIS (Ren, *et al*., 2006).

The distinctive advantage of the Mamdani approximator over the Sugeno one lies in its unique ability to use not only numerical data but also verbal data obtained from human knowledge and experience (Tütmez and Tercan, 2007). Ying *et al*. (1999) proposed that Sugeno fuzzy systems can be more economical in the number of input fuzzy sets and fuzzy rules than the Mamdani fuzzy systems, if non trapezoidal/non triangular input fuzzy sets are used. They concluded that, when trapezoidal or triangular input fuzzy sets are used, Sugeno and Mamdani fuzzy systems have comparable minimal system configurations. Tütmez and Tercan (2007) compared the performance of Mamdani (linguistic) and Sugeno (clustering-based) fuzzy models in the spatial interpolation of the mechanical properties of rocks. Their results indicate that prediction performance of the clustering-based Sugeno fuzzy modelling approach is better than that of the Mamdani model.

A typical fuzzy rule in a Sugeno fuzzy model has the form (Tütmez and Tercan, 2007):

If x is A and y is B then z = f(x, y) 2.25

Where *A* and *B* are fuzzy sets in the antecedent, while *z = f(x, y)* is a crisp function in the consequent. Usually, *f(x, y)* is a polynomial in the input variables x and y, but it can be any function as long as it can appropriately describe the output of the model within the fuzzy region specified by the antecedent of the rule. When f(x, y) is a first-order polynomial, the resulting fuzzy inference system is called a first-order Sugeno fuzzy model (Kosko, 1999). When *f* is a constant, we then have a zero-order Sugeno fuzzy model, which can be viewed either as a special case of the Mamdani Fuzzy inference system, in which each rule's consequent is specified by a fuzzy singleton (or a pre-defuzzified consequent), or a special case of the Tsukamoto fuzzy model, in which each rule's consequent is specified by an MF of a step function centre at the constant. Moreover, a zero-order Sugeno fuzzy model is functionally equivalent to a radial basis function network under certain minor constraints (Kosko, 1999).

The final output result of a zero-order Sugeno model is a smooth function of its input variables as long as the neighbouring membership functions in the antecedent have enough overlap. Similarly, the overlap of MFs in the consequent of a Mamdani model does not have a decisive effect on the smoothness; it is the overlap of the antecedent MFs that determines the smoothness of the resulting input-output behaviour (Lin and Namin, 2005). Azero-order Sugeno model has unlimited approximation power for matching well any nonlinear function arbitrarily on a compact set if the number of rules is not restricted,. Figure 2.22 shows the fuzzy reasoning procedure for a first-order Sugeno fuzzy model.

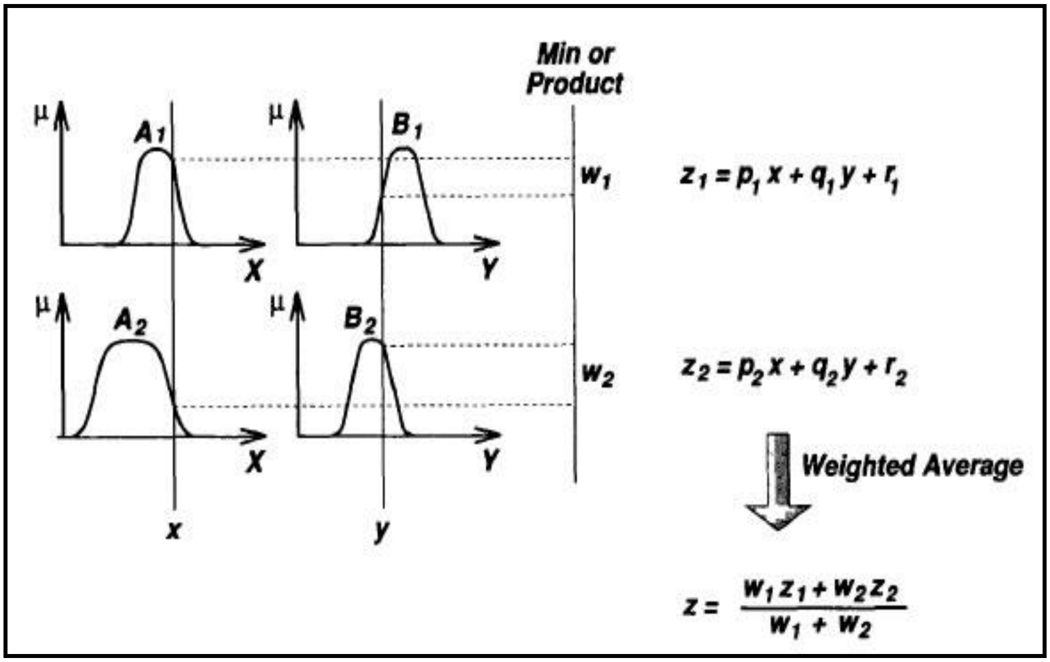


Figure 2.22: First order Sugeno Fuzzy Model (Lin and Namin, 2005)

2.7.3 ANFIS Optimization Algorithms

ANFIS uses two types of optimization algorithms to fine-tune its parameters. They are hybrid optimization and backpropagation optimization. It uses hybrid learning algorithm to fine-tune the parameters of a Sugeno-type fuzzy inference system (FIS) which is a combination of the least-squares and back-propagation gradient descent methods to model a training data set. Gradient descent implies going downhill in small steps until you reach the bottom of error surface and is the learning technique used in back propagation. The back propagation weight update is equal to the slope of the energy function that is further scaled by a learning rate η and the steeper the slope, the bigger the update but may cause a slow convergence (Parveen, 2012). ANFIS also helps to validate models using a checking data set to test for overfitting of the training data. Backpropagation helps to find the optimal values of the nonlinear learning parameters e.g., the parameters of the Gaussian member ship function.

Therefore, a hybrid learning algorithm combines the backpropagation gradient descent and the least squares estimate method, which outperforms the original backpropagation algorithm (Rumelhart, 1986). The consequent parameters (parameters after fuzzification) are updated first using the least squares algorithm and the antecedent parameters (parameters before fuzzification) are then updated by back propagating the errors that still exist. This hybrid algorithm also involves forward and backward passes.In the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameters are identified by the least squares method. In the backward pass, the error signals propagate backward and the premise parameters (consequent parameters with error function) are updated by gradient descent. Jang and Sun (1995) gives more details about hybrid learning algorithm.

2.7.4 Data Training, Validation and Testing using ANFIS

As ANN can be trained, ANFIS can also be trained to learn from given data. Training the neural network is the process of finding the minimum of a complicated nonlinear function called 'error function'. This function helps to describe the error the neural network makes in approximating or classifying the training data, as a function of the weights of the network; w. For an ANFIS model to be configured for a specific problem, the fuzzy rules and the activation functions (i.e. membership functions) of fuzzification neurons need to be specified. In terms of the fuzzy rules, the antecedent fuzzy sets can be specified once we know the specific problem domain; while for the consequents of the fuzzy rules, the parameters (e.g.) are formed and adjusted by certain learning algorithm in the training process. Also, the shapes of activation functions can be formed and adjusted in the training process (Jang *et al.,* 1997). For ANFIS models, the bell-shaped function is the one most used activation function. This is described in Figure 2.23 and represented by the Equation 2.26:

$$y = \frac{1}{1+\left\{(x-\frac{s}{r})2\right\}t} \qquad 2.26$$

Where *r*, *s* and *t* are the parameters that control the slope, center and width of the bell-shaped function respectively. In the training process, these parameters can be specified and adjusted by the learning algorithm.

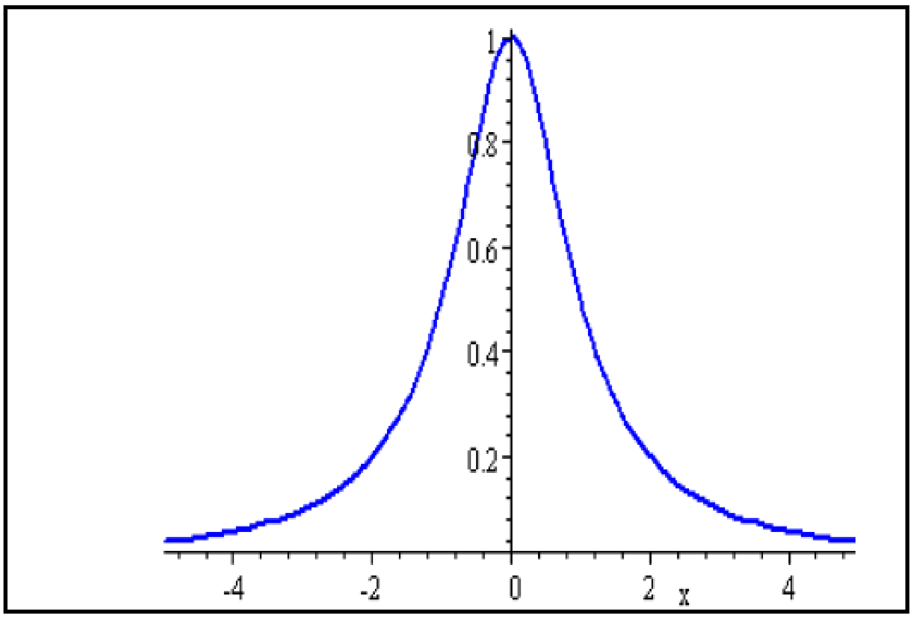


Figure 2.23: A Bell-Shaped Function with r = t = 1 and s = 0 (Jang *et al.*, 1997)

Model Validation using checking and testing data sets is the process by which the input vectors from input/output data sets on which the FIS was not trained, are presented to the trained FIS model, to see how well the FIS model predicts the corresponding data set output values. You use a validation data set to check and control the potential for the model overfitting the data. A major problem with model validation using adaptive techniques is how to select a data set that will both represent data that will follow the trend of the training model and at the same time will not be used in the training phase (Jang *et al.*, 1997). This helps in order not to render the validation process trivial. Also, if you expect to be presenting noisy measurements to your model, it is possible the training data set does not include all of the representative features you want to model.

2.7.5 ANFIS Initialization

In ANFIS, determination of antecedent and consequent parameters are very difficult tasks. Two methods are used for ANFIS initialization to overcome these problems. They are grid partitioning and subtractive clustering. Grid partition and fuzzy clustering are the two methods most often used to identify the antecedent membership functions (MFs), whereas the linear least-squares method is used to compute the consequent parameters (Chi, 1994). The antecedent parameters are the parameters before applying the training algorithm and the consequent parameters are the parameters obtained after applying the training algorithm like linear least-squares method. Grid partition is the process of breaking complex input-output data structures into simpler ones that can easily be handled by ANFIS (Chi, 1994).

Data can also be clustered and it forms the basis of many classification and system modeling algorithms. The sole aim of data clustering is to identify natural groupings of data from a large data set to produce a concise representation of a system's behaviour (Chi, 1994). The Fuzzy Logic Toolbox has many preset tools used to find clusters in input-output training data. Data clustering information can also be used to generate a Sugeno-type fuzzy inference system that best models the data behaviour using a minimum number of rules. These rules partition themselves (grid partioning) according to the fuzzy qualities associated with each of the data clusters (Figure 2.24).

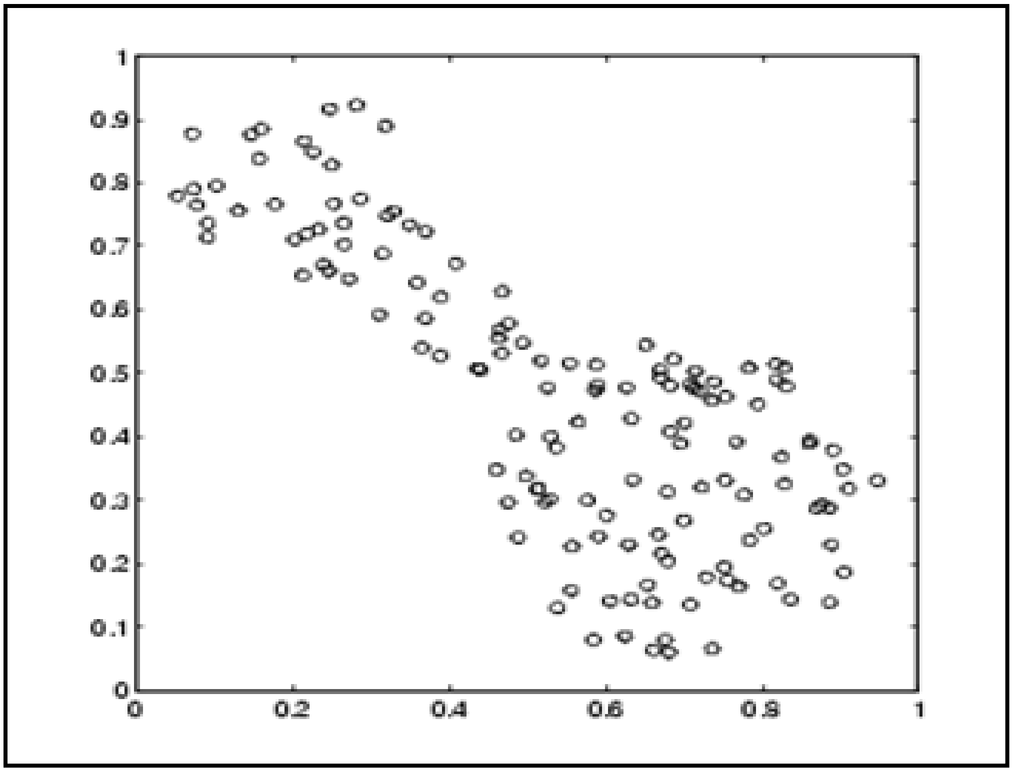


Figure 2.24: Example of 2-D cluster data (Bezdek, 1981)

2.7.6 Merits and Demerits of ANFIS

The following are the merits of using ANFIS over fuzzy logics and Artificial Neural Networks (Aqil *et al.*, 2007):

i. ANFIS can be easily implemented for a given input/output task and hence it is attractive for many application purposes.

ii. ANFIS model integrates the ANN and FIS tools into a 'compound', meaning that there are no boundaries to differentiate the respective features of ANN and FIS.

iii. ANFIS has the ability to model linear and non linear data sets.

iv. It has a fast learning capability. After proper training, an ANFIS completely bypasses the repeated use of complex iterative process for new cases presented to it. In short, an ANFIS is very fast after training.

v. It has a great adaptability to data sets.

vi. It can build complex non linear relationship between inputs and outputs data sets.

vii. Like fuzzy logic, it has no prior knowledge of the system model.

viii. It has the ability to self-organize network structures and to tune the parameters of the fuzzy system.

The major demerit of ANFIS is the time consumed to build the structure model including fuzzy logic and ANN reasoning capabilities (Aqil *et al.*, 2007). The ANFIS is highly more complex than the ordinary fuzzy inference systems. It is also not available for all the options of the fuzzy inference system. Ideally, the ANFIS only supports Sugeno-type systems (zero or first order), and these must have the following properties (Aqil *et al.*, 2007):

i. It must either be a first or zeroth order Sugeno-type systems.

ii. It must have a single output function which is obtained using weighted average defuzzification.

iii. All the output membership functions must be the same type and can either be linear or constant.

iv. There must be no rule sharing. Different rules cannot share the same output membership function.

v. The number of output membership functions must be equal to the number of rules.

vi. It must have unity weight for each rule.

If these constraints are not strictly followed, error functions will occurred.

2.7.7 Application of ANFIS in Water Resources and Hydrology

ANFIS is a well-known artificial intelligence technique that has been used currently in hydrological processes (Zounemat-Kermani and Teshnehalb, 2008). Use of this technique for rainfall–runoff and river flow time series predictions has been reported by many researchers (Valenca and Ludermir, 2000; Chang and Chen, 2001, 2006; Nayak *et al.*, 2004, 2005). Very few studies are available on rainfall–runoff modelling and river flow forecasting using ANFIS together with its performance comparison with an ANN technique (Chau *et al.*, 2005; Aqil *et al.*, 2007). Tayfur and Singh (2006) used ANFIS models for simulating event-based rainfall–streamflow. Mukarji *et al* (2009) applied the ANN and ANFIS mode to forecast stream flow for Ajay River Basin in Jharkhand, India and results observed that ANFIS model predicts better than the ANN model in most of the cases.

ANFIS has also been used in the following areas: Optimal operation of multipurpose reservoir (Mehta and Rain, 2009), hydrological time series prediction (Zounemat-Kermani and Teashnehalb, 2008), municipal water consumption modeling (Yurdusev and Firat, 2009), stream flow forecasting (Swain and Umamahesh, 2004), sediment volume prediction (Cigizoglu and Alp, 2006), ground water flow prediction (Gungor, 2007), reconstruction of missing precipitation events (Abebe et al., 2002, Dastorani *et al.*, 2010), short term water level prediction (Erinawati and Fento, (2012), prediction of scour depth at culvert outlets (Azamathulla and Ghani, 2011) and Evapotranspiration prediction (Firat, 2007).

2.8. Performance Evaluation Criteria

There are a number of reasons why hydrologists and water resources engineers need to evaluate model performance: (1) to provide a quantitative estimate of the model's ability to reproduce historic and future watershed behaviour; (2) to provide a means for evaluating improvements to the modeling approach through adjustment of model parameter values, model structural modifications, the inclusion of additional observational information, and representation of important spatial and temporal characteristics of the watershed; (3) to compare current modeling efforts with previous study results.

In this study, the performances of ANN and ANFIS network models are compared by use of five statistical goodness of fit measures:

1. Correlation coefficient (*R*)
2. Coefficient of determination (R^2)
3. Mean square error (MSE)
4. Nash-Sutcliffe efficiency (E)
5. Index of Agreement (IOA)

2.8.1. Coefficient of Correlation (R)

Coefficient of correlation measures the strength of association between two variables. It is generally denoted by r or R (Legates and McCabe, 1999).

$$r = \frac{\sum_{i=1}^{n}\left(\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2 \sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}} \qquad 2.27$$

Where: x_i = raw observed values for x, $\bar{X}$ = mean value of x_i, y_i = raw observed values for y

$\bar{y}$ = mean value of y_i, n = number of data

- The value of a correlation coefficient ranges between -1 and 1.
- The greater the absolute value of a correlation coefficient, the stronger the *linear* relationship.

- The strongest linear relationship is indicated by a correlation coefficient of -1 (negative correlation or 1 (positive correlation).
- The weakest linear relationship is indicated by a correlation coefficient equal to 0.

A correlation of 0 does not mean zero relationship between two variables; rather, it means zero linear relationship. It is possible for two variables to have zero linear relationship and yet have a strong curvilinear relationship.

2.8.2. Coefficient of Determination (R^2)

The coefficient of determination (denoted by R^2 or r^2) is a key output of regression analysis. It is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable. It is the square of the coefficient of correlation (r). Coefficient of correlation is also called coefficient of efficiency (CE). The formula for computing the coefficient of determination for a linear regression model with one independent variable is given below (Legates and McCabe, 1999).

$$r^2 = \left(\frac{\sum_{i=1}^{n} (O_i - \bar{O})(P_i - \bar{P})}{\sqrt{\sum_{i=1}^{n} (O_i - \bar{O})^2} \sqrt{\sum_{i=1}^{n} (P_i - \bar{P})^2}} \right)^2 \qquad 2.28$$

Where n is the number of observations used to fit the model, Σ is the summation symbol, o raw observed data, $\bar{o}$ the mean of the observed data and p predicted values and $\bar{p}$ mean of predicted value.

- The coefficient of determination is the square of the correlation coefficient (R) between predicted scores and actual scores; thus, it ranges from 0 to 1.

- An r^2 of 0 means that the dependent variable cannot be predicted from the independent variable.
- An r^2 of 1 means the dependent variable can be predicted without error from the independent variable.
- An r^2 between 0 and 1 indicates the extent to which the dependent variable is predictable. An r^2 of 0.10 means that 10 percent of the variance in the dependent variable is predictable from the independent variable; an R^2 of 0.20 means that 20 percent is predictable; and so on.

2.8.3. Mean Square Error (MSE)

Mean-square error (MSE) or the mean-square deviation (MSD) measures the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed. Basically, the MSD represents the sample variance of the differences between predicted values and observed values. These individual differences are called **residuals** when the calculations are performed over the data sample that was used for estimation, and are called **prediction errors** when computed out-of-sample. The MSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. MSD is a good measure of accuracy, but only to compare forecasting errors of different models for a particular variable and not between variables, as it is scale-dependent.

The MSD of predicted values y_t for times t of a regression's dependent variable y is computed for n different predictions as the the mean of the squares of the deviations and is show below (Legates and McCabe, 1999).

$$MSD = \frac{\sum_{t=1}^{n}(y_t - y)^2}{n} \tag{2.29}$$

2.8.4. Nash-Sutcliffe or Modeling Efficiency (E)

Nash-Sutcliffe or modeling efficiency (E) proposed by Nash and Sutcliffe (1970) is defined as one minus the sum of the absolute squared differences between the predicted and observed values normalized by the variance of the observed values during the period under investigation. It is calculated as:

$$E = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} \left(O_i - \bar{O}\right)^2} \qquad 2.30$$

Where: O_i = observed values, P_i = predicted values, $\bar{O}$ = mean of O values, n = number of observed data.

Value of E lies the range between 1.0 (perfect fit) and - ∞ (infinity). An efficiency value lower than zero indicates that the mean value of the observed time series would have been a better predictor than the model (Legates and McCabe, 1999).

2.8.5. Index of Agreement (IOA)

The index of agreement ((IOA) also denoted by (*d*) was proposed by Willmot (1981) to overcome the insensitivity of Nash-Sutcliffe efficiency E and coefficient of determination r^2 to differences in the observed and predicted means and variances (Legates and McCabe, 1999). The index of agreement represents the ratio of the mean square error and the potential error (Willmot, 1984) and is defined as:

$$d = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} \left(\left| P_i - \bar{O} \right| + \left| O_i - \bar{O} \right| \right)^2} \qquad 2.31$$

All symbols are as defined previously. The potential error in the denominator represents the largest value that the squared difference of each pair can attain. With the mean square error in the numerator, d is also very sensitive to peak flows and insensitive for low flow conditions as it is Nash-Sutcliffe efficiency E. The range of d is similar to that of r^2 and lies between 0 (no correlation) and 1 (perfect fit). Practical applications of d show that it has some disadvantages (Legates and McCabe, 1999): (1) relatively high values (more than 0.65) of d may be obtained even for poor model fits, leaving only a narrow range for model calibration; and (2) despite Willmot's intention, d is not sensitive to systematic model over- or under prediction (Willmot, 1984).

CHAPTER 3

3.0. RESEARCH METHODOLOGY

3.1. The Study Area

Ikpoba River is situated within the rainforest belt of Edo State and the Western Littoral hydrological area (HA-6) of Nigeria (Akintola, 1986). The Western Littoral hydrological area (HA-6) is one of the eight hydrological areas (HAs) into which Nigeria is subdivided. The gauging station from which the data for this study was collected is located along Ikpoba River at Benin City which is about 160km due East of Lagos. The River rises from the Ishan Plateau in the Northern part and flows in south west direction in a steeply incised valley and through sandy areas before passing through Benin City and joining the Ossiomo River Basin. Related to the town, the river is situated between Latitude 6°19'41.71" to 6°54'53.41" and Longitude: 5°38'45.39" to 5°55'49.77" and its drainage area is 922km^2 **(**BORBDA, 2005).

The Ikpoba dam and reservoir site is located, spanning from Okhoro to Teboga, along the Ikpoba river running through Egor and Ikpoba-Okha local government area in Benin City, Edo state (Anyata *et al.*, 2013). The dam has a reservoir storage capacity of 1.5 x 10^6 m^3, reservoir catchment area of 120 km^2 and a reservoir surface area of 1.07 x 10^6 m^2 (www.wds.worldbank.org/external/defal.). The dam is 610m long with a crest level height of 35m above mean sea level (MSL) (Ehiorobo, 2008). It has a spillway length (weir) of 60m and an emergency spillway length of 4m (Anyata *et al.*, 2013). The storage capacity may have changed due to siltation and effect of climate change.

The dam reservoir, a single purpose reservoir is the main source of water supply for Benin City. The proposed ultimate capacity 160,000 m^3/day (Ehiorobo, 2008). This accounts for about 60% of the water supply requirement for Benin City with a population of about 1.5 million as at 2015 with an annual growth rate (2006-2015) of 3.9% (http://population.city/nigeria/benin-city/). At present, problems associated with the reservoir are siltation and growth of weeds over the years (Edo State Urban Water Board, 2007) and underutilization of the reservoir (Anyata *et al.*, 2013). Hence, there is need for effective modelling and forecasting of the river discharge in order to assess and evaluate the possibility of converting the reservoir into a multipurpose reservoir which can be used for water supply, hydropower, irrigation and recreation. Figure 3.1 shows the location map of the study area and Table 3.1 shows the parameters of the hydrological gauging station.

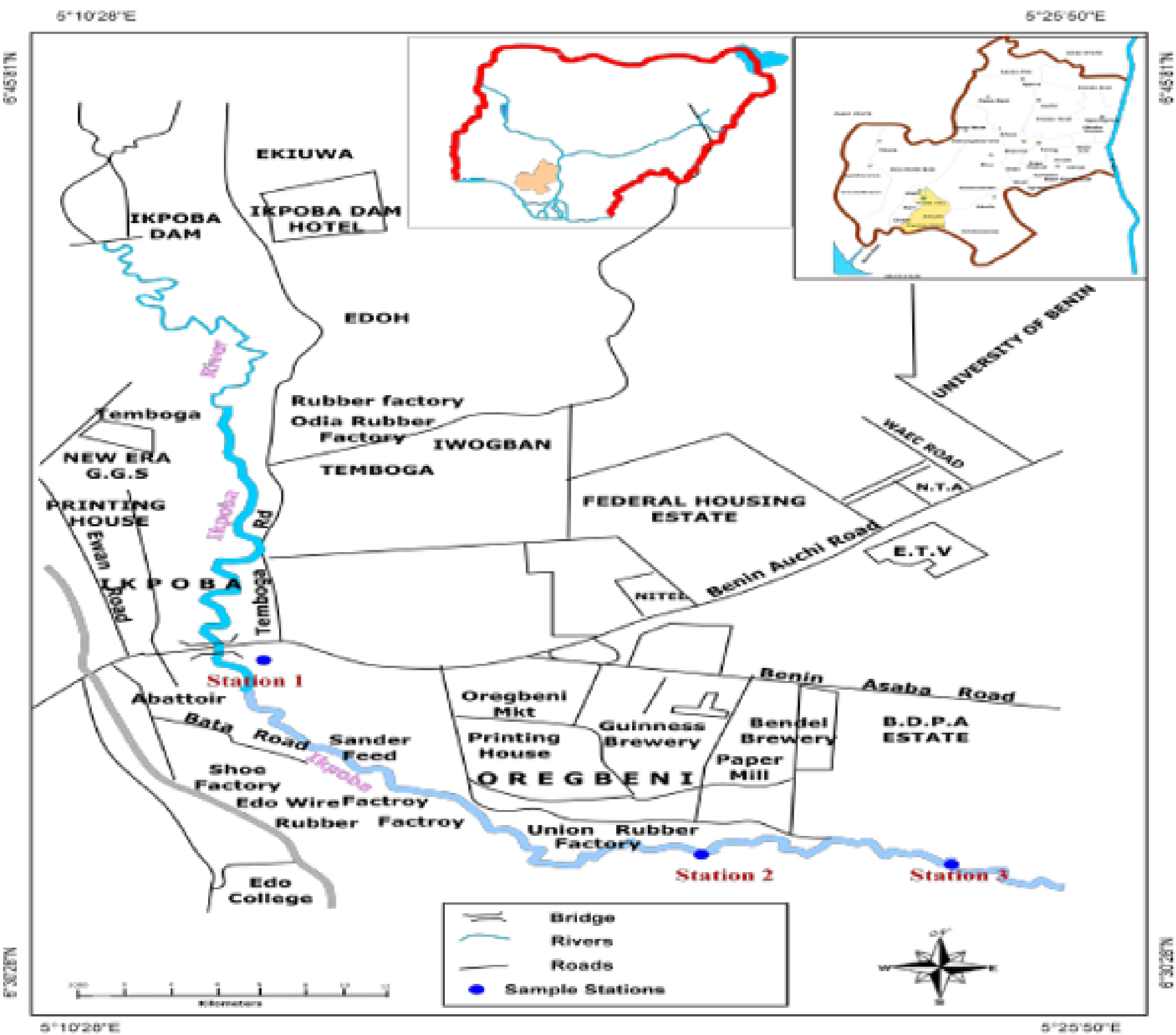


Figure 3.1: Location Map of Ikpoba dam along Ikpoba River (www.researchgate.net/publications)

Table 3.1: Ikpoba River Hydrological Gauging Station Parameters

Location of Station	State of Location	Basin	Latitude	Longitude	Drainage Area (km^2)
Ikpoba River at Benin City	Edo, Nigeria	Ossiomo	$6^0$20'N	$5^0$39'E	922

Source: (BORDA, 2005)

3.2. Collection and Pre-processing of the Data Used

The discharge and the run off data sets used for this study were obtained from Owena-River Basin in Benin-City, Edo State while the precipitation and temperature date sets were obtained from NIMET, Lagos. The only available discharge and runoff data sets used were the daily, monthly and annual discharges and run-off volumes of Ikpoba river from 1991 to 1995 (5 years) and 1999 to 2000 (2 years). The precipitation data sets from 1991 to 2010 (20 years) and the temperature data sets from 1991 to 2000 (10 years) were used to analyze the effect of climate change on the discharge forecasting. Figure 3.2 shows the mean daily discharge of Ikpoba River from 1991-1995, Figure 3.3 shows the mean daily precipitation from 1991-2010 and Figure 3.4 shows the mean daily temperature from 1991-2000. For these data sets to be used in hydrological context, they were first preprocessed. Firstly, the discharge data sets from October to December 1991 and January 1992 were missing and to make the data set complete, these missing data were filled by the mean values of their corresponding data sets using mean imputation method (Enders, 2010). 1825 daily discharge data for the five years (1991-1995) were used for training, testing and validation and the data set from 1999 to 2000 (730 data) were used independently for further testing and validation. The effect of climate change on the discharge forecasting indexed by precipitation and temperature were considered by incorporating the precipitation and temperature data sets (from 1991-1994) into the discharge data sets as inputs.

Secondly, the input and target output variable data sets used in the models were selected based on discharge, temperature and precipitation. This is one of the potential steps in developing accurate model because these variables determine the structures of the ANFIS and ANN models and affect the weighted coefficients and results of the models. Five models were developed. The first four models were based on the discharge only while the fifth model was based on the discharge, the temperature and the precipitation. The models were developed with the following inputs and target outputs data sets.

Model-1 used the daily discharge data set of 1994 (one elemental variable) as input and the 1995 daily discharge data set as the target output given rise to 365 x 1 matrix form.

Model-2 used the daily discharge data sets of 1993 and 1994 (two elemental variables) as inputs and the 1995 daily discharge data set as the target output given rise to 365 x 2 matrix form.

Model-3 used the daily discharge data sets of 1992, 1993 and 1994 (three elemental variables) as inputs and the 1995 daily discharge data set as the target output given rise to 365 x 3 matrix form.

Model-4 used the daily discharge data sets of 1991, 1992, 1993 and 1994 (four elemental variables) as inputs and the 1995 daily discharge data set as the target output given rise to 365 x 4 matrix form.

Model-5 used the mean monthly discharge, temperature and precipitation data sets of 1991, 1992, 1993 and 1994 (twelve elemental variables) as inputs and the 1995 daily discharge data set as the target output given rise to 12 x 12 matrix form.

Finally, outliers were checked for by visualizing the data sets. In statistical hydrology, an **outlier** is an observation point that is far distant from other observations. An **outlier** may be due to variability in the measurement or it may indicate experimental error. Major outliers were not found in the data set under consideration though minor one were noticed in the temperature date.

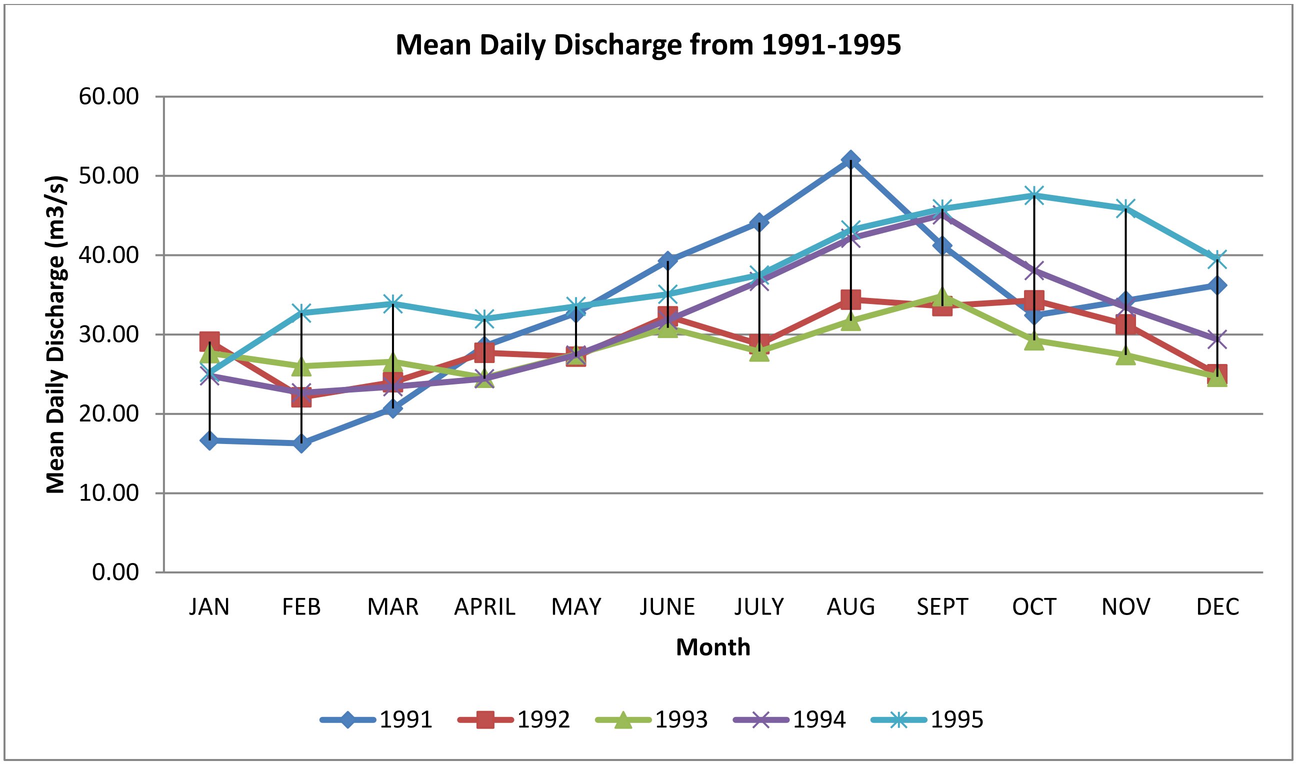


Figure 3.2: Mean Daily Discharge of Ikpoba River from 1991-1995

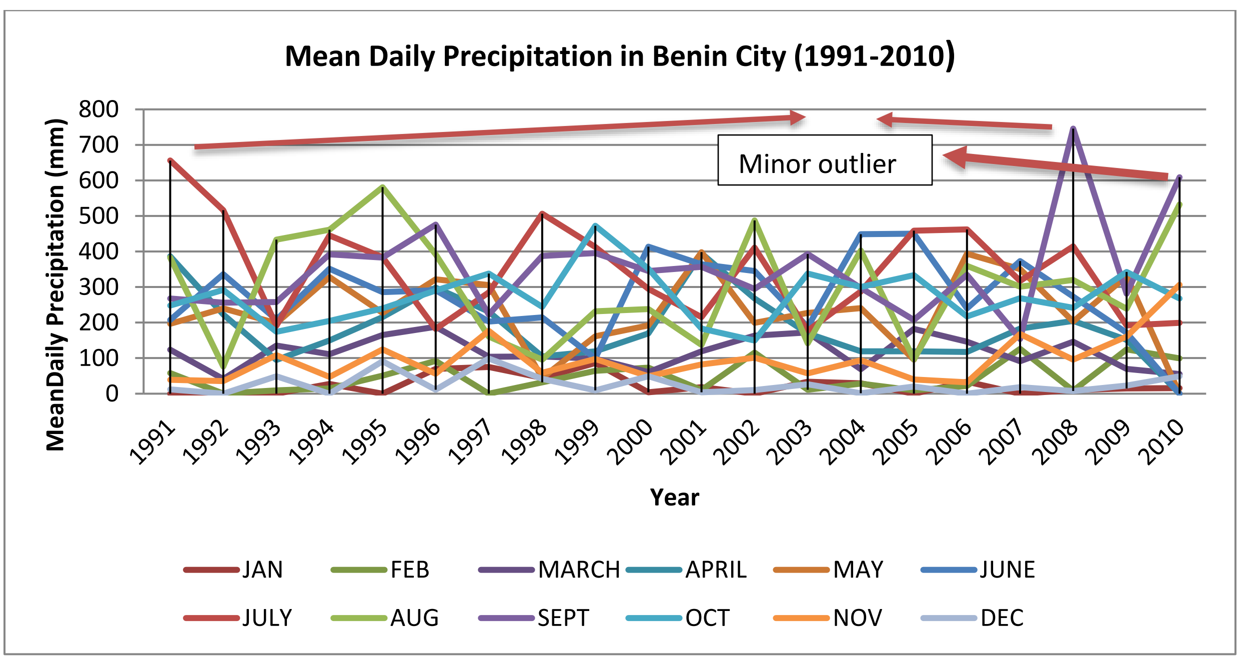


Figure 3.3: Mean Daily Precipitation in Benin City from 1991-2010

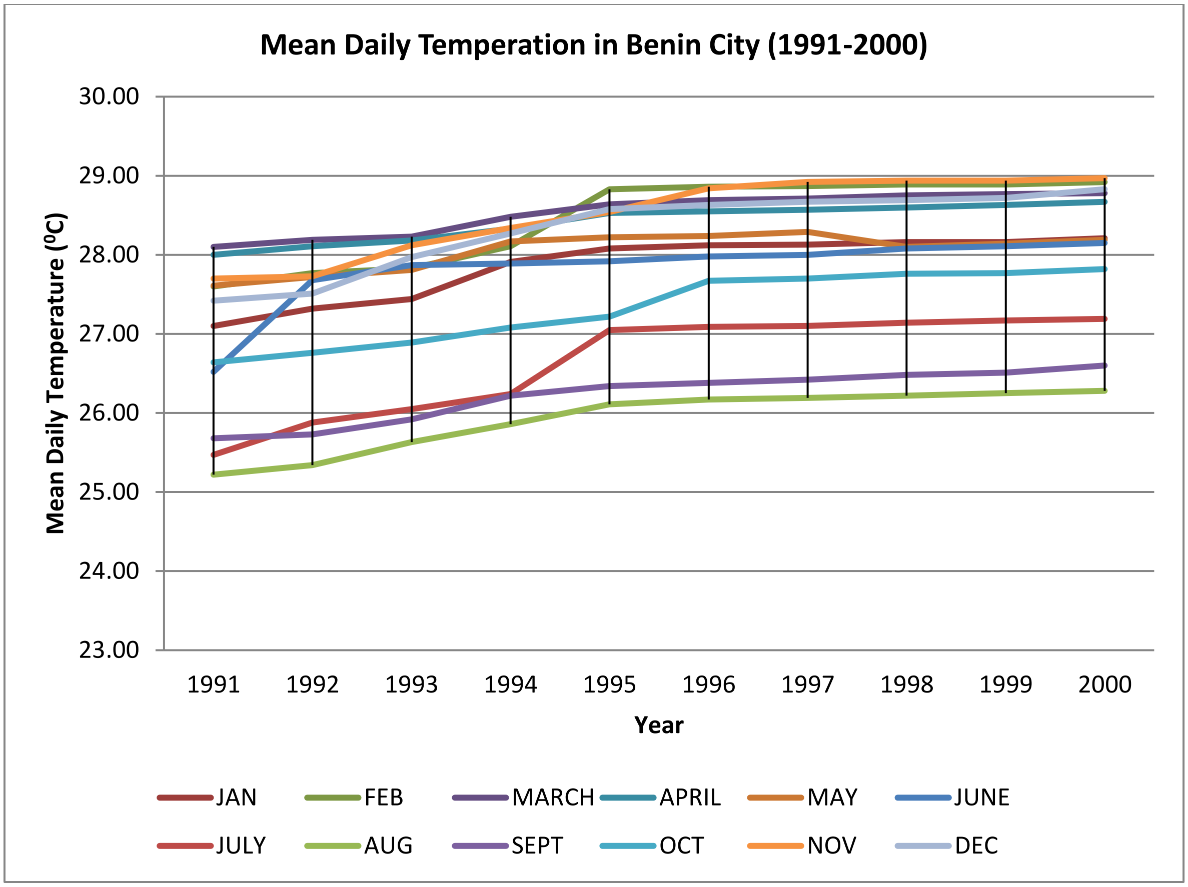


Figure 3.4: Mean Daily Temperature in Benin City from 1991-2000

3.3. Development of Models

Firstly, only the discharge data sets were considered in the forecasting model and after which the effect of climate change on the discharge forecasting indexed by temperature and precipitation were considered and interpreted. The first four models used different combinations of the antecedent discharges of the Ikpoba River to construct appropriate input structures of the time series forecasting models while the fifth model add temperature and precipitation. The general structure of the forecasting models is given in equation 3.1 below and the structures of the forecasting models are given in table 3.2 (Firat, 2007).

$$Q(t) = Q(t-1) + Q(t-2) + \ldots Q(t-n) \qquad 3.1$$

Table 3.2: Structure of the forecasting models

S/N	Model	Input Parameters	Input Structure	No of Input Variables	Target Output
1	Model-1	Q	Q(t-1)	1	Q(t)
2	Model-2	Q	Q(t-1), Q(t-2)	2	Q(t)
3	Model-3	Q	Q(t-1), Q(t-2), Q(t-3)	3	Q(t)
4	Model-4	Q	Q(t-1), Q(t-2), Q(t-3), Q(t-4)	4	Q(t)
5	Model-5	T P Q	T(t-1), T(t-2), T(t-3), T (t-4) P(t-1), P(t-2), P(t-3), P(t-4) Q(t-1),Q(t-2), Q(t-3), Q(t-4)	12	Q(t)

Where , Q(t) represents the river discharge at time (t), 1995; T(t-1), T(t-2), T(t-3), T (t-4); P(t-1), P(t-2), P(t-3), P(t-4) and Q(t-1), Q(t-2), Q(t-3), Q(t-4) represent the temperature, the precipitation and the river discharge at one time step (1994), second time steps (1994 and 1993), third time steps (1994, 1993 and 1992) and fourth time steps (1994, 1993, 1992 and 1991) lag respectively.

3.4. Modelling Procedures

Two different modelling and forecasting techniques were used to model Ikpoba river discharge. They are the adaptive neuro-fuzzy inference system (ANFIS) and the artificial neural network (ANN). MATLAB software was used for the modelling and forecasting. Different statistical performance criteria such as coefficient of correlation (R), coefficient of determination (R^2), mean square error (MSE), Nash-Sutcliffe or modelling efficiency (E) and index of agreement (IOA) were used to make comparisons among the network models using the two methods.

3.4.1. Model Application Using ANFIS

The following standard procedures for designing networks models to solve problems for time series analysis and forecast using ANFIS in MATLAB were utilized (Anctil and Rat, 2005):

Firstly, the selected input data sets and the target data sets for each of the five models Model-1, Model-2, Model-3, Model-4 and Model-5 were fed into ANFIS graphical user interface (GUI) of MATLAB.

Secondly, the fuzzy logic inference for fuzzification processes (crisps-fuzzy sets) was specified. The first order Sugeno fuzzy logic inference (linear function) was used for the five models. The number of fuzzy rules and the optimal number of fuzzy parameters required to define the FLI for the best result were decided based upon the number of inputs used and their types, as well as on the number of fuzzy membership function employed in the models. The number of fuzzy membership functions for each input model was considered to be 3 and the generalized bell (gbell) type of membership function which is a direct generalization of the Cauchy distribution as used in the probability theory with three parameters was used for all the five models. Due to its smoothness and concise expression, it is popularly used in many hydrological applications to specify the fuzzy sets (Jang and Sun., 1995). The outputs function of the ANFIS models were considered as a linear type for all the five models.

Thirdly, the learning algorithm to initialize the input data structures to generate the fuzzy sets used the grid partitioning method which helped to partition the data into various groups such as training, testing and validation (checking). Various percentages for the training data set, testing data set and validation data set were randomly selected. The partitioning of 70% training data set, 15% testing data set and 15% validation data set were found to give optimal results. The (70%) training data set were imputed into the ANFIS network during training and the networks were adjusted according to their errors. The

(15%) validation data set were used to measure the networks generalizations and to halt the training when generations stopped improving. The remaining (15%) testing data set have no effects on the training and so provide an independent measure of the network performance during and after training. The daily discharge data sets of (1999-2000) (730 data) were also further used to test the ANFIS models independently. This helped to ensure proper validation processes. All the various models were trained, tested and validated. The parameters of the membership functions were adjusted and trained using hybrid algorithm which is a combination of least square- Levenberg-Marquardt algorithm and gradient descent backpropagation algorithm.

Finally, the models were deffuzified. The number of Epochs used was 10, the error tolerance limit used was 0, the AND implication method used was PROD (product) and the deffuzification method used was the weighted average (wtaver) for all the five models.

Model-5 which incorporated temperature and precipitation data sets into the discharge data set was used to evaluate the effect of climate change on the discharge forecast. All the ANFIS procedures used above with discharge data sets only were also used but the mean daily data sets were used instead of the daily data set. This is to reduce the computational time since the input data sets are now 12.

Figures 3.5 to 3.15 show the network models for all the five models using ANFIS.

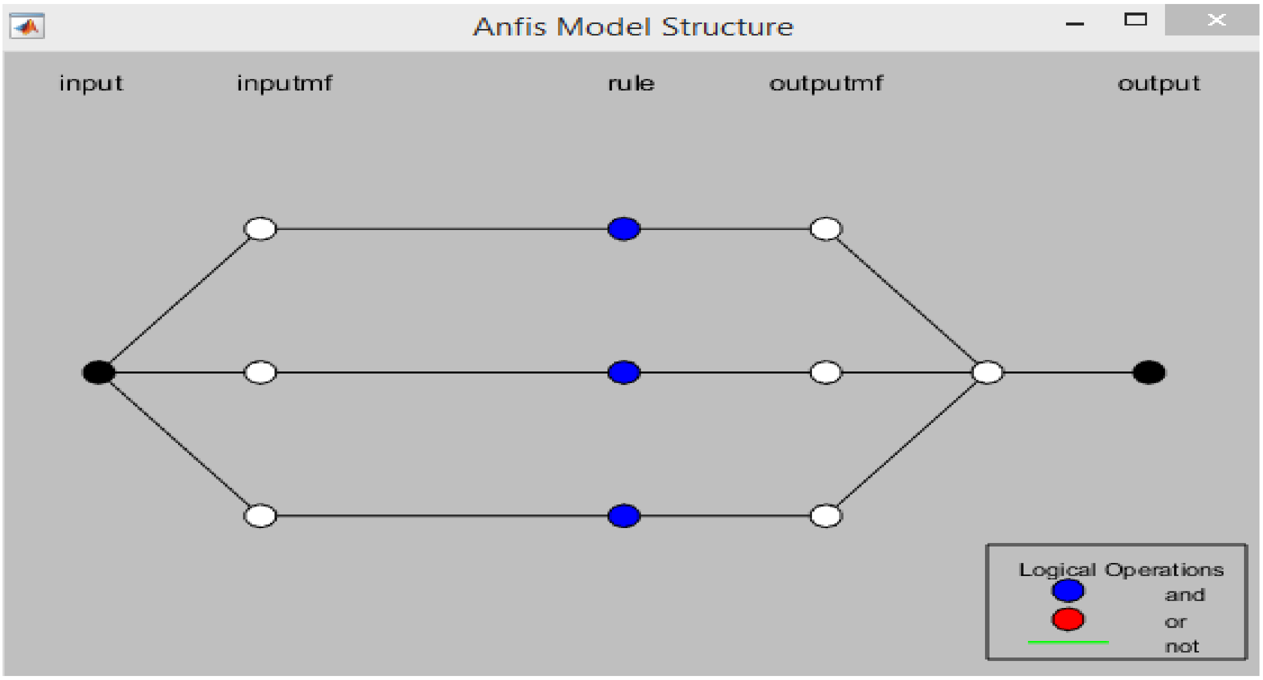


Figure 3.5: ANFIS model network for model-1 (1 input data)

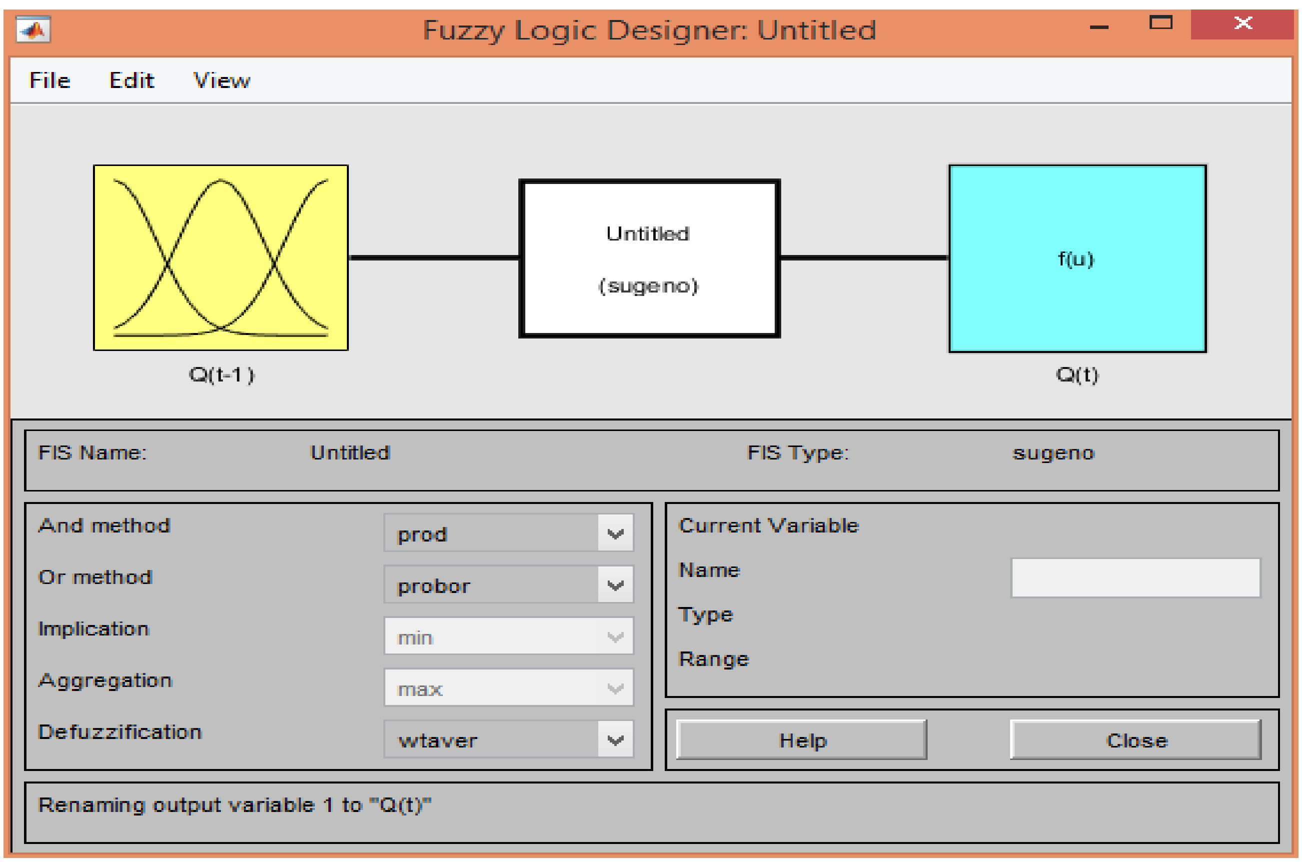


Figure 3.6: ANFIS deffuzification process for model-1

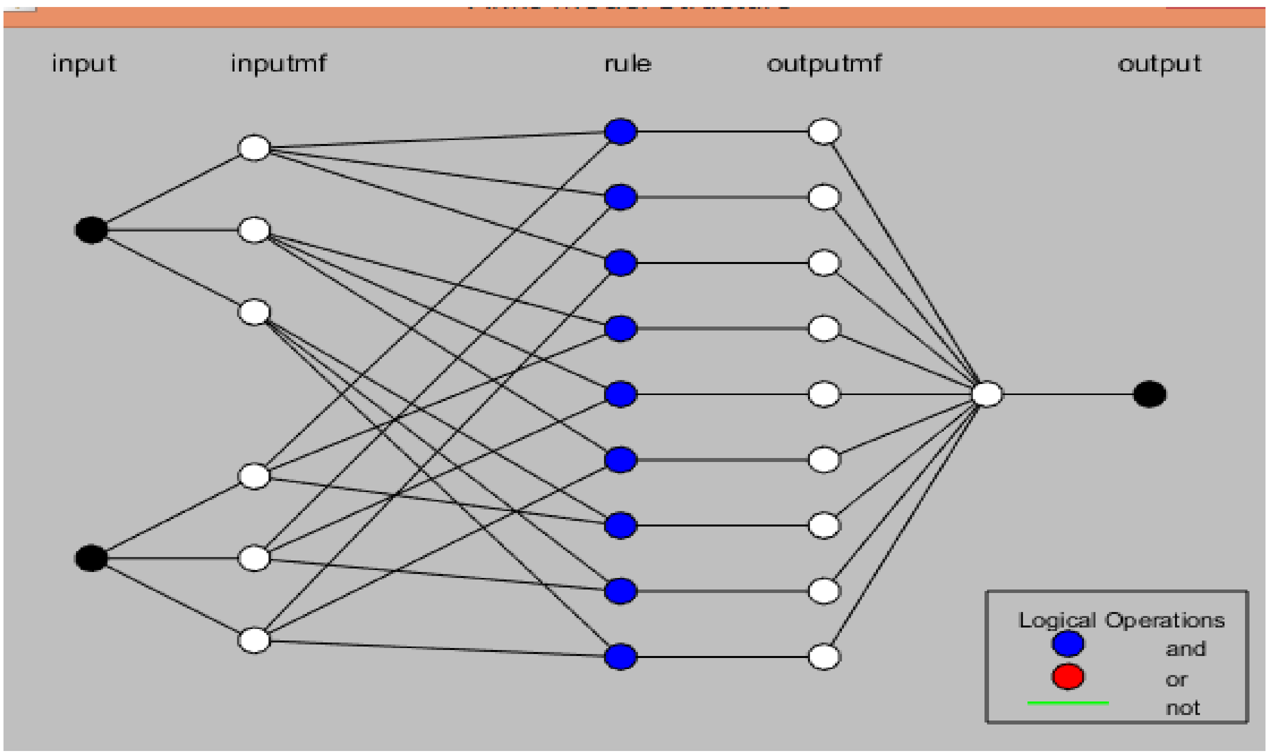


Figure 3.7: ANFIS model network for model-2 (2 inputs data)

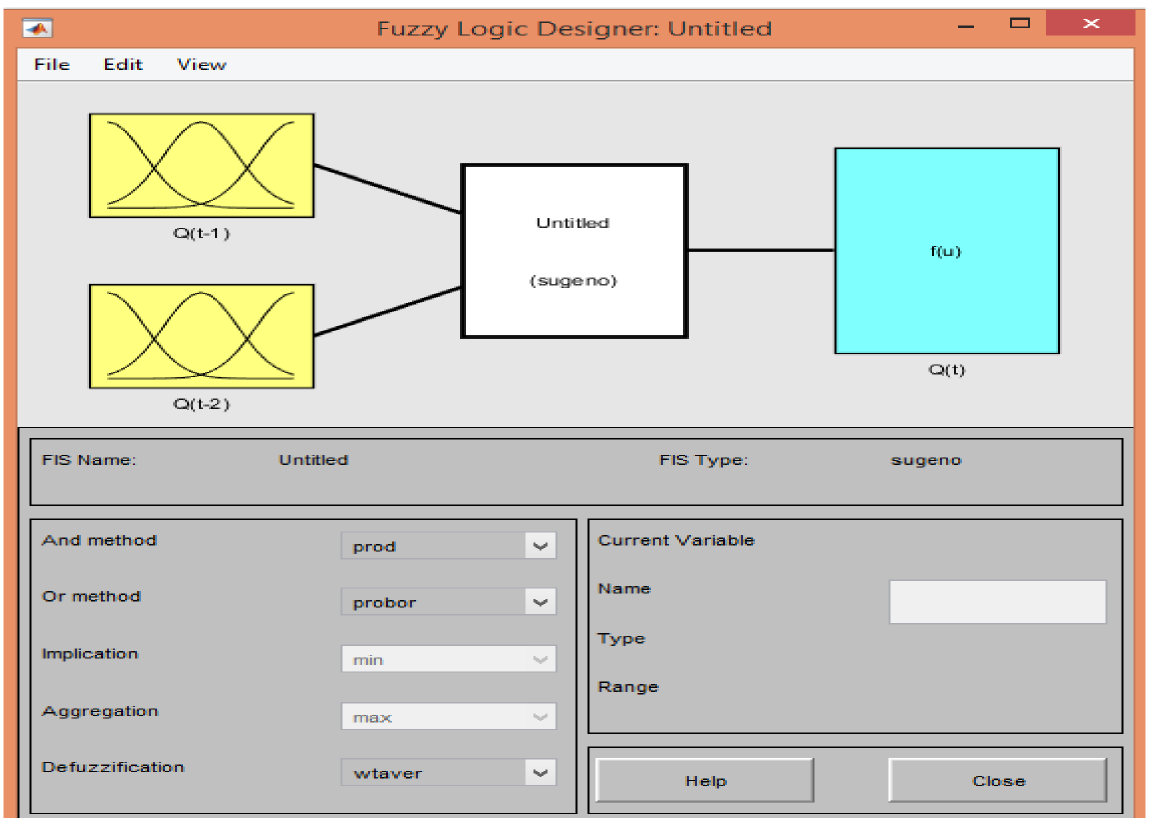


Figure 3.8: ANFIS deffuzification process for model-2

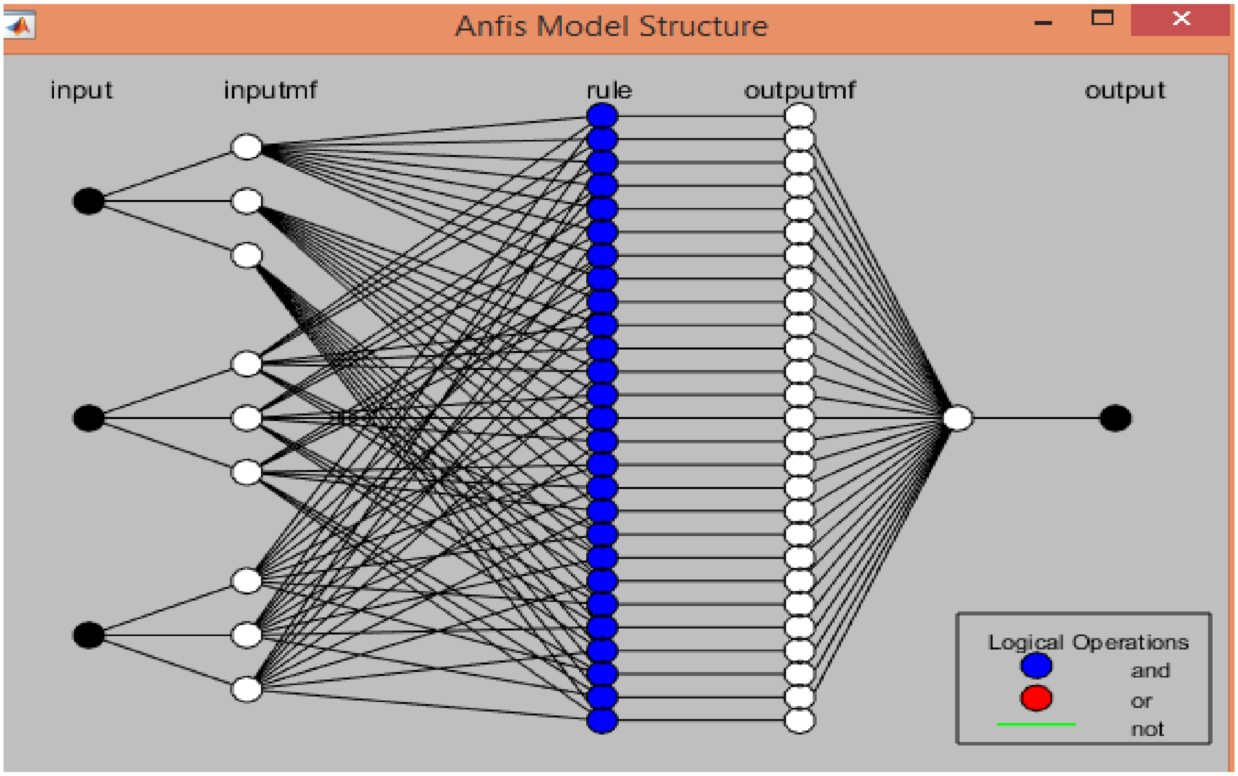


Figure 3.9: ANFIS model network for model-3 (3 inputs data)

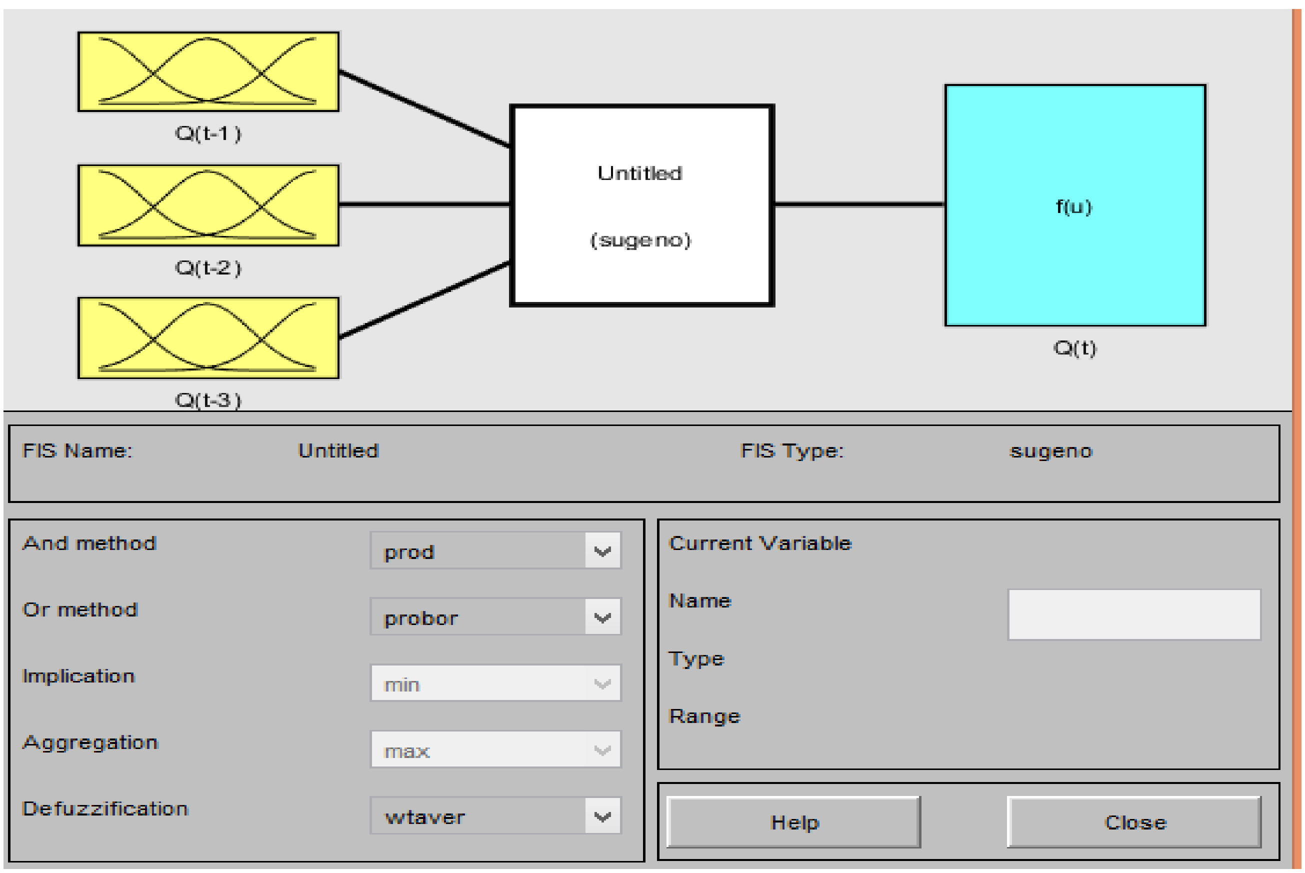


Figure 3.10: ANFIS deffuzification process for model-3

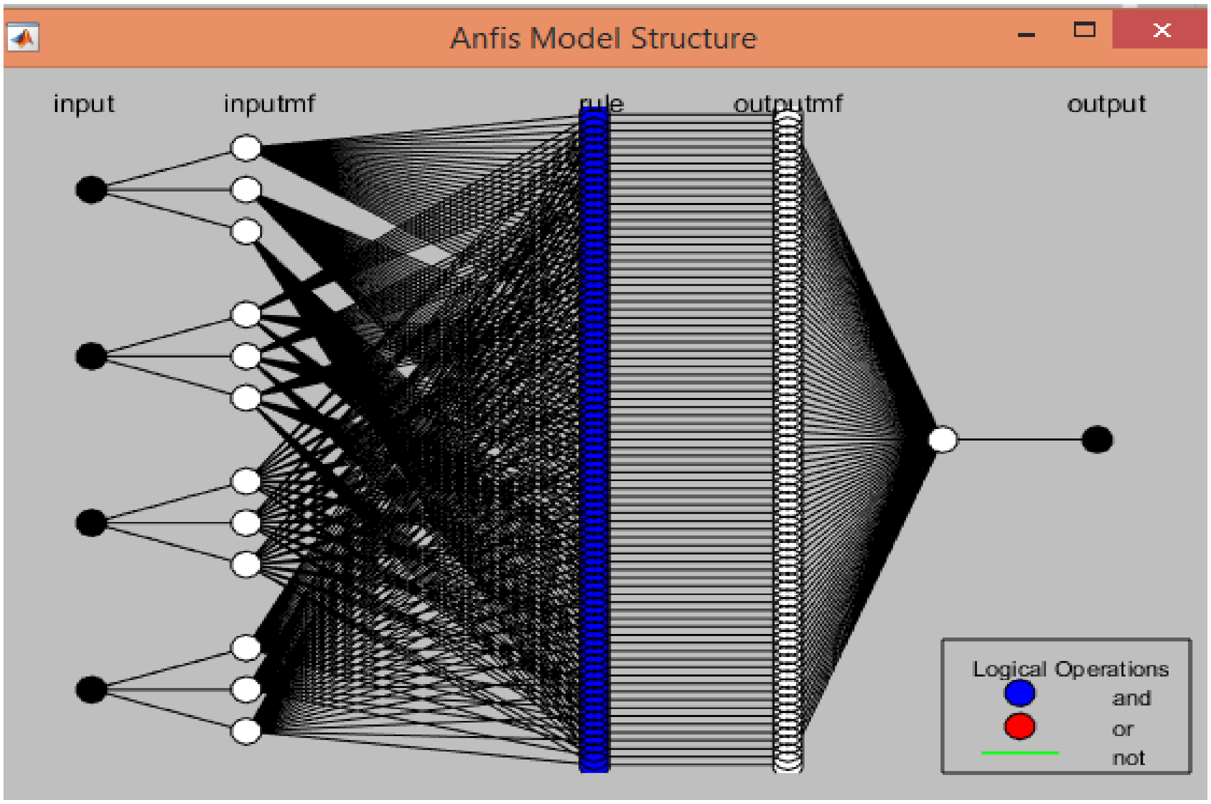


Figure 3.11: ANFIS model network for model-4 (4 inputs data)

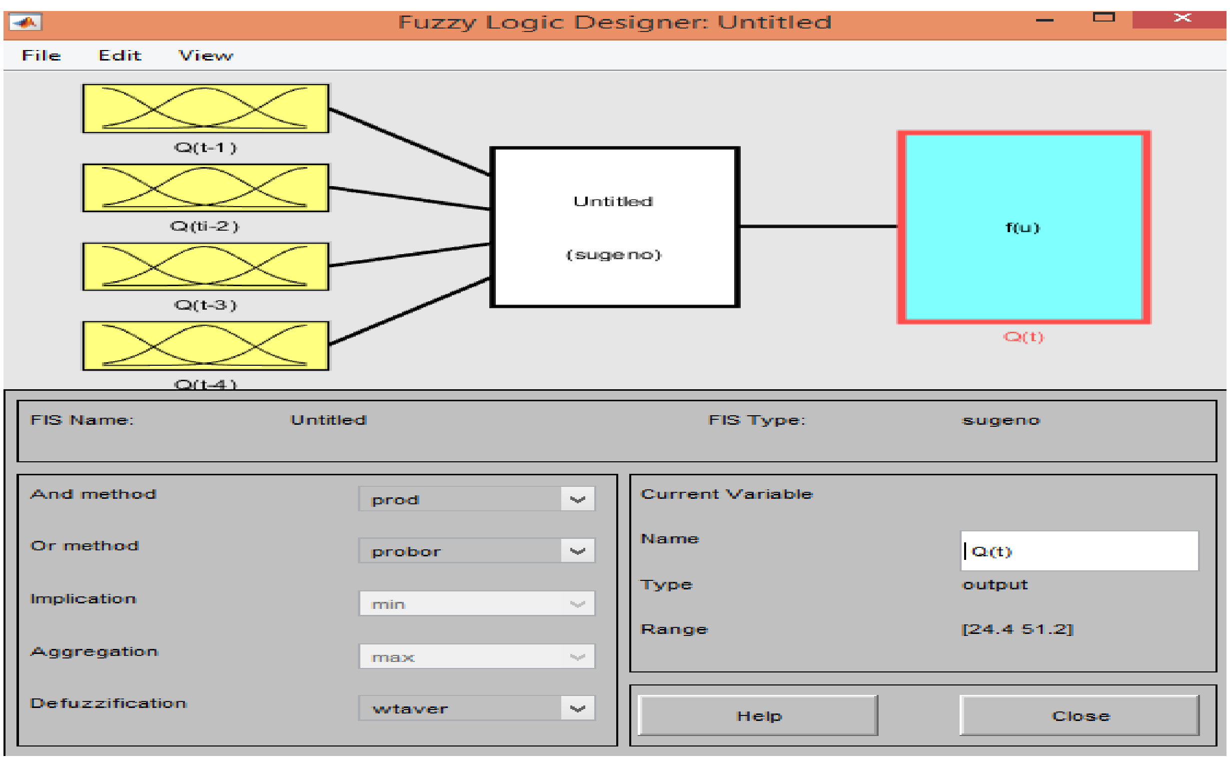


Figure 3.12: ANFIS deffuzification process for model-4

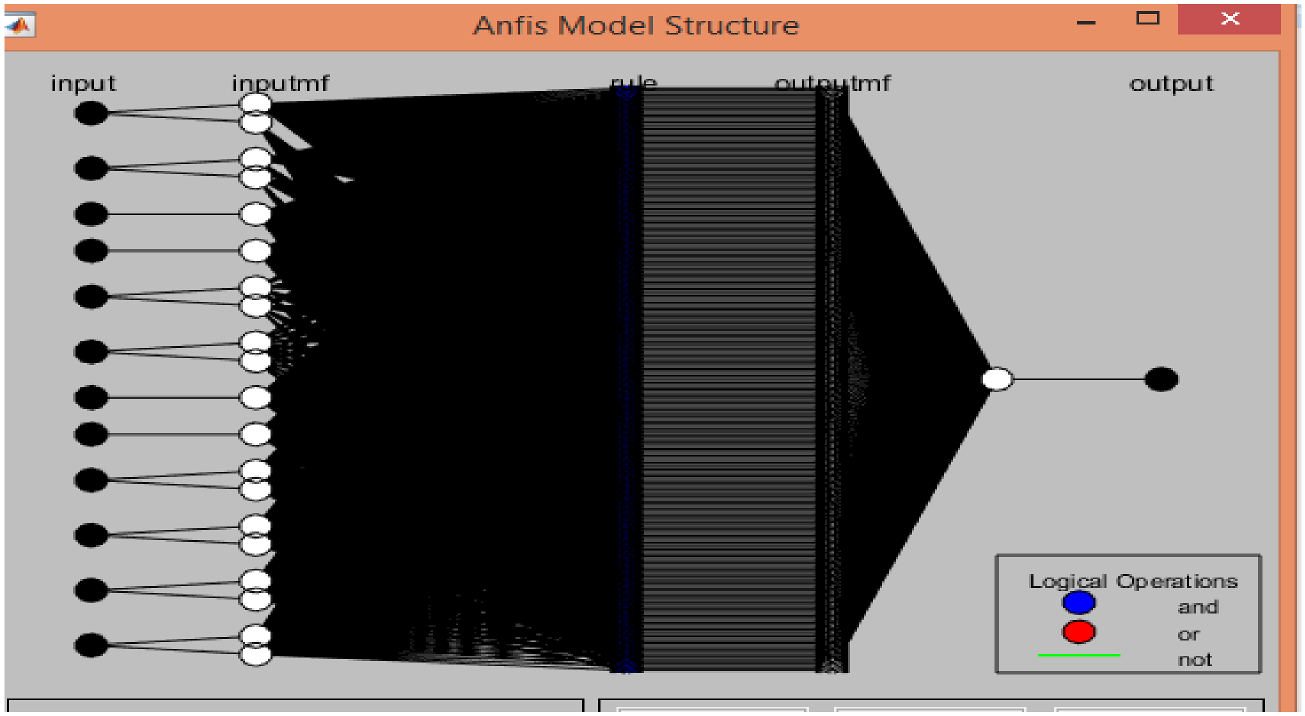


Figure 3.13: ANFIS model network for model-5 (12 inputs data)

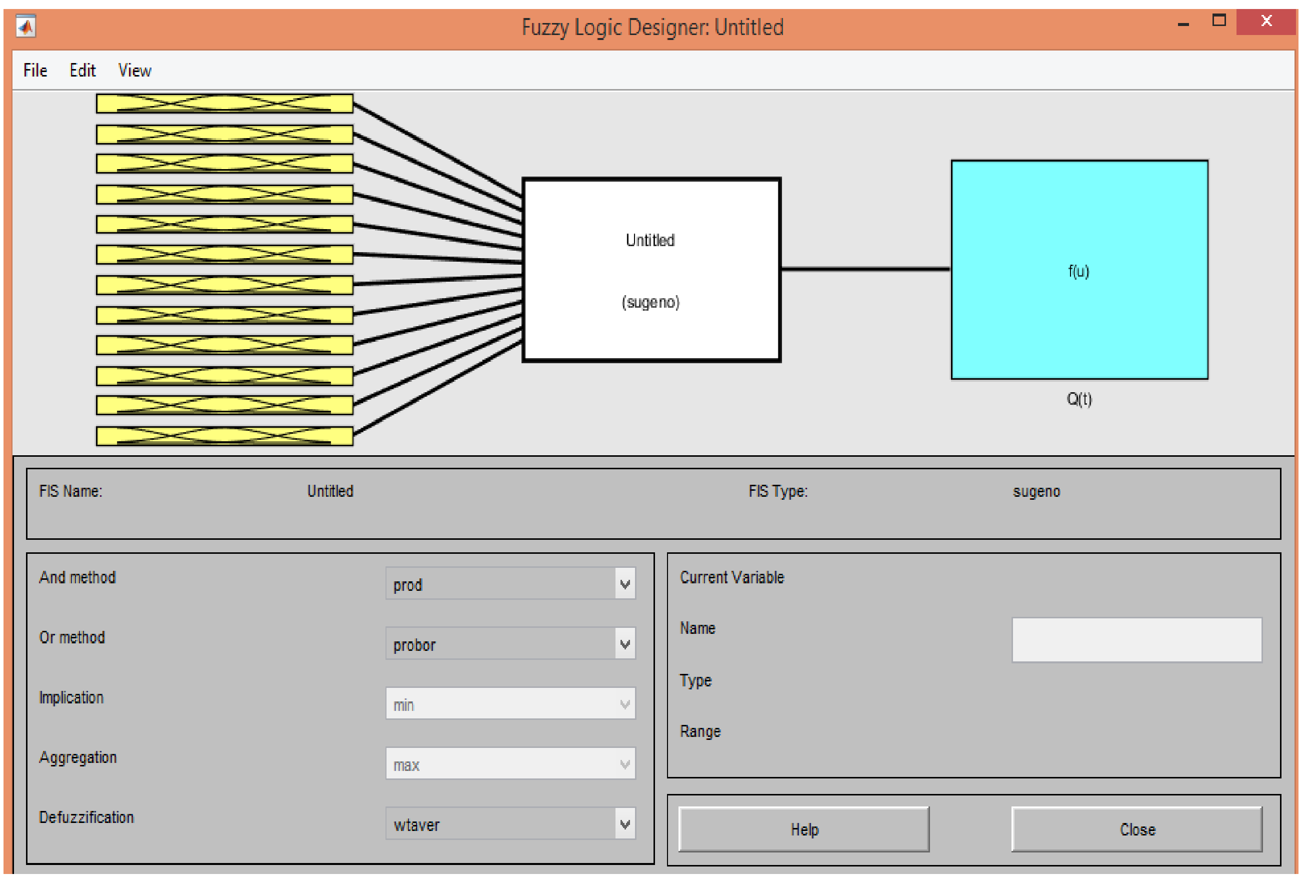


Figure 3.14: ANFIS deffuzification process for model-5

3.4.2. Model Application Using ANN

The following standard steps for designing networks models to solve problems for time series analysis and forecast using ANNs in MATLAB were utilized (Shamseldin, 2006):
Firstly, the selected input and target output data sets for each of the models: Model-1, Model-2, Model-3, Model-4 and Model-5 as used with ANFIS were fed into ANN graphical user interface (GUI) of MATLAB.

Secondly, the multilayer perceptron type was specified. All the ANN models developed used a three-layered perception with a number of functional nodes and connections including weights and biases. One hidden layer was used for all the ANN models for training. The number of hidden neurons (HN) in the hidden layer of the ANN architecture was varied till the best performance was obtained. 10 numbers of hidden neurons were found to be adequate. Sigmoid transfer function was used in the hidden layer while the linear transfer function was used in the output layer. The learning algorithm used was the backpropagation algorithm because it is very fast and easy to understand. Various percentages for the training data set, testing data set and validation data set as used with ANFIS models were also used. The (70%) training data set were presented to the ANN network during training and the networks were adjusted according to their errors. The (15%) validation data set were used to measure the networks generalizations and to halt the training when generations stopped improving. The remaining (15%) testing data set have no effects on the training and so provide an independent measure of the network performance during and after training.

Thirdly, the various ANN models were trained, tested and validated using three training algorithms in order to make comparative analysis. The training algorithms are Levenberg-Marquardt (LM) algorithm with back propagation, Scaled Conjugate Gradient (SCG) algorithm with back propagation and Bayesian Regularization (BR) algorithm with back propagation. Finally, the models were also tested with separated independent data sets for further testing and validation. The data sets of (1999-2000) (730 data) were used to test the ANN models independently. This helped to ensure proper validation processes.

Figures 3.15 to 3.19 show the network models for all the five models using ANN.

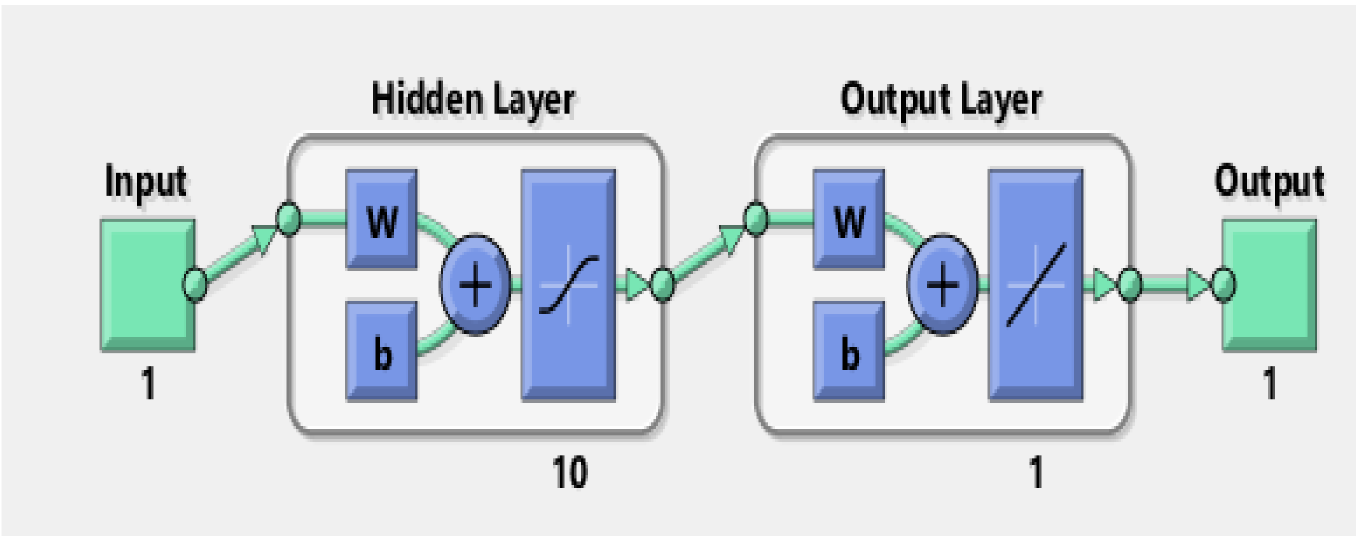


Figure 3.15: ANN network model for Model-1 (1 input data)

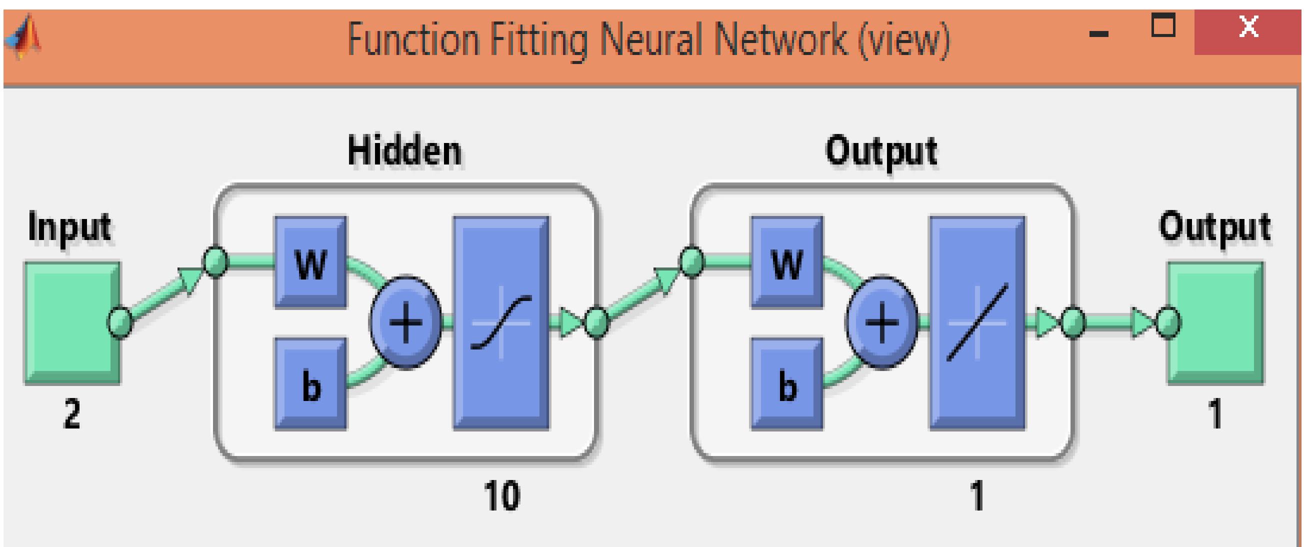


Figure 3.16: ANN network model for Model-2 (2 inputs data)

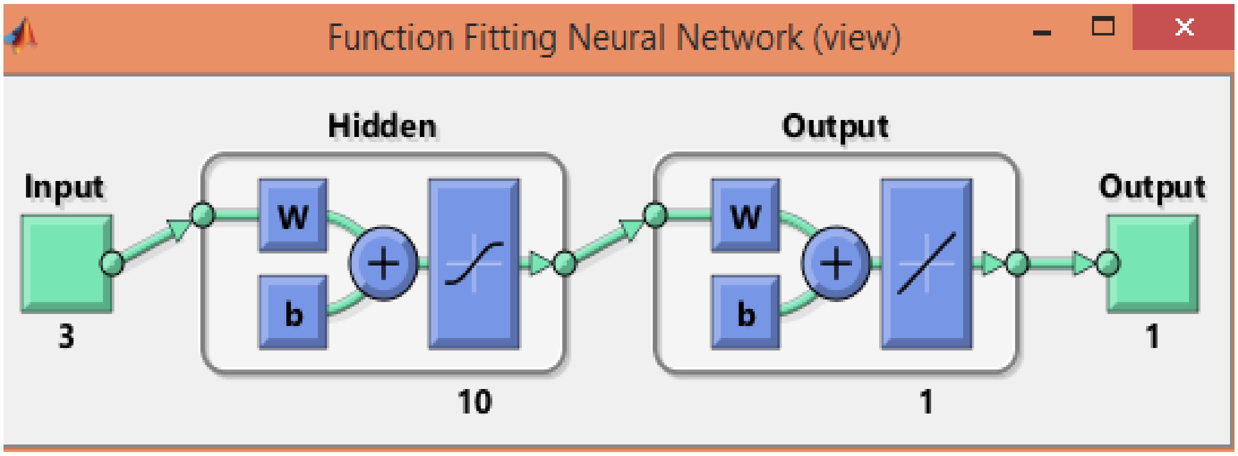


Figure 3.17: ANN network model for Model-3 (3 inputs data)

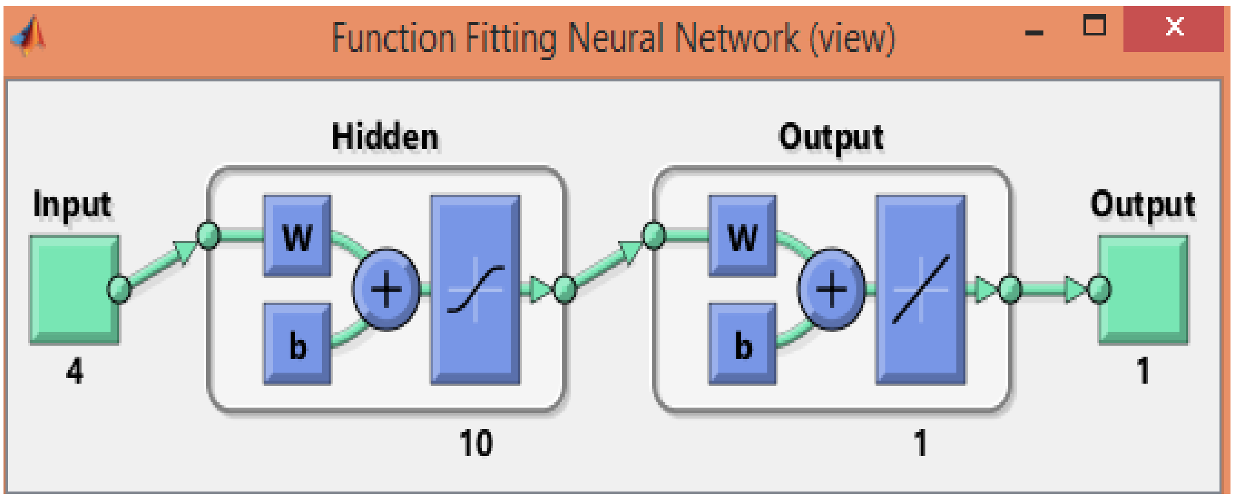


Figure 3.18: ANN network model for Model-4 (4 inputs data)

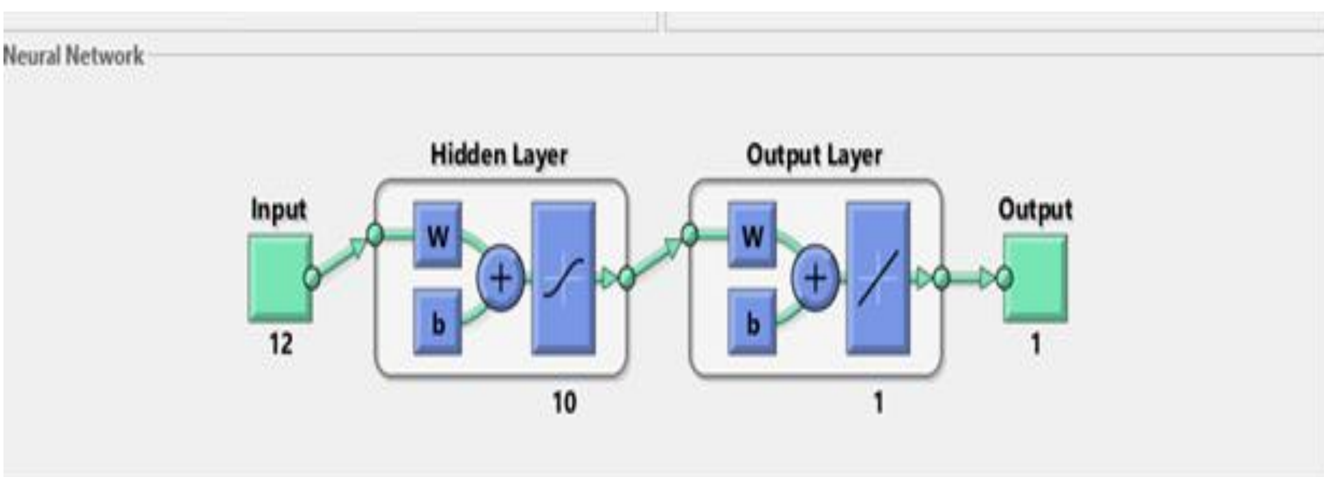


Figure 3.19: ANN network model for Model-5 (12 inputs data)

3.5. Performance Evaluation Criteria

The performances of the different models for both ANFIS and ANN were evaluated using goodness of fit criteria after each of the model structures were calibrated using the training, testing and validation data set. The performances of the network models and the algorithms used in this study for both ANFIS and ANN were compared using the following five criteria:

1. Coefficient correlation (*R*)
2. Coefficient of determination (R^2)
3. Mean square error (MSE)
4. Nash-Sutcliffe or modeling efficiency (E)
5. Index of Agreement (IOA)

The coefficient of correlations (R) for all the model were reported by each of the method (ANFIS and ANN) used in MATLAB. From the value of R generated, the coefficient of determinations (R^2) were calculated as the square of R. The values of the mean square errors (MSE), the Modelling efficiencies (E) and the index of agreements (IOA) were computed using Equation 2.29, 2.30 and 2.31 as reported in the literature review.

Comparisons were made among the various criteria to evaluate the best performance model in the following order:

- The larger the values of **R** the better the performance results.
- The larger the value of $\mathbf{R^2}$ the better the performance results.
- The smaller the values of **MSE** the better the performance results.
- The larger the values of **E** the better the performance results.
- The larger the values of **IOA** the better the performance results.

CHAPTER 4

4.0. RESULTS AND DISCUSSIONS

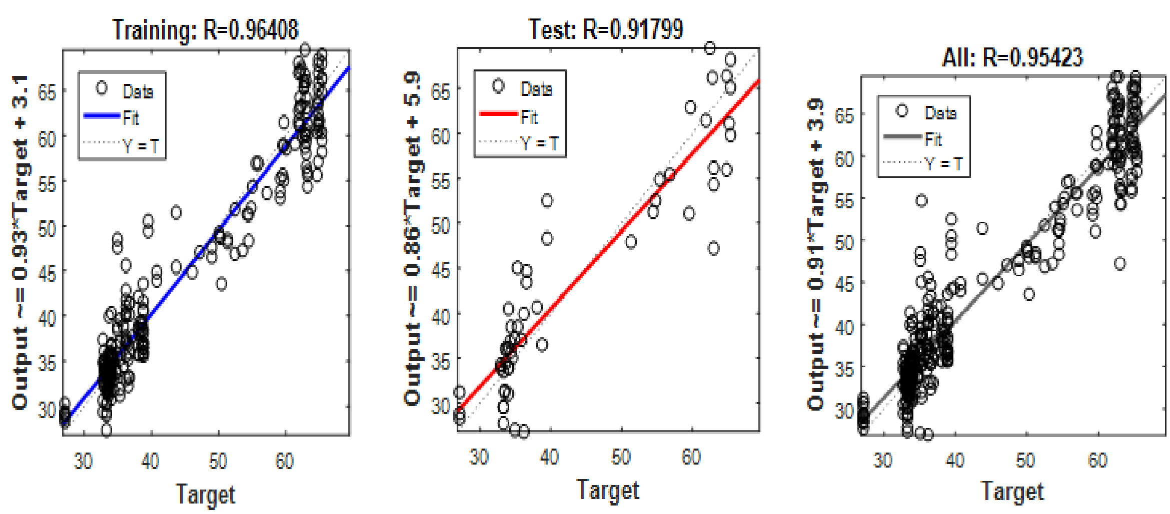
The results from the modelling and forecasting of the Ikopba river discharge using ANFIS and ANN are presented below.

4.1. Results and Discussion from ANFIS Models

The results obtained for model 1 to model 5 using ANFIS are presented and discussed below: The coefficient of correction (R) and the mean square error (MSE) were used to monitor the performance during training phase, testing phase and combined phase (training, testing and validation) using MATLAB training interface. Validation data sets were used to check and control the potential of the network models to over fit the data during training, hence they are not discussed separately like the training and testing data sets. Coefficient of determination (R^2), modelling efficiency (E) and index of agreement (IOA) were used to make detail comparison.

Model-1

The results from model 1 are shown in Figures 4.1 and 4.2.



(a) Training (b) Testing (c) All

Figure 4.1: Performance in model-1 using R as an evaluation criteria

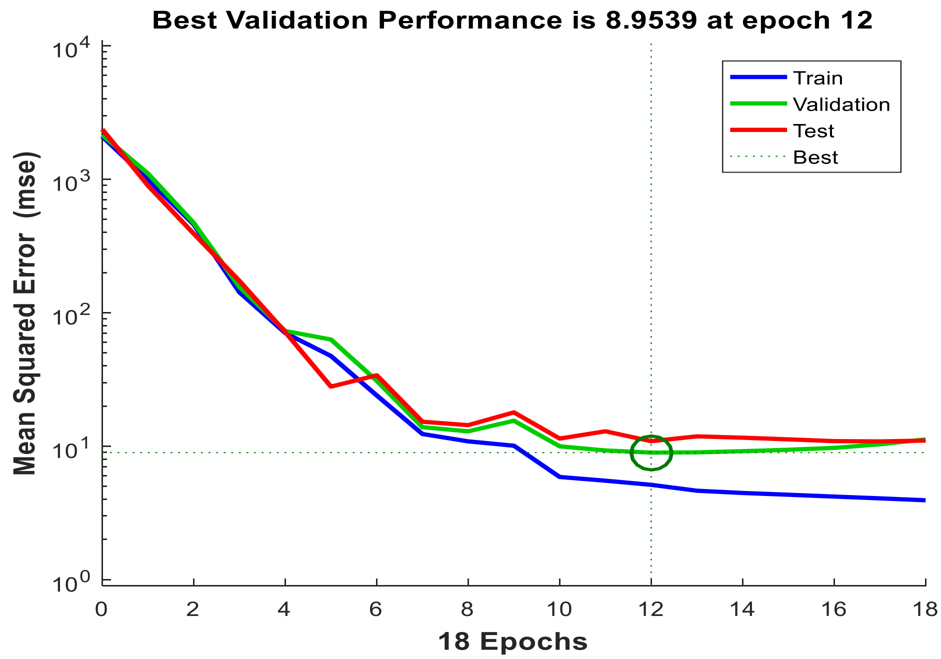
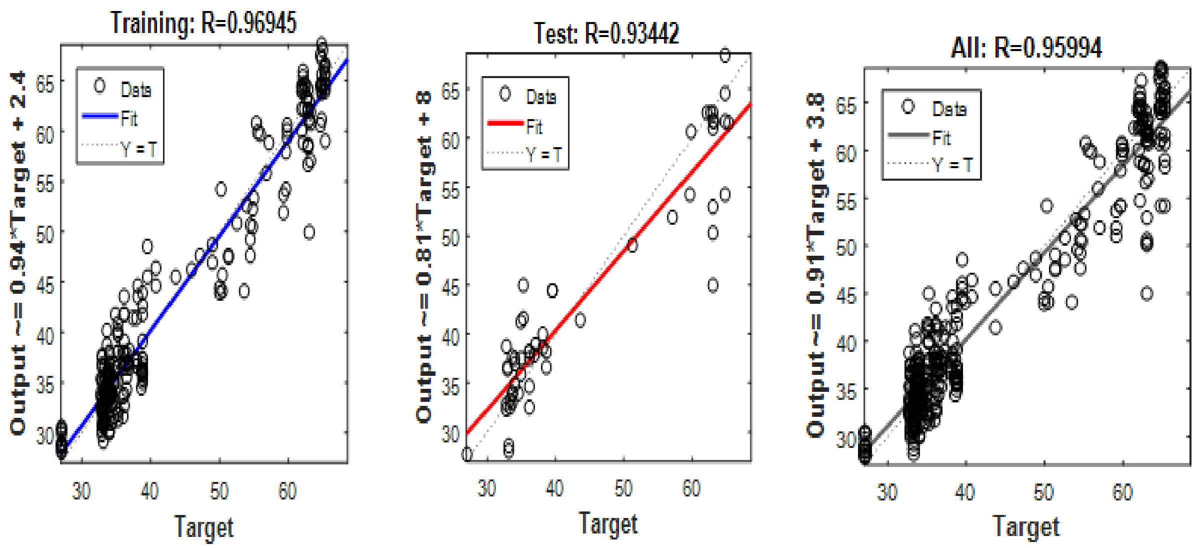


Figure 4.2: Performance in model-1 using MSE as an evaluation criteria

In Figure 4.1, the values of R in the training, testing and combined phases are 0.964, 0.918 and 0.954 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase, follow by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.2) 6.50 (m^3/s), 10.55 (m^3/s) and 8.950 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.929, E = 0.932 and IOA = 0.731), testing phase (R^2 = 0.843, E = 0.911 and IOA = 0.710) and combined phase (R^2 = 0.910, E = 0.921 and IOA = 0.730). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Based on the criteria used, the results showed that Model 1 was able to predict the river discharge with more than 70% degree of accuracy. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively.

Model-2

The results from model 2 are shown in Figures 4.3 and 4.4.



(a) Training (b) Testing (c) All

Figure 4.3: Performance in model-2 using R as an evaluation criteria

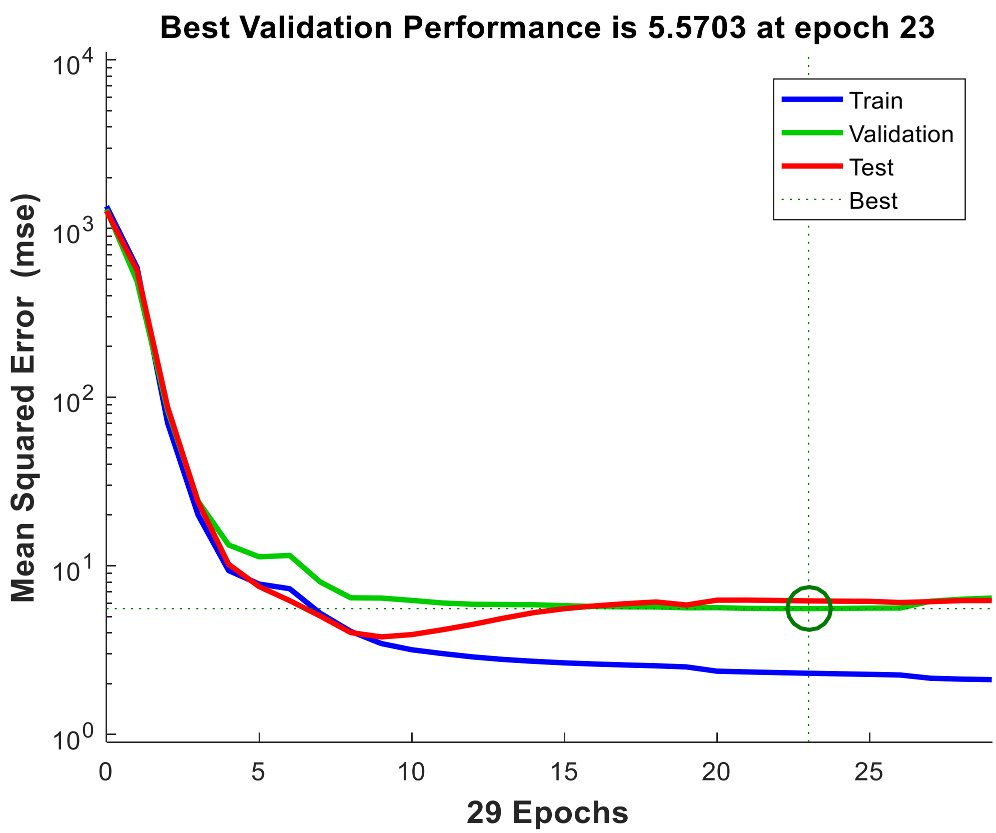
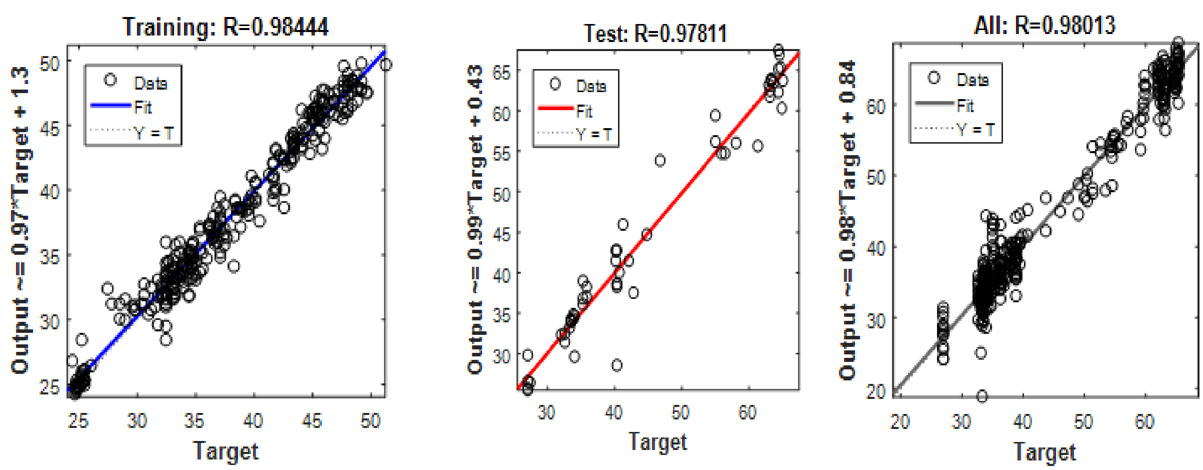


Figure 4.4: Performance in model-2 using MSE as an evaluation criteria

In Figure 4.3, the values of R in the training, testing and combined phases are 0.969, 0.934 and 0.960 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase, follow by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.4) 3.77 (m^3/s), 6.68 (m^3/s) and 5.57 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.939, E = 0.952 and IOA = 0.735), testing phase (R^2 = 0.872, E = 0.942 and IOA = 0.726) and combined phase (R^2 = 0.922, E = 0.948 and IOA = 0.732). These confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. But the results from model 2 improved better than that of model 1 when the five performance criteria were compared as indicated by high values of (R, R^2, E and IOA) and lower values of (MSE). This indicated that the more the dataset, the better the results.

Model-3

The results from model 3 are shown in Figures 4.5 and 4.6.



(a) Training (b) Testing (c) All

Figure 4.5: Performance in model-3 using R as an evaluation criteria

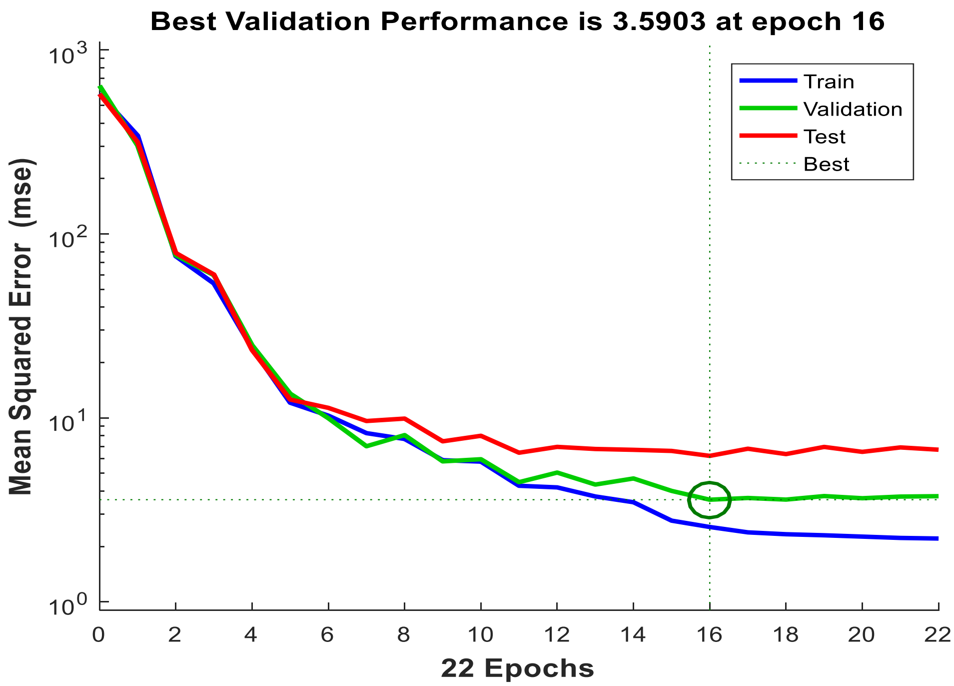
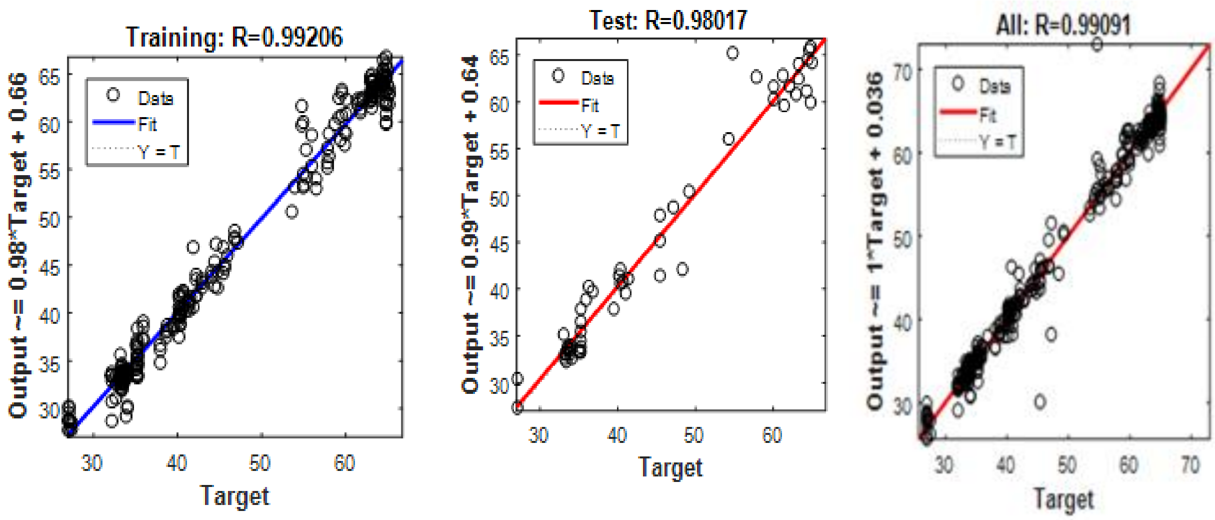


Figure 4.6: Performance in model-3 using MSE as an evaluation criteria

In Figure 4.5, the values of R in the training, testing and combined phases are 0.984, 0.978 and 0.980 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase, follow by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.6) 3.15 (m^3/s), 5.85 (m^3/s) and 3.59 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.968, E = 0.962 and IOA = 0.744), testing phase (R^2 = 0.956, E = 0.953 and IOA = 0.738) and combined phase (R^2 = 0.960, E = 0.961 and IOA = 0.743). These confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. But the results from model 3 improved better than that of model 1 and model 2 when the five performance criteria were compared as indicated by high values of (R, R^2, E and IOA) and lower values of (MSE). This indicated that the more the dataset, the better the results.

Model-4

The results from model 4 are shown in Figures 4.7 and 4.8.



(a) Training (b) Testing (c) All

Figure 4.7: Performance in model-4 using R as an evaluation criteria

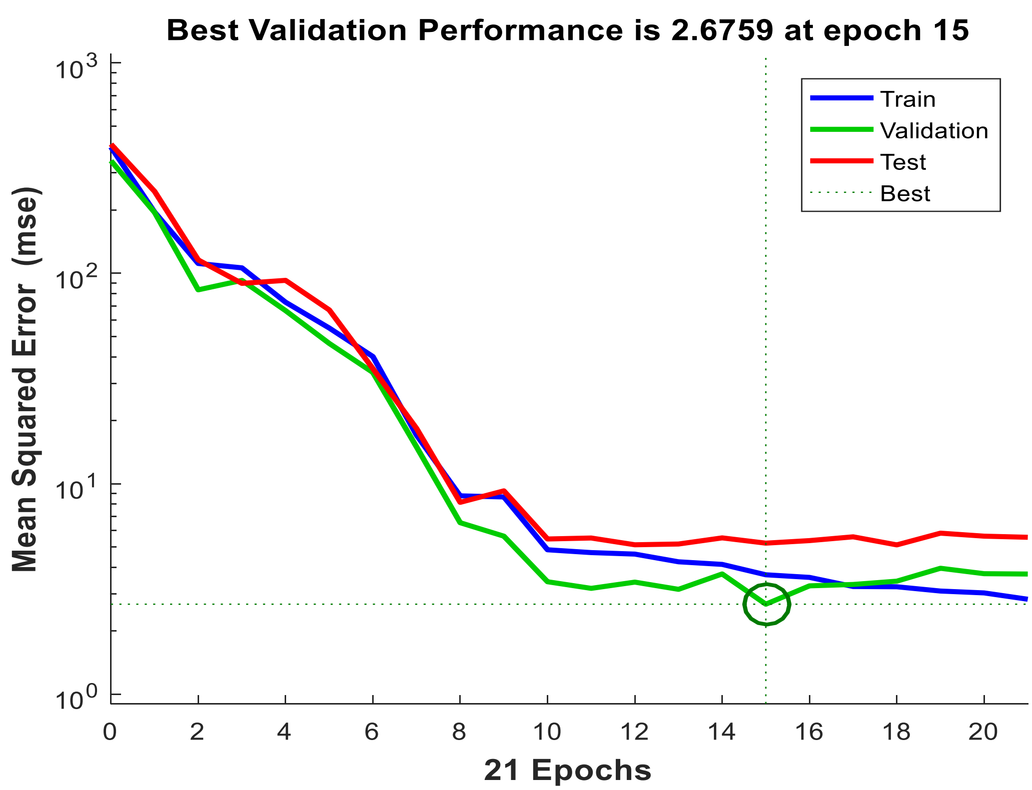
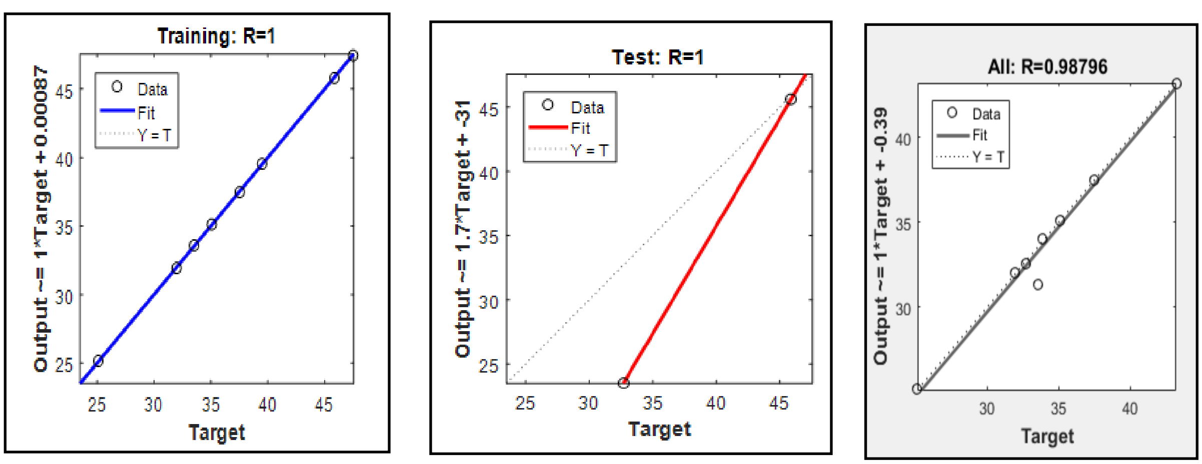


Figure 4.8: Performance in model-4 using MSE as an evaluation criteria

In Figure 4.7, the values of R in the training, testing and combined phases are 0.992, 0.980 and 0.991 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase, follow by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.8) 2.67 (m^3/s), 4.58 (m^3/s) and 2.68 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.984, E = 0.981 and IOA = 0.751), testing phase (R^2 = 0.960, E = 0.978 and IOA = 0.741) and combined phase (R^2 = 0.982, E = 0.980 and IOA = 0.750). These confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. But the results from model 4 improve better than that of model 1, model 2 and model 3 when the five performance criteria were compared as indicated by high values of (R, R^2, E and IOA) and lower values of (MSE). This indicated that the more the dataset, the better the results.

Model-5

The results from model 5 are shown in Figures 4.9 and 4.10.



(a) Training (b) Testing (c) All

Figure 4.9: Performance in model-5 using R as an evaluation criteria

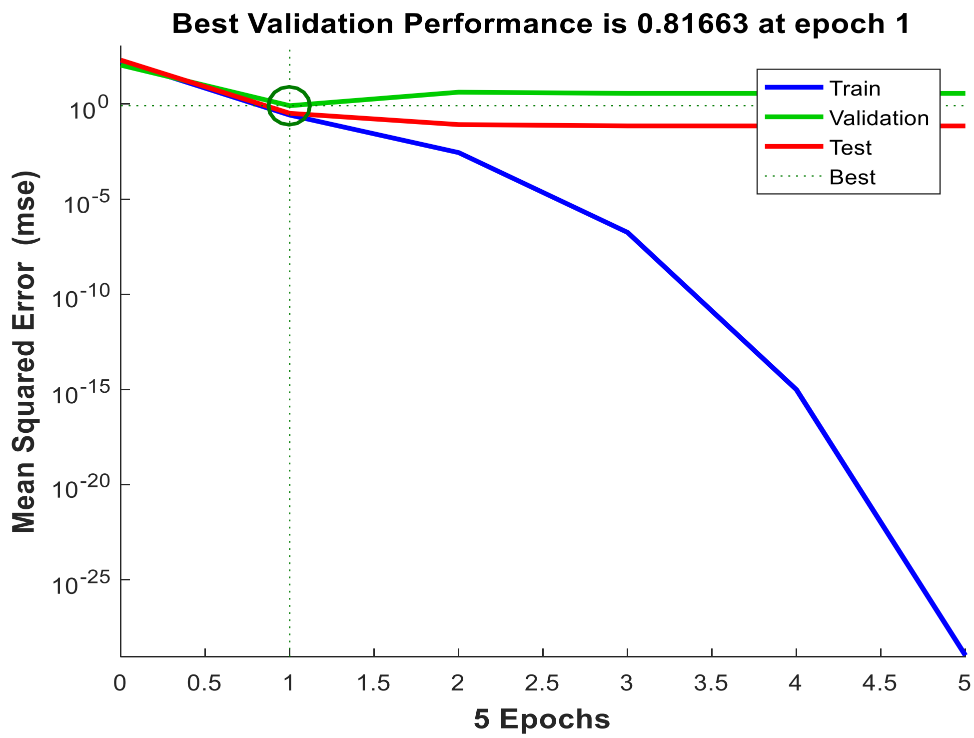


Figure 4.10: Performance in model-5 using MSE as an evaluation criteria

In Figure 4.9, the values of R in the training, testing and combined phases are 1.0, 1.0 and 0.988 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase, followed by testing phase and least in the combined phase. These values were further confirmed with the values of MSE (Figure 4.10) 0.57 (m^3/s), 0.57 (m^3/s) and 0.987 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 1.0, E = 0.991 and IOA = 0.872), testing phase (R^2 = 1.0, E = 0.992 and IOA = 0.873) and combined phase (R^2 = 0.975, E = 0.987 and IOA = 0.842). These confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. But the results from model 5 improved better than that of model 1, model 2, model 3, and model 4 when the five performance criteria were compared as indicated by high values of (R, R^2, E and IOA) and lower values of (MSE).

This indicated that the more the dataset, the better the results. The effects of climate change incorporated into the data set in model 5 helped improved the results very considerably as indicated by the value of R and $R^2 = 1$ both in the training and testing phases (Table 4.1).

Table 4.1: Comparison among the five models using performance evaluation criteria using ANFIS

	TRAINING					TESTING					ALL				
MODEL	R	R^2	MSE (M3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA
MODEL 1	0.964	0.929	6.50	0.932	0.731	0.918	0.843	10.55	0.911	0.710	0.954	0.910	8.950	0.921	0.730
MODEL 2	0.969	0.939	3.77	0.952	0.735	0.934	0.872	6.68	0.942	0.726	0.960	0.922	5.570	0.948	0.732
MODEL 3	0.984	0.968	3.15	0.962	0.744	0.978	0.956	5.85	0.953	0.738	0.980	0.960	3.590	0.961	0.743
MODEL 4	0.992	0.984	2.67	0.981	0.751	0.980	0.960	4.58	0.978	0.741	0.991	0.982	2.680	0.980	0.750
MODEL 5	1.00	1.00	0.57	0.991	0.872	1.00	1.00	0.57	0.992	0.873	0.988	0.976	0.82	0.987	0.842

Table 4.1 showed the performance criteria data used to make comparison. Also, Figures 4.11-4.13 showed the graphs of training, testing and combined phases respectively for the five models using performance evaluation criteria.

Looking at the table and the figures very well, it can be deduced that all the five ANFIS models performed well using training data sets, testing data sets and all the effect of the whole data sets. However, model-5 performed better than the other models when the effect of climate change was incorporated. Figure 4.14 showed the graph of the predicted values and the measured values in the training, testing and combined phases. The graph followed similar trend.

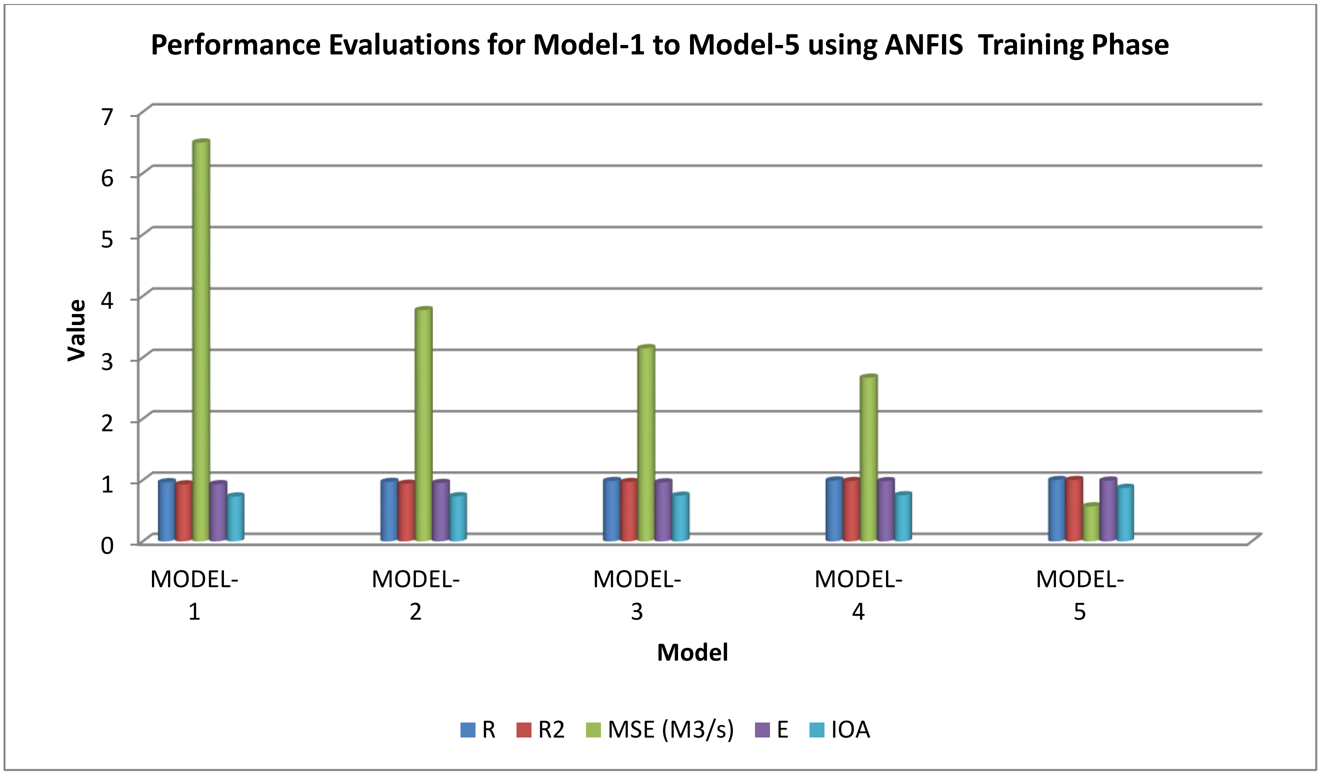


Figure 4.11: Performance evaluation in model-1 to model-5 using ANFIS (training phase)

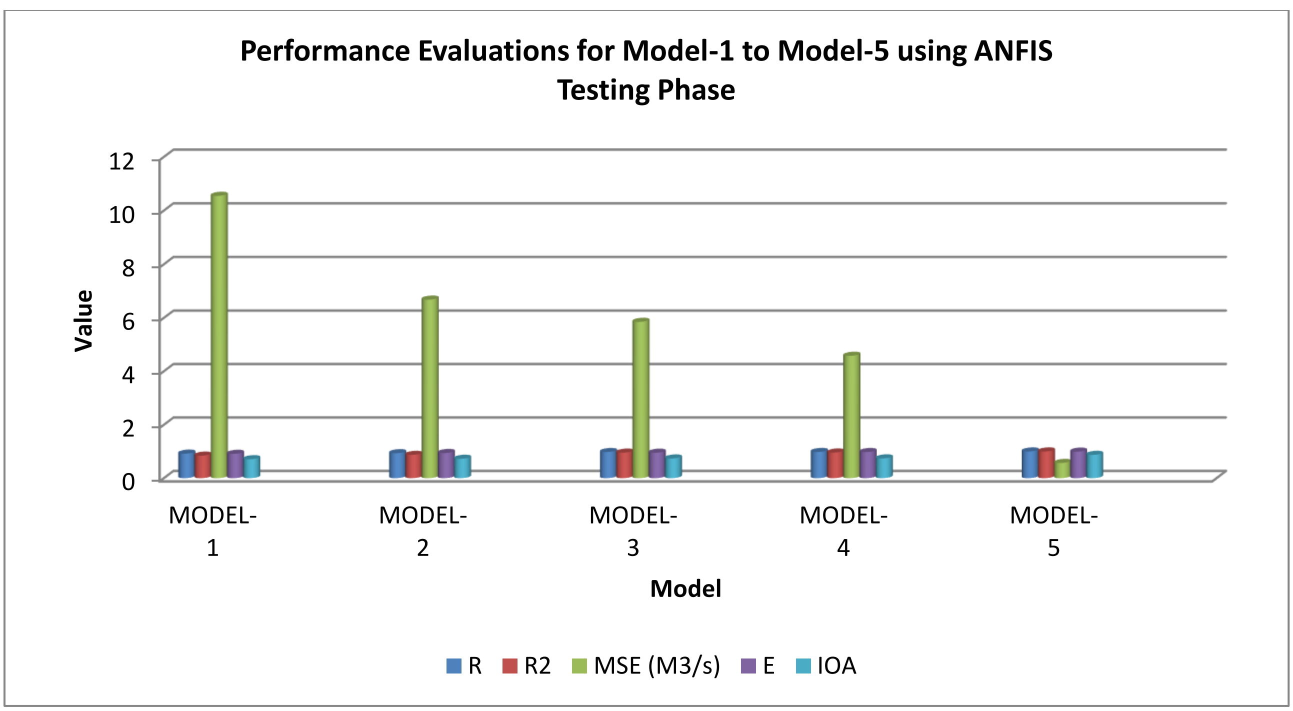


Figure 4.12: Performance evaluation in model-1 to model-5 using ANFIS (testing phase)

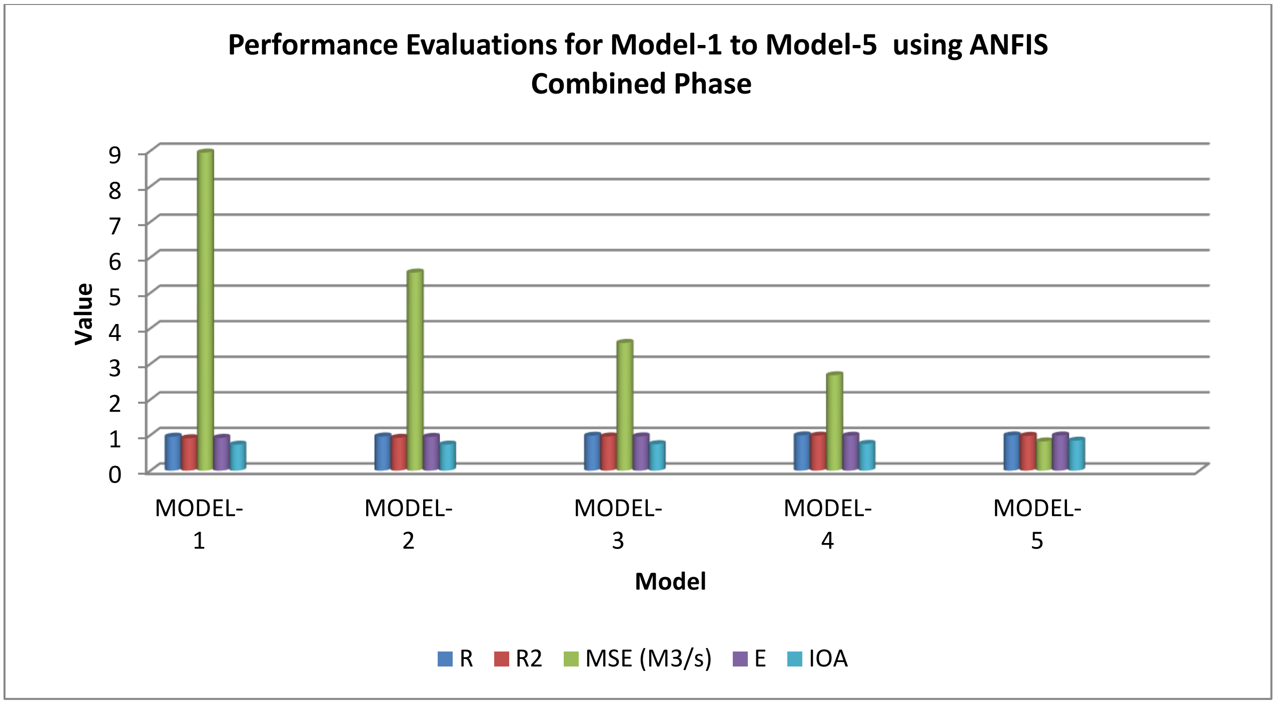


Figure 4.13: Performance evaluation in model-1 to model-5 using ANFIS (combined phase)

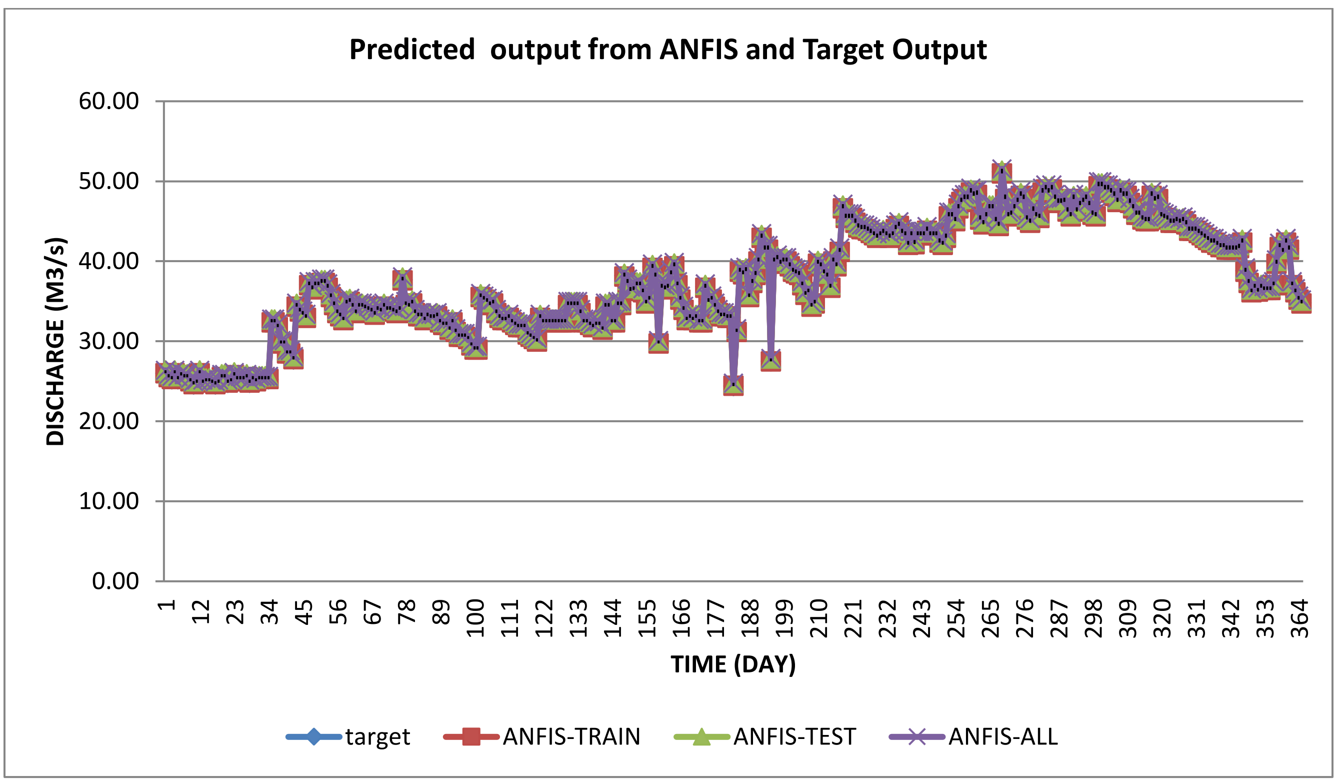


Figure 4.14: Comparing predicted discharge using training, testing and all data sets against the target data sets (ANFIS)

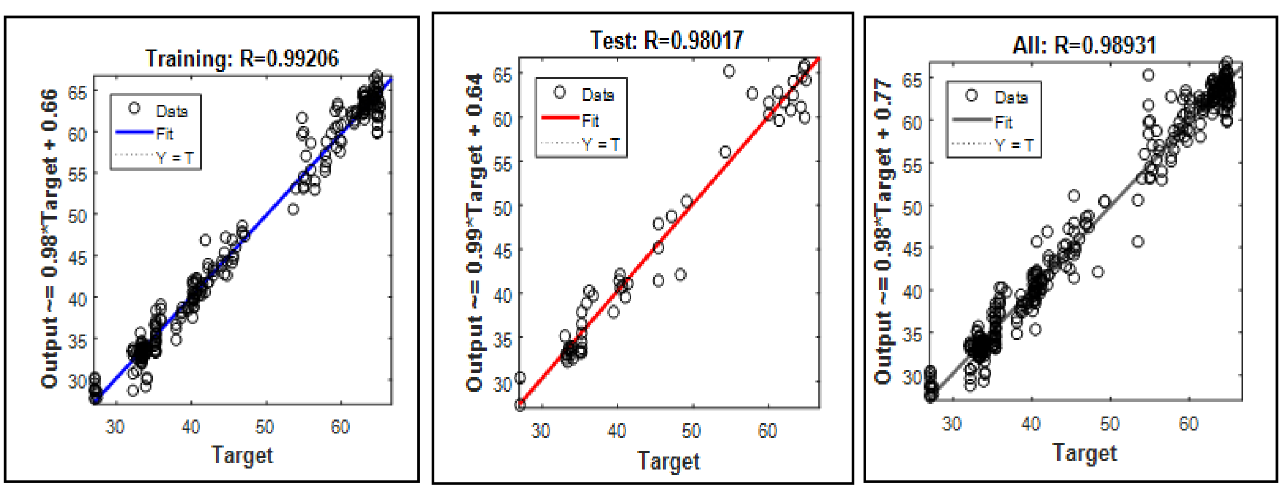
Further Testing and Validation of the best Model (Model-5) using 1999 and 2000 data sets

1999 data sets

To further validate the predictive power of the ANFIS models to predict future data sets, year 1999 and 2000 data sets were further tested with the only model 5 being the best model. The results from year 1999 data are presented and discussed in Figures 4.15 and 4.16. The tabulated results are presented in Table 4.2.

Table 4.2: Comparison of model-5 with 1999 data sets using performance evaluation criteria, ANFIS

	TRAINING					TESTING					ALL				
MODEL	R	R^2	MSE (M3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA
MODEL 5, 1999	0.992	0.884	2.58	0.965	0.866	0.980	0.960	5.73	0.932	0.869	0.989	0.978	5.18	0.958	0.831
MODEL 5	1.00	1.00	0.57	0.991	0.872	1.00	1.00	0.57	0.992	0.873	0.988	0.976	0.82	0.987	0.842



(a) Training (b) Testing (c) All

Figure 4.15: Performance in model-5 using R as an evaluation criteria (1999 data set)

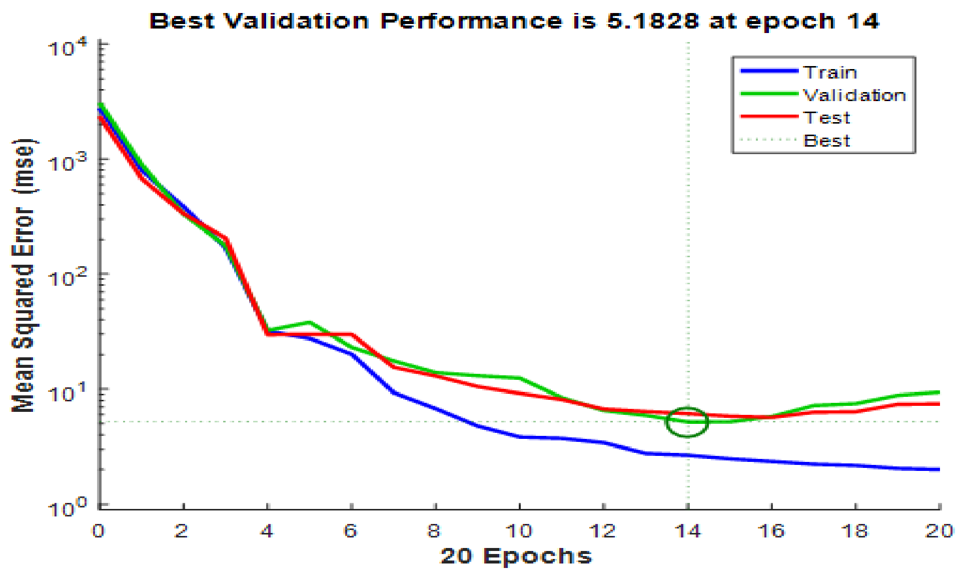


Figure 4.16: Performance in model-5 (1999 data sets) using MSE as an evaluation criteria

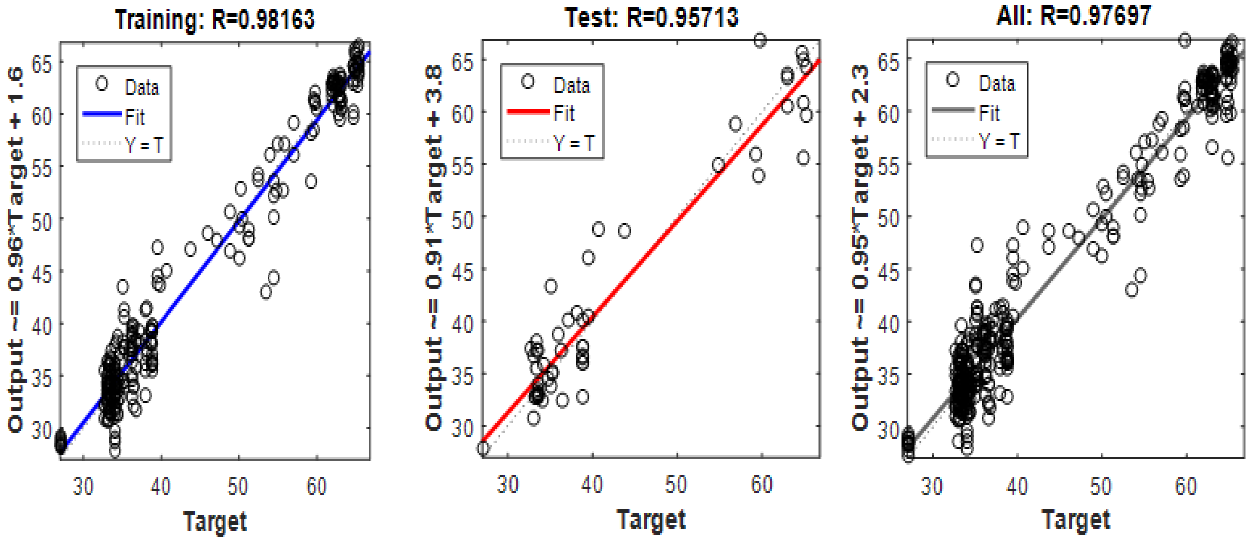
From Table 4.2, and Figures 4.15 and 4.16, the results from the five evaluation criteria showed that ANFIS was able to predict the river discharges of year 1999 reasonably though the predictive power reduced as can be seen in the lower value of R, R^2, E, IOA and higher value in the value of MSE.

2000 data sets

The results from year 2000 data are presented and discussed in Figures 4.17 and 4.18. The tabulated results are presented in Table 4.3.

Table 4.3: Comparison of model-5 with 2000 data sets using performance evaluation criteria, ANFIS

	TRAINING					TESTING					ALL				
MODEL	R	R²	MSE (M3/s)	E	IOA	R	R²	MSE (M³/s)	E	IOA	R	R²	MSE (M³/s)	E	IOA
MODEL 5, 2000	0.982	0.964	8.11	0.951	0.795	0.957	0.916	10.83	0.891	0.797	0.977	0.955	9.28	0.943	0.788
MODEL 5	1.00	1.00	0.57	0.99	0.872	1.00	1.00	0.57	0.992	0.873	0.988	0.976	0.82	0.987	0.842



(a) Training (b) Testing (c) All

Figure 4.17: Performance in model-5 (2000 data sets) using R as an evaluation criteria

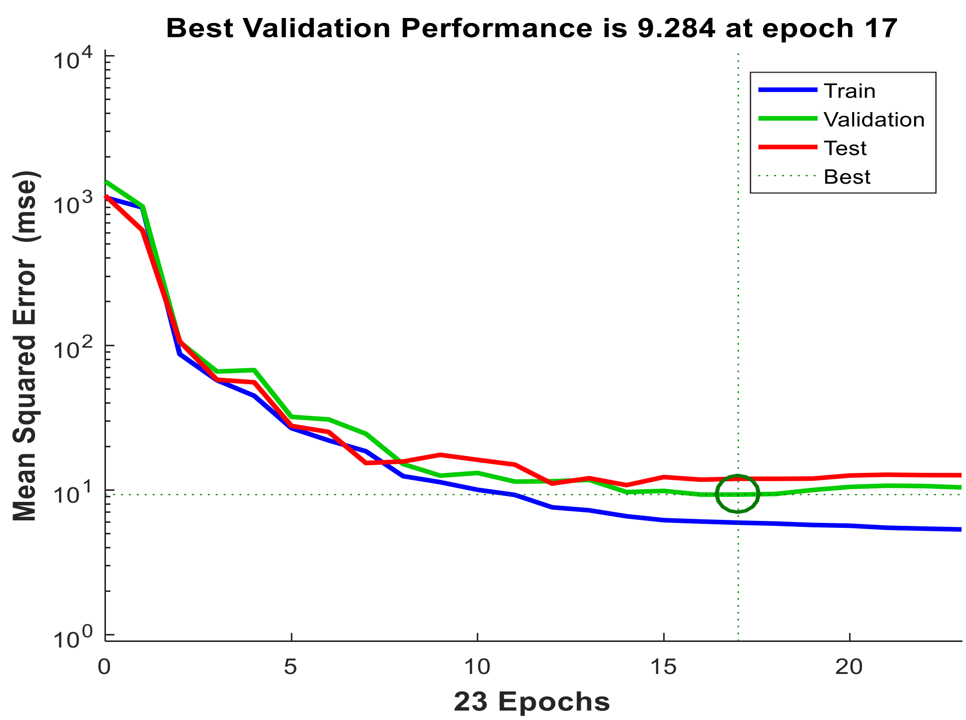


Figure 4.18: Performance in model-5 (2000 data sets) using MSE as an evaluation criteria

From Table 4.3 and Figures 4.17 and 4.18, the results from the five evaluation criteria showed that ANFIS was able to predict the river discharges of year 2000 though the predictive power reduced as can be seen in the lower value of R, R^2, E, IOA and higher value in the value of MSE. When the evaluation criteria for model 5 using 1999 and 2000 data sets were compared, year 1999 data sets performed better than that of year 2000. This indicate that the longer the predictive period, the less the accuracy of the predictive power of ANFIS as noticed with 1999 and 2000 data set. In such case, a little percentage increase (factor) can be factored in to increase the prediction accuracy. This cannot be validated as at the time of this study due to lack of data and also, the length of time to which the models can accurately predict the river discharge cannot be validated due to lack of long span data set. Generally, model 5 gave better results than the other four models. The training phase in model-5 showed an over-estimation of 0.043% of the observed target data sets (Figure 4.11) while an over-estimation of 0.044% was observed in the testing phase (Figure 4.12). This suggests that Ikpoba River discharge can be better modeled and forecasted if climate change is incorporated in the model.

The fact that ANFIS could not train the network models perfectly with MSE of zero could be due to the small size of the data sets and the effect of noise (white noise) in the sample data. In this case, the training data sets will not include all the representative features of the data sets the ANFIS wanted to model. It should be noted that, the sample data sets used in this research were small (5 years data) and the effect of noise expected could be due to the infilling of the missing discharge data in 1991 and 1992 by mean imputation method. White noise is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean and finite variables. They are independent from one another. If actually white noise is present in the data sets, it will be confirmed under ANN using Bayesian regularization algorithm because this algorithm has the capability of detecting white noise in sample data sets.

In general, this type of modelling works well if the training data presented to ANFIS for training (estimating) membership function parameters is fully representative of all the features of the data that the trained FIS is intended to model.

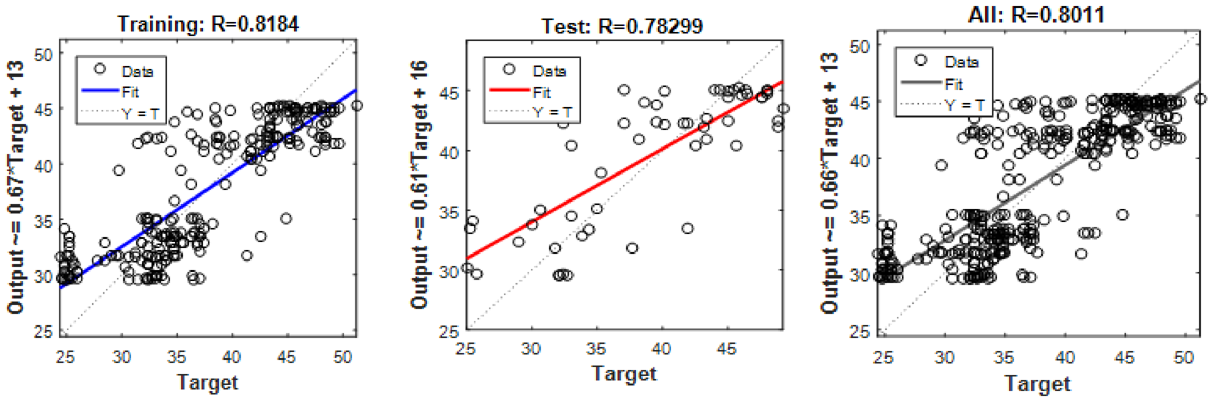
4.2. Results and Discussion from ANN Models

The five built models (model 1-5) were also applied to ANN to compare its predictive power with respect to that of ANN. In applying the ANN, three different training algorithms were used. The results from the ANN models are presented and discussed below.

4.2.1. Results and Discussion from Levenberg-Marquardt Algorithm (LM)

MODEL -1

The results of ANN_LM from model 1 are shown in Figures 4.19 and 4.20.



(a) Training (b) Testing (c) All

Figure 4.19: Performance in model-1 using R as an evaluation criteria (LM)

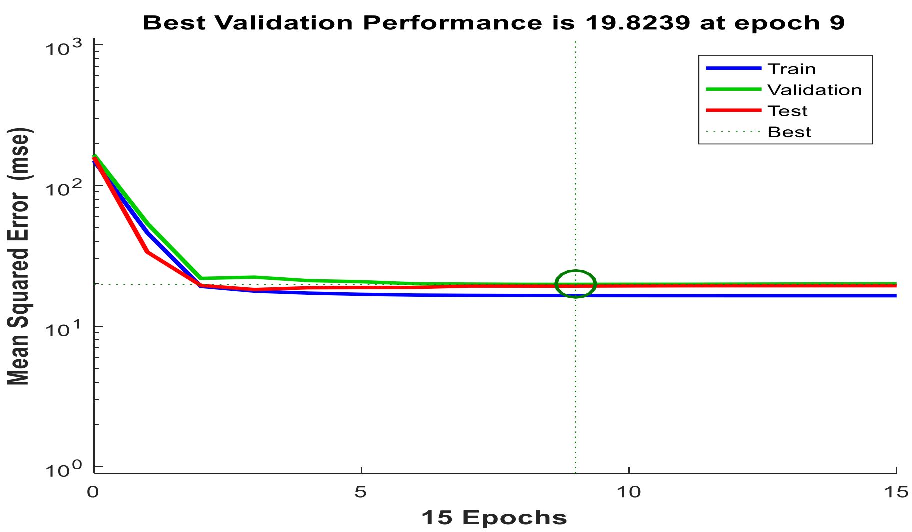
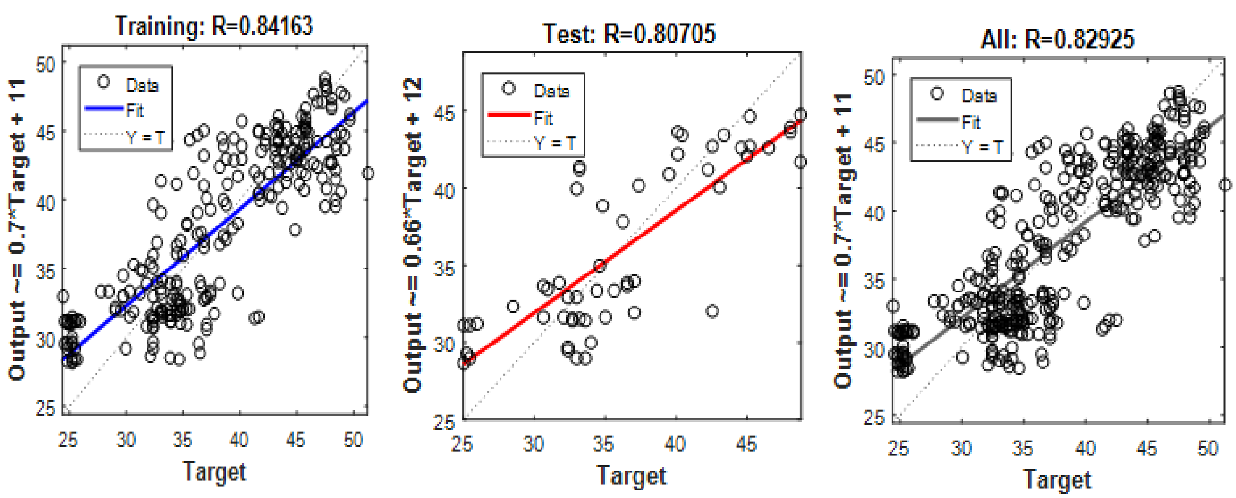


Figure 4.20: Performance evaluation using MSE as an evaluation criteria (LM)

In Figure 4.19, the values of R in the training, testing and combined phases are 0.76, 0.768 and 0.743 respectively. The higher the value of R the better the result. This showed that the result was better in the testing phase, follow by training phase and least in the combined phase. These values were further confirmed with the values of MSE (Figure 4.20) 20.05 (m^3/s), 19.91 (m^3/s) and 20.30 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.578, E = 0.84 and IOA = 0.61), testing phase (R^2 = 0.59, E = 0.85 and IOA = 0.63) and combined phase (R^2 = 0.552, E = 0.84 and IOA = 0.61). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively.

MODEL-2

The results of ANN_LM from model 2 are shown in figure 4.21 and 4.22 below.



(a) Training (b) Testing (c) All

Figure 4.21: Performance in model-2 using R as an evaluation criteria (LM)

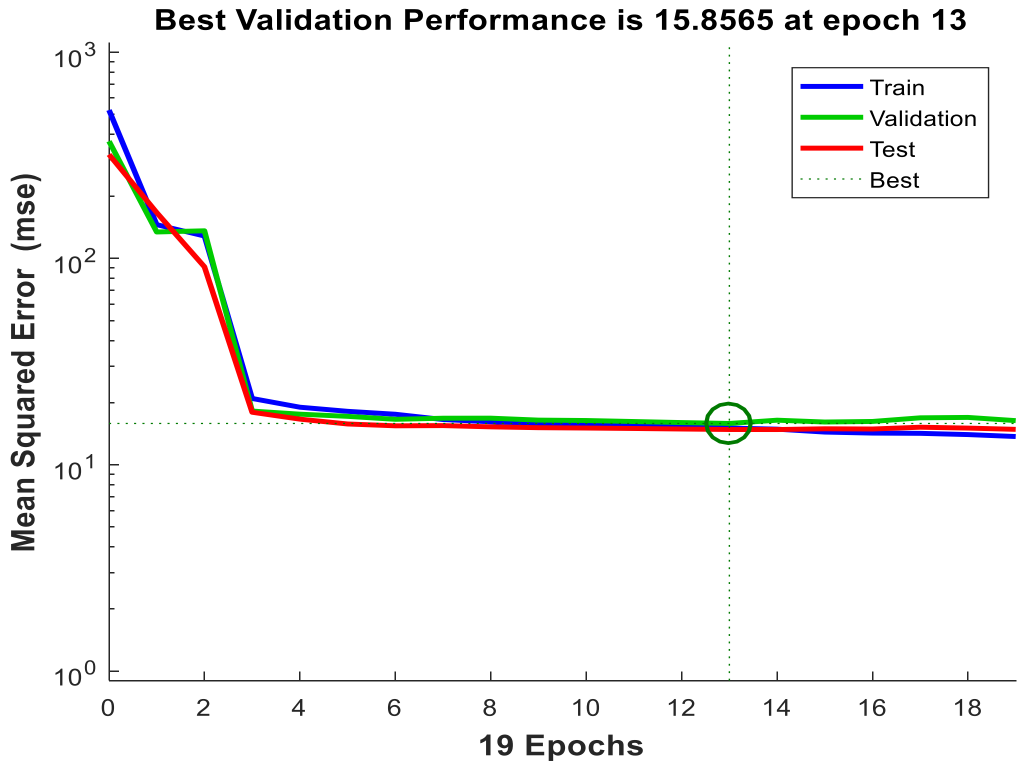
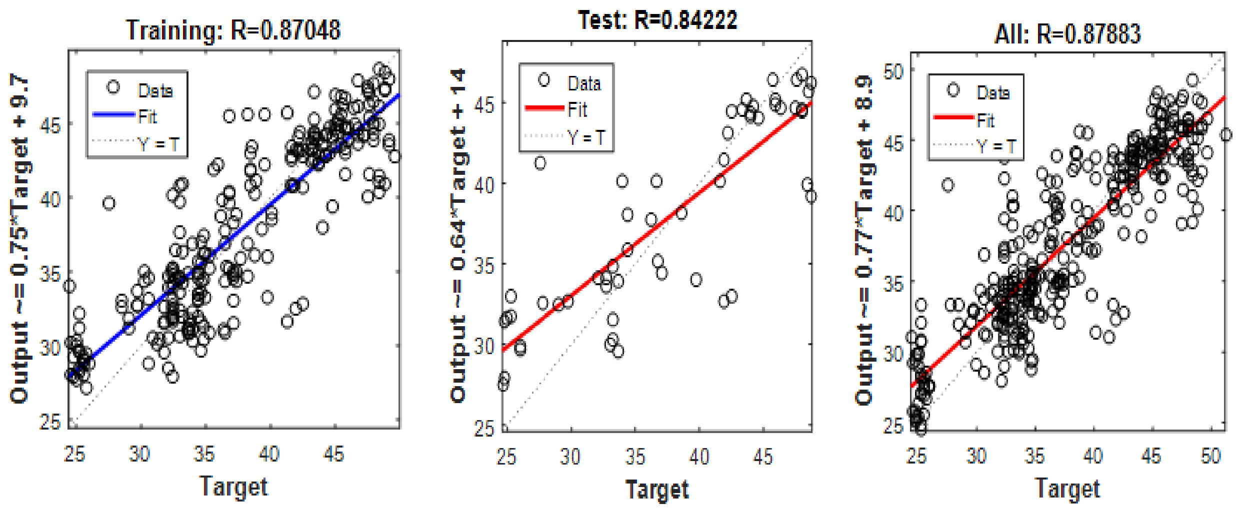


Figure 4.22: Performance evaluation using MSE (LM) as an evaluation criteria

In Figure 4.21, the values of R in the training, testing and combined phases are 0.783, 0.819 and 0.793 respectively. The higher the value of R the better the result. This showed that the result was better in the testing phase, follow by training phase and least in the combined phase. These values were further confirmed with the values of MSE (Figure 4.22) 18.52 (m^3/s), 18.98 (m^3/s) and 18.84 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.613, E = 0.860 and IOA = 0.63), testing phase (R^2 = 0.671, E = 0.87 and IOA = 0.64) and combined phase (R^2 = 0.629, E = 0.86 and IOA = 0.62). These confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. But the results from this model improved better than the one in model 1 indicating more data set gives better results.

MODEL-3

The results of ANN_LM from model 3 are shown in Figures 4.23 and 4.24.



(a) Training (b) Testing (c) All

Figure 4.23: Performance in model-3 using R as an evaluation (LM)

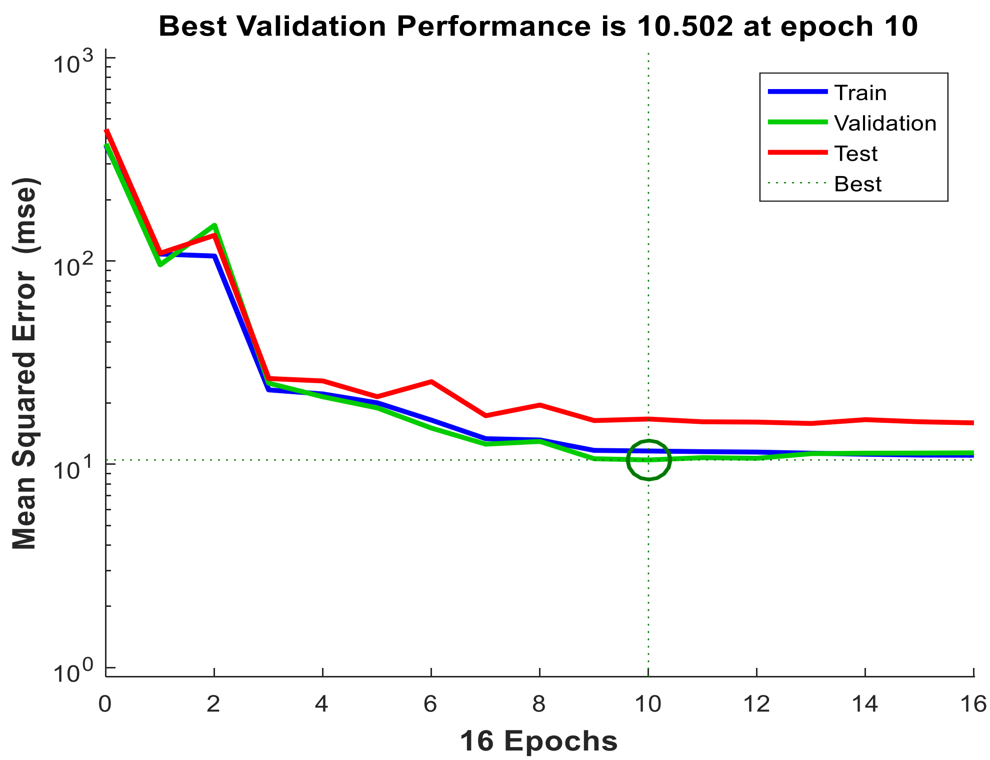
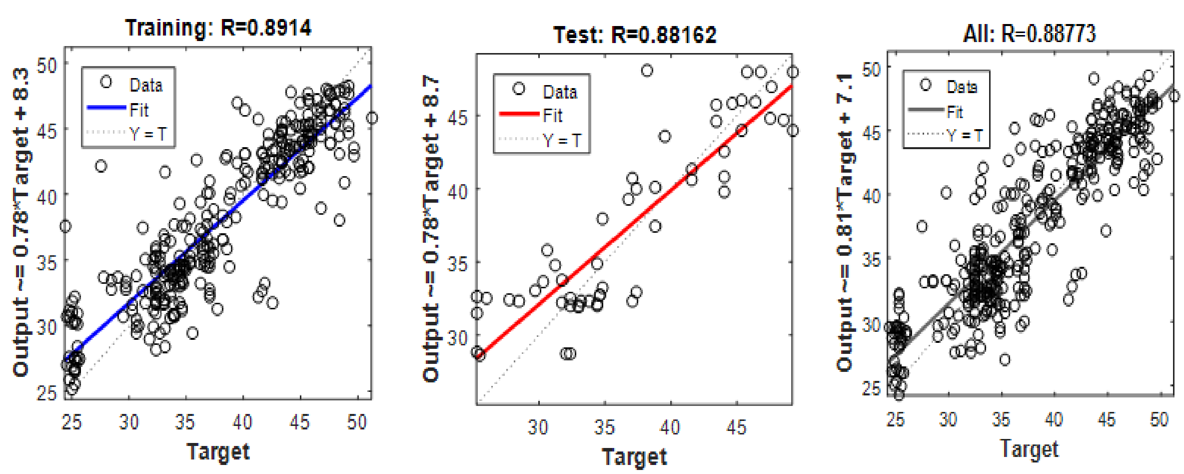


Figure 4.24: Performance evaluation using MSE as an evaluation (LM)

In figure 4.23, the values of R in the training, testing and combined phases are 0.856, 0.831 and 0.856 respectively. The higher the value of R the better the result. This showed that the result was better in the testing phase, followed by training phase and least in the combined phase. These values are further confirmed with the values of MSE (Figure 4.24) 13.76 (m^3/s), 13.87 (m^3/s) and 13.76 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.733, E = 0.91 and IOA = 0.652), testing phase (R^2 = 0.691, E = 0.89 and IOA = 0.651) and combined phase (R^2 = 0.733, E = 0.91 and IOA = 0.652). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. But the results from this model improved better than the one in model 1 and model 2 indicating more data set gives better results.

MODEL-4

The results of ANN_LM from model 4 are shown in Figures 4.25 and 4.26.



(a) Training (b) Testing (c) All

Figure 4.25: Performance in model-4 using R as an evaluation criteria (LM)

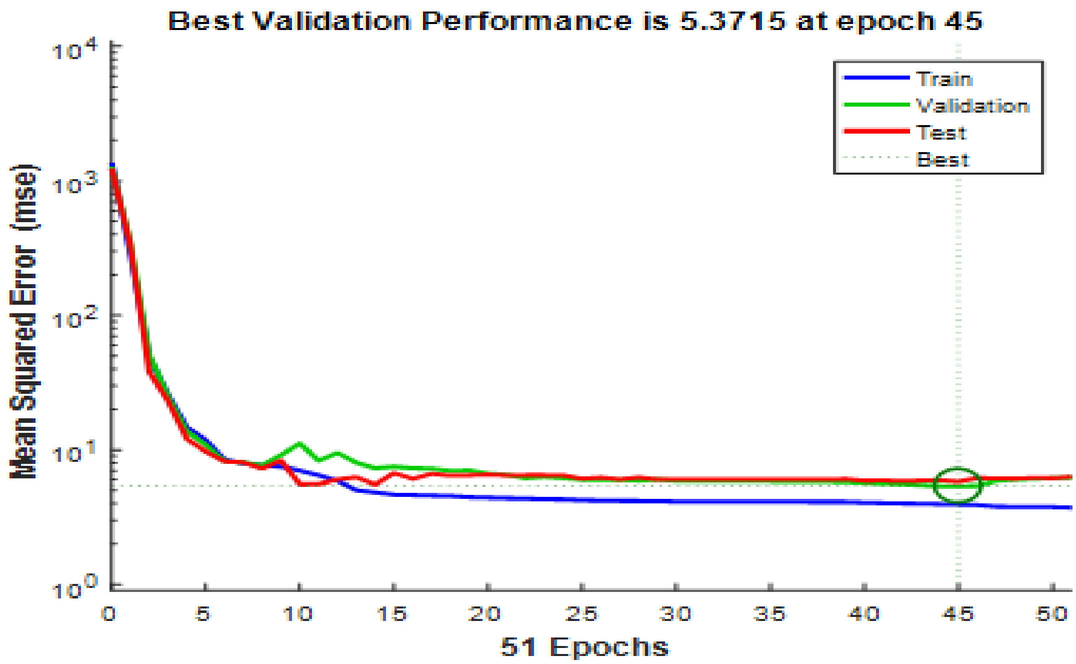
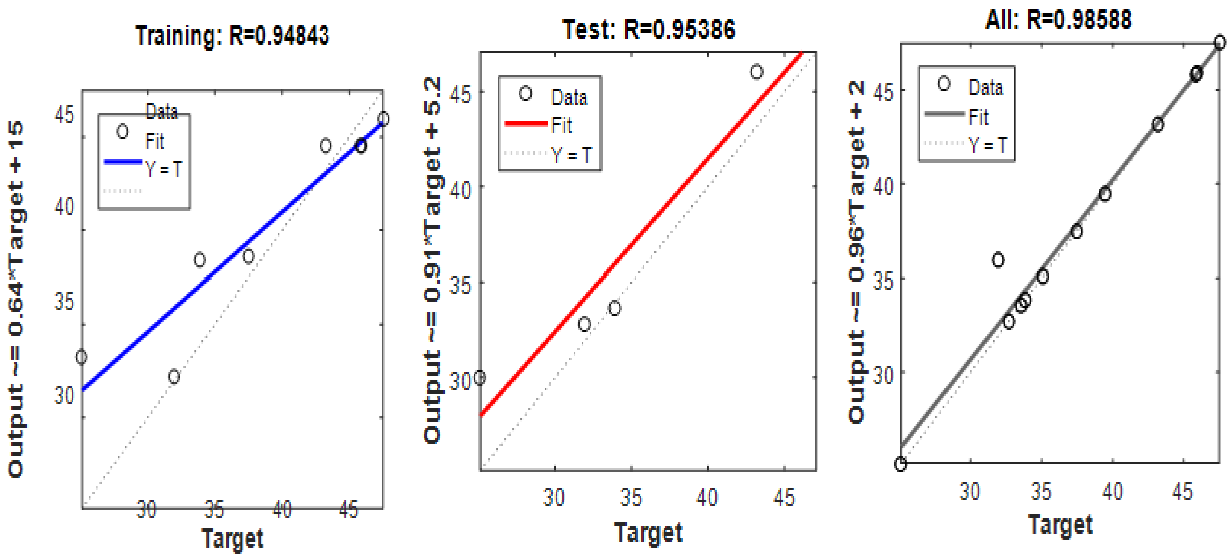


Figure 4.26: Performance evaluation using MSE as an evaluation (LM)

In Figure 4.25, the values of R in the training, testing and combined phases are 0.872, 0.85 and 0.865 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase, follow by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.26) 11.27 (m^3/s), 12.88 (m^3/s) and 11.83 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.760, E = 0.93 and IOA = 0.661), testing phase (R^2 = 0.731, E = 0.90 and IOA = 0.654) and combined phase (R^2 = 0.748, E = 0.92 and IOA = 0.655). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. But the results from this model improve better than the one in model 1, model 2 and model 3 indicating more data set gives better results.

Model-5

The results of ANN_LM from model 5 are shown in Figures 4.27 and 4.28.



(a) Training (b) Testing (c) All

Figure 4.27: Performance in model-5 using R as an evaluation criteria (LM)

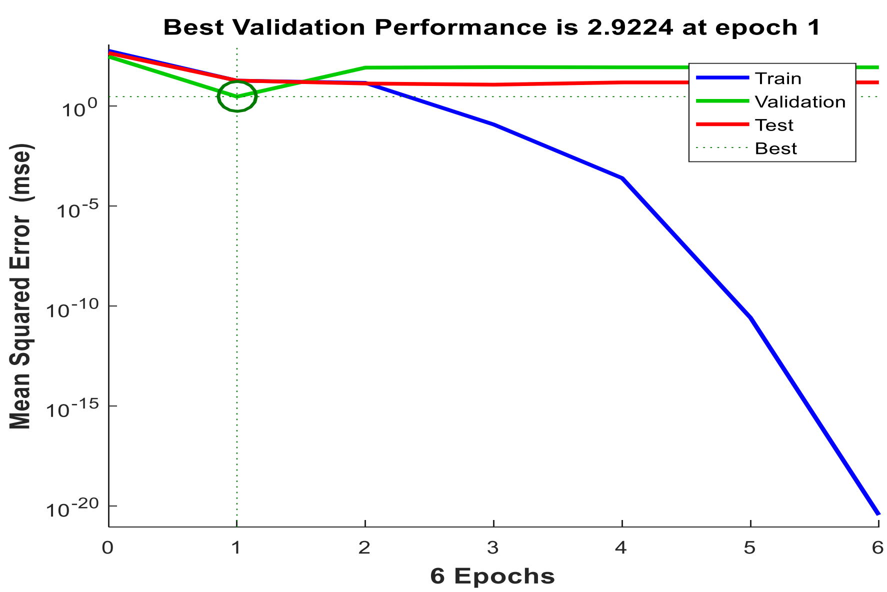


Figure 4.28: Performance in model-5 using MSE as an evaluation (LM)

In Figure 4.27, the values of R in the training, testing and combined phases are 0.948, 0.954 and 0.986 respectively. The higher the value of R the better the result. This showed that the result was better in the combined phase, followed by testing phase and least in the training phase. These values were further confirmed with the values of MSE (Figure 4.28) 3.58 (m^3/s), 3.55 (m^3/s) and 2.92 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.899, E = 0.968 and IOA = 0.732), testing phase (R^2 = 0.910, E = 0.970 and IOA = 0.736) and combined phase (R^2 = 0.972, E = 0.981 and IOA = 0.752). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. But the results from this model improved better than the one in model 1, model 2, model 3, model 4 and model 5 indicating more data set gives better results as can be seen in Table 4.4 below.

Table 4.4: Comparison of ANN_LM models using performance evaluation criteria

LM	TRAINING					TESTING					ALL				
MODEL	R	R^2	MSE (M3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA
MODEL 1	0.818	0.669	19.21	0.88	0.64	0.783	0.613	19.88	0.81	0.62	0.801	0.642	19.82	0.88	0.63
MODEL 2	0.842	0.709	15.23	0.91	0.65	0.807	0.651	16.08	0.86	0.64	0.829	0.687	15.86	0.91	0.64
MODEL 3	0.870	0.757	10.58	0.94	0.66	0.842	0.709	12.18	0.87	0.65	0.879	0.773	10.50	0.93	0.67
MODEL 4	0.891	0.794	5.18	0.95	0.68	0.882	0.778	5.77	0.89	0.67	0.888	0.976	5.37	0.94	0.68
Model 5	0.948	0.899	3.58	0.968	0.732	0.954	0.910	3.55	0.970	0.736	0.986	0.972	2.92	0.981	0.752

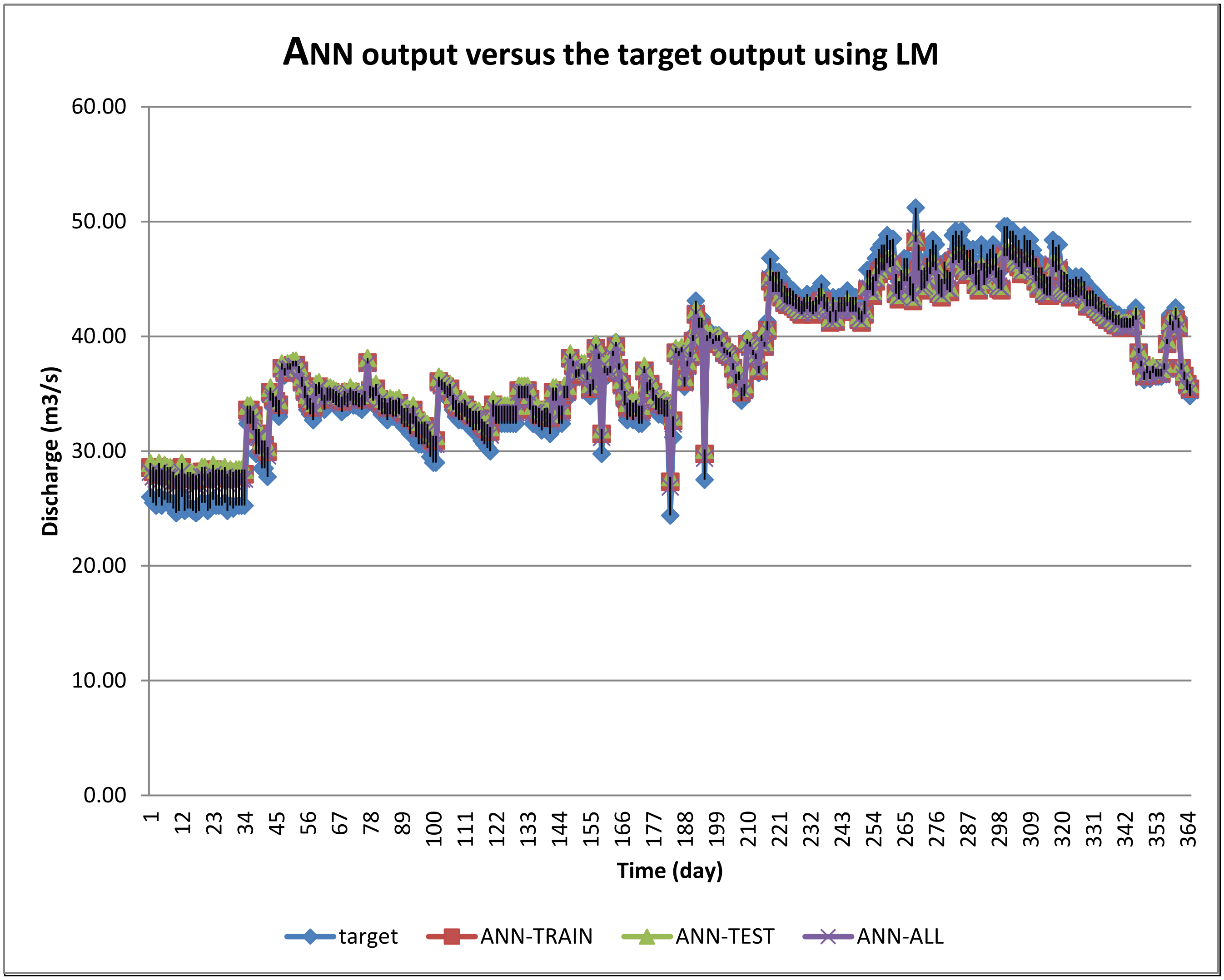


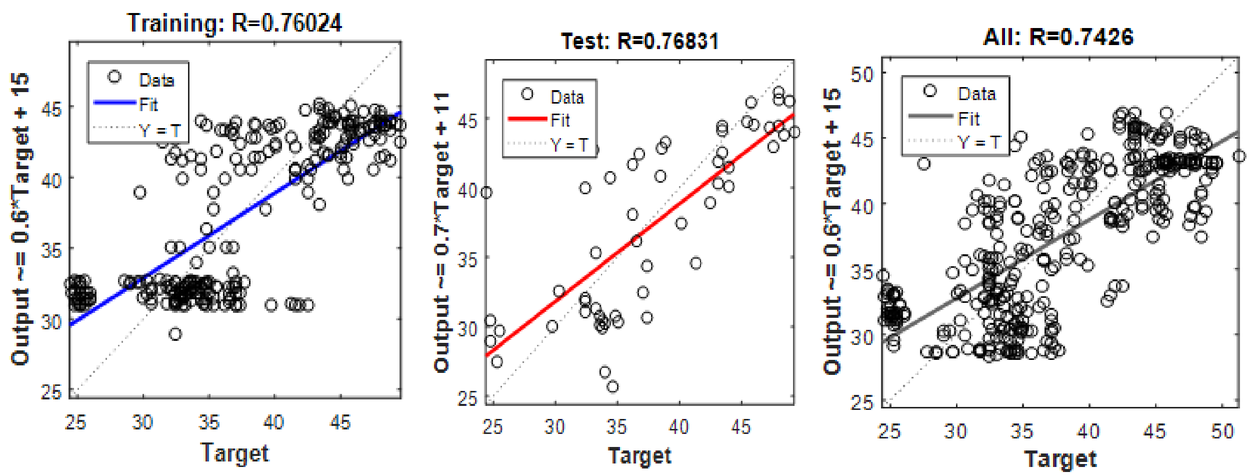
Figure 4.29: Comparing predicted discharge using train, test and all data sets against the target data sets (LM)

Figure 4.29 showed the graph of the predicted data and the measured data both in the training, testing and combined phases. The graph showed similar trend. The ANN models showed superiority in the combined phase. The training phase in model-5 showed an over-estimation of 0.14% of the observed target data sets while an over-estimation of 0.11% was observed in the testing phase (Table 4.4).

4.2.2. Results and Discussion from Scaled Conjugate Gradient Algorithm (SCG)

MODEL -1

The results of ANN_SCG from model 1 are shown in Figures 4.30 and 4.31.



(a) Training (b) Testing (c) All

Figure 4.30: Performance in model-1 using R as an evaluation criteria (SCG)

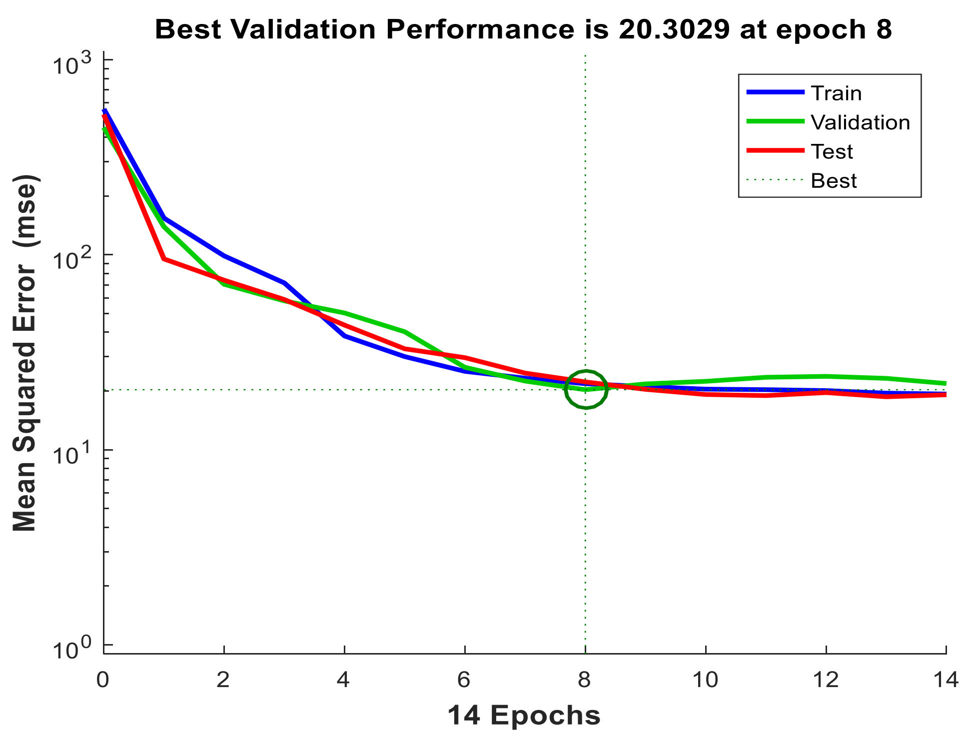
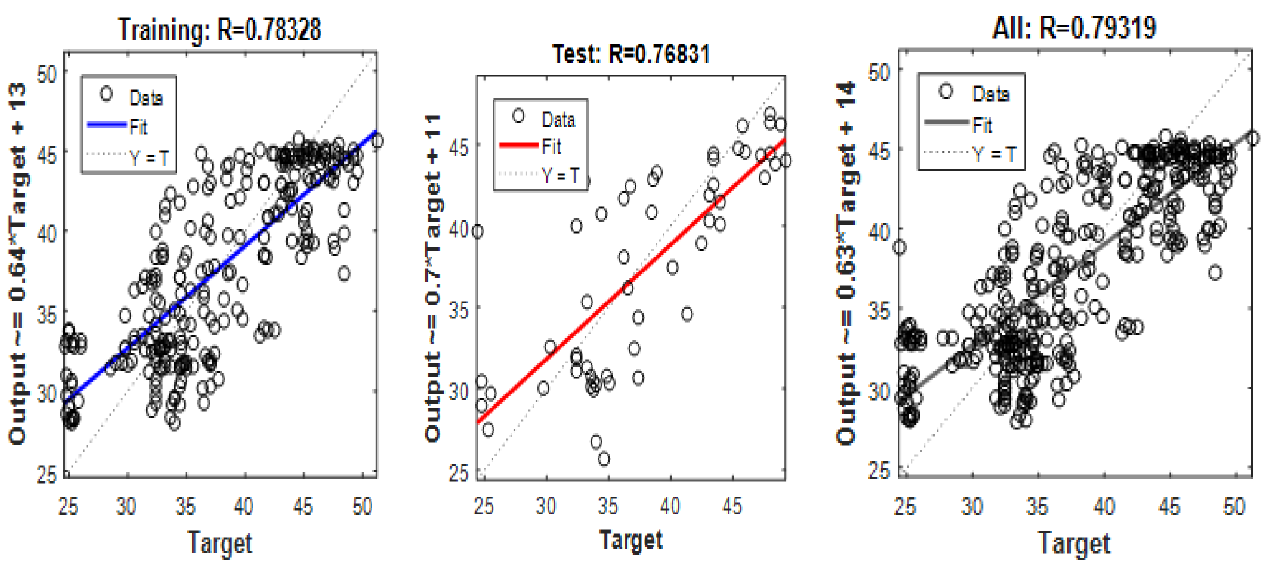


Figure 4.31: Performance evaluation using MSE as an evaluation criteria (SCG)

In Figure 4.30, the values of R in the training, testing and combined phases are 0.760, 0.768 and 0.743 respectively. The higher the value of R the better the result. This showed that the result was better in the testing phase, follow by training phase and least in the combined phase. These values were further confirmed with the values of MSE (Figure 4.31) 20.05 (m^3/s), 19.91 (m^3/s) and 20.30 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.578, E = 0.840 and IOA = 0.610), testing phase (R^2 = 0.590, E = 0.850 and IOA = 0.630) and combined phase (R^2 = 0.552, E = 0.840 and IOA = 0.610). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively.

MODEL-2

The results of ANN_SCG from model 2 are shown in Figures 4.32 and 4.33.



(a) Training (b) Testing (c) All

Figure 4.32: Performance in model-2 using R as an evaluation criteria (SCG)

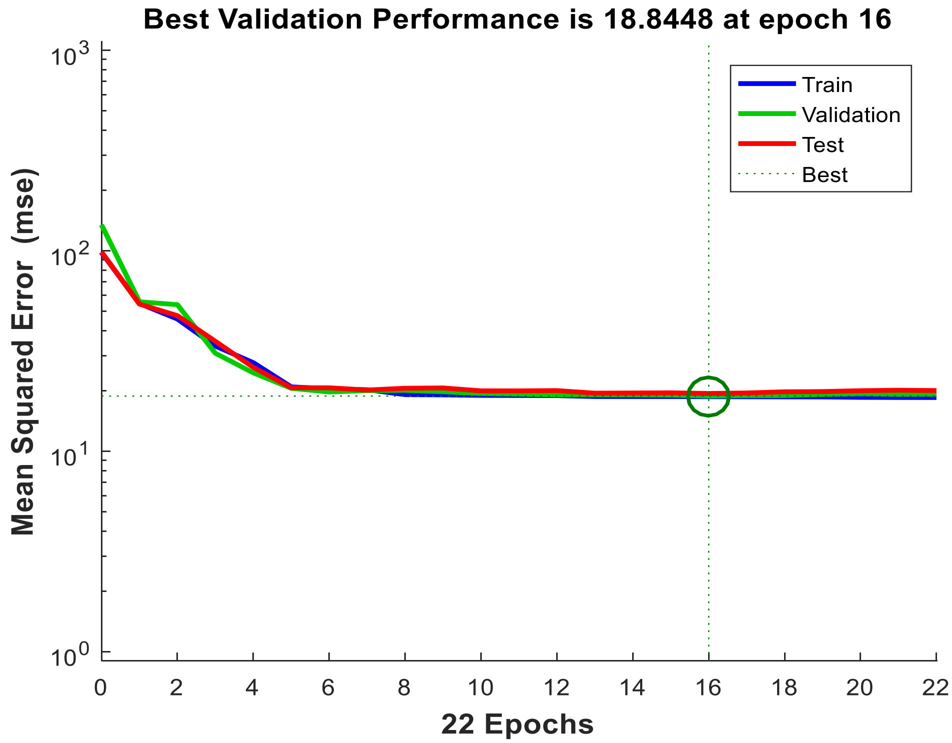
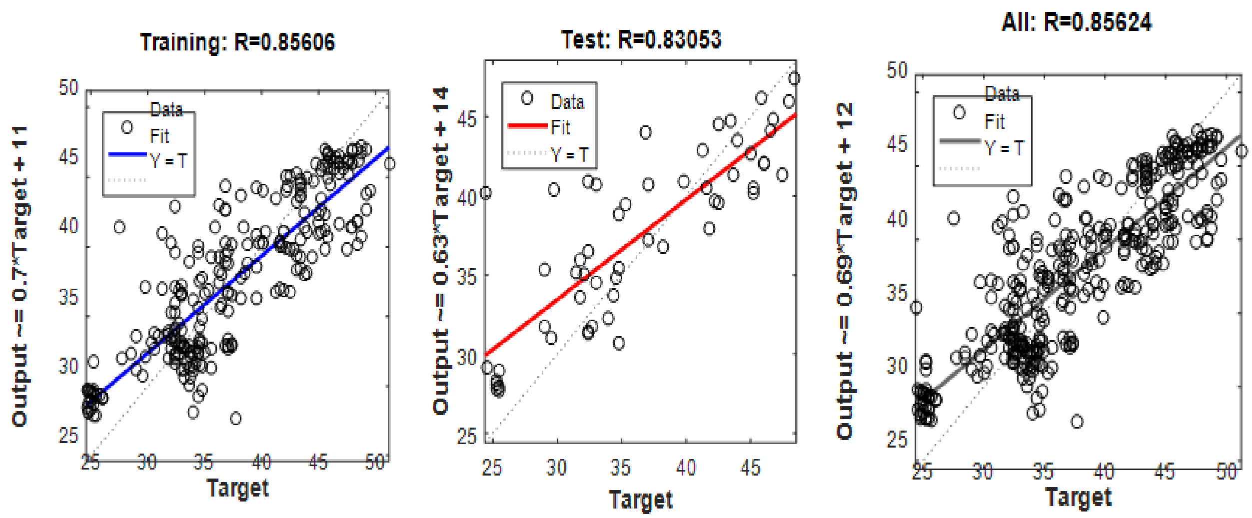


Figure 4.33: Performance evaluation using MSE as an evaluation criteria (SCG)

In Figure 4.32, the values of R in the training, testing and combined phases are 0.783, 0.768 and 0.793 respectively. The higher the value of R the better the result. This showed that the result was better in the combined phase, follow by training phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.33) 18.52 (m^3/s), 18.98 (m^3/s) and 18.50 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.613, E = 0.860 and IOA = 0.630), testing phase (R^2 = 0.590, E = 0.850 and IOA = 0.630) and combined phase (R^2 = 0.629, E = 0.860 and IOA = 0.620). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the results from model 2 improved better than that of model 1.

MODEL -3

The results of ANN_SCG from model 3 are shown in Figures 4.34 and 4.35.



(a) Training (b) Testing (c) All

Figure 4.34: Performance in model-3 using R as an evaluation criteria (SCG)

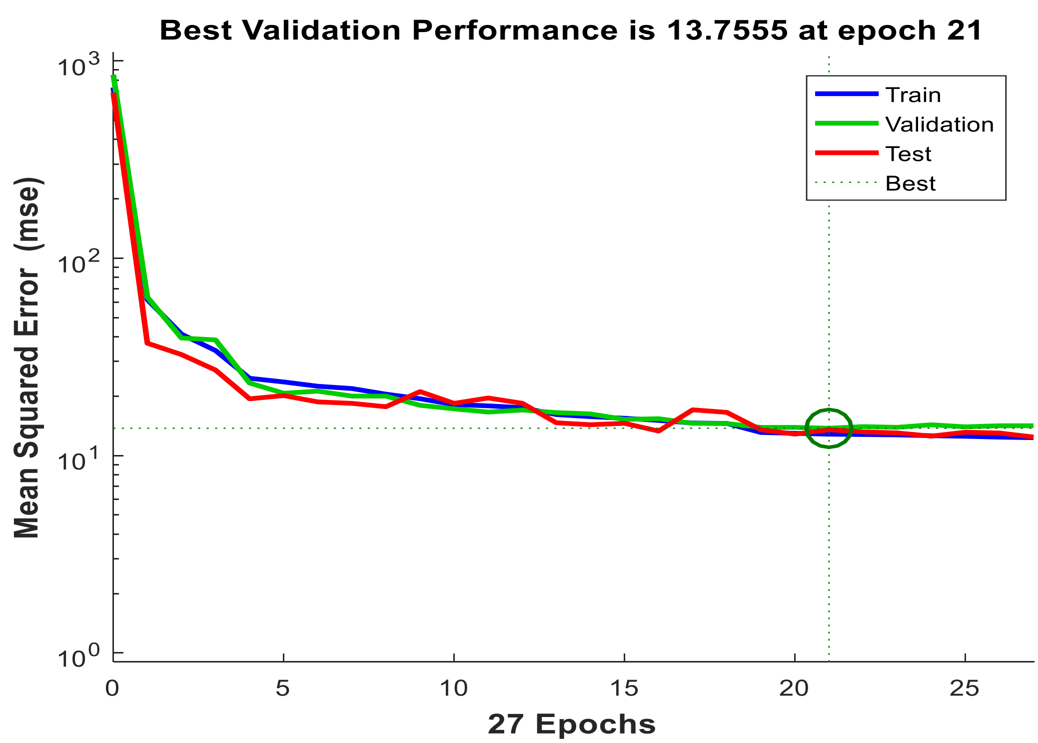
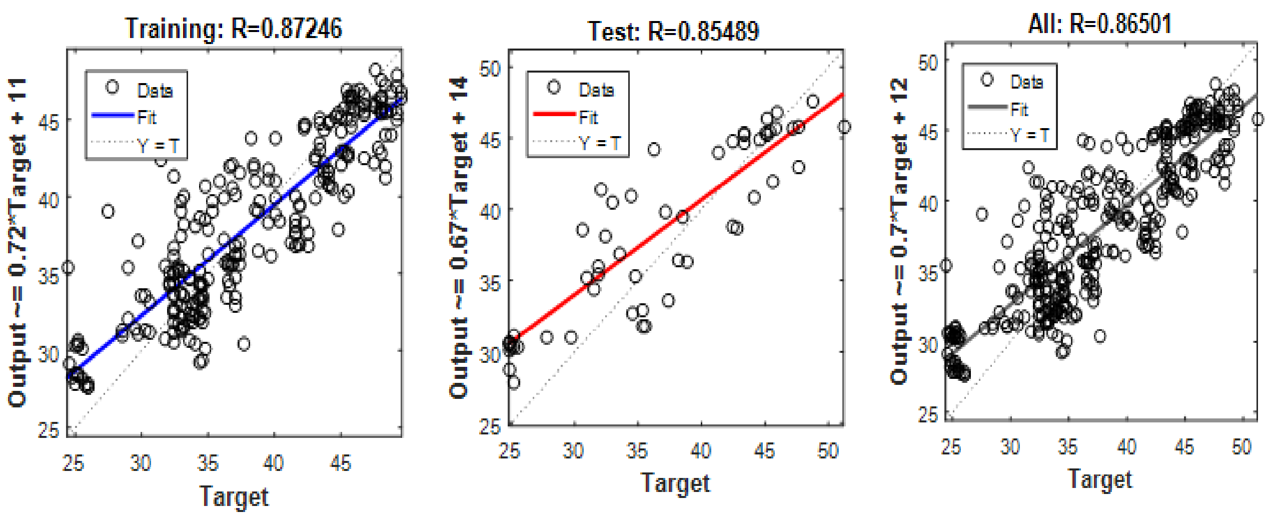


Figure 4.35: Performance evaluation using MSE as an evaluation criteria (SCG)

In Figure 4.34, the values of R in the training, testing and combined phases are 0.856, 0.831 and 0.856 respectively. The higher the value of R the better the result. This showed that the result was better in the combined phase and training phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.35) 13.76 (m^3/s), 13.87 (m^3/s) and 13.76 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.733, E = 0.910 and IOA = 0.652), testing phase (R^2 = 0.691, E = 0.890 and IOA = 0.651) and combined phase (R^2 = 0.733, E = 0.910 and IOA = 0.652). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the results from model 3 improved better than that of model 1 and model 2.

MODEL -4

The results of ANN_SCG from model 4 are shown in Figures 4.36 and 4.37.



(a) Training (b) Testing (c) All

Figure 4.36: Performance in model-4 using R as an evaluation criteria (SCG)

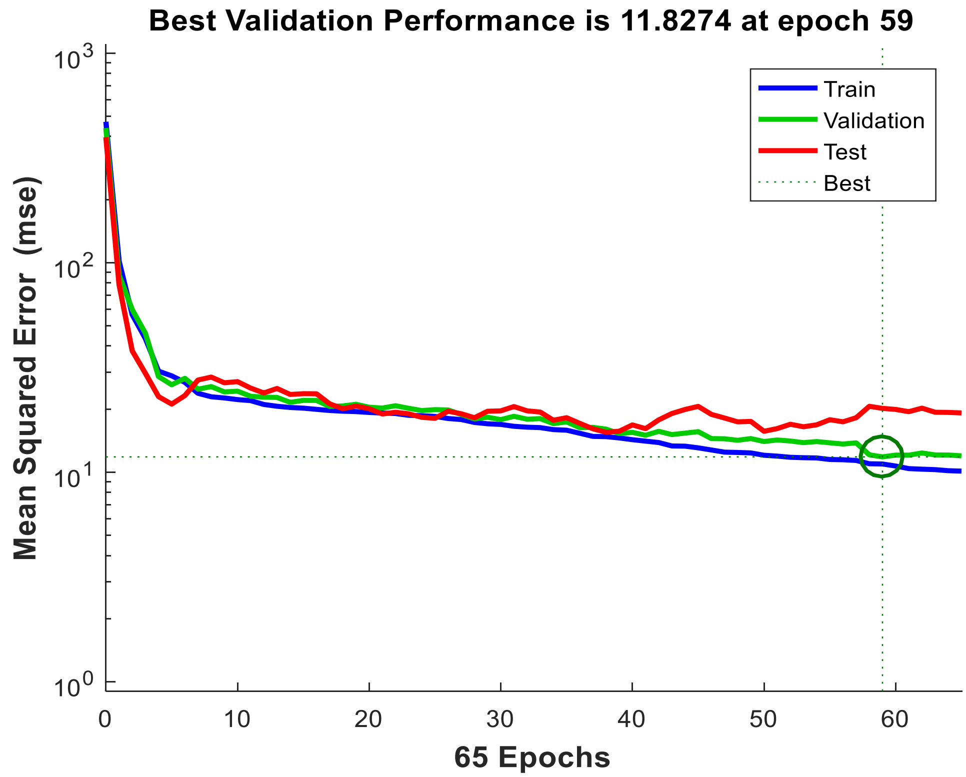
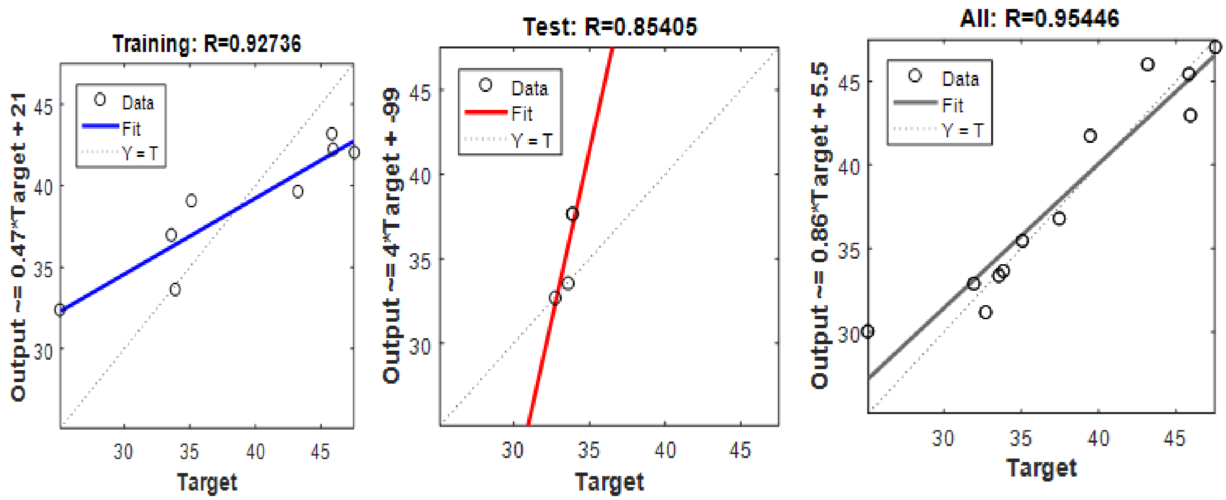


Figure 4.37: Performance evaluation using MSE as an evaluation criteria (SCG)

In Figure 4.36, the values of R in the training, testing and combined phases are 0.872, 0.855 and 0.865 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase followed by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.37) 11.27 (m^3/s), 12.88 (m^3/s) and 11.83 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.760, E = 0.930 and IOA = 0.661), testing phase (R^2 = 0.731, E = 0.900 and IOA = 0.654) and combined phase (R^2 = 0.748, E = 0.920 and IOA = 0.655). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the results from model 4 improved better than that of model 1, model 2 and model 3.

Model-5

The results of ANN_SCG from model 5 are shown in Figures 4.38 and 4.39.



(a) Training (b) Testing (c) All

Figure 4.38: Performance in model-5 using R as an evaluation criteria (SCG)

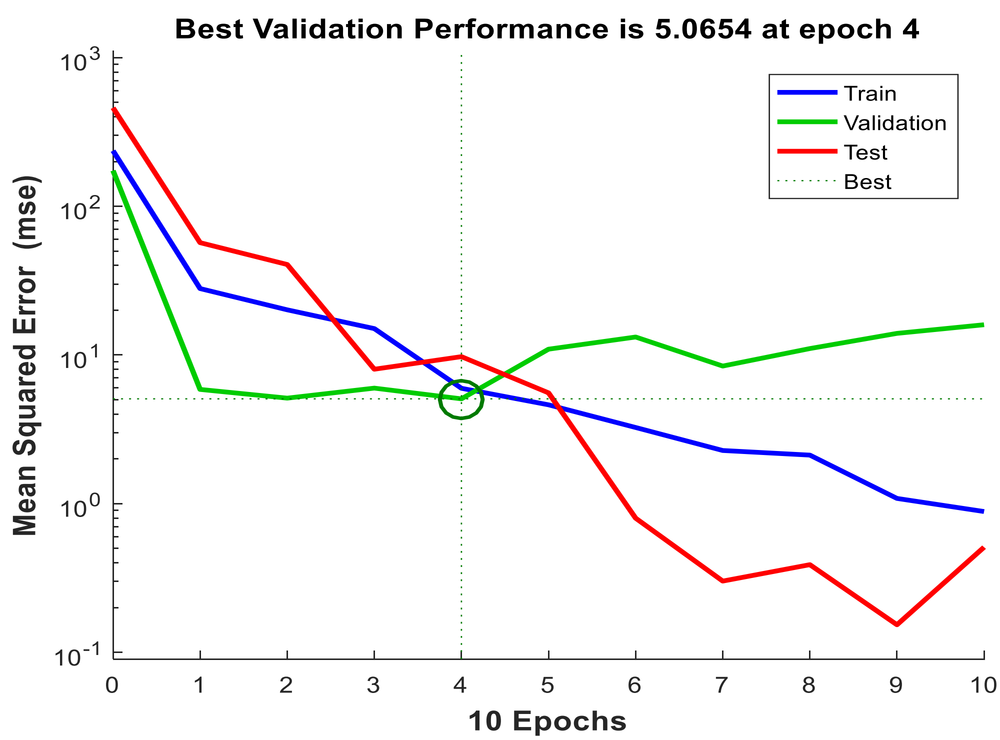


Figure 4.39: Performance in model-5 using MSE as an evaluation criteria (SCG)

In Figure 4.38, the values of R in the training, testing and combined phases are 0.927, 0.854 and 0.954 respectively. The higher the value of R the better the result. This showed that the result was better in the combined phase followed by training phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.39) 5.27 (m^3/s), 9.73 (m^3/s) and 5.07 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.859, E = 0.943 and IOA = 0.722), testing phase (R^2 = 0.729, E = 0.919 and IOA = 0.652) and combined phase (R^2 = 0.910, E = 0.948 and IOA = 0.731). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the results from model 5 improved better than that of model 1, model 2, model 3 and model 4 as seen in Table 4.5.

Table 4.5: Comparison of ANN_SCG models using performance evaluation criteria (SCG)

SCD	TRAINING					TESTING					ALL				
MODEL	R	R^2	RMSE (M3/s)	E	IOA	R	R^2	RMSE (M^3/s)	E	IOA	R	R^2	RMSE (M^3/s)	E	IOA
MODEL 1	0.760	0.578	20.05	0.84	0.61	0.768	0.590	19.91	0.85	0.63	0.743	0.552	20.30	0.84	0.61
MODEL 2	0.783	0.613	18.52	0.86	0.63	0.819	0.671	18.98	0.87	0.64	0.793	0.629	18.84	0.86	0.62
MODEL 3	0.856	0.733	13.76	0.91	0.652	0.831	0.691	13.87	0.89	0.651	0.856	0.733	13.76	0.91	0.652
MODEL 4	0.872	0.760	11.27	0.93	0.661	0.855	0.731	12.88	0.90	0.654	0.865	0.748	11.83	0.92	0.655
Model 5	0.927	0.859	5.27	0.943	0.722	0.854	0.729	9.73	0.191	0.652	0.954	0.910	5.07	0.948	0.731

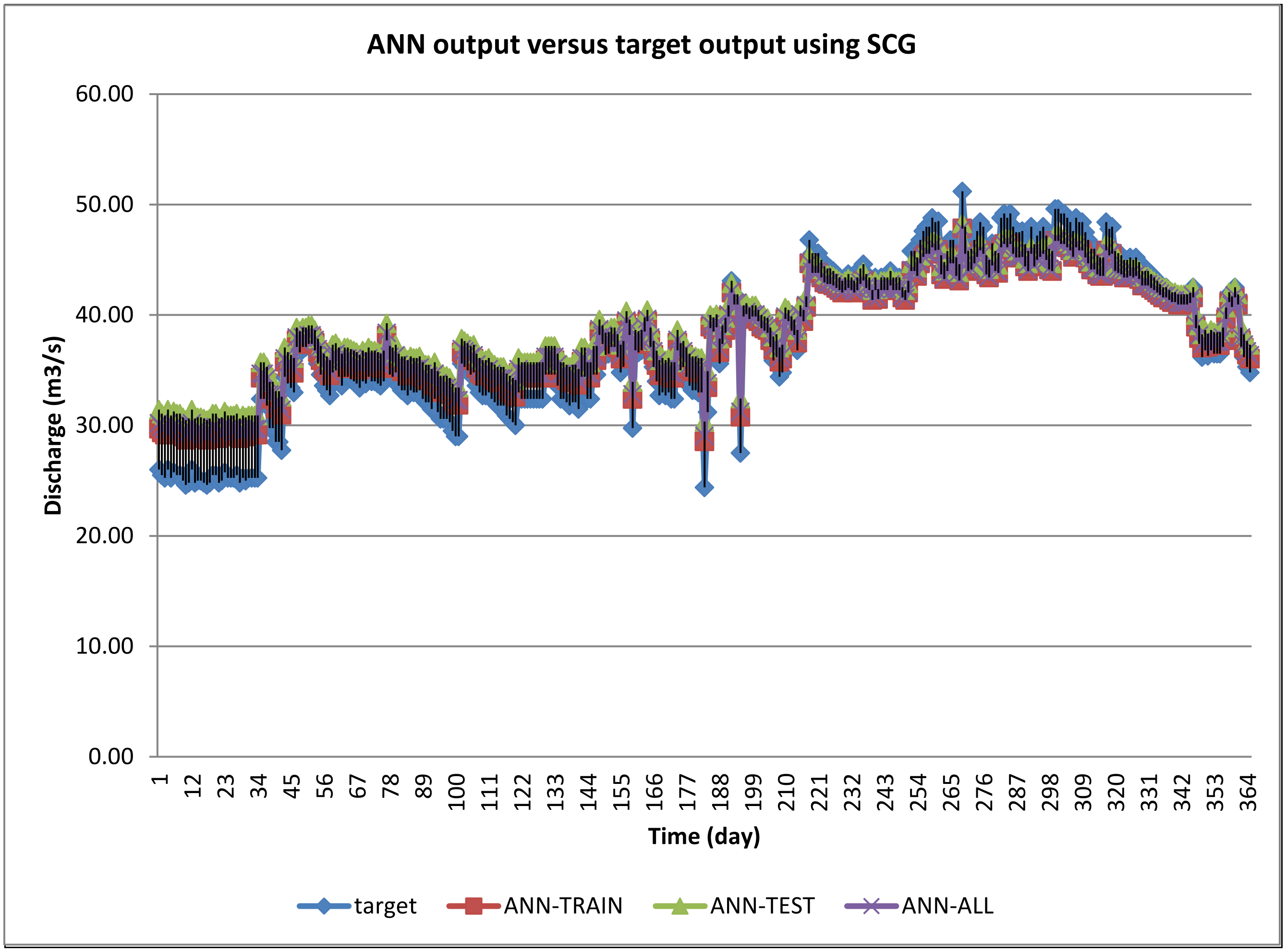


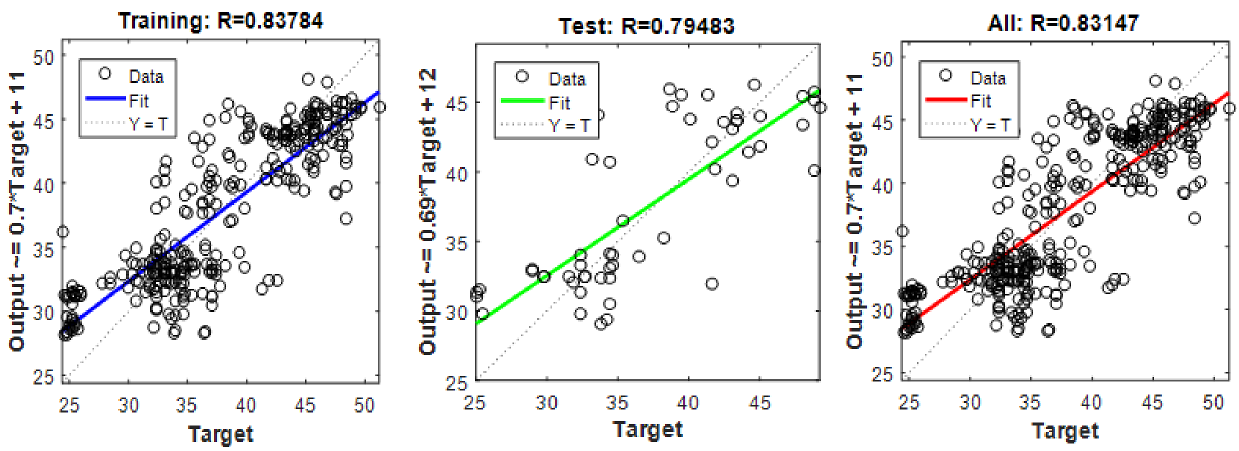
Figure 4.40: Comparing predicted discharge using train, test and all data sets against the target data sets (SCG)

Figure 4.40 showed the graph of the predicted data and the measured data both in the training, testing and combined phases. The graph showed similar trend. The ANN models showed more superiority in the combined phase. The training phase in model-5 showed an over-estimation of 0.21% of the observed target data sets while an over-estimation of 0.31% was observed in the testing phase (Table 4.5). Comparing the five evaluation criteria of ANN_LM and ANN_SCG algorithms, it was discovered that the LM algorithm performed better than SCG algorithm in the training, testing and combined phases.

4.2.3. Results and Discussion from Bayesian Regularization Algorithm (BR)

MODEL -1

The results of ANN_BR from model 1 are shown in Figures 4.41 and 4.42.



(a) Training (b) Testing (c) All

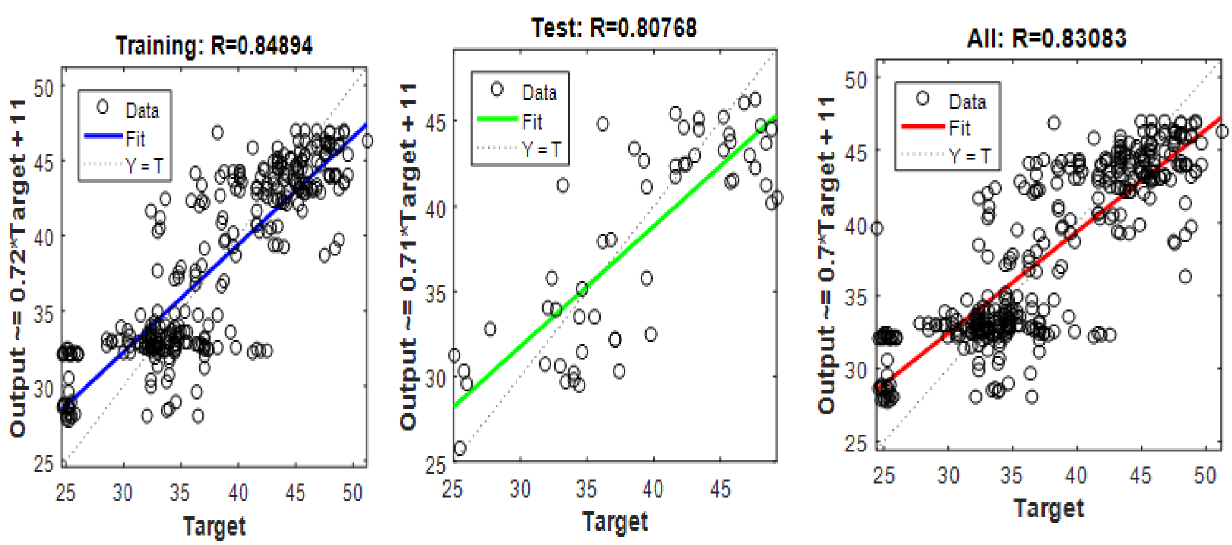
Figure 4.41: Performance in model-1 using R as an evaluation criteria (BR)

Figure 4.42: Performance evaluation using MSE as an evaluation criteria (BR)

In Figure 4.41, the values of R in the training, testing and combined phases are 0.838, 0.795 and 0.831 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase followed by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.42) 16.68 (m^3/s), 18.03 (m^3/s) and 16.72 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.702, E = 0.910 and IOA = 0.65), testing phase (R^2 = 0.632, E = 0.880 and IOA = 0.60) and combined phase (R^2 = 0.691, E = 0.90 and IOA = 0.63). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively.

MODEL -2

The results of ANN_BR from model 2 are shown in Figures 4.43and 4.44.



(a) Training (b) Testing (c) All

Figure 4.43: Performance in model-2 using R as an evaluation criteria (BR)

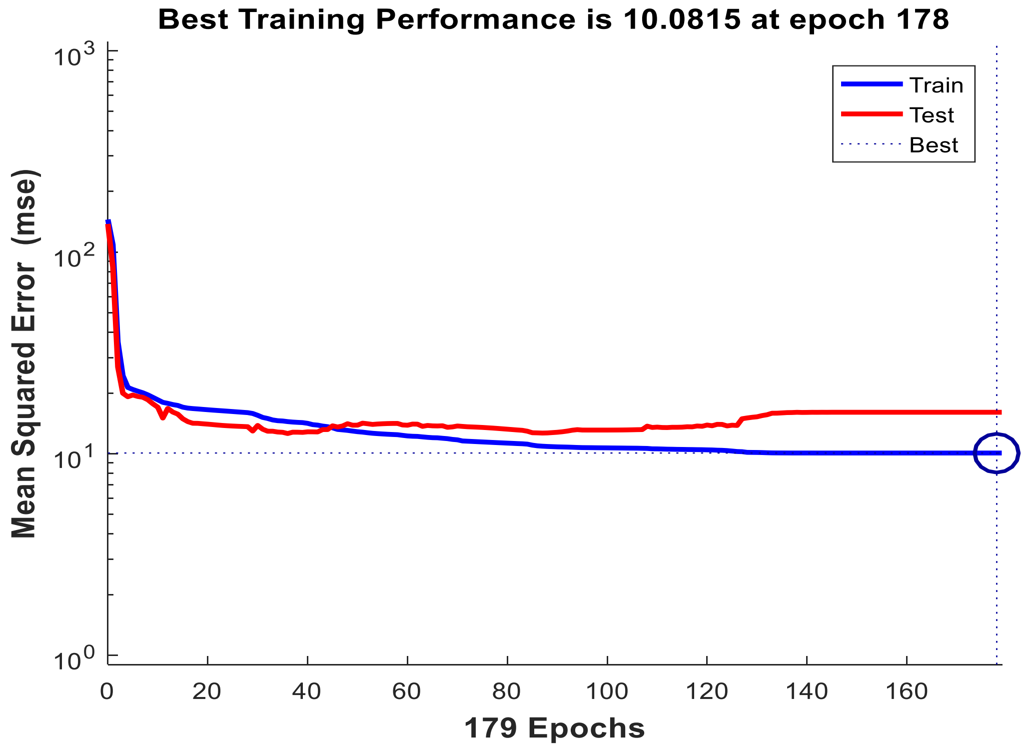
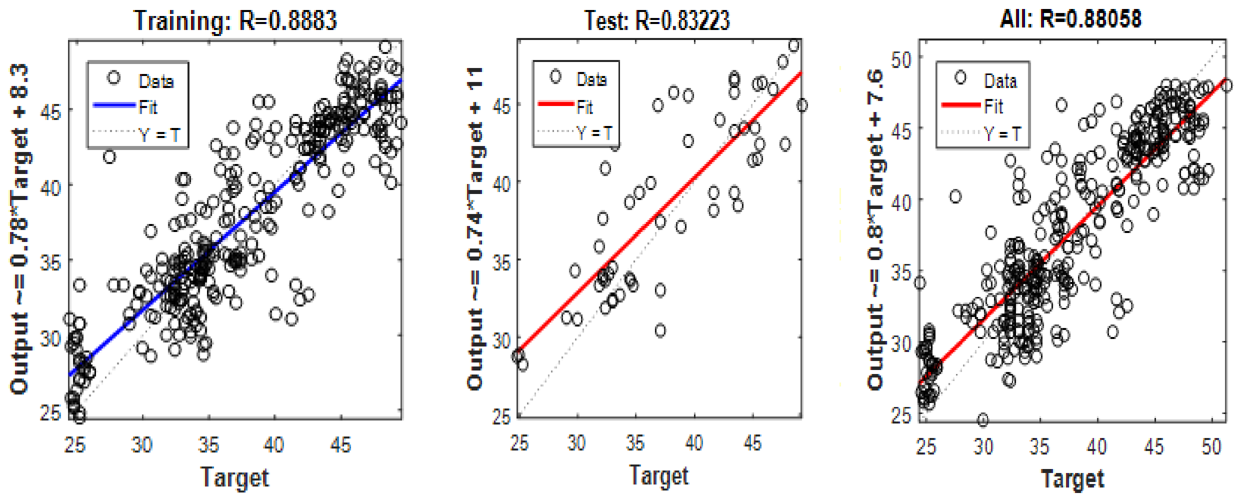


Figure 4.44: Performance evaluation using MSE as an evaluation criteria (BR)

In Figure 4.43, the values of R in the training, testing and combined phases are 0.849, 0.808 and 0.833 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase followed by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.44) 9.97 (m^3/s), 14.36 (m^3/s) and 10.08 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.721, E = 0.920 and IOA = 0.66), testing phase (R^2 = 0.653, E = 0.89 and IOA = 0.61) and combined phase (R^2 = 0.694, E = 0.91 and IOA = 0.64). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the result, from model 2 improved better than that of model 1.

MODEL -3

The results of ANN_BR from model 3 are shown in Figures 4.45 and 4.46.



(a) Training (b) Testing (c) All

Figure 4.45: Performance in model-3 using R as an evaluation criteria (BR)

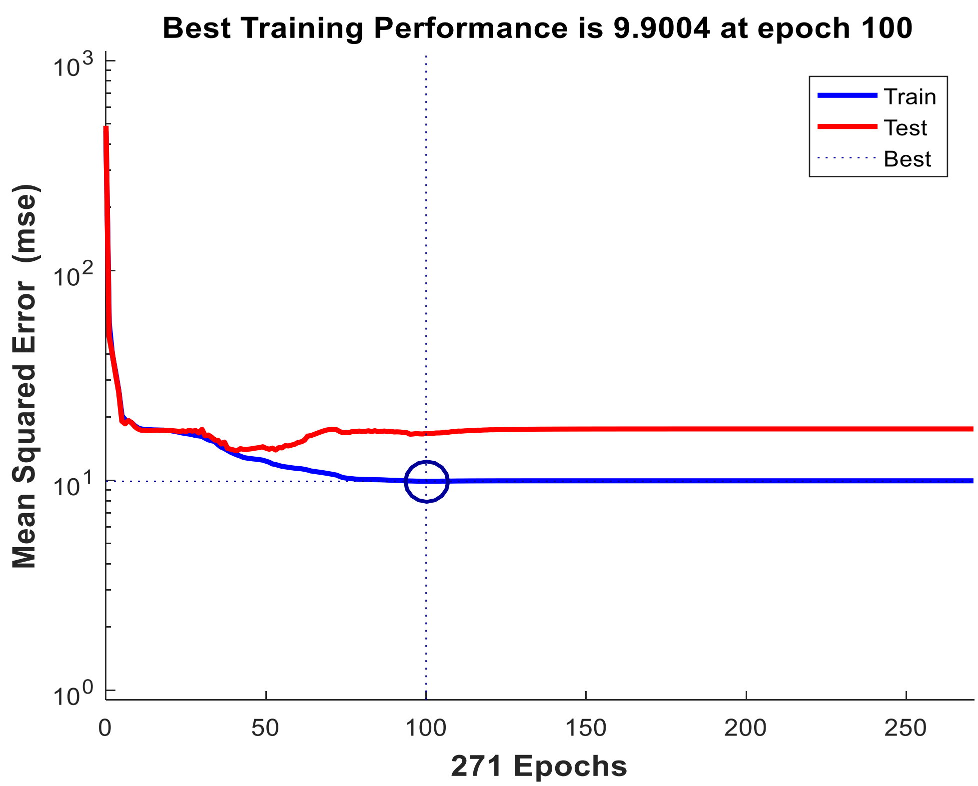
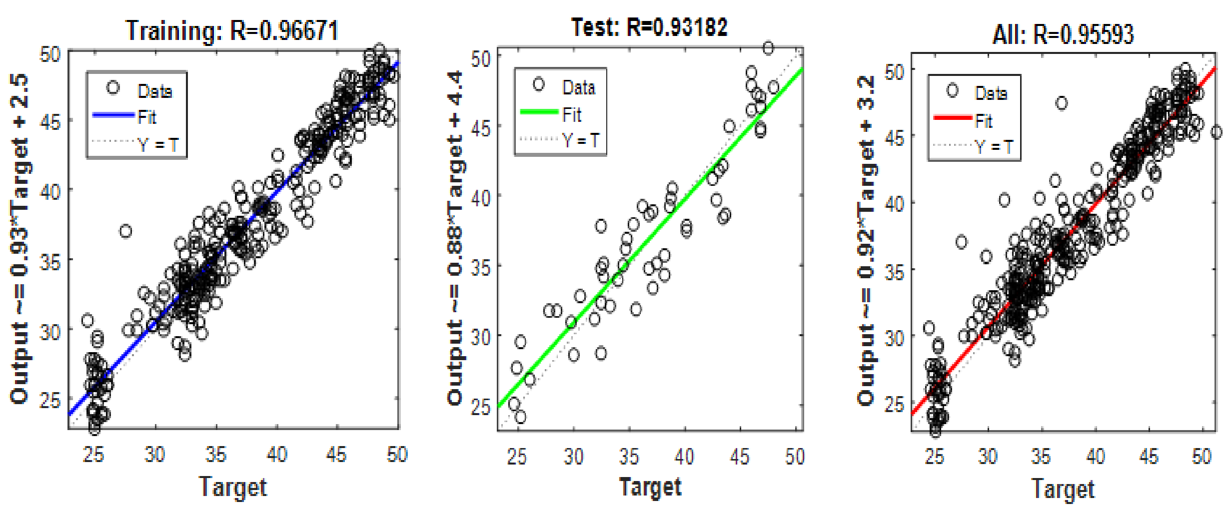


Figure 4.46: Performance evaluation using MSE as an evaluation criteria (BR)

In Figure 4.45, the values of R in the training, testing and combined phases are 0.888, 0.832 and 0.881 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase followed by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.46) 9.88 (m^3/s), 12.68 (m^3/s) and 9.90 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.789, E = 0.95 and IOA = 0.69), testing phase (R^2 = 0.692, E = 0.93 and IOA = 0.64) and combined phase (R^2 = 0.776, E = 0.96 and IOA = 0.67). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the result, from model 3 improved better than that of model 1 and model 2.

MODEL -4

The results of ANN_BR from model 4 are shown in Figures 4.47 and 4.48.



(a) Training (b) Testing (c) All

Figure 4.47: Performance in model-4 using R as an evaluation criteria (BR)

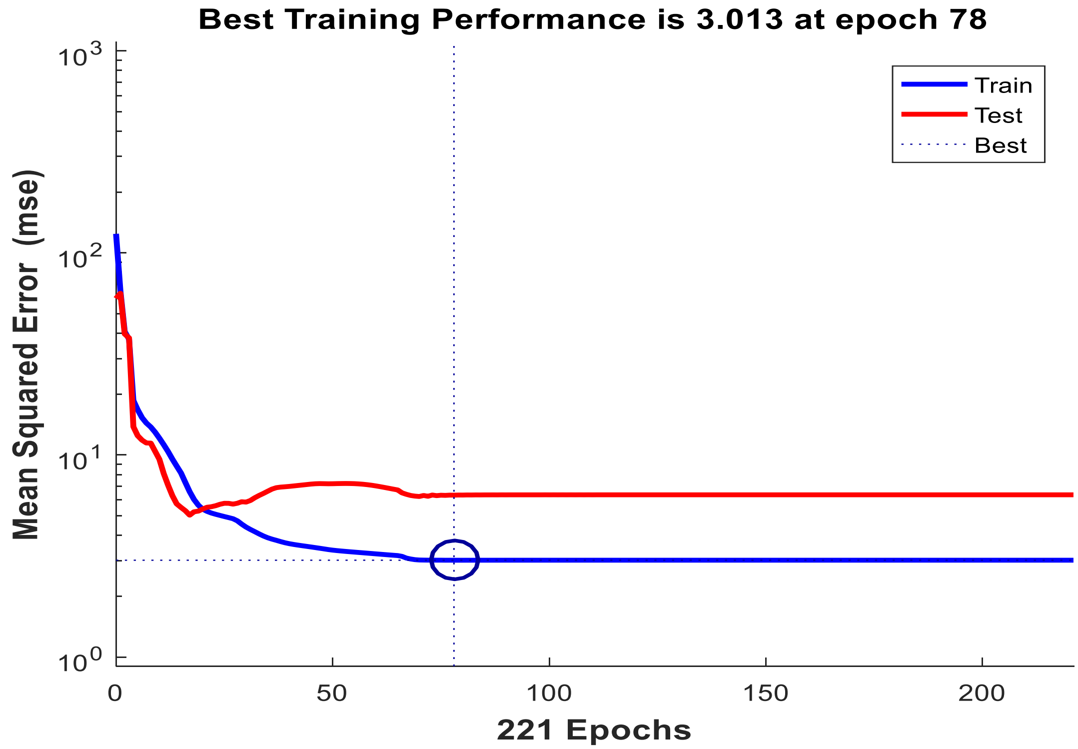
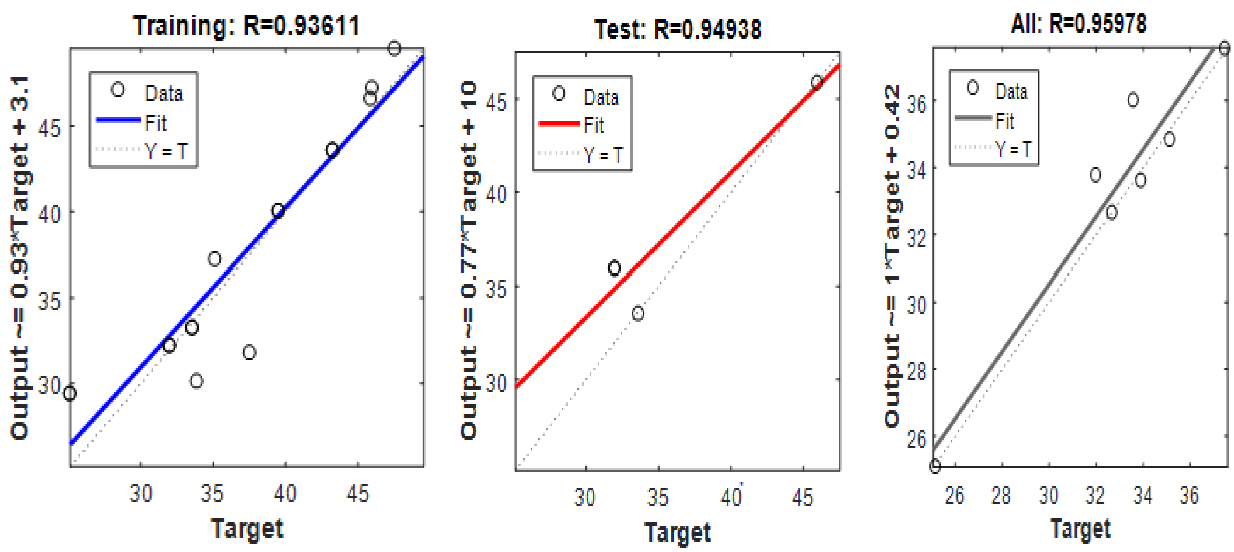


Figure 4.48: Performance evaluation using MSE as an evaluation criteria (BR)

In Figure 4.47, the values of R in the training, testing and combined phases are 0.967, 0.932 and 0.956 respectively. The higher the value of R the better the result. This showed that the result was better in the training phase followed by combined phase and least in the testing phase. These values were further confirmed with the values of MSE (Figure 4.48) 2.97 (m^3/s), 5.25 (m^3/s) and 3.01 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.935, E = 0.980 and IOA = 0.730), testing phase (R^2 = 0.869, E = 0.960 and IOA = 0.71) and combined phase (R^2 = 0.914, E = 0.970 and IOA = 0.72). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the result, from model 4 improved better than that of model 1, model 2 and model 3.

Model-5

The results of ANN_BR from model 5 are shown in Figures 4.49 and 4.50.



(a) Training (b) Testing (c) All

Figure 4.49: Performance in model-5 using R as an evaluation criteria (BR)

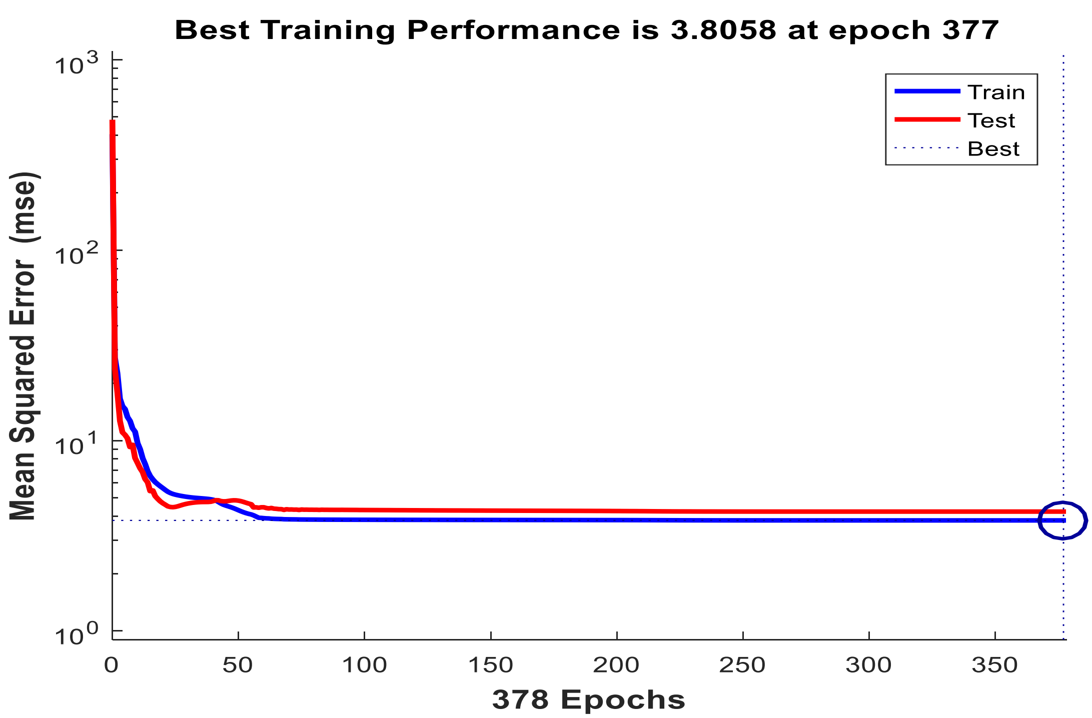


Figure 4.50: Performance in model-5 using MSE as an evaluation criteria (BR)

In Figure 4.49, the values of R in the training, testing and combined phases are 0.936, 0.950 and 0.960 respectively. The higher the value of R the better the result. This showed that the result was better in the combined phase followed by testing phase and least in the training phase. These values were further confirmed with the values of MSE (Figure 4.50) 4.25 (m^3/s), 3.87 (m^3/s) and 3.81 (m^3/s) in the training, testing and combined phase respectively. The lower the value of MSE the better the result. Three other performance criteria (coefficient of determination, R^2; modeling efficiency, E and index of agreement, IOA) were further used to confirm the validity of these results. In the training phase, the results are (R^2 = 0.876, E = 0.951 and IOA = 0.728), testing phase (R^2 = 0.903, E = 0.956 and IOA = 0.730) and combined phase (R^2 = 0.922, E = 0.959 and IOA = 0.738). These again confirmed the previous results as the higher the value of R^2, E and IOA, the better the result. Equations 2.30 and 2.31 were used to estimate the value of E and IOA respectively. However, the result, from model 5 improved better than that of model 1, model 2, model 3 and model 4 as seen in Table 4.6.

Table 4.6: Comparison of ANN BR models using performance evaluation criteria (BR)

BR	TRAINING					TESTING					ALL				
MODEL	R	R^2	RMSE (M3/s)	E	IOA	R	R^2	RMSE (M^3/s)	E	IOA	R	R^2	RMSE (M^3/s)	E	IOA
MODEL 1	0.838	0.702	16.68	0.91	0.65	0.795	0.632	18.03	0.88	0.60	0.831	0.691	16.72	0.90	0.63
MODEL 2	0.849	0.721	9.97	0.92	0.66	0.808	0.653	14.36	0.89	0.61	0.833	0.694	10.08	0.91	0.64
MODEL 3	0.888	0.789	9.88	0.95	0.69	0.832	0.692	12.68	0.93	0.64	0.881	0.776	9.90	0.96	0.67
MODEL 4	0.967	0.935	2.97	0.98	0.73	0.932	0.869	5.25	0.96	0.71	0.956	0.914	3.01	0.97	0.72
MODEL 5	0.936	0.876	4.25	0.951	0.728	0.950	0.903	3.87	0.956	0.730	0.960	0.922	3.81	0.959	0.738

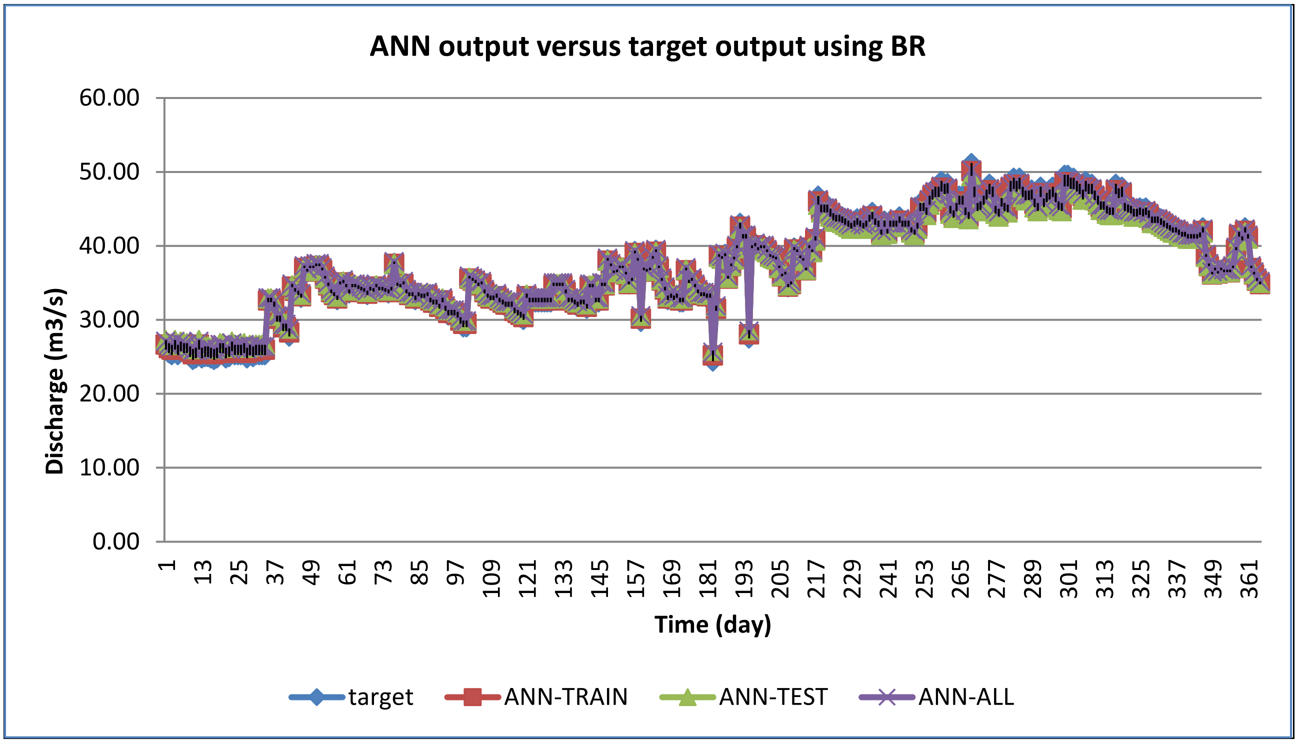


Figure 4.51: Comparing predicted discharge using train, test and all data sets against the target data sets (BR)

Figure 4.51 showed the graph of the predicted data and the measured data both in the training, testing and combined phases. The graph showed similar trend. The ANN models showed more superiority in the combined phase. The training phase in model-5 showed an over-estimation of 0.19% of the observed target data sets while an over-estimation of 0.17% was observed in the testing phase (Table 4.6). Comparing the five evaluation criteria of LM, SCG and BR algorithms with model 5, it was discovered the LM algorithm performed better than SCG and BR algorithms in the training, testing and combined phases. This indicated that the data set used with model 5 did not contain white noise since average values were used. But when model 1- 4 were used, BR performed better than LM and SCG. This indicates that the data used for model 1- 4 contained white noise which might be due to the infilling of missing discharge data using mean imputation method. BR is highly sensitive to white noise.

Comparing ANFIS models with ANN models using model-5 only

Since model 5 performed better than the other four models in all the analyses, comparison between ANFIS and ANN_LM (the best model) was only done with model 5. The comparison are showed in Table 4.7 and Figures 4.52- 4.54. The three phases, training, testing and combined were all used to make comparison.

Table 4.7: Comparison of ANFIS model-5 with ANN model-5 incorporating climate change using performance evaluation criteria

	TRAINING					TESTING					ALL				
MODEL	R	R^2	MSE (M3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA	R	R^2	MSE (M^3/s)	E	IOA
MODEL-5_ANFIS	1.00	1.00	0.57	0.991	0.872	1.00	1.00	0.57	0.992	0.873	0.988	0.976	0.82	0.987	0.842
MODEL-5_ANN_LM	0.948	0.899	3.58	0.968	0.732	0.954	0.910	3.55	0.970	0.736	0.986	0.972	2.92	0.981	0.752
MODEL-5_ANN_SCG	0.927	0.859	5.27	0.943	0.722	0.854	0.729	9.73	0.919	0.652	0.954	0.910	5.07	0.948	0.731
MODEL-5_ANN_BR	0.936	0.876	4.25	0.951	0.728	0.950	0.903	3.87	0.956	0.730	0.960	0.922	3.81	0.959	0.738

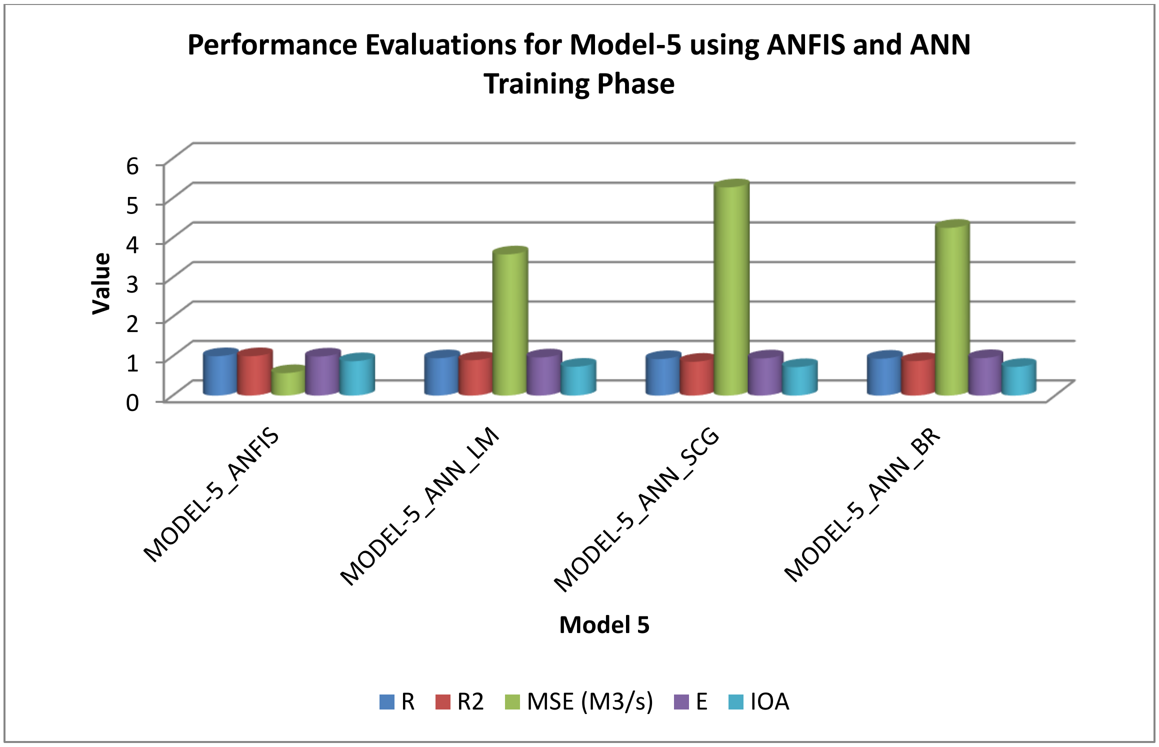


Figure 4.52: Performance evaluation of ANFIS and ANN using model-5 (training phase)

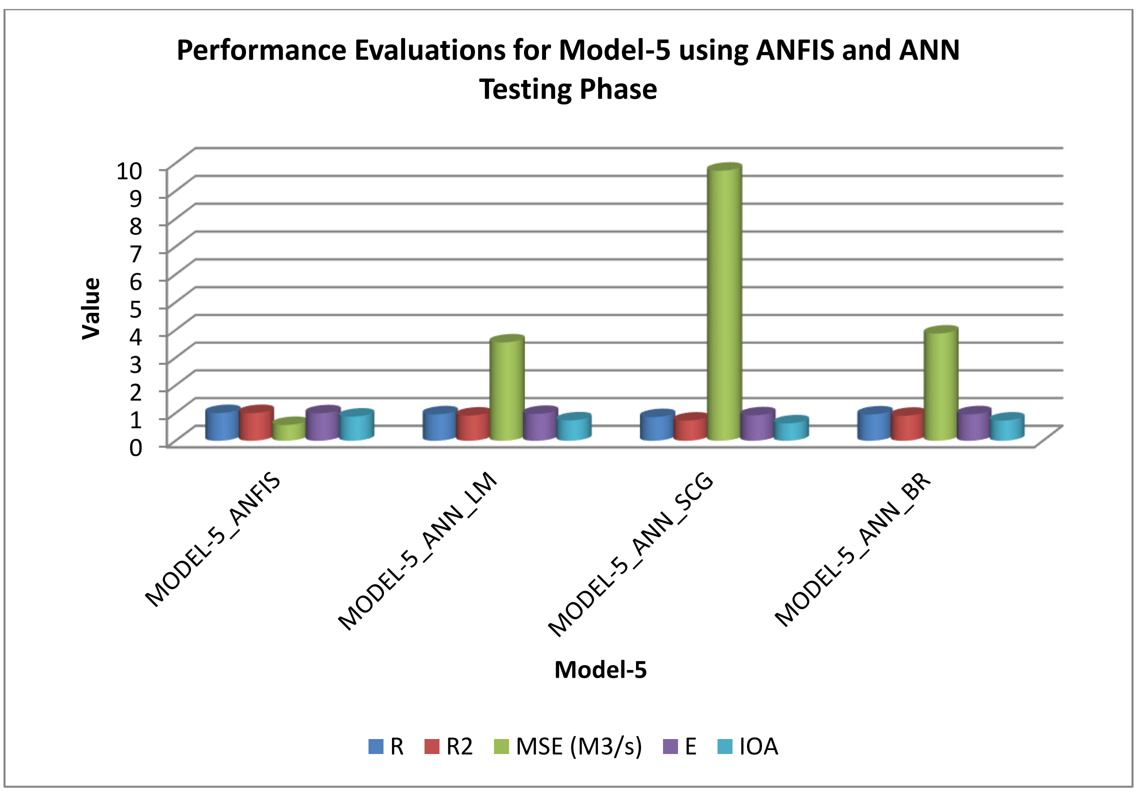


Figure 4.53: Performance evaluation of ANFIS and ANN using model-5 (testing phase)

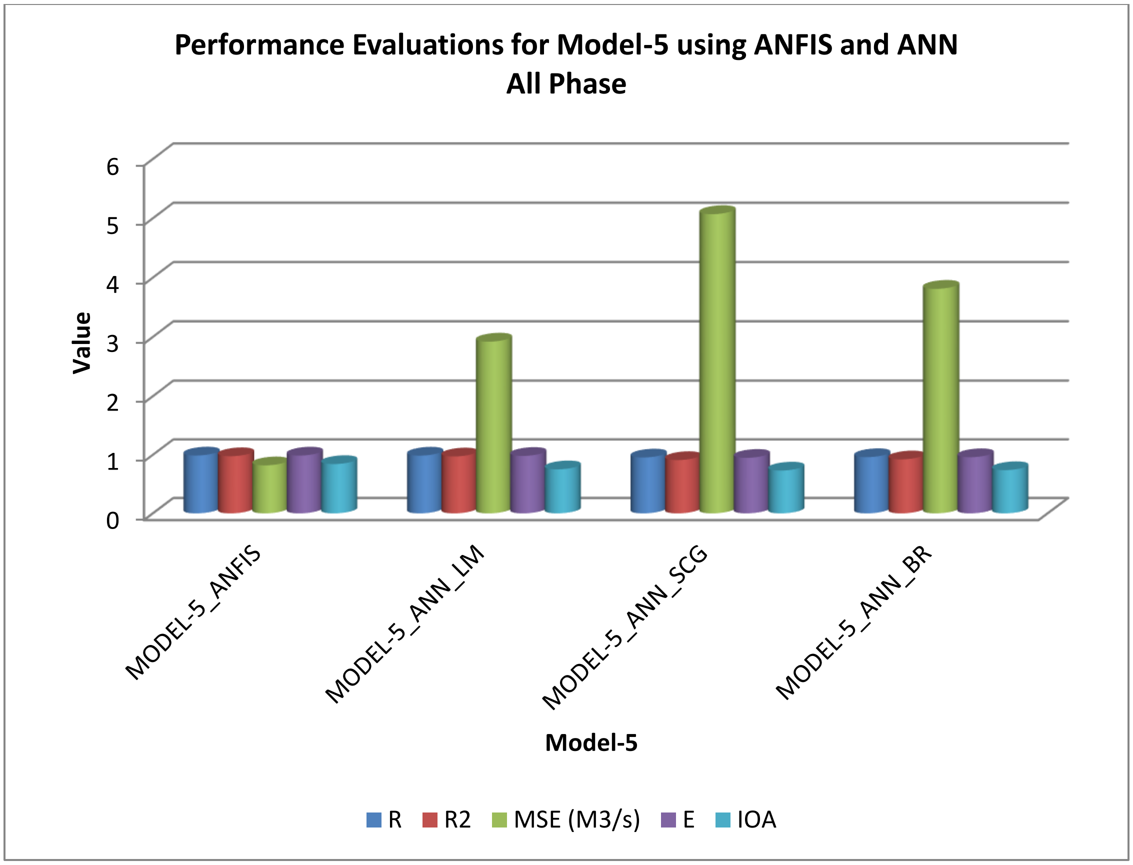


Figure 4.54: Performance evaluation of ANFIS and ANN using model-5 (combined phase)

Looking at Table 4.7 and the Figures 4.52-4.54 very closely, it was deduced that ANFIS models performed better than all the ANN models using training data sets, testing data sets and all the data sets.

Although all the algorithms of ANN in model 5 gave good results, Levenberg Marquardth algorithm performed best followed by Bayesian regulation and scaled conjugate gradient unlike in model-1 to 4 where Bayesian regularization performed the best. This is an indication that when the mean daily data sets (averaging variables) of the input and target variables were used in model-5, the effect of white noise was eliminated making Bayesian regularization inability to perform better than LM which is popularly used as default in many optimization and forecasting applications because of its greater computational efficiency. White noise is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean and finite variables. They are independent from one another. The superiority of ANFIS technique to ANN may be due to the fuzzy partitioning of the input space and for creating a rule-base inference system to generate the output.

Figure 4.55 showed the percentage of over estimation of the predicted data sets as compared to the measured ones.

The training phase of model-5_ANFIS, model-5_ANN_LM, model-5_ANN_SCG and model-5_ANN_BR, showed over-estimation of 0.043%, 0.14%, 0.21% and 0.19% of the observed target data sets while over-estimation of 0.044%, 0.11%, 0.31%, 0.17% were observed in the testing phase (Figure 4.55) respectively.

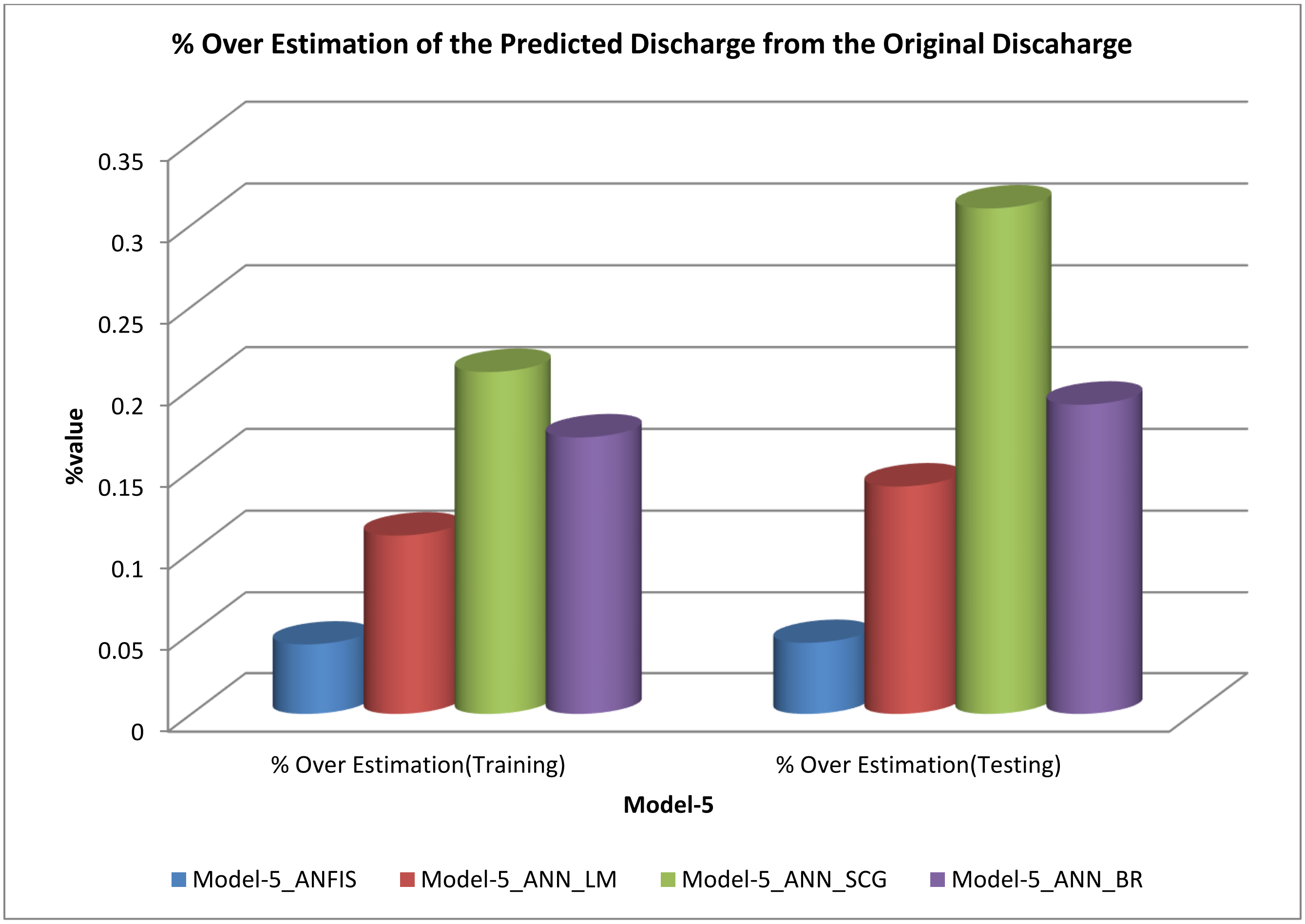


Figure 4.55: % over estimation of the discharge in model-5 using ANFIS and ANN

4.3. Evaluation of the Potential of Ikpoba Dam/Reservoir being used as a Multi-Purpose one

From the river discharge data (1991-1995) used for the prediction, the average monthly discharge computed is 31.9m^3/s as shown in Figure 4.56 and Table 4.8.

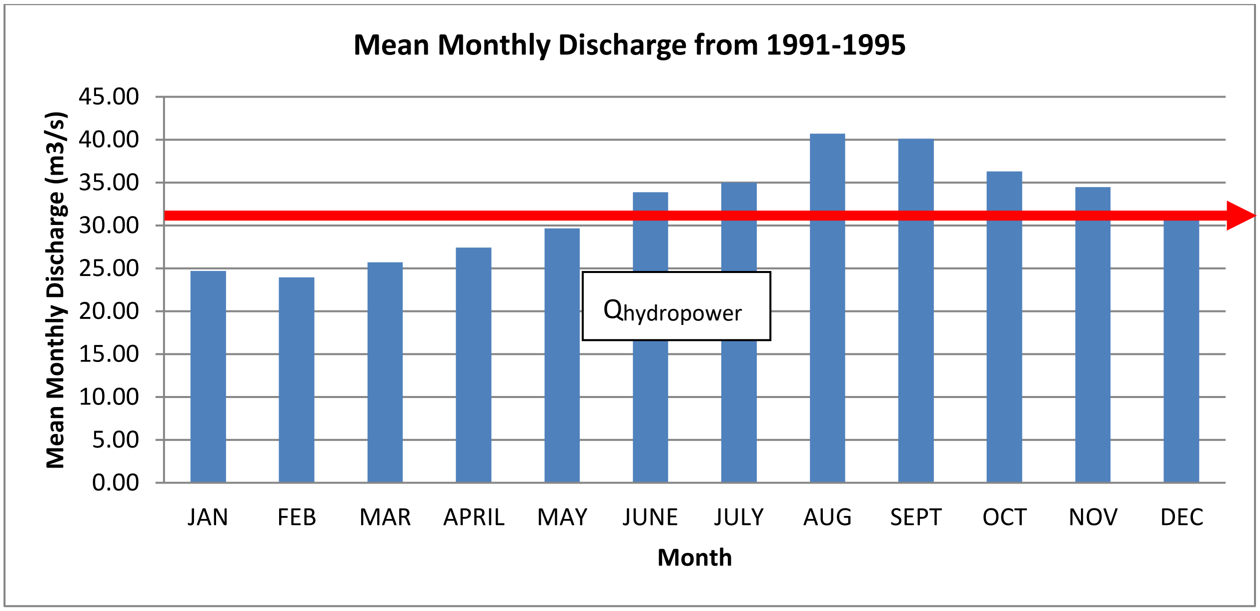


Figure 4.56: Mean Monthly discharge of Ikpoba River (1991-1995)

Table 4.8: Mean Daily Discharge of Ikpoba River (1991-1995)

YEAR	JAN	FEB	MAR	APRIL	MAY	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC	Mean
1991	16.66	16.27	20.66	28.53	32.70	39.29	44.13	52.02	41.20	32.41	34.28	36.21	32.86
1992	29.10	22.10	24.00	27.70	27.20	32.30	28.80	34.40	33.60	34.30	31.30	25.00	29.15
1993	27.64	26.01	26.55	24.56	27.50	30.83	27.85	31.73	34.86	29.26	27.41	24.67	28.24
1994	24.78	22.65	23.43	24.42	27.41	31.80	36.69	42.15	45.08	38.06	33.41	29.38	31.61
1995	25.23	32.69	33.87	31.96	33.55	35.10	37.49	43.20	45.85	47.54	45.90	39.48	**37.66**
Mean	24.68	23.94	25.70	27.43	29.67	33.86	34.99	40.70	40.12	36.31	34.46	30.95	**31.90**

The following are the hydraulic and hydrologic data of Ikpoba dam/reservoir (Ehiorobo, 2008; Anyata, 2013):

Capacity of the reservoir = $1.5 \times 10^6 m^3$

Reservoir surface area = $1.07 \times 10^6 m^2$

Catchment area of the reservoir = $120 km^2$

Length of the dam = 610m

Crest level height of the dam = 35m (above mean sea level)

Spilled way length (weir) = 60m

Emergency spilled length = 4m

Ultimate Pumping capacity of the dam = 160,000,000l/day

Table 4.9: Annual Total Run-off Volume of Ikpoba River (1991-1995) (BORDA, 2005)

S/N	YEAR	ANNUAL TOTAL RUN-OFF VOLUME ($10^9 m^3$)
1	1991	0.786
2	1992	0.824
3	1993	0.878
4	1994	0.968
5	1995	1.1867
	Average Annual run-off volume	**0.9285**

The average annual run-off volume of Ikpoba River was evaluated to be $0.9285 \times 10^9 m^3$ as shown in Table 4.9 and Figure 4.57. The total annual run-off volume increased yearly, an indication that the river is under siltation. The annual total run-off volume for year 1999 and 2000 are also shown in Table 4.10 for further comparison.

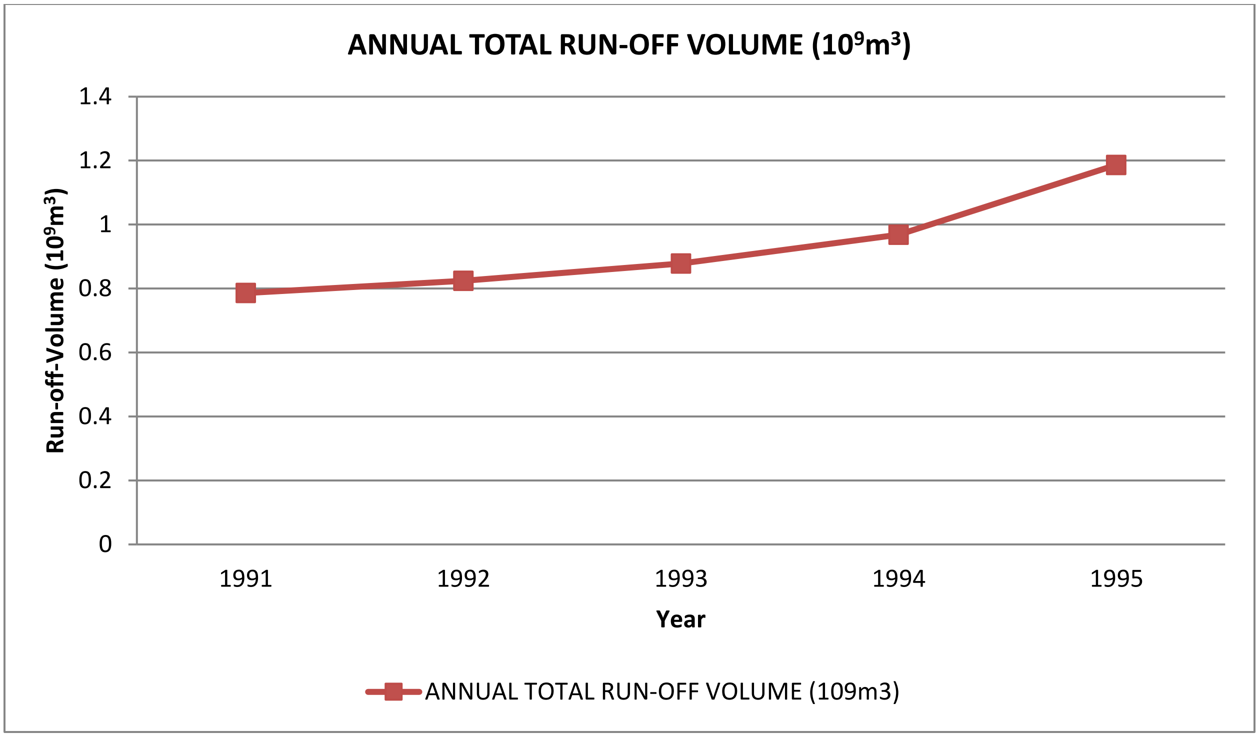


Figure 4.57: Annual Total Run-off Volume of Ikpoba River (1991-1995)

Table 4.10: Annual Total Run-off Volume of Ikpoba River (1999-2000) (BORDA, 2005)

S/N	YEAR	ANNUAL TOTAL RUN-OFF VOLUME (10^9m^3)
1	1999	1.375
2	2000	1.443

4.3.1. Possibility of using Ikpoba River Dam for Hydropower Generation

In Figure 4.56 which showed the mean monthly discharge data of Ikpoba River from 1991-1995, the design discharge for the hydropower plant was taken as the nominal (average) monthly discharge, 31.90m^3/s for safety purposes. But the maximum and minimum discharges of the river are 40.70m^3/s and 23.94m^3/s respectively. Based on the height of the dam (35m), the following can be assessed and assumed:

Gross head of the dam = 35-3 = 32m (3 = free board)

Hydraulic loss = 4% of the gross head = 1.28m

Net head of the dam = 32 – 1.28 = 30.72m

Efficiency of turbine = 0.90

Efficiency of penstock = 0.95

Efficiency of generator = 0.85

Correction factor = 0.75

Nominal (average) monthly discharge = 31.90m^3/s

Maximum monthly discharge = 40.70m^3/s

Minimum monthly discharge = 23.94m^3/s

Assume the hydropower plant runs continuously for 24 hours, the instantaneous power for any given month that could be obtained from the power plant (http://www.renewablesfirst.co.uk),

$P = \eta*\rho*g*h*Q*\zeta$

Where:

η = total efficiency of the plant, = 0.90*0.95*0.85 = 0.73

ρ = density of water, 1000kg/m^3

g = acceleration due to gravity, 9.81m/s^2

h = net height of dam, 30.72m

ζ = correction factor = 0.75

Assumptions:

12 months in a year

30 days in a month

24 hours in a day

Maximum power (P_{max}) that can be installed = 0.73*1*9.81*30.72*40.70*0.75 = **6715.4KW = 6.72MW**

Maximum annual energy capacity (E_{max}) of the power plant = (P_{max} * t) = 6.72*12*30*24

= **58060.8MWh** (Where t = time period under consideration, 24 hours per day)

Nominal (average) power (P_a) that can be installed = 0.73*1*9.81*30.72*31.90*0.75 = **5263.4KW**

= 5.26MW

The nominal (average) annual energy capacity (E_a) of the power plant, (P_a * t) = 5.26*12*30*24

= **45446.4MWh**

Minimum power (P_{min}) that can be installed = 0.73*1*9.81*30.72*23.94*0.75 = **3950.0KW**

= 3.95MW

The minimum annual energy capacity (E_{min}) of the power plant, (P_{min} * t) = 3.95*12*30*24

= **34128.0MWh**

From table 4.9 and figure 4.57, average total annual run-off volume of Ikpoba river downstream (Table 4.9):

= **0.9285 x 10^9m^3** **(a)**

The nominal (average) discharge of the river = 31.9m^3/s

The total annual run-off volume of the river at this discharge value

= 31.9m^3/s * 60seconds *60minutes * 24hours * 365days = **1.006 x 10^9m^3** **(b)**

Therefore, the water volume balance in the river is the difference between (b) and (a)

1.006 x 10^9m^3 - 0.9285 x 10^9m^3 = 0.0775 x 10^9m^3 **(c)**

The maximum designed capacity utilization for the water supply to Benin-City = 1.6 x 10^5m^3/day

For one year, the volume of water required = 1.6 x 10^5m^3/day * 365days = 5.84 x 0^7m^3

= 0.0584 x 0^9m^3 **(d)**

The annual water balance in the river going to the reservoir after abstraction for portable water supply is the difference between (c) and (d).

= 0.0775 x 10^9m^3 - 0.0584 x 0^9m^3 **= 0.0191 x 0^9m^3/year**

$= 0.0191 \times 10^9/365 = \mathbf{0.523 \times 10^6 m^3}$

But the reservoir capacity = $\mathbf{1.5 \times 10^6 m^3}$

It was therefore inferred that Ikpoba dam at ultimate pumping capacity of 160,000,000 liters/day could also be utilized to produce an average monthly power of 5.26MW using hydropower plant. The annual volume of water that will remain in the dam reservoir (needed for this hydropower scheme) after the abstraction for portable water supply is $\mathbf{0.523 \times 10^6 m^3}$

The above value is the annual volume of water that will be available for hydropower development in Ikpoba dam since plan might not be necessary for irrigation scheme within the river sub basin because of non-availability of large cultivable land spaces and ruggedness of the terrain. A 5.26MW power was therefore proposed for Ikpoba dam. The hydropower scheme proposed will consist of a forebay from which the flow of water can be regulated into and through a penstock to a power house. The power house will house the electro-mechanical equipment for the generation of electricity. It was also observed that the run-off volume increases linearly from year 1991 to 1995 (figure 4.57) and even becomes high ($1.375 \times 10^9 m^3$ and $1.443 \times 10^9 m^3$) in year 1999 and 2000 respectively (Table 4.10). This could be due to effect of climate change and also an indication that the reservoir is under high siltation which could lead to a high reduction in the storage capacity of the reservoir. It is therefore advisable to de-silt the reservoir regularly in order to return it to its original storage capacity else, its full optimization would not be achieved. The detailed design of the proposed hydropower scheme and its environmental impact/social assessments (EISA) are not covered in this research.

CHAPTER 5

5.0. FINDINGS, CONCLUSIONS, RECOMMENDATIONS AND CONTRIBUTIONS TO KNOWLEDGE

From the discussion of results, the following detailed findings, conclusions and recommendations are presented. Contributions to knowledge and further research areas are also presented.

5.1. Findings

The following were the findings from the study:

1. The ANFIS and ANN models developed were able to predict the river discharge of Ikpoba river especially model-5 when the effect of climate change was incorporated.
2. The training, testing and validation of the data sets carried out were able to predict the data to a reasonable degree of accuracy of error tolerance of less than $\pm$ 0.5%.
3. The five statistical evaluation criteria used were able to test the performances of the developed models accurately with model-5 performing better than the others.
4. The ANFIS models performed better than ANN models with the three ANN algorithms used. ANFIS model-5 was selected as the best forecast technique and model to be used.
5. Ikpoba dam/reservoir has the potential of being used for hydropower generation producing 5.26MW of power (with an average discharge of 31.9m^3/s) monthly. The annual volume of water in the reservoir available for this hydropower scheme is 0.523 x $10^6 m^3$.

5.2. Conclusions

The following conclusions were derived from this study:

1. Adaptive neuro-fuzzy inference system (ANFIS) and Artificial neural network (ANN) are adequate forecasting methods for predicting river discharge. However, ANFIS has a high predicting power than a stand-alone ANN.
2. Climate change is affecting Ikpoba river discharge as revealed by the seasonal variation in temperature and precipitation and should always be taken into account whenever the discharge prediction is being made.
3. Ikpoba dam reservoir is undergoing siltation as revealed by the increase in total annual runoff volume.

5.3. Recommendations

The following are the recommendations from the study:

1. The ANFIS and ANN network models developed should be applied for future prediction of the discharge data of Ikpoba River until more data are available.
2. River basins should be properly gauged so as to make discharge data available to the public needed in water resources management.
3. When input data sets for forecast contained effect of white of noise, BR training algorithm in ANN should always be used. This algorithm is very sensitive to effect of white noise which other algorithms could not detect.
4. When input data sets for forecast are very large, SCG training algorithm in ANN should be used because it is very fast in training large data set. LM and BR take longer time in training large input data sets.
5. River discharge records should be properly managed and kept by the basin/river authorities.

6. Where possible, pump storage scheme should be installed in the river to store excessive discharge flow during raining season (periods of high discharge) which could be turbined down for use during dry season (periods of low discharge).
7. Dredging should be carried out periodically in the reservoir to avoid siltation.

5.4. Contributions to Knowledge

The study contributed to knowledge in the following ways:

1. Artificial intelligence forecasting models of Ikpoba river discharge have be developed for use within Ikpoba river basin/catchment. This is useful for future forecast due to the poorly gauged nature of the catchment.
2. Detailed evaluation of the use of Ikpoba river dam for both water supply and hydropower generation has been carried out.

5.5. Recommendations for Future Research

The applications of ANFIS with hybrid algorithm and ANN with three training algorithms (LM, SCG and BR) were only investigated in this research. The application of ANFIS with backpropagation algorithm and subtractive clustering technique and ANN with other training algorithms (such as Conjugate Gradient Descent, Gradient Descent with momentum factor and others) should further be investigated with Ikpoba river discharge by future researchers. The potential of ANFIS and ANN in the following areas could also be investigated by future researchers:

1. Prediction and modelling of runoff flow in a Ikpoba river
2. Prediction and modelling of sediment loads transport in a Ikpoba river
3. Prediction and modelling of groundwater flow/infiltration in a Ikpoba river
4. Prediction and modelling of contaminant plumes in a Ikpoba river
5. Prediction and modelling of Ikpoba reservoir outflow from the storage
6. Detailed design of the hydropower plant to be used and its financial analysis should be further investigated.

REFERENCES

Akintola, J.O. (1986). Rainfall Distribution in Nigeria. 1892-1983. Impact Publisher, Ibadan.

Anctil, F. and Rat, A. (2005): Evaluation of neural network stream flow forecasting on 47 watersheds; *J. Hydrol. Eng.* **10** 85–88.

Anyata, B.U., Ezugwu, C.N. and Ekenta, E.O (2013): Estimation of the useful life of Ikpoba River, International Journal of Engineering Research & Technology (IJERT Vol. 2 Issue 8, August – 2013 ISSN: 2278- 0181)

Aqil, M., Kita, I., Yano, A. and Nishiyama, S. (2007) A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. *J. Hydrol*. **337**, 22–34.

Asefa, T., Kemblowski, M. McKee, M. and Khalil, A. (2006), Multi-time scale stream flow predictions: The support vector machines approach, J. Hydrol., 318(1–4), 7–16, doi:10.1016/j.jhydrol.2005.06.001.

Azamathulla, H.M.D. and Ghani, A. (2011) ANFIS-based approach for predicting the scour depth at culvert outlets. ASCE J Pipeline Syst Eng Pract 2(1):35–40

Babuška, R. and Verbruggen H (2003). Neuro-fuzzy methods for nonlinear system identification. Annu. Rev. Control, 27(1): 73-85.

Baraldi, A. N. and Enders, C. K. (2010). An introduction to modern missing data analyses. *Journal of School Psycholology 48*(1): 5-37.

Barry, R., (2000). Artificial Neural Network Prediction of Wavelet Sub-bands for Audio Compression, Thesis for Bachelor of Science, Dept. of Electrical & Computer Engineering, http://www.ilos.net/~rbarry/Thesis.pdf

Bezdek, J.C., (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms,* Plenum Press, New York,

Bilgili, M. (2010) Prediction of soil temperature using regression and artificial neural network models. Meteorol Atmos Phys 110: 59–70

Birikundavyi, S., Labib, R., Trung, H.T. and Roussele, J. (2002) Performance of Neural Network in Daily Streamflow Forecasting. J. Hydrol. Eng., ASCE, 7(5), 393-398.

Bishop, C.M., (1995a). Neural Networks for Pattern Recognition. Oxford University Press, New York.

Bishop, C.M., (1995b). "Training with noise is equivalent to Tikhonov regularization", Neural Computation, 7: 108-116.

Benin-Owena River Basin Authority, BORBDA (2005), "Benin Owena River Basin Development Authority Hydrological year Book", 1989 to 2008

Boucher, M. A., Perreault, L. and Anctil, F. (2009): Tools for the assessment of hydrological ensemble forecasts obtained by neural networks. J. Hydroinform. 11(3–4), 297–307.

Cassuto, A. E., and Ryan, S. (1979), Effect of price on the residential demand for water within an agency, Water Resour. Bull., 15(2), 345–353, doi:10.1111/j.1752-1688.1979.tb00337.

City of Montreal (2010), The Montreal Community Sustainable Development Plan 2010-2015, Montreal, QC.

Chang, F.J. and Chen, Y. C. (2001): A counter propagation fuzzy-neural network modeling approach to real time stream flow prediction. J. Hydrol. 245(1–4), 153–164.

Chang, F. J., Hu, H. F. and Chen, Y.C. (2001): Counter propagation fuzzy–neural network for river flow reconstruction, Hydrol. Process., 15, 219–232,

Chang, F. J., Chang, L. C. and Huang, H. L. (2002): Real-time recurrent learning neural network for stream-flow forecasting, Hydrol. Process., 16, 2577–2588,

Chang, L. C., Chang, F. J. and Sai, Y. H. T. (2005). Fuzzy exemplar-based inference system for flood forecasting. *Water Resource Res.*, 41.

Chang, F. J. and Chang, Y. T. (2006): Adaptive neuro-fuzzy inference system for prediction of water level in reservoir, Adv. Water Res., 29, 1–10,

Chantasut, N., Charoenjit, C. and Tanprasert, C. (2005). Predictive Mining of Rainfall Predictions Using Artificial Neural Networks for Chao Phraya River, *4th International Conference of The Asian Federation of Information Technology in Agriculture and The 2nd World Congress on Computers in Agriculture and Natural Resources*, August 9-12.

Chau, K. W., Wu, C. L. and Li, Y. S. (2005) Comparison of several flood forecasting models in Yangtze River. *J. Hydrol. Engng ASCE* **10**(6), 485–491.

Chen, S. H., Lin, Y. H., Chang, L. C. and Chang, F. J. (2006): The strategy of building a flood forecast model by neuro-fuzzy network, Hydrol. Process, 20, 1525–1540, Chen, P. W., Wang, J. Y. and Lee, H. M. (2004) Model selection of SVMs using GA approach. In: *2004 IEEE International Joint Conference on Neural Networks* vol. 3, 2035–2040. IEEE, New York, USA.

Chiu, S., (1994). "Fuzzy Model Identification Based on Cluster Estimation,"*Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3,

Cigizoglu, H. K. and Alp, M. (2006): Generalized regression neural network in modeling river sediment yield, Adv. Eng. Softw., 37, 63–68,

Cimen,M., and Kisi, O. (2009), Comparison of two different data driven techniques in modeling surface water level fluctuations of Lake Van, Turkey, J. Hydrol., 378(3–4), 253–262, doi:10.1016/j.jhydrol.2009.09.029.

Cogger, K. O. (2010), Nonlinear regression methods: A survey and extensions, Intell. Syst. Acc. Fin. Manage. 17, 19–39, doi:10.1002/isaf.311

Collins, L. M., Schafer, J. L. and Kam, C. M (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods 6*(4): 330-351.

Coppla, Jr., E.A. (2002). Fuzzy rule-based methodology for estimating monthly groundwater recharge in a temperature watershed, Journal of Hydrologic engineering, July-August. .

Coulibaly, P., Anctil, F. and Bobe B. (2000). Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. J. Hydrol. 230(3–4), 244–257.

Coulibaly, P., Anctil, F. and Bobee, B. (2001). Multivariate reservoir inflow forecasting using temporal neural networks. Journal of Hydrologic Engineering, 6(5): 367-376.

Coulibaly, P., Babovic, V., Cluckie, I., Mynett, A. and Ball, J. (eds) (2009) Advances in Hydroinformatics. IWA Publishing, London.

Dastorani, M.T., Moghadamnia, A.R., Piri J, and Rico-Ramirez, M (2010) Application of ANN and ANFIS models for reconstructing missing flow data. Environ Monit Assess 166:421–434

Dawson, W. (1998). An artificial neural network approach to rainfall-runoff modeling. Journal of Sciences-des Sciences Hydrologigues, 43(1): 47-66.

Dawson, C. W. and Wilby, R. (1998) "An artificial neural network approach to rainfall-runoff modelling". *Hydrological Sci. Journal.* 43 (1). pp. 47-66.

Dawson, C. W. and Wilby, R. L. (2001) Hydrological modelling using artificial neural networks. Prog. Phys. Geogr. 25(1), 80–108. Department for Environment, Food and Rural Affairs (DEFRA)/Environmental Agency (EA) 2005 The Social Performance of Flood Warning Communications Technologies. Technical Report W5C-016. Environment Agency, Bristol.

Dawson, C. W., See, L. M., Abrahart, R. J. and Heppenstall, A. J. (2006) Symbiotic adaptive neuro-evolution applied to rainfall–runoff modelling in northern England. Neural Networks 19(2), 236–247.

Dawson, C. W., Abrahart, R. J. and See, L. M. (2007) HydroTest: a web-based toolbox of evaluation metrics for the standardized assessment of hydrological forecasts. Environ. Model. Softw. 22(7), 1034–1052.

Dawson, C. W., Abrahart, R. J. and See, L. M. (2009). HydroTest: further development of a web resource for the standardised assessment of hydrological models. Environ. Model. Softw.

Deka, P. and Chandramouli, V. (2003) A fuzzy neural network model for deriving the river stage-discharge relationship. *Hydrol. Sci. J.* **48**(2), 197–209.

Dixon, B. (2005). “Applicability of neuro-fyzzy techniques in predicting ground-wateer vunerability: a GIS-based sensitivity analysis”. *Journal of Hydrology*. 309. pp.17-38.

Duan, Q., Sorooshian, S. and Gupta, V.K. (1992). Effective and efficient global optimization for conceptual rainfall runoff models. Water Resources Research. 28 (4):1015-1031.

Dubrovin, T., Jolma, A. and Turunen, E. (2002) Fuzzy model for real-time reservoir operation. *J. Water Resour. Plan. Manage*. **128**(1), 66–73.

Dubitzky, W; Granzow, M and Berrar, D (2007). Fundamentals of data mining in genomics and proteomics. Springer Science & Business Media. p. 178.

Edo State Urban Water Board, 2007.

Efron, B and Tibshirani, R. (1997). "Improvements on cross-validation: The .632 + Bootstrap Method". Journal of the American Statistical Association **92** (438): 548–560. doi:10.2307/2965703. JSTOR 2965703. MR 1467848.

Eekhout, I., Hoitink, A.J.F. and Vet, C. W. (2013). Missing data in a multi-item instrument were best handled by multiple imputations at the item score level. Journal of Clinical Epidemiology, 67(3), 335-42.

Ehiorobo, J.O. (2008). 'Robustness Analysis of a GPS Network for Earth Dam Deformation Monitoring'; Unpublished Ph.D Thesis, Department of Civil Engineering, University of Benin, Benin City, Nigeria.

Emiroglu, M., Bilhan, O. and Kisi, O. (2011), Neural networks for estimation of discharge capacity of triangular labyrinth side-weir located on a straight channel, Expert Syst. Appl., 38(1), 867–874, doi:10.1016/j.eswa.2010.07.058.

Enders, C. K. (2010). Applied Missing Data Analysis. New York, NY, The Guilford Press.

Er, M.J. and Sun, Y.L. (2001) "Hybrid Fuzzy Proportional Integral Plus Conventional Derivative Control of Linear and Nonlinear Systems", IEEE Trans. On Industrial Electronics, Vol.48, No.6 pp. 1109-1117.

Erinawati, R. and Fenton, J. (2012). Short-term Water Level Prediction Using Fuzzy Adaptive System and Artificial Neural Networks Approaches. 1799-1805.

Feng, L., and Hong, W. (2007). The practical application of fuzzy information analysis in flood forecasting. *ICCS 2007*, 1056-1061.

Fernando, T.M.K.G, Sivakumar, B and Jayawardena, A.W (2002) River flow forecasting –use of phase-space reconstruction and artificial neural network approaches. J.Hydrol. 265, 225-245.

Fernando, A. and Shamseldin, A. Y. (2009): Investigation of the internal functioning of the radial basis function neural network river flow forecasting models. J. Hydrol. Eng. 14(3), 1–7.

Firat, M. and Gungor, M. (2007): River flow estimation using adaptive neuro fuzzy inference system, Math. Comput. Simulat., 75(3–4), 87–96, Firat, M. (2007): Watershed modeling by adaptive Neuro- fuzzy inference system approach. Doctor of Philosophy Thesis, Pamukkale University, Turkey, (in Turkish).

Foresee, F.D and Hagan, M.T (1997). Proceedings of the International Joint Conference on Neural Networks.

From Wikipedia, the free encyclopedia: (www.wds.worldbank.org/external/defal.).

Gallant, S.I., (1993). Neural Network Learning and Expert Systems. MIT Press, Cambridge.

Geisser, S (1993). Predictive Inference. New York, NY: Chapman and Hall. ISBN 0-412-03471-9.

Gowda, C.C. and Moyya, S.D. (2014). Runoff Modelling using different member ship functions in Adaptive Neuro fuzzy inference system. International Journal of Advances in Engineering Sciences.4(4):48-51

Goswami, M., O'Connor, K. M., Bhattarai, K. P. and Shamseldin, A. Y. (2005) Real-time river flow forecasting for the Brosna catchment in Ireland using eight updating models. J. Hydrol. Earth Syst. Sci. 9, 394–411.

Grossman, R; Seni, G; Elder, J; Agarwal, N; and Liu, H (2010). Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. Morgan and Claypool. doi:10.2200/S00240ED1V01Y200912DMK002.

Haddad, O.B., Sharife, F. and Alimohammadi, S. (2005). ANN in river flow forecasting . Proceedings of the 6th WSEA Int. Conf. on Evolutionary Computing, Lisbon, Portugal, pp: 316-324.

Hagan, M. T., and Menhaj, M. (1994), Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks, 5(6), 989–993, doi:10.1109/72.329697.

Hagan, M.T., Demuth, H.B. and Beale, M.H. (1996), Neural Network Design, Boston, MA: PWS Publishing.

Han, D., Chan, L. and Zhu, N. (2007), Flood forecasting using support vector machines, Journal of Hydroinformatics, 9(4), 267–276, doi:10.2166/hydro.2007.027.

Hansen, L.K., and Salamon, P. (1990). Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12: 993-1001

Haykin, S. (2009). Neural Networks and Learning Machines. 3rd edition, Pearson Education, Inc.,New Jersey.

Hong, Y. S., Rosen, M. R. and Reeves, R. R. (2002). Dynamic fuzzy modeling of storm water infiltration in urban fractured aquifers. *J. Hydroloiy. Eng.* 7(5):380–391.

Hughes, T. (1980), Peak period design standards for small western U.S. water supply, Water Resour. Bull., 16(4), 661–667, doi:10.1111/j.1752-1688.1980.tb02446.

Ivakhnenko, A. G. (1970), Heuristic self-organization in problems of engineering cybernetics, Automatica, 6(2), 207–219, doi:10.1016/0005-1098(70)90092-0.

Jain, S. K. and Srivastva, D. K. (2005). Application of ANN for reservoir in flow prediction and operation. *J Water Resources Planning and Manage*. 125: 23-271.

Jang, J.S.R. and Sun, C.T. (1995). "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence", Prentice-Hall, USA,

Karul, C., Soyupak, S. Cilesiz, A. F. Akbay, N. and Germen, E. (2000), Case studies on the use of neural networks in eutrophication modeling, Ecol. Model., 134(2–3), 145–152, doi:10.1016/S0304-3800(00)00360-4.

Karunanithi, N., Grenney, W.J., Whitley, D. and Bovee, K., (1994). Neural Networks for River Flow Prediction", J. Comp. Civil Eng., ASCE 8, 201{220,

Khan, M. S., and P. Coulibaly (2006). Application of support vector machine in lake water level prediction, J. Hydrol. Eng., 11(3), 199–205, doi:10.1061/(ASCE)1084-0699(2006)11:3(199).

Kisi, O. (2004) River flow modeling using artificial neural network. ASCE J Hydrol Eng 9(1):60–63

Kisi, O., (2005). Daily river flow forecasting using artificial neural networks and auto-regressive models. Turkish J.Eng. Sci, 29(2005): 9-20.

Kisi, O. (2007) Stream flow forecasting using different artificial neural network algorithms. ASCE J Hydrol Eng 12(5):532–539

Kisi, O. (2008) River flow forecasting and estimation using different artificial neural network techniques. Hydrol Res 39(1): 27–40

Kisi O, Nia, A.M, Gosheh M.G, Tajabadi M.R.J, and Ahmadi, A (2012) Intermittent stream flow forecasting by using several data driven techniques. Water Resour Manag 26(2):457–474

Kisi, O. (2008), Stream flow forecasting using neuro-wavelet technique, Hydrol. Processes, 22, 4142–4152, doi:10.1002/hyp.7014.

Kisi, O. (2009), Neural networks and wavelet conjunction model for intermittent streamflow forecasting, J. Hydrol. Eng., 14(8), 773–782, doi:10.1061/(ASCE)HE.1943-5584.0000053

Kisi, O. and Cimen, M. (2011), A wavelet-support vector machine conjunction model for monthly streamflow forecasting, J.Hydrol., 399(1–2), 132–140, doi:10.1016/j.jhydrol.2010.12.041.

Kisi, O and Uncuoghlu, E (2005), “Comparison of three backpropagation training algorithms for two case studies,” Indian Journal of Engineering & Materials Sciences, Vol. 12, pp 434-442

Khan, E. (1999) “Neural fuzzy based intelligent systems and application”. *Fusion of Neural Networks, Fuzzy Sets, and Genitic Algorithms:Industrial Applications.* CRC Press, Washington. pp.331.

Kohavi, R. (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (San Mateo, CA: Morgan Kaufmann) **2** (12): 1137–1143. CiteSeerX: 10.1.1.48.529.

Kosko, B. (1999). “Fuzzy systems as universal approximators”. *Proc. IEEEInt.Conf.Fuzzy Sys.*pp.1153-1162.

Kumar A.P.S., Sudheer, K.P., Jain, S.K and Agarwal, P.K. (2005) Rainfall runoff modeling using artificial neural networks comparison of network types. Hydrol Process 19:1277–1291

Leahy, P., Kiely, G. and Corcoran, G. (2008): Structural optimization and input selection of an artificial neural network for river level prediction. J. Hydrology, 355 192-201.

Legates, D. R., and McCabe Jr., G. J. (1999), Evaluating the Use of "Goodness-of-Fit" Measures in Hydrologic and Hydroclimatic Model Validation, Water Resour. Res., 35(1), 233–241

Li, G. and Shi, J. (2012) “Applications of Bayesian methods in wind energy conversion systems”, Renewable Energy, Vol. 43, pp 1-8

Lin, B. and Namin, M. M. (2005): "Modelling suspended sediment transport using an integrated numerical and ANNs model". *J. Hydraul. Res.*, 43(3).. pp.302-310.

Liong, S. Y., Lim, W. H., Kojiri, T. and Hori, T. (1995.): Advance Flood forecasting for Flood stricken Bangladesh with a fuzzy reasoning method, Hydrol. Process. 14, 431–448, 2000.

Lorrai, M. and Sechi, G.M. Neural Nets for Modelling Rainfall-Runoff Transformations, In: *Water resources management*, Vol. 9, pp. 299-313,

Lu, W., Wang, W. Leung, A. Y. T., Yuen, K. K. Xuand, Z. and Fan, H. (2002), Air pollutant parameter forecasting using support vector machines, Neural Networks Proceedings of the 2002 International Joint Conference, 1, 630–635, doi:10.1109/IJCNN.2002.1005545.

Luger, G.F., and Stubblefield, W.A. (1993). Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 2nd Edition, Benjamin/Cumming Publishing, Redwood City, California.

MacKay, D.J.C. (1992), Neural Computation, Vol. 4, No. 3, pp. 415–447

Maier, H. R. and Dandy, G. C. (2000): Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environ. Model. Softw. 15, 101–124.

Maier, H. A., Jain, G. C. and Dandy, K. P. S (2010), Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions, Environ. Model. Software, 25, 891–909

Mahabir, C., Hicks, F. E. and Fayek, A. R. (2000): Application of fuzzy logic to the seasonal runoff, Hydrol. Process. 17, 3749–3762, Mahabir, C., Hicks, F. E. and Fayek, A.R. (2003) Application of fuzzy logic to forecast seasonal runoff. *Hydrol. Process*es **17**(18), 3749–3762.

Maidment, D. and Miaou, S. (1986), Daily water use in nine cities, Water Resour. Res., 22(6), 845–851, doi:10.1029/WR022i006p00845.

Maidment, D. and Parzen, E. (1984), Monthly water use and its relationship to climatic variables in Texas, Water Resour. Bull., 19(8), 409–418.

Maidment, D., Miaou, S and Crawford, M. (1985), Transfer function models of daily urban water use, Water Resour. Res., 21(4), 425–432, doi:10.1029/WR021i004p00425.

Maier, H. R. and Dandy, G. C. (1996). The use of artificial neural networks for the prediction of water quality parameters. Water Resources Research. 32:1013-1022.

Maier, H. R. and Dandy, G. C. (1997). Determining inputs for neural network models of multivariate time series. Microcomputers in Civil Engg. 12:353-368.

Maity, R., Bhagwat, P. P. and Bhatnagar, A. (2010), Potential of support vector regression for prediction of monthly streamflow using endogenous property, Hydrol. Processes, 24(7), 917–923, doi:10.1002/hyp.7535.

Marquardt, D.W. (1963). An Algorithm for the Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal of Applied Mathematics*, 11(2):431– 441.

Marsh, T.J. (2002). Capitalising on river flow data to meet changing national needs - a UK perspective. *Flow Measurement and Instrumentation*, **13**, 291-298.

Maskey, S., Guinot, V. and Price, R. K., (2004). Treatment of precipitation uncertainty in rainfall-runoff modeling: A fuzzy set approach. Adv. Water Resour., 27(9):889–898.

Masters, T. (1993): Practical Neural Networks Recipes in Cþþ. Academic, New York.

McCulloch, W.S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, 5: 115-133.

Mehta, R and Jain, S. (2009). Optimal operation of a multipurpose reservoir using neuro-fuzzy technique. Water Res. Manage., 23(3): 509-529.

Miaou, S. (1990). A class of time series urban water demand models with non-linear climatic effects, Water Resour. Res., 26, 169–178.

Mitra B., Scott, H.D., Dixon, J.C. and Mckimmey, J.M. (1998). Application of fuzzy logic to the prediction of soil erosion in a large watershed. Geoderma, 86: 183-209

Miyagishi, K., Ohsako, M. and Ichihashi, H. (1999), Temperature prediction from regional spectral model by neurofuzzy GMDH, Proc. of The Second Asia-Pacific Conference on Industrial Engineering and Management Systems, Kanazawa, Japan, 705–708.

Møller, M. F. (1993) "A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks," Volume 6, Issue 4, Pages 525-533

Moghaddamnia, A., Ghafari Gousheh, M., Piri J., Amin S. and Han, D. (2009) Evaporation estimation using artificial neural networks and adaptive neuro-fuzzy inference system technique. Adv Water Resour 32(1):88–97

Msiza, I. S., Nelwamondo, F. V. and Marwala, T. (2008), Water demand prediction using artificial neural networks and support vector regression, J. Comput., 3(11), 1–8, doi:10.4304/jcp.3.11.1-8.

Mukarji, A., Chandranath, C. and Narendra, S. R. (2009). Flood Forecasting Using ANN, Neuro-Fuzzy, and Neuro-GA Models. *J of Hydrologic Engineering*, 14(6):647-652.

Mutlu, E., Chaubey, I., Hexmoor, H. and Bajwa, S.G. (2008) Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. Hydrol Process 22(26):5097–5106

Nagy, H. M., Watanabe, K. and Hirano, M. (2002) Prediction of sediment load concentration in rivers using artificial neural network models, Journal of Hydro. Eng, 128, 588-595

Nayak, P. C., Sudheer, K. P., and Ramasastri, K. S. (2004a): Fuzzy computing based rainfall-runoff model for real time flood forecasting, Hydrol. Process., 17, 3749–3762.

Nayak, P. C., Sudheer, K. P., Rangan, D. M, Ramasastri, K. S. (2004b): A Neuro Fuzzy computing technique for modeling hydrological time series, J. Hydrol., 29, 52–66.

Nayak, P. C., Sudheer, K. P., Rangan, D. M. and Ramasastri, K. S. (2004) A neuro-fuzzy computing technique for modeling hydrological time series. *J. Hydrol*. **291**, 52–66.

Nayak, P. C., Sudheer, K. P., Rangan, D. M. & Ramasastri, K. S. (2005) Short-term flood forecasting with a neurofuzzy model. *Water Resour. Res.* **41**, 2517–2530.

Nayak, P. C., Sudheer, K. P. and Ramasastri, K. S. (2005). Fuzzy computing based rainfall–runoff model for real time flood forecasting. *J of Hydrologic process*. 9: 955-968.

Owen, J. S., Eccles, B. J., Choo, B. S. and Woodings, M. A. (2001): The application of auto–regressive time series modeling for the time frequency analysis of civil engineering structures, Eng. Struct., 23, 521–536.

Pan, X., Lee, T.L. Chunrong, B. (2013) "A comparison of neural network backpropagation algorithms for electricity load forecasting," Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on , vol., no., pp.22,27, 14-14.

Partal, T. (2009), River flow forecasting using different artificial neural network algorithms and wavelet transform, Can. J. Civil Eng., 36(1), 26–38, doi: 10.1139/L08-090.

Parveen, S., Sangeeta, G. and Dharminder, K., (2011), *"*A Study of Prediction Modeling Using Multilayer Perceptron (MLP) With Error Back Propagation*",* Proceedings of AICTE sponsored National Conference on Knowledge Discovery & Network Security: pp. 17-20.

Parveen, S., Sangeeta, G. and Dharminder, K., (2012), "Minimization of Error in Training a Neural Network Using Gradient Descent Method", International Journal of Technical Research, Vol. 1,Issue 1, pp.10-12

Pongracz, R., Bogardi, I.and Duckstein, L. (1999) Application of fuzzy rule-based modeling technique to regional drought. *J. Hydrol.* **224**, 100–114.

Pramanik, N. and Panda, R. K. (2009): Application of neural network and adaptive neuro-fuzzy inference systems for river flow prediction. Hydrol. Sci. J. 54(2), 247–260.

Pulido-Calvo, I., Portela, M.M., (2007): Application of neural approaches to one-step daily flow forecasting in Portugues watersheds. J. Hydrology 332 1-15.

Raghuwanshi1, N. S., R. Singh. and Reddy, L. S. (2006). Runoff and Sediment Yield Modeling Using Artificial Neural Networks: Upper Siwane River, India. *J. Hydrol. Eng*.11:71-79.

Rajasekaran, S., Gayathri, S. and Lee, T.L. (2008), Support vector regression methodology for storm surge predictions, Ocean Eng., 35, 1578–1587, doi:10.1016/j.oceaneng.2008.08.004.

Raman, H. and Sunilkumar, N. (1995). Multivariate modeling of water resources time series using artificial neural networks. J. Hydrol. Sci. 40:145-163.

Ram´ırez, M. C. V, Velho, H. F. C and Ferreira, N. J (2005) Artificial neural network technique for rainfall forecasting applied to the S˜ao Paulo region; *J. Hydrol.* **301** 146–162.

Reed, R.D., and Marks II, R.J., (1999). Neural Smithing, MIT Press, Cambridge, Mass.

Ren, Q., Baron, L. and Balazinski, M. (2006,). Type-2 Takagi-Sugeno-Kang fuzzy logic modeling using subtractive clustering. Proceeding of Annual meeting of the North American fuzzy information processing society, June. 3-6 Montreal, Canada, pp. 120-125.

Rezaeianzadeh, M., Amin, S., Khalili, D. and Singh, V.P. (2010) Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions. Water Resour Manag 24(11):2673–2688

Rezaeianzadeh, M., Stein, A., Tabari, H., Abghari, H., Jalalkamali, N., Hosseinipour, E.Z. and Singh, V.P. (2013) Assessment of a conceptual hydrological model and artificial neural networks for daily outflows forecasting. Int J Environ Sci Technol. doi:10.1007/s13762-013-0209-0

Rivard, H., Yang, J., and Zmeureanu, R. (2005). Building Energy Predication with Adaptive Artificial Neural Networks, *Ninth International IBPSA Conference Montréal*, August 15- 18. www.intechopen.

Rojas, R., (1996). Neural Networks - A Systematic Introduction, Chapter 7, The Backpropagation Algorithm, http://www.inf.fu-berlin.de/~rojas/neural/chap7.ps

Rubin, D. (2005), Down the drain: Dealing with Montreal's water wastage, McGill Reporter, 37(17), Montreal, QC.

Rubin, D. B. (1976). Inference and missing data. *Biometrika 63*(3): 581-592.

Rubin, D. B. (1987). Multiple Imputation for Nonresponse in Surveys. New York, J. Wiley & Sons.

Russell, S. O. and Campbell, P. F. (1996) Reservoir operating rules with fuzzy logic programming. *J. Water Resour. Plan. Manage*. **122**(4), 262–269.

Sabziparvar, A.A. and Tabari, H. (2010) Comparison of artificial neural network models and non-linear regression methods for estimation of potato crop evapotranspiration in a semi-arid region of Iran. In: The international conference on intelligent network and computing, Nov 26–28, Kuala Lumpur, Malaysia

South Africa Development Community, SADC (2011): Dam Synchronisation and Flood Releases in the Zambezi River Basin Project. Final Report. Transboundary Water Management.

Sahooa, G.B., Raya, C. and Wadeb, H.F (2005): Pesticide prediction in ground water in North Carolina domestic wells using artificial neural networks. Ecological Modelling 183 29–46

Samani, N.M., Gohari, M. *and* Safavi, A. A. (2007). A simple neural network model for the determination of aquifer parameters, *J. Hydrol.*, 340, 1–11, *doi:*10.1016/j.jhydrol.2007.03.017. Cross Ref, Web of Science. Times Cited: 22, ADS

Sarycheva, L. (2003), Using GMDH in ecological and socio-economical monitoring problems, Syst. Anal. Model. Simul., 43(10), 1409–1414, doi:10.1080/02329290290024925.

Schafer, J. L. (1997). Analysis of Incomplete Multivariate Data. London, UK, Chapman & Hall.

Schafer, J. and Hox, J.J. (1999), Software for Multiple Imputation :A Review of Current Software for Handling Missing Data, Kwantitatieve Methoden, 62, 123-138.

Şen, Z. (1998) Fuzzy algorithm for estimation irradiation from sunshine duration. *Solar Energy* **63**(1), 39–49.

Sen, Z. (2001): Fuzzy Logic and Foundation", ISBN 9758509233, Bilge K¨ult¨ur Sanat Publisher, Istabul, 172 pp.

Senthil Kumar, A. R., Ojha, C. S. P., Singh, R. D. and Swamee, P. K. (2012). Modeling of Suspended Sediment Concentration at Kasol in India Using ANN, Fuzzy Logic, and Decision Tree Algorithm. *J. of Hydrologic Engineering*.@ ASAE:295-403.

Shamseldin, A. Y., Nasr, A. E., and O'Connor, K. M. (2002) Comparison of different forms of the Multi-layer Feed-Forward Neural Network method used for river flow forecast combination. J. Hydrol. Earth Syst. Sci. 6(4), 671–684.

Shamseldin, A. Y. (2006). Topics related to rainfall–runoff models. In River Basin Modelling for Flood Risk Mitigation (ed. D. W. Knight and A. Y. Shamseldin). Taylor and Francis, London, Chapter 10.

Shamseldin, A. Y., O'Connor, K. M. and Nasr, A. E. (2007). A comparative study of three neural network forecast combination methods for river flow forecasting. Hydrol. Sci. J. 55(2), 898–916.

Shiri, J., and Kisi, O. (2010), Short-term and long-term streamflow forecasting using a wavelet and neuro-fuzzy conjunction model, J. Hydrol., 394(3–4), 486–493.

Shouyu, C. and Guangtao, F. (2003) A DRASTIC-based fuzzy pattern recognition methodology for groundwater vulnerability evaluation. *Hydrol. Sci. J.* **48**(2), 211–220.

Shivakuma, B., Jayawardhane, A. W. and Fernando, T. M. K. G. (2002). River flow forecasting: use of phase space reconstruction and artificial neural network approaches. 265: 225-245.

Shrestha, R.R. (2003). Flood Routing using Artificial Neural Network in: proc.IAHR Congress, Thessaloniki, Greece.

Singh, P.and Deo, M.C. (2007). Suitability of different neural networks in daily flow forecasting. Appl. Soft Comput. J., 7(3): 968-978.

Sinha Jitendra, R. K., Sahu, A. A., Pali, A. K. and Sinha, B. L. (2013). Rainfall runoff modeling using multi layer Perceptron Technique – A case study of the Upper Kharun Catchments in Chhattishgarh. *J of Agricultural Engineering*, 50(2): 43-51.

Smith, J. (1988), A model of daily municipal water use for short-term forecasting, Water Resour. Res., 24(2), 201–206, doi: 10.1029/WR024i002p 00201.

Solomatine, D., and Ostfeld, A. (2008), Data driven modeling: Some past experiences and new approaches, J. Hydroinf., 10(1), 3–22.

Solomatine, D.P. and Dibike, Y.B., (2001) Neural network approximation of a hydrodynamic model in optimizing reservoir operation, In:*Proceedings of the second international conference on hydroinformatics*, Zurich. pp. 201-206.

Sorooshian, S., Duan, Q. and Gupta, V.K. (1993). Calibration of rainfall-runoff models: Application of global optimization to the Sacramento soil moisture accounting model. Water Resources Research. 29(4):1185-1194.

Strandhagen, E., Marcus, W.A., and Meacham, J.E. (2006). Views of the rivers: representing streamflow of the greater Yellowstone ecosystem (hotlink to http://geography.uoregon.edu/amarcus/Publications/Strandhagen-et-al_2006_Cart_Pers.pdf). Cartographic Perspectives, no. 55, Fall.

Swain, P. and Umamahesh, N. (2004). Stream flow forecasting using neuro-fuzzy inference system. Indian J. Power River Valley Dev., 53(5/6): 81-85.

Tabari, H., Marofi, S., Abyaneh, H.Z., Sharifi, M.R. (2010) Comparison of artificial neural network and combined models in estimating spatial distribution of snow depth and snow water equivalent in Samsami basin of Iran. Neural Comput Appl 19:625–635

Tabari, H., Marofi, S. and Sabziparvar, A.A. (2010) Estimation of daily pan evaporation using artificial neural network and multivariate non-linear regression. Irrig Sci 28:399–406

Tabari, H., Sabziparvar, A.A. and Ahmadi, M. (2011) Comparison of artificial neural network and multivariate linear regression methods for estimation of daily soil temperature in an arid region. Meteorol Atmos Phys 110:135–142

Tayfur, G., Özdemir, S. and Singh, V. P. (2003) Fuzzy logic algorithm for runoff-induced sediment transport from bare soil surfaces. *Adv. Water Resour.* **26**, 1249–1256.

Tayfur, G., Moramorco, T. and Singh, V. P. (2003). Predicting and forecasting flow discharge at sites receiving significant lateral inflow; *Hydrol. Process*. 21 1848–1859.

Tayfur, G and Singh., V P (2006). ANN and fuzzy logic models for simulating event-based rainfall–runoff; *J. Hydrol. Engg*. 132 1321–1330.

Tilmant, A., Vanclooster, M., Duckstein, L, and Persoons, E. (2002). Comparison of fuzzy and nonfuzzy optimal reservoir operating policies. *J. Water Resour. Plann. Manage*. 128(6): 390–398.

Tokar, A.S. and Johnson, P.A. (1999). Rainfall-runoff modeling using artificial neural networks. J. Hydrologic Engg., ASCE. 4(3):232-239

Tütmez, B. and Tercan, A.E. (2007), "Assessment of Uncertainty in Geological Sites Based on Data Clustering and Conditional Probabilities", Journal of Uncertain Systems, 1, 3, 206-221

van Buuren, S. (2012). Flexible Imputation of Missing data. New York, Chapman & Hall/CRC.

Varma, S and Simon, R (2006). "Bias in error estimation when using cross-validation for model selection". BMC Bioinformatics **7**: 91. doi:10.1186/1471-2105-7-91. PMC 1397873. PMID 16504092.

Vitkovsky´, J.P., Simpson, A.R., Lambert, M.F., M.ASCE, and Liggett, J. A. (2007). Experimental Observation and Analysis of Inverse Transients for Pipeline Leak Detection. *Journal of water resources planning and management*, ASCE, 0733-9496.

Wang, Y. M, Kerh, T. and Traore, S. (2009) Neural network approach for modelling river suspended sediment concentration due to tropical storms; *J. Global Nest* **11** 457–466.

Wang, W. C., Chau, K. W., Cheng, C. T. and Qiu, L. (2009). A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series, J. Hydrol. 374(3–4), 294–306.

Wang, Y .M, Traore, S., Kerh, T. and Leu, J .M. (2011) Modelling reference evapotranspiration using feed forward back propagation algorithm in arid regions of Africa; *Irrigation Drainage* **60** 404–417.

Wang, W., and Ding, J. (2003), Wavelet network model and its application to the prediction of hydrology, Nat. Sci., 1(1), 67–71.

World Commission on Dams, WCD (2000): Dams and Development: A New Framework for Decision-making. Earthscan Publications Ltd., London

Willmott, C. J. (1981). On the validation of models. Physical Geography, 2, 184–194

Willmott, C. J. (1984). On the evaluation of model performance in physical geography. Spatial Statistics and Models, G. L. Gaile and C. J. Willmott, eds., 443-460

Wong, J. S., Zhang, Q. and Chen, Y. D. (2010), Statistical modeling of daily urban water consumption in Hong Kong: Trend, changing patterns,and forecast, Water Resour. Res., 46, W03506, doi:10.1029/2009WR 008147

Wu, J .S., Han, J., Annambhotla, S. and Bryant, S. (2005) Artificial neural networks for forecasting watershed runoff and stream flows; *J. Hydrol. Eng.* **10** 216–222.

Wu, J. C. and Liu, T. S. (1996)"A Sliding-Mode Approach to Fuzzy Control Design", IEEE Trans. On Control Systems Technology, Vol.4, No.2, pp. 141-151.

Yeshewatesfa, H., Bardossey, A. and Hanse, W.T. (2001). Development of fuzzy logic based rainfall runoff model. *J. of Hydrological Sciences*. 46(3):363-376.

Ying, B., Hanqi, Z. and Zvi. S.R, (2005) "Fuzzy Logic Control to Suppress Noises and Coupling Effects in a Laser Tracking System", *IEEE Trans on Control SystemsTechnology*, Vol.13, No.1, pp.113-121.

Yonaba, H., Anctil, F. and Fortin, V. (2010) Comparing sigmoid transfer functions for neural network multistep ahead stream flow forecasting. ASCE J Hydrol Eng 15(4):275–283

Yu, X. Y., Liong, S. Y. & Babovic, V. (2004) EC-SVM approach for real-time hydrologic forecasting. *J. Hydroinf.* **6**(3), 209–233.

Yu, P. S., and Chen, S. T. (2005). Updating real-time flood forecasting using a fuzzy rule-based model. *Hydrological Sciences-Journal-des Science Hydrologiques, 50*(2), 256-278.

Yu, P.S., Chen, S.T. and Chang, I.F. (2006), Support vector regression for real-time flood stage forecasting, J. Hydrol., 328(3–4), 704–716, doi:10.1016/j.jhydrol.2006.01.021.

Yue, Z., Zhao, S; Liu, T, (2011) "Bayesian regularization BP Neural Network model for predicting oil-gas drilling cost," Business Management and Electronic Information (BMEI), 2011 International Conference on , vol.2, no., pp.483,487.

Yurdusev, M.and Firat, M. (2009). Adaptive neuro fuzzy inference system approach for municipal water consumption modeling: An application to Izmir, Turkey. J. Hydrol., 365(3-4): 225-234.

Zadeh, L. A. (1965) Fuzzy sets. *Information and Control* **8**, 338–353.

Zhou, S., McMahon, T. Walton, A. and Lewis, J. (2000), Forecasting daily urban water demand: A case study of Melbourne, J. Hydrol., 236(3), 153–164, doi:10.1016/S0022-1694(00)00287-0

Zounemat-Kermani, M. and Teshnehlab, M. (2008). Using adaptive neuro-fuzzy inference system for hydrological time series prediction. Appl. Soft Comput. J. 8(2): 928-936.

(http://water.usgs.gov/edu/streamflow3.html)

(http://www.vanguardngr.com/2013/04/river-basins-how-unending-policy-reversals-abet-inefficiency/).

(*http://www.nigeria.gov.ng/fed_min_water_resources.aspx).*

http://www.s-cool.co.uk/a-level/geography/river-profiles/revise-it/storm-hydrographs-and-river-discharge