PREDICTIVE MODELLING OF KALITI WASTEWATER TREATMENT PLANT PERFORMANCE USING ARTIFICIAL NEURAL NETWORKS

A thesis submitted to the school of graduate studies of Addis Ababa University in partial fulfillment of the requirements for the degree of Master of Science in Chemical Engineering with Specialization in Environmental Engineering

By

Getnet Sewnet Kassahun

Advisor

Dr.-Ing Berhanu Assefa

ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FEBRUARY 2012

Approval Sheet

Thesis entitled **Predictive Modelling of kaliti Wastewater Treatment Plant Performance Using Artificial Neural Networks** by **Getnet Sewnet Kassahun** is approved for the degree of Master of Science, in Chemical Engineering.

Approved by Board of Examiners

_____	_____	_____
Chairman, Dept's Graduate Committee	Signature	Date
Dr.-Ing Berhanu Assefa	_____	_____
Advisor	Signature	Date
Dr.-Ing Agizew Nigusie	_____	_____
External Examiner	Signature	Date
Dr.-Ing Zebene Kifle	_____	_____
Internal Examiner	Signature	Date

Place: Addis Ababa University, Addis Ababa Institute of Technology (AAU-AAiT)

Abstract

Artificial neural networks are a form of artificial intelligence that have the capability of learning, growing, and adapting within dynamic environments. A reliable model for any wastewater treatment plant is essential in order to provide a tool for predicting its performance and to form a basis for controlling the operation of the process. This would enable to assess the stability of environmental balance at minimized operational costs.

Wastewater treatment process is complex and attains a high degree of nonlinearity due to the presence of bio-organic constituents that are difficult to model using mechanistic approaches. Predicting the plant operational parameters using conventional experimental techniques is also a time consuming step and is an obstacle in the way of efficient control of such processes. In this work, a soft computing approach based on back propagation artificial neural networks, which employed genetic algorithm and partial mutual information to perform input selection, was used to acquire the knowledge base of Kaliti wastewater treatment plant and has been applied for predicting and optimizing selected plant performance variables viz. pH, BOD_5, COD, NH_3, and TDS effluent concentration of the plant.

In the model structure of the treatment plant performance, two different functional structures in the configuration of the network are constructed and compared. In the first configuration Multiple-Input-Single-Output (MISO), structures differing in the type of data used, raw operational data and outlier removed data in the input layer, are used to build models for each of the five performance indicators selected in this work. Partial Mutual Information-based Input Selection (PMIS) and Genetic Algorithm (GA) based input selection algorithms are applied for both above mentioned MISO configuration based models. In the second configuration Multiple-Input-Multiple-Output (MIMO), GA based input selection is applied for both the raw and outlier removed data. The model performance was evaluated with statistical parameters and the simulation results indicates that the MISO modelling approach achieves much more accurate predictions as compared with the MIMO modelling approach.

Optimum model architecture of 14-43-1 for pH, 16-29-1 for BOD_5, 14-50-1 for COD, 6-28-1 for NH_3, and 10-48-1 for TDS were selected for predicting the performance of Kaliti wastewater treatment plant using MISO configuration. The linear correlation between predicted outputs and target outputs for the optimum model architecture described above are 0.97 for pH, 0.94 for BOD_5, 0.98 for COD, 0.94 for NH_3, and 0.98 for TDS.

Keywords: Artificial neural networks; Back propagation; Genetic algorithm; Partial mutual information; Wastewater treatment

Acknowledgments

It is with great humbleness that I would like to acknowledge my Good Lord as the true creator of all knowledge. This work is the result of my borrowing a small piece of knowledge from Him. A work such as this which involves the expenditure of large amounts of time and effort can never be accomplished through the labors of a single individual. As I sit down to complete this thesis, it is difficult to find the words to adequately express appreciation for the contributions of the many people who have made the completion of the task possible.

Special thanks go to Dr.-Ing Berhanu Assefa, my thesis supervisor. His optimism and ability to put things into perspective has been a great source of inspiration. I thank him for his confidence in my ability to meet higher standard of scholarly research and writing, which was meant much to me. I wish to express my deep sense of gratitude to Addis Ababa Water and Sewerage Authority (AAWSA) for their kind help in providing the necessary data for the thesis ,and to Horn of Africa Regional Environment Center and Network (HoA-REC/N) for supporting the thesis financially. Thanks also to all the authors of the articles listed in the bibliography of this thesis.

I would very much like to acknowledge the encouragement, patience and support provided by my family and friends of mine who have also shared in all the pain, frustration, and fun of producing the thesis. May God Bless You All!

Finally, I would like to dedicate this thesis to all those who have sacrificed things in their own lives to accommodate things in mine.

Getnet S. Kassahun

Addis Ababa, Ethiopia

Table of Contents

ABSTRACT I

ACKNOWLEDGMENTS II

TABLE OF CONTENTS III

LIST OF FIGURES V

LIST OF TABLES VII

LIST OF ACRONYMS VIII

CHAPTER 1- INTRODUCTION 1

1.1 BACKGROUND 1

1.2 STATEMENT OF THE PROBLEM 3

1.3 OBJECTIVES 4

1.3.1 General Objective 4

1.3.2 Specific Objectives 4

1.4 ORGANIZATION OF THE THESIS 5

1.5 UNIQUENESS OF THE THESIS 5

CHAPTER 2- LITERATURE REVIEW 6

2.1 ARTIFICIAL NEURAL NETWORKS FUNDAMENTALS 6

2.1.1 Biological Neuron 6

2.1.2 Artificial Neuron 9

2.1.3 Architecture and Elements of Artificial Neural Network 10

2.2 BACK PROPAGATION ARTIFICIAL NEURAL NETWORKS 14

2.2.1 Back propagation Algorithm 15

2.3 APPLICATION OF NEURAL NETWORKS 17

2.3.1 Pattern Classification 17

2.3.2 Clustering 17

2.3.3 Function Approximation 19

2.3.4 Forecasting 20

2.3.5 Optimization 20

2.3.6 Association 20

2.3.7 Control 20

2.4 ARTIFICIAL NEURAL NETWORK MODEL DEVELOPMENT 20

CHAPTER 3- DESCRIPTION OF THE WASTEWATER TREATMENT PLANT 24

3.1 DESCRIPTION OF THE STUDY AREA 24
3.2 DESCRIPTION OF THE WASTEWATER TREATMENT PROCESS 25
3.3 DESCRIPTION OF THE NIGHT SOIL TREATMENT 29
3.4 EFFLUENT GUIDELINES AND STANDARDS 29

CHAPTER 4- MATERIALS AND METHODS 31

4.1 MATERIALS 31
4.1.1 Historical Data 31
4.1.2 Software 32
4.2 METHODS 33
4.2.1 Selection of Appropriate Model Outputs 33
4.2.2 Data Pre-processing 33
4.2.3 Input Selection 33
4.2.4 Data Division 34
4.2.5 Model Architecture Selection 34
4.2.6 Model Structure Selection 34
4.2.7 Model Training 37
4.2.8 Model Evaluation 37

CHAPTER 5- RESULTS AND DISCUSSIONS 38

5.1 DATA PRE-PROCESSING 38
5.2 INPUT SELECTION 45
5.3 DATA PARTITIONING 51
5.4 MODEL TRAINING AND TESTING 51
5.4.3 Modelling Results of MISO Configuration 52
5.4.4 Modelling Results of MIMO Configuration 63

CHAPTER 6- CONCLUSION AND RECOMMENDATIONS 69

6.1 CONCLUSION 69
6.2 RECOMMENDATIONS 71

REFERENCES 72

APPENDIX 74

APPENDIX-A: ESTIMATION OF PARTIAL MUTUAL INFORMATION 74
APPENDIX-B: VBA CODE FOR ESTIMATION OF PARTIAL MUTUAL INFORMATION 77
APPENDIX-C: PMI SCORE FOR OUTLIER REMOVED DATA 87
APPENDIX-D: GRAPHICAL PLOT OF POTENTIAL INPUT VARIABLES AND TARGET VARIABLES FOR OUTLIER REMOVED DATA 90
APPENDIX-E: NETWORK DESIGN AND SOFTWARE SETTINGS 92
APPENDIX-F: BASIC STATISTICS 95

List of Figures

Figure 2.1: (a) Schematic of biological neuron. (b) Mechanism of signal transfer between two biological neurons. Reproduced from. 7

Figure 2.2: Block diagram of the nervous system. Reproduced from. 8

Figure 2.3: Signal interaction from neurons and analogy to signal summing in an artificial neuron comprising the single layer perceptron. Reproduced from. 9

Figure 2.4: Structure of a typical multilayer neural network.Reproduced from 10

Figure 2.5: Single node anatomy. Reproduced from 11

Figure 2.6: Commonly used transfer functions. Reproduced from. 13

Figure 2.7: Notation and index labeling used in back propagation ANNs. Reproduced from . 15

Figure 2.8: Problems solved by ANNs.(a) pattern classification (b) clustering, (c) function approximation, (d) forecasting, (e) association (e.g., image completion). Reproduced from. . 19

Figure 2.9: Steps in ANN model development process. Reproduced from. 23

Figure 3.1: Satellite image of Kaliti wastewater treatment plant. 24

Figure 5.1: Potential input variables for the Raw operational Data of Kaliti wastewater treatment plant (a) TDS,TVS,TSS,BOD and COD,(b) NO_2,NO_3,pH and PO_4,(c) SO_4,NH_3,DO and EC 40

Figure 5.2: Appropriate Output variables for the raw operational Data of Kaliti wastewater treatment plant.(a) BOD.NH_3,COD,and TDS.(b) pH. 41

Figure 5.3: Box diagrams for the potential input data and appropriate target data 42

Figure 5.4: Prediction of pH based on the raw operational data. 54

Figure 5.5: Prediction of pH based on outlier removed data. 55

Figure 5.6 : Prediction of BOD_5 based on the raw operational data. 56

Figure 5.7: Prediction of BOD_5 based on outlier removed data. 57

Figure 5.8: Prediction of COD based on the raw operational data. 58

Figure 5.9: Prediction of COD based on outlier removed data. 58

Figure 5.10: Prediction of NH_3 based on the raw operational data. 59

Figure 5.11: Prediction of NH_3based on outlier removed data. 60

Figure 5.12: Prediction of TDS based on the raw operational data. 61

Figure 5.13: Prediction of TDS based on outlier removed data. 62

Figure 5.14: Prediction of performance indicators of the WWTP based on raw operational data 65

Figure 5.15: Prediction of performance indicators of the WWTP based on outlier removed data 67

List of Tables

Table 3.1: Characteristics of the biological wastewater treatment plant 27

Table 4.1: Selected operational variables of Kaliti WWTP.....31

Table 4.2: Network specification used for PMIS based input selection 36

Table 4.3: Network specification used for GA based input selection 36

Table 5.1: Basic statistics descriptors for appropriate selected model outputs of raw operational data.....43

Table 5.2: Basic statistics descriptors for potential model input variables of raw operational data.....43

Table 5.3 : Basic statistics descriptors for appropriate selected model outputs of outlier removed data.....44

Table 5.4: Basic statistics descriptors for potential model input variables of outlier removed data.....44

Table 5.5: Partial mutual information score for pH output variable.....46

Table 5.6: Partial mutual information score for BOD_5 output variable.....47

Table 5.7: Partial mutual information score for COD output variable 48

Table 5.8: Partial mutual information score for NH_3 output variable 49

Table 5.9: Partial mutual information score for TDS output variable 50

Table 5.10: Selected input variables using PMIS for outlier removed data 50

Table 5.11: Data partition set for both the raw operational and outlier removed data 51

Table 5.12: Performance Statistics of selected models for pH prediction.....53

Table 5.13: Performance Statistics of selected models for BOD_5 prediction.....55

Table 5.14: Performance Statistics of selected models for COD prediction 57

Table 5.15: Performance Statistics of selected models for NH_3 prediction.....59

Table 5.16: Performance Statistics of selected models for TDS prediction 61

Table 5.17: Performance Statistics of selected model for MIMO configuration for the raw operational data.....63

List of Acronyms

AAWSA	Addis Ababa Water and Sewerage Authority
AIC	Akaike Information Criterion
ANNs	Artificial Neural Networks
BOD	Biochemical Oxygen Demand
BPANNs	Back propagation Artificial Neural Networks
COD	Chemical Oxygen Demand
DO	Dissolved Oxygen
EC	Electrical Conductivity
GA	Genetic Algorithm
GRNN	Generalized Regression Neural Networks
HoA-REC/N	Horn of Africa Regional Environment Center and Network
MI	Mutual Information
MIMO	Multiple-Input-Multiple-Output
MISO	Multiple-Input-Single-Output
MLPs	Multilayer Perceptrons
NH_3	Ammonia
NO_2	Nitrate
NO_3	Nitrite
PE	Processing Element
PMIS	Partial Mutual Information-based Input Selection

PO_4^{3-}	Phosphate
SO_4^{2-}	Sulphate
TDS	Total Dissolved Solids
TSS	Total Suspended Solids
TVS	Total Volatile Solids
VBA	Visual Basic for Applications
WWTP	Wastewater Treatment Plant

Chapter 1- Introduction

1.1 Background

Proper operation and control of Wastewater Treatment Plant (WWTP) is vital in producing effluent which meets quality requirements of regulatory agencies such as the environmental protection authority and minimizes detrimental effects on the environment as well as public health. Though modelling of a WWTP is a very difficult task due to the non-linear characteristics of the physical, biological and chemical processes involved; a better operation and control of a WWTP can be achieved by developing robust models for predicting the plant performance based on the past observation (Moreno et.al.,2001).

Owing to their high accuracy, adequacy and quite promising applications in engineering, Artificial Neural Networks (ANNs) can be used for modelling WWTP processes (Maier et.al.,2000). The ANN, by relying on representative historical data of the process they can be used for better prediction of the process performance.

Artificial Neural Networks are computational modelling tools that have recently emerged and found extensive acceptance in many disciplines for modelling complex real-world problems. ANNs may be defined as structures comprised of densely interconnected adaptive simple processing elements (called artificial neurons or nodes) that are capable of performing massively parallel computations for data processing and knowledge representation (Schalkoff,1997).

The attractiveness of ANNs comes from the remarkable information processing characteristics of the biological system such as non linearity, high parallelism, robustness, fault and failure tolerance, learning, ability to handle imprecise and fuzzy information, and their capability to generalize (Basheer et.al.,2000). Artificial models possessing such characteristics are desirable because (i) non linearity allows better fit to the data, (ii) noise-insensitivity provides accurate prediction in the presence of uncertain data and measurement errors, (iii) high parallelism implies fast processing and hardware failure-tolerance, (iv) learning and adaptivity allow the system to update (modify) its internal structure in response to changing environment, and (v) generalization enables application of the model to unlearned data (Zhang,1999).

In addition to their super capability to acquire non-linear characteristics of a process, ANNs provide different advantages over mechanistic modelling of a WWTP. For instance, when ANNs are applied to prediction of WWTP performance task they will result in reduction of cost for undertaking laboratory tests. And also save time-taken for undergoing lengthy laboratory tests like BOD_5 determination in addition to the advantage that efficiently predicted values of parameters will provide for proper operation and control of the WWTP.

In this work, soft sensors based on GA-PMIS sets and neural network were developed to predict the effluent concentration of pH, BOD_5, COD, NH_3, and TDS of the WWTP and in turn to predict the performance of Kaliti WWTP.

1.2 Statement of The Problem

The characteristics of influent to the WWTPs vary from one plant to another depending on the type of community life style. Not only this, the type of influent for any plant is also time dependent and it is difficult to have a homogeneous influent to a WWTP (Hong et.al.,2003). This may result in an operational risk impact on the plant. In addition to this, serious environmental and public health problems may result from improper operation and control of a WWTP, as discharging contaminated effluent to a receiving water body can cause or spread various diseases to human beings. Accordingly, environmental regulations set restrictions on the quality of effluent that must be met by any WWTP. These stringent discharge standards and time-dependant non uniform influent characteristics make the proper management of treatment systems an issue.

Kaliti WWTP cannot be an exception to the situation described above i.e. unless the plant is properly operated and controlled, it will result in serious environmental and public health problems. A better (safer) operation and control of the WWTP can be achieved by developing a robust mathematical tool for predicting the plant performance based on past observations of certain key parameters. To this end, in this thesis work, an artificial neural network modelling technique was used to acquire the knowledge base of Kaliti WWTP and predictive models are presented to provide accurate predictions of the performance of the WWTP.

1.3 Objectives

1.3.1 General Objective

The general objective of this thesis work was to develop an intelligent models that enables to predict the performance of Kaliti wastewater treatment plant using Artificial Neural Networks.

1.3.2 Specific Objectives

i. To develop ANN models that are capable of predicting treated wastewater quality parameters given raw wastewater quality parameters.

ii. To evaluate fitness of the ANN models to the data.

iii. To assess the performance of the ANN models developed.

1.4 Organization of the Thesis

This thesis has been organized into six chapters and supporting appendices. This chapter outlines the scope of this research which is further detailed in subsequent chapters.

Chapter 2 is a review of literature detailing back propagation artificial neural networks. Application of neural networks and a framework of neural network model development process are also discussed in this section.

Chapter 3 briefly introduces Kaliti wastewater treatment plant and also describes the biological wastewater treatment process while Chapter 4 discusses the material and software used and the methods followed to predict the performance of the wastewater treatment plant. This work utilizes actual operating data from Kaliti wastewater treatment plant located in Addis Ababa, Ethiopia. And NeuralWorks Predict® and MATLAB®, JMP PRO® are applied to accomplish the task.

Chapter 5 presents the developed predictive models for the wastewater treatment plant. Moreover this chapter discusses the performance of the candidate models developed to predict the performance of the treatment plant using different performance statistics and graphical analysis.

Chapter 6 concludes the thesis by pointing out some recommendations for future research.

1.5 Uniqueness of the Thesis

In contrast to several previous studies in using artificial neural network technology for wastewater application, this work drives its uniqueness from the following points:

- Development of neural networks in conjunction with genetic algorithm, partial mutual information estimation ,and statistical techniques for predictive modelling of domestic wastewater treatment plant.
- This work utilizes actual operating data collected (not simulated data) that covers the entire seasonal variation for all the studied variables.

Chapter 2- Literature Review

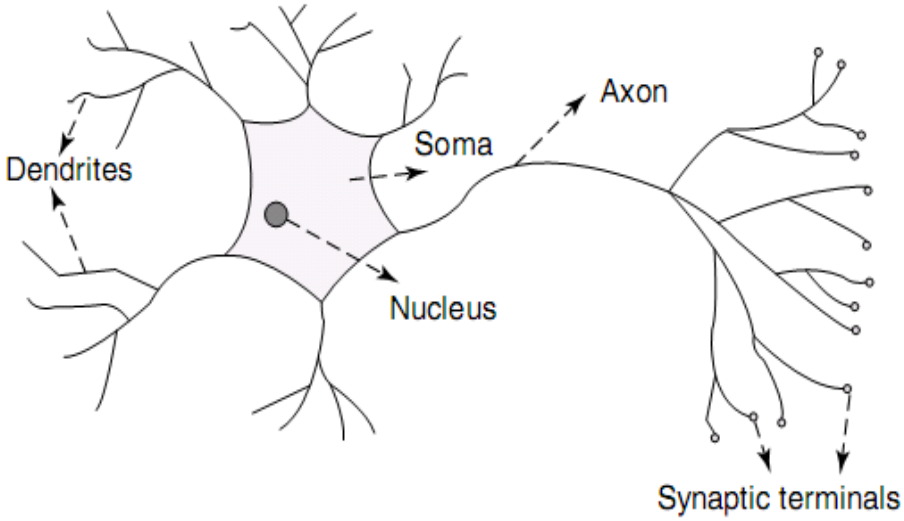
2.1 Artificial Neural Networks Fundamentals

An artificial neural networks, or simply neural network, is a type of artificial intelligence (computer system) that attempts to mimic the way the human brain processes and stores information. It works by creating connections between mathematical processing elements, called neurons. Knowledge is encoded into the network through the strength of the connections between different neurons, called weights, and by creating groups, or layers, of neurons that work in parallel. The system learns through a process of determining the number of neurons or nodes and adjusting the weights for the connections based upon training data.

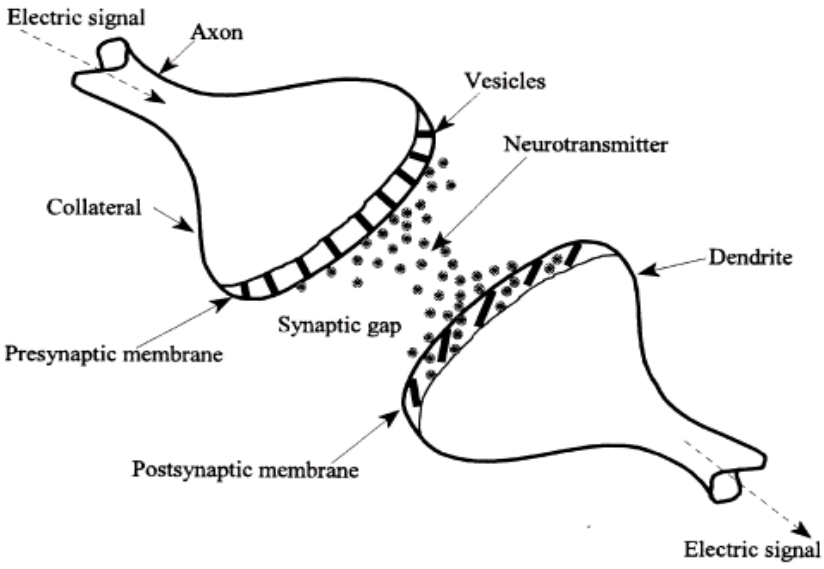
Although ANNs are drastic abstractions of the biological counterparts, the idea of ANNs is not to replicate the operation of the biological systems but to make use of what is known about the functionality of the biological networks for solving complex problems. The main objective of ANN based computing (neurocomputing) is to develop mathematical algorithms that will enable ANNs to learn by mimicking information processing and knowledge acquisition in the human brain. ANN based models are empirical in nature, however they can provide practically accurate solutions for precisely or imprecisely formulated problems and for phenomena that are only understood through experimental data and field observations (ASCE,2000).

2.1.1 Biological Neuron

The human nervous system consists of billions of neurons of various types and lengths relevant to their location in the body (Schalkoff,1997). Fig.1a shows a schematic of an over simplified biological neuron with three major functional units dendrites, cell body (soma), and axon. The cell body has a nucleus that contains information about heredity traits, and a plasma that holds the molecular equipment used for producing the material needed by the neuron (Jain et.al.,1996).The dendrites receive signals from other neurons and pass them over to the cell body. The total receiving area of the dendrites of a typical neuron is approximately 0.25mm (Zupan and Gasteiger,1993).



(a)



(b)

Figure 2.1: (a) Schematic of biological neuron. (b) Mechanism of signal transfer between two biological neurons. Reproduced from (Schalkoff,1997).

The axon, which branches into collaterals, receives signals from the cell body and carries them away through the synapse (a microscopic gap) to the dendrites of neighboring neurons. A schematic illustration of the signal transfer between two neurons through the synapse is shown in Fig.2.1b. An impulse, in the form of an electric signal, travels with in the dendrites and through the cell body towards the presynaptic membrane of the synapse.

Up on arrival at the membrane, a neurotransmitter (chemical) is released from the vesicles in quantities proportional to the strength of the incoming signal. The neurotransmitter diffuses within the synaptic gap towards the post-synaptic membrane, and eventually into the dendrites of neighboring neurons, thus forcing them (depending on the threshold of the receiving neuron) to generate a new electrical signal. The generated signal passes through the second neuron(s) in a manner identical to that just described. The amount of signal that passes through a receiving neuron depends on the intensity of the signal emanating from each of the feeding neurons, their synaptic strengths, and the threshold of the receiving neuron (Basheer et.al.,2000).

Because a neuron has a large number of dendrites/synapses, it can receive and transfer many signals simultaneously. These signals may either assist (excite) or inhibit the firing of the neuron. This simplified mechanism of signal transfer constituted the fundamental step of early neurocomputing development and the operation of the building unit of ANNs.

Figure 2.2 shows the block diagram of the nervous system. The ability of the nervous system to adjust to signals is a mechanism of learning, and the rate of firing an output (response) is altered by the activity in the nervous system. Simply, a single neuron processes information by receiving signals from its dendrites, and produces an output signal which is then transmitted to other neurons.

Figure 2.2: Block diagram of the nervous system. Reproduced from (Basheer et.al.,2000).

The crude analogy between artificial neuron and biological neuron is that the connections between nodes represent the axons and dendrites, the connection weights represent the synapses, and the threshold approximates the activity in the soma (Jain et.al.,1996). Fig.2.3 illustrates biological neurons with various signals of intensity x and synaptic strength w feeding into a neuron with a threshold of b, and the equivalent artificial neurons system. Both the biological network and ANN learn by incrementally adjusting the magnitudes of the weights or synapses' strengths (Zupan and Gasteiger,1993).

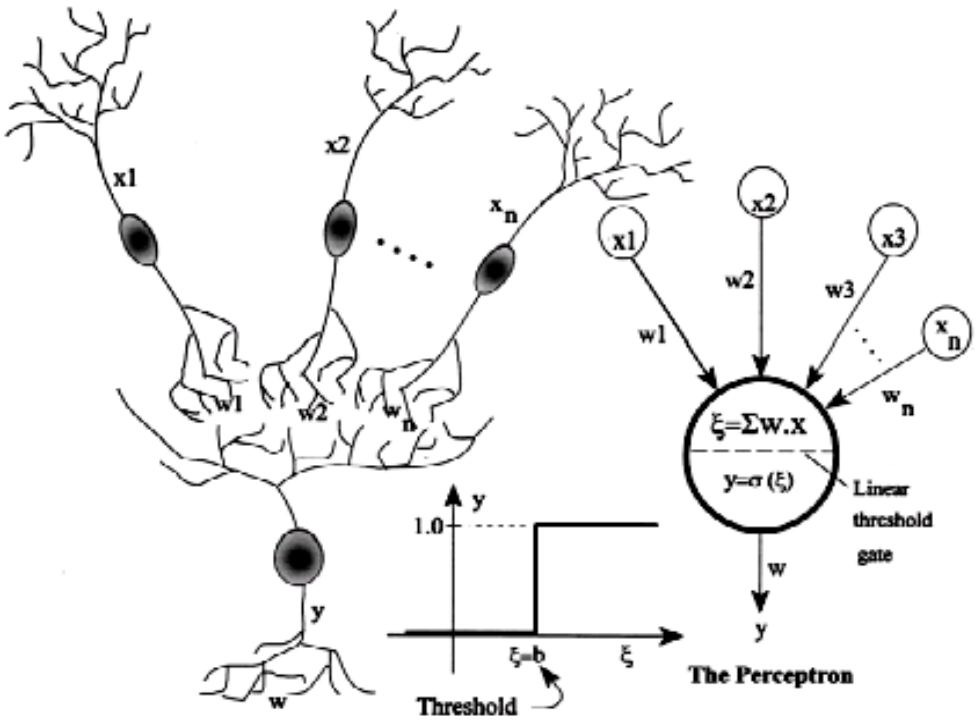


Figure 2.3: Signal interaction from neurons and analogy to signal summing in an artificial neuron comprising the single layer perceptron. Reproduced from (Jain et.al.,1996).

2.1.2 Artificial Neuron

In 1958, Rosenblatt introduced the mechanics of the single artificial neuron and introduced the 'Perceptron' to solve problems in the area of character recognition (Hecht-Nielsen,1990). Basic findings from the biological neuron operation enabled early researchers (e.g., McCulloch and Pitts,1943) to model the operation of simple artificial neurons. An artificial processing neuron receives inputs as stimuli from the environment, combines them in a special way to form a 'net' input (ε), passes that over through a linear threshold gate, and transmits the (output, y) signal forward to another neuron or the environment, as shown in Fig.2.2. Only when ε exceeds (i.e., is stronger than) the neuron's threshold limit (also called bias, b),will the neuron fire (i.e. becomes activated). Commonly, linear neuron dynamics are assumed for calculating ε (Haykin,1994).The net input is computed as the inner (dot) product of the input signals (x) impinging on the neuron and their strengths (w). For n signals, the perceptron neuron operation is expressed as

$$y = \begin{cases} 1, & if \sum_{i=1}^{n} w_i x_i \geq b, \\ 0, & if \sum_{i=1}^{n} w_i x_i < b, \end{cases} \quad \text{.....(2.1)}$$

With 1 indicating ‘on’ and 0 indicating ‘off’ (Fig.2.3), or class A and B, respectively, in solving classification problems. Positive connection weights ($W_i > 0$) enhance the net signal (ε) and excite the neuron, and the link is called excitory, where as negative weights reduce ε and inhibit the neuron activity, and the link is called inhibitory. The system comprised of an artificial neuron and the inputs as shown in Fig.2.3 is called the Perceptron which establishes a mapping between the inputs activity (stimuli) and the output signal. In Eq.(2.1), the neuron threshold may be considered as an additional input node whose value is always unity (i.e. x=1) and its connection weight is equal to b. In such case, the summation in Eq.(2.1) is run from 0 to n, and the net signal ε is compared to 0.

2.1.3 Architecture and Elements of Artificial Neural Network

Neural networks can be thought of as “Black Box” devices that accept inputs and produces outputs (Hassoun,1995). Figure 2.4 shows a typical neural network structure.

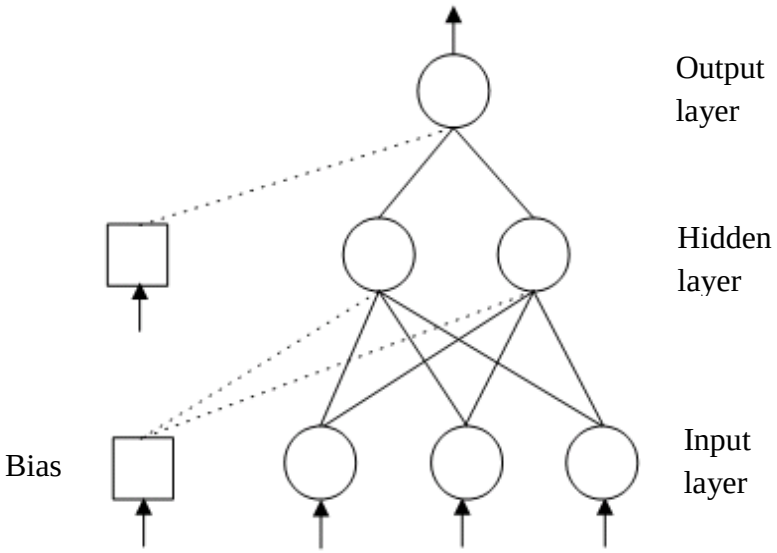


Figure 2.4: Structure of a typical multilayer neural network.Reproduced from (Hassoun,1995)

Input Layer: A layer of neurons that receives information from external sources, and passes this information to the network for processing. These may be either sensory inputs or signals from other systems outside the one being modeled.

Hidden Layer: A layer of neurons that receives information from the input layer and processes them in a hidden way. It has no direct connections to the outside world (inputs or outputs). All connections from the hidden layer are to other layers within the system.

Output Layer: A layer of neurons that receives processed information and sends output signals out of the system.

Bias: Acts on a neuron like an offset. The function of the bias is to provide a threshold for the activation of neurons. The bias input is connected to each of the hidden and output neurons in a network.

The number of input neurons corresponds to the number of input variables into the neural network, and the number of output neurons is the same as the number of desired output variables. The number of neurons in the hidden layer(s) depends on the application of the network.

As inputs enter the input layer from an external source, the input layer becomes "activated" and emits signals to its neighbors (hidden layer) without any modification. Neurons in the input layer act as distribution nodes and transfer input signals to neurons in the hidden layer. The neighbors receive excitation from the input layer, and in turn emit an output to their neighbors (second hidden layer or output layer). Each input connection is associated to a quantity, called "a weight factor" or "a connection strength".

The strength of a connection between two neurons determines the relative effect that one neuron can have on another. The weight is positive if the associated connection is excitatory, and negative if the connection is inhibitory.

The basic component of a neural network is the neuron, also called "node", or the "processing element, PE". Nodes contain the mathematical processing elements which govern the operation of a neural network. Figure 2.5 illustrates a single node of a neural network.

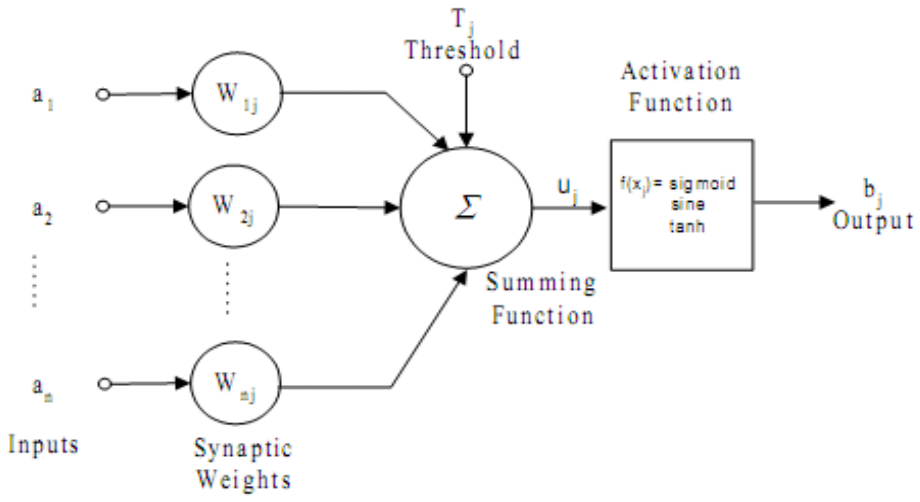


Figure 2.5: Single node anatomy. Reproduced from (Hassoun,1995).

i) Inputs and Outputs

We represent inputs by a_1, a_2 and a_n, and the output by b_j. Just as there are many inputs to a neuron, there should be many input signals to our PE. The PE manipulates these inputs to give a single output signal.

ii) Weighting Factors

The values W_{1j}, W_{2j}, and W_{nj}, are weight factors associated with each input to the node. This is something like the varying synaptic strengths of biological neurons. Weights are adaptive coefficients within the network that determine the intensity of the input signal. Every input (a_1, a_2, …, a_n) is multiplied by its corresponding weight factor (W_{1j}, W_{2j}, …, W_{nj}), and the node uses this weighted input (W_{1j} a_1, W_{2j} a_2, …, W_{nj} a_n) to perform further calculations. If the weight factor is positive, then (W_{ij} a_i) tends to excites the node. If the weight factor is negative, then (W_{ij} a_i) inhibits the node.

In the initial setup of a neural network, we choose weight factors according to a specified statistical distribution. We then adjust the weight factors in the development of the network or "learning" process (Khawla,1998).

iii) Internal Threshold

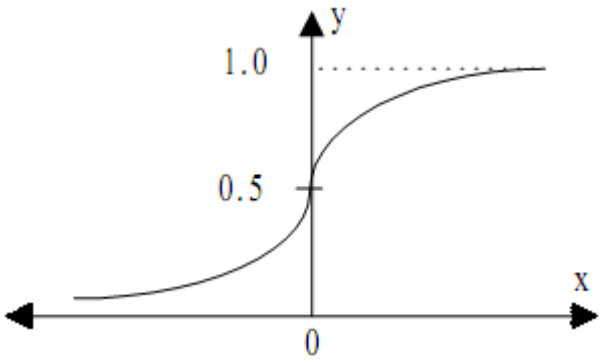
The other input to the node, T_j, is the node's internal threshold. This is a randomly chosen value that governs the "activation" or total input of the node through the following equation.

Total Activation= $x_i = \sum_{i=1}^{n}(w_{ij}a_i) - T_j$……..…..(2.2)

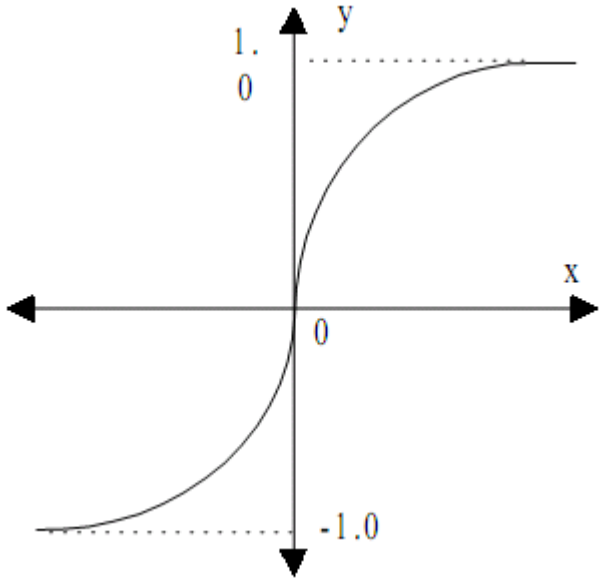
The total activation depends on the magnitude of the internal threshold T_j. If T_j is large or positive, the node has a high internal threshold, thus inhibiting node-firing. If T_j is zero or negative, the node has a low internal threshold, which excites node-firing. If no internal threshold is specified, we assume it to be zero.

iv) Transfer Functions

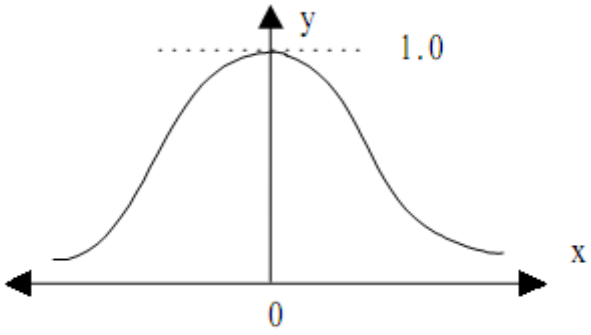
We determine node's output using a mathematical operation on the total activation of the node. This operation is called a transfer function. The transfer function can transform the node's activation in a linear or nonlinear manner. Figure 2.6 shows several types of commonly used transfer functions.



(a) A sigmoidal transfer function



(b) A hyperbolic tangent transfer function



(c) A Gaussian transfer function

Figure 2.6: Commonly used transfer functions. Reproduced from (Hecht-Nielsen,1990).

The corresponding equation for the transfer function are as follows:

- Sigmoid transfer function

$$f(x) = \frac{1}{1+e^{-x}} \; ; \; 0 \leq f(x) \leq 1 \quad \text{.....(2.3)}$$

- Hyperbolic tangent transfer function

$$f(x) = tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \; ; \; -1 \leq f(x) \leq 1 \quad \text{.....(2.4)}$$

- Gaussian transfer function

$$f(x) = exp\left(\frac{-x^{2}}{2}\right); \; 0 \leq f(x) \leq 1 \quad \text{.....(2.5)}$$

The, output, b_j, is found by performing one of these functions on the total activation, x_i,

2.2 Back propagation Artificial Neural Networks

The Back Propagation artificial neural networks (BPANNs) are discussed in more detail, for their popularity, and their flexibility and adaptability in modelling a wide spectrum of problems in many application areas. The feed forward error back propagation learning algorithm is the most famous procedure for training ANNs. Back propagation is based on searching an error surface (error as a function of ANN weights) using gradient descent for point(s) with minimum error. Each iteration in BP constitutes two sweeps: forward activation to produce a solution, and a backward propagation of the computed error to modify the weights. In an initialized ANN (i.e., an ANN with assumed initial weights), the forward sweep involves presenting the network with one training example (Wythoff,1993).

This starts at the input layer where each input node transmits the value received forward to each hidden node in the hidden layer. The collective effect on each of the hidden nodes is summed up by performing the dot product of all values of input nodes and their corresponding interconnection weights, as described in Eq.(2.1). Once the net effect at one hidden node is determined, the activation at that node is calculated using a transfer function (e.g., sigmoidal function) to yield an output between 0 and +1 or -1 and +1. The amount of activation obtained represents the new signal that is to be transferred forward to the subsequent layer (e.g., either hidden or output layer). The same procedure of calculating the net effect is repeated for each hidden node and for all hidden layers. The net effect(s) calculated at the output node(s) is consequently transformed into activation(s) using a transfer function.

The activation(s) just calculated at the output node(s) represents the ANN solution of the fed example, which may deviate considerably from the target solution due to the arbitrary selected inter connection weights. In the backward sweep, the difference (i.e.,error) between the ANN and target outputs is used to adjust the interconnection weights, starting from the output layer, through all hidden layers, to the input layer, as will be described in the following section. The forward and back ward sweeps are performed repeatedly until the ANN solution agrees with the target value with in a prespecified tolerance. The BP learning algorithm provides the needed weight adjustments in the backward sweep.

2.2.1 Back propagation Algorithm

In order to be able to run the algorithm, it is essential to define the interlayer as the gap between two successive layers that encloses the connection weights and contains only the neurons of the upper layer, as shown in Fig.2.7 (assuming that all layers are positioned above the input layer).Consider a Multi Layer Perceptron (MLP) network with L interlayers. For interlayer l {1,2, ...,L} there are N_l nodes and $N_l \times N_{l-1}$ connection links with weights $W \in R^{Nl_x Nl-1}$,where N_l and N_{l-1} are the numbers of nodes (including thresholds) in interlayers l and l-1,respectively (Fig.2.7).A connection weight is denoted by W_{ji}^{l} if it resides in interlayer l and connects node j of interlayer l with node i of lower (preceding) interlayer l-1(node i is the source node and node j is the destination node).In any interlayer l, a typical neuron j integrates the signals, x_j ,impinging onto it, and produces a net effect, ε_j,according to linear neuron dynamics:

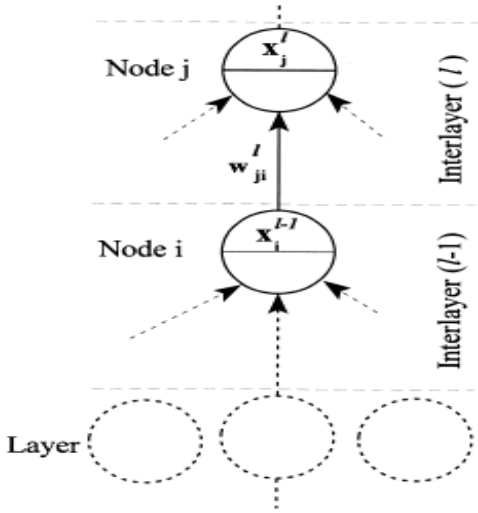


Figure 2.7: Notation and index labeling used in back propagation ANNs. Reproduced from (Wythoff,1993).

$$\xi_j^l = \sum_{i=1}^{N_{l-1}} w_{ji}^l x_i^{l-1} \quad \text{(2.6)}$$

The corresponding activation, x_j^l ,of the neuron is determined using a transfer function, σ, that converts the total signal into a real number from abounded interval:

$$x_j^l = \sigma(\xi_j^l) = \sigma\left(\sum_{i=1}^{N_{l-1}} w_{ji}^l x_i^{l-1}\right) \quad \text{(2.7)}$$

One popular function used in BP is the basic continuous sigmoid:

$$\sigma(\xi) = \frac{1}{1+e^{-\xi}} \quad \text{(2.8)}$$

where $-\infty < \xi < \infty +$ and 0.0≤ σ ≤1.0. Eqs.(2.6) – (2.8) are used for all nodes to calculate the activation. For the input nodes the activation is simply the raw input. In any interlayer, an arbitrary weight W_{ji}^l at iteration(t) will be updated from its previous state (t-1) value according to:

$$\mathrm{w_{ji}^l}(t) = w_{ji}^l(t-1) + \Delta w_{ji}^l(t) \quad \text{(2.9)}$$

where ΔW_{ji}^l is the(+/-) incremental change in the weight. The weight change is determined via the modified delta rule (Zupan and Gasteiger,1993),which can be written as

$$\Delta \mathrm{w_{ji}^l} = \eta \delta_j^l x_i^{l-1} + \mu \Delta w_{ji}^{l(previous)} \quad \text{(2.10)}$$

where η is the learning rate controlling the update step size, μ is the momentum coefficient, and x_i^{l-1} is the input from the $l-1$th interlayer. The first part of the right-hand side of Eq.(2.10) is the original delta rule. The added momentum term helps direct the search on the error hyperspace to the global minimum by allowing a portion of the previous updating (magnitude and direction) to be added to the current updating step. Note that Eq.(2.10) can also be applied to any neuron threshold (bias) which can be assumed as a link, with weight equal to the threshold value, for an imaginary neuron whose activation is fixed at 1.0.The weight change can also be determined using a gradient descent written in generalized form for an interlayer l:

$$\Delta \mathrm{w_{ji}^l} = -k\left(\frac{\partial \varepsilon^l}{\partial w_{ji}^l}\right) \quad \text{(2.11)}$$

Therefore, in order to determine the incremental changes for the lth interlayer, the main task is to quantify the error gradient ($\partial \varepsilon^l / \partial W_{ji}^l$). Using Eqs.(2.10) and (2.11),the required weight

change can be derived with different expressions depending on whether the considered neuron is in the output layer or in a hidden layer. If the neuron is in the output layer, then l=L in Eq.(2.10),with δ_j^l calculated from

$$\delta_j^L = \left(x_j^L - y_j\right)x_j^L(l - x_j^L) \quad \text{..(2.12)}$$

If the neuron is in a hidden layer, the weight change is also calculated using Eq.(2.10) with δ_j^l determined from

$$\delta_j^l = x_j^l\left(1 - x_j^l\right)\left(\sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1}\right) \quad \text{..(2.13)}$$

where δ_k^{l+1} is calculated for a given non-output layer (l) beginning with a layer one level up ($l + 1$) and moving down layer by layer. That is, for the last (uppermost) hidden layer in a network, δ_j^l is determined via δ_k^{l+1} of the output layer calculated using Eq.(2.12)**.**The above delta equations ,Eqs.(2.12) and(2.13), are based on the sigmoid transfer function given in Eq.(2.8)**.** For a different function, the terms $x_j^L\left(1 - x_j^L\right)$ and $x_j^l\left(1 - x_j^l\right)$ in Eqs.(2.12) and (2.13), respectively, should be replaced with the relevant first derivative of the used function. This technique of distributing backward the errors starting from the output layer down through the hidden layer gave the method the name back propagation of error with the modified delta rule (Rumel hart et.al.,1995).

2.3 Application of Neural Networks

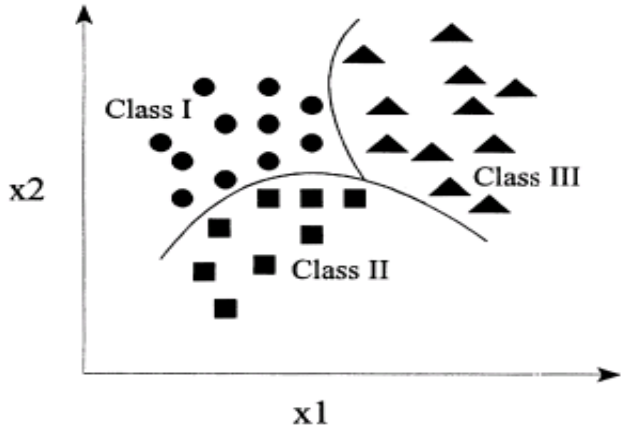
Neural networks can deal with problems that are complex, nonlinear, and uncertain, due to their properties and capabilities.

2.3.1 Pattern Classification

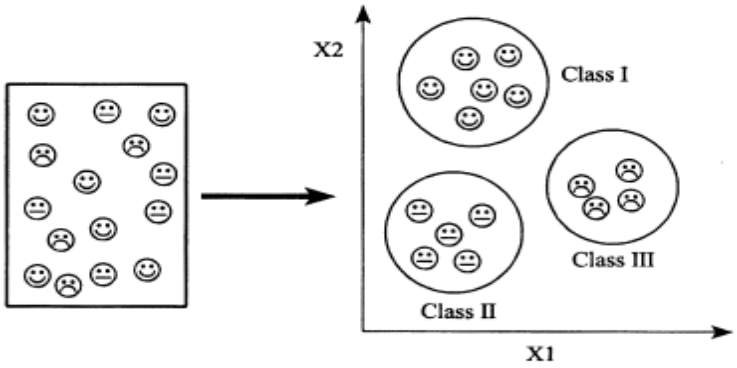
Pattern classification deals with assigning an unknown input pattern, using supervised learning, to one of several prespecified classes based on one or more properties that characterize a given class, as shown in Fig.2.8a.

2.3.2 Clustering

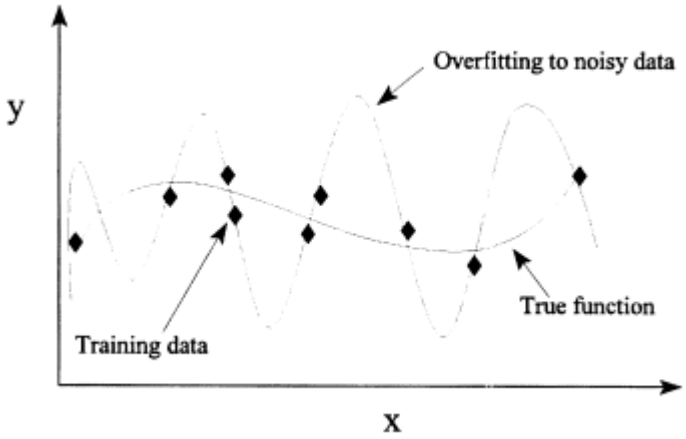
Clustering is performed via unsupervised learning in which the clusters (classes) are formed by exploring the similarities or dissimilarities between the input patterns based on their inter correlations (Fig.2.8b).The network assigns ‘similar’ patterns to the same cluster.



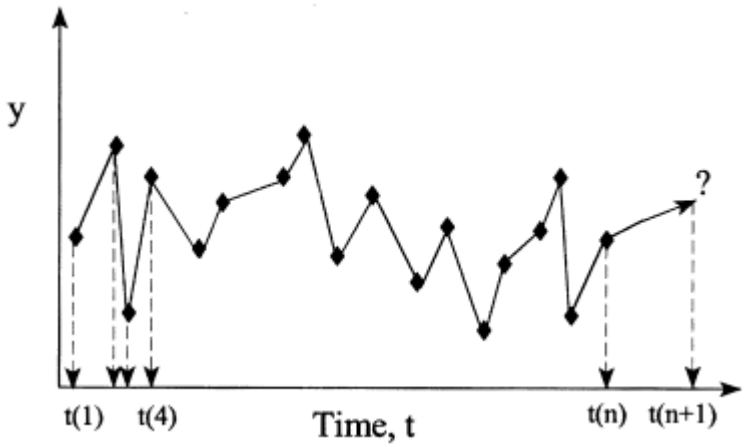
(a) Pattern Classification



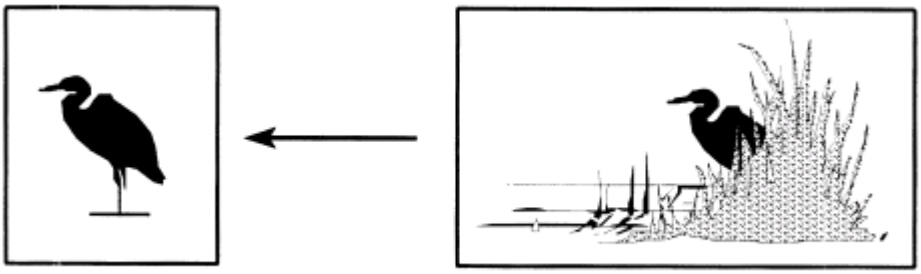
(b) Clustering



(c) Function Approximation



(d) Forecasting



(e) Image Completion

Figure 2.8: Problems solved by ANNs.(a) pattern classification (b) clustering, (c) function approximation, (d) forecasting, (e) association (e.g., image completion). Reproduced from (Jain et.al.,1996).

2.3.3 Function Approximation

Function approximation (modelling) involves training ANN on input–output data so as to approximate the underlying rules relating the inputs to the outputs (Fig.2.8c).Multilayer ANNs are considered universal approximators that can approximate any arbitrary function to any degree of accuracy (Hecht-Nielsen,1990),and thus are normally used in this application. Function approximation is applied to problems (i) where no theoretical model is available, i.e., data obtained from experiments or observations are utilized, or (ii) to substitute theoretical models that are hard to compute analytically by utilizing data obtained from such models.

2.3.4 Forecasting

Forecasting includes training of an ANN on samples from a time series representing a certain phenomenon at a given scenario and then using it for other scenarios to predict (forecast) the behavior at subsequent times (Fig.2.8d).That is, the network will predict Y(t+1) from one or more previously known historical observations (e.g., Y(t-2), Y(t-1),and Y(t),where t is the time step).

2.3.5 Optimization

Optimization is concerned with finding a solution that maximizes or minimizes an objective function subject to a set of constraints. Optimization is a well-established field in mathematics, however ANNs, such as the Hopfield network, were found to be more efficient in solving complex and nonlinear optimization problems (Pham,1994).

2.3.6 Association

Association involves developing a pattern associator ANN by training on ideal noise-free data and subsequently using this ANN to classify noise-corrupted data (e.g., for novelty detection). The associative network may also be used to correct (reconstruct) the corrupted data or completely missing data (or image), as shown in Fig.2.8e.Hopfield and Hamming networks are especially used for this application, and to a lesser degree multilayer back propagation ANN strained on patterns with identical input and output (Fu,1995).

2.3.7 Control

Control is concerned with designing a network, normally recurrent, that will aid an adaptive control system to generate the required control inputs such that the system will follow a certain trajectory based on system feedback (Jain et.al.,1996).

2.4 Artificial Neural Network Model Development

The main steps in the development of ANN prediction models, as well as the way the data flow through, and the outcomes achieved at, different steps, are given in Fig.2.9.The model development steps covered here represent a subset of the 10 steps presented by Jake man et.al.(2006).The first step in the model development process presented here is the choice of appropriate model output(s) (i.e. the variable(s) to be predicted) and a set of potential model input variables from the available data.

Although ANNs are data driven models, it is up to the modeler to choose which input variables should be considered as part of the model development process. This can be done based on a priori knowledge and/or the availability of data. The resulting data set constitutes the "Selected Data (Unprocessed)" (Fig.2.9).It should be noted that once the model outputs have been chosen, the number of nodes in the output layer has also been determined (Fig.2.9).Next, the unprocessed data, which consist of measured values of the potential model input variables, as well as the model output variable(s),have to be processed (e.g. scaled, lagged) so that they are in a suitable form for the subsequent steps of the model development process.

Once the processed database of potential model inputs and outputs has been assembled ("Selected Data(Processed)"),the actual model can be developed. All ANN prediction models take the following form:

$$Y = f(X, W) + \varepsilon \quad \text{.....(2.14)}$$

Where Y is the vector of model outputs; X, the vector of model inputs; W, vector of model parameters (connection weights); f(•),functional relationship between model outputs, inputs and parameters;ε vector of model errors.

Consequently, in order to develop an ANN model, the vector of model inputs (X),the form of the functional relationship (f(•)),which is governed by the network architecture (e.g. multilayer perceptron) and geometry (e.g. the number of hidden layers and nodes, type of transfer function) and the vector of model parameters (W),which includes the connection and bias weights, need to be defined. The vector of appropriate model inputs is determined during the "Input Selection" step (Fig.2.9).This can be achieved either by using a model free approach, which uses statistical measures of significance, or a model based approach, as part of which appropriate inputs are selected based on the performance of models with different sets of inputs. In the latter case, steps 5 and 6 in Fig.2.9 have to be repeated for each set of model inputs tried. Once the vector of model inputs has been selected, the number of model inputs, and hence the number of nodes in the input layer of the ANN model, are known (Fig.2.9).

The resulting "Model Development Data" are usually divided into calibration and validation subsets. The calibration data are used to estimate the unknown model parameters (connection weights) and the validation data are used to validate the performance of the calibrated model on an independent dataset. If cross validation is used as the stopping criterion, the calibration

data are divided into training and testing subsets (Fig.2.9).Next, the functional form of the model, f(•),needs to be selected, which depends on the model architecture (e.g. multilayer perceptron, radial basis function),as well as an appropriate number of hidden nodes and how they are arranged (e.g. single layer, two layers).It should be noted that while the selection of an appropriate model Structure is required for most ANN architectures, it is superfluous for some, such as Generalized Regression Neural Networks (GRNN),which have a fixed structure.

While the choice of an appropriate model architecture is a function of modeler preference, the optimal model structure generally needs to be determined using an iterative process. This involves selecting a network with a certain structure (e.g. number of hidden nodes, transfer functions),calibrating (training) the selected ANN model, as part of which an estimate of the vector of model parameters (W) is obtained, evaluating its performance and then repeating the calibration and evaluation steps for different network configurations (Fig.2.9).Once the network configuration that performs best on the calibration data is identified, the calibrated model needs to be validated using an independent data set. As ANNs are prone to over fitting the calibration data, cross validation is generally used, as part of which the calibration data are divided in to training and testing subsets, which enables the performance of models with different network configurations to be validated during the model calibration phase to ensure overfitting of the training data has not occurred.

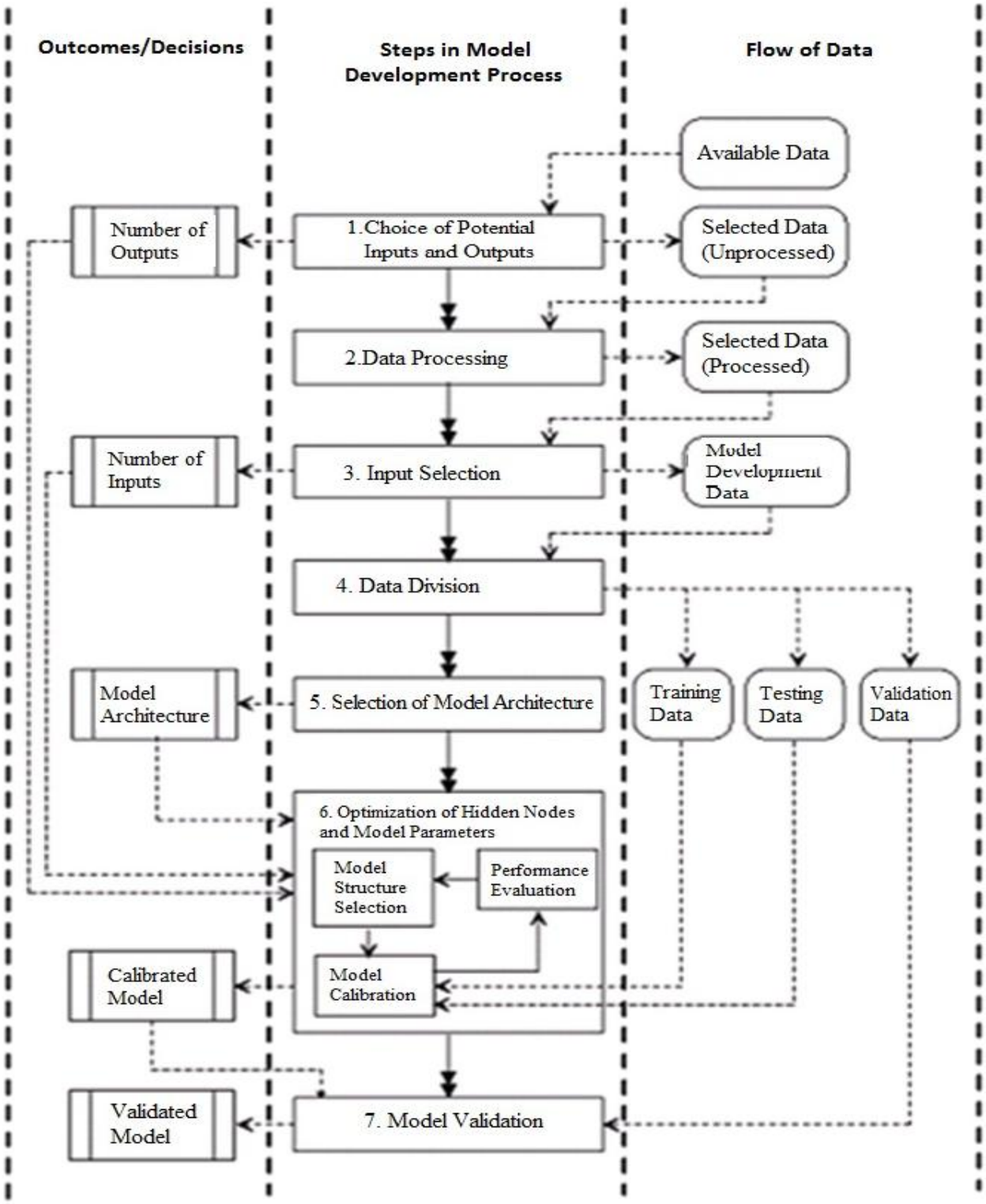


Figure 2.9: Steps in ANN model development process. Reproduced from (May et.al,2008).

Chapter 3- Description of The Wastewater Treatment Plant

3.1 Description of The Study Area

Kaliti WWTP is located in the southern part of Addis Ababa, the capital city of Ethiopia. The Kaliti wastewater treatment plant was commissioned in 1981 with a design capacity of 7,600 m^3/day flow and 3,500 kg/day biochemical oxygen demand. Treatment consists of inlet screens and grit chambers, two settling cambers, and two parallel pond systems, each made up of a facultative pond, a maturation pond and two polishing ponds. Sludge lagoons and drying beds were constructed in 1999 with treatment capacity of 110,000 m^3/year of sludge.

The wastewater from housing units connected to the city's sewer system is conveyed to the treatment plant by the sewer network while the sewage from residences and different institutions is transported by vacuum truck. According to AAWSA (2002) records, there are about 1800 connections on to the existing pipe sewer system of 120 km. The number of people connected to the existing sewer system is very low, amounting 13,000. The sewerage system was designed on the basis of an average water consumption of 150 liters per capita per day to serve an equivalent population of 200,000. Some 3000 connections discharge about 6000-7000 m^3/day into the sewer system corresponding to 4.8% percent of volume of billed water. The sanitation master plan also recognizes onsite septic tanks and pit latrines, and calls for sludge to be collected by vacuum trucks and taken to drying beds, disposed in sanitary landfills, injected into the sewer network at selected sites, or applied to forestry lands.

Figure 3.1: Satellite image of Kaliti wastewater treatment plant.

The plant was also designated to handle the night soils collected in the city, whose average estimated characteristics are as follows

- Input flow rate: 250 m^3/d
- BOD_5:2500 kg/d
- SS: 7500 kg/d

3.2 Description of The Wastewater Treatment Process

The treatment and disposal of wastewater in developing countries is of prime importance for environmental and public health reasons. The simplest method of municipal wastewater treatment is through the use of waste stabilization ponds or lagoons. Lagoons are simple earthen basins in which wastewater is treated by the removal of particulate matter and the biological degradation of settled solids. Waste stabilization ponds rely on lengthy detention times and environmental factors (wind, solar radiation) for treatment efficiency.

Kaliti Wastewater treatment plant has two triangular slanty ponds which are in parallel and eight drying beds. Each line of the pond consists of one facultative pond with a depth of 1m – 3m, one maturation pond with a depth of 1m and two polishing ponds with a depth of 1m. The hydraulic retention time of the wastewater in the stabilization ponds was approximately 30 days at maximum flow rate and the effluent from the ponds flow by gravity and finally discharged to little Akaki river.

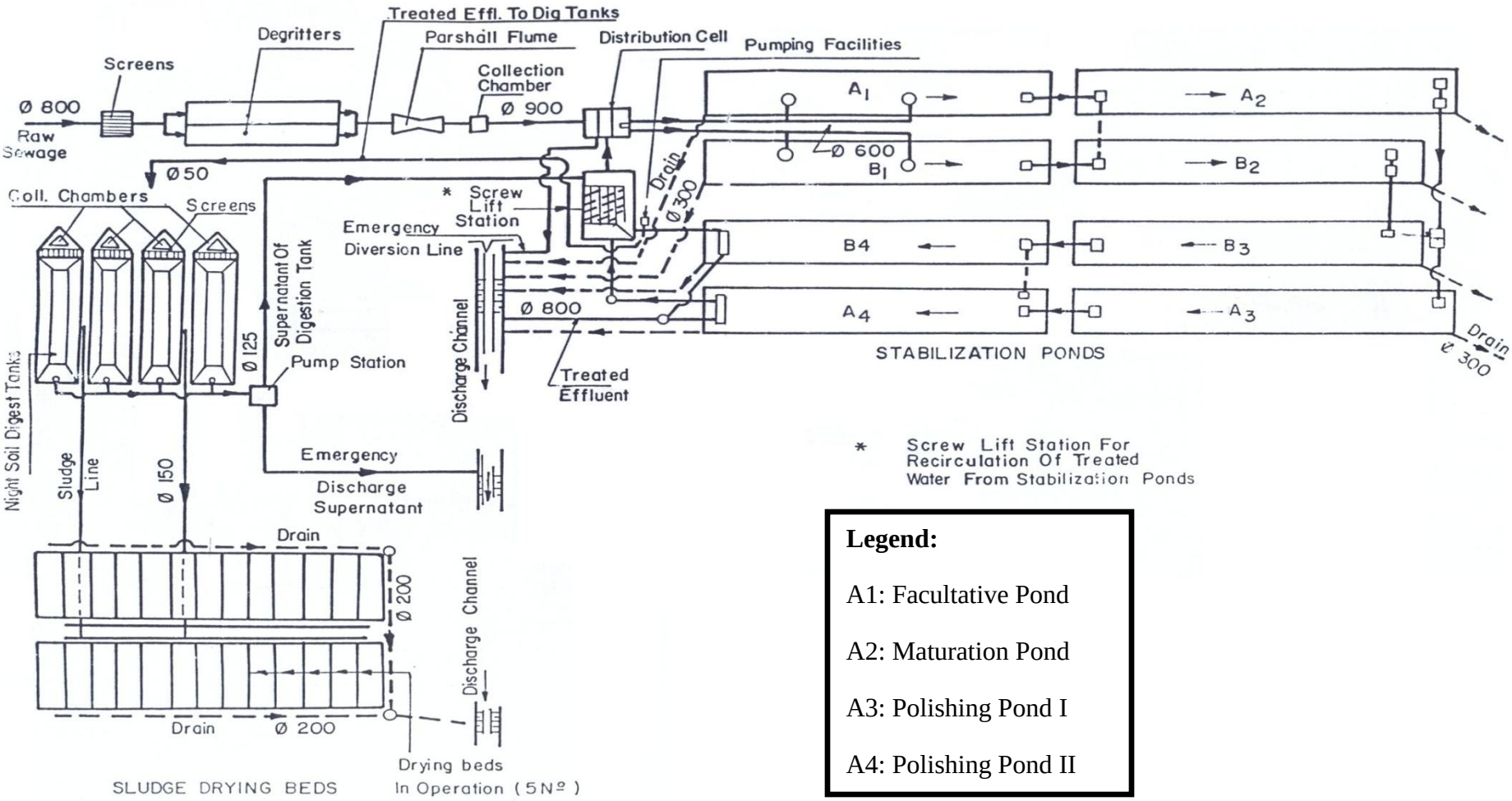


Figure 3.2: Kaliti Sewage Treatment Plant Flow Diagram. Reproduced From (AAWSA,2002).

The treatment plant consists of a set of wastewater treatment work comprising the following:

i. Screening and Degritting Channel

The treatment plant has a screening and degritting channel dimensioned from the beginning for treating an effluent of 200,000 equivalent inhabitants. This piece of equipment is composed of two canals designed to function in parallel which can be isolated by means of coffer dams each channel is equipped with an inclined screen (65 degree) of a width of 2.5m leaving a free space of 25mm between bars.

The length of each degritting channel is 10.5m. The outlet of each channel is equipped with a linear weir (Eiffel tower type) and allows the maintenance of a constant degritting speed. The downstream part of this apparatus includes a venture flume with a recording and totalizing flow meter which is installed in a small cabinet on the edge of the canal.

ii. Distribution Cell

The treatment plant has also a distribution cell that allows a partial or total feeding of all the stabilized ponds. The volume of water allowed to recirculate in each treatment path is measured by an indicating, recording, and totalizing flow meter.

iii. Stabilization Pond

The treatment plant has two rows of paralleled biological treatment comprising 4 stabilization ponds each whose global characteristics are as following

Table 3.1: Characteristics of the biological wastewater treatment plant

	Pond Type	Volume (m^3)	Retention time at 7500m^3/d	Average height (m)	Area (m^2)
A1+B1	Facultative pond	95149	12.7	2.04	46493
A2+B2	Maturation pond	44176	5.9	0.95	46577
A3+B3	Polishing pond I	44586	5.9	1.02	43563
A4+B4	Polishing pond II	44237	5.9	0.95	46482
Total		228,148	30		183,115

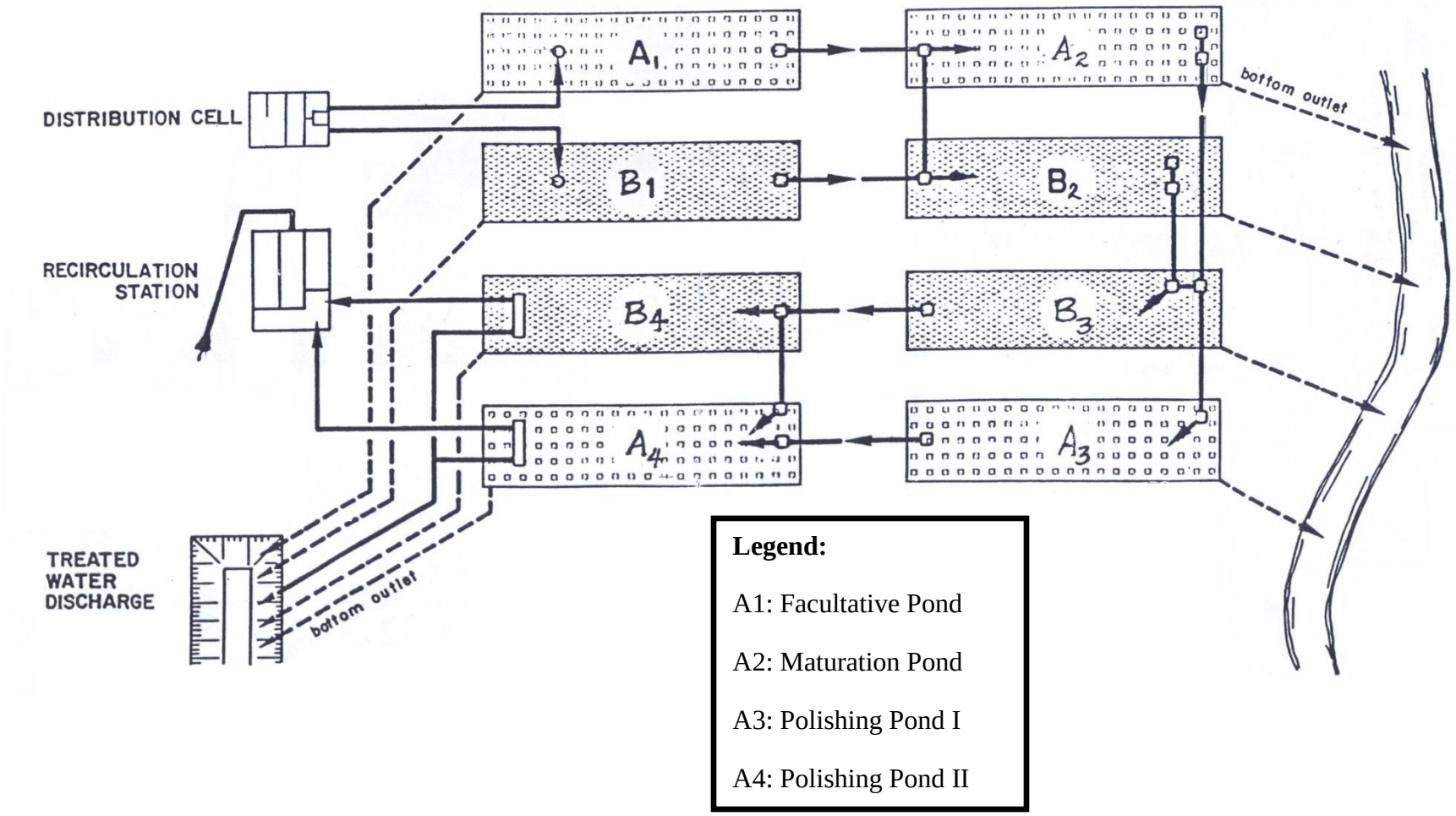


Figure 3.3: Process flow sheet of Kaliti wastewater treatment plant. Reproduced From (AAWSA,2002).

iv. Treated Water Recirculation Station

The treatment plant has 3 Archimedes screws, each capable of raising an incoming flow of 80l/s with the following specification;

- Diameter: 800mm
- Speed: 51rpm
- Motor: 7.5 and 10Hp
- Max first phase recycling capacity: 13,824m^3/d

During the second phase, the rotation speed increased to 610rpm and the input flow to 105l/s.

3.3 Description of The Night Soil Treatment

The treatment plant consists of a set of night soils treatment work comprising:

i. Four night soil digestion tanks

The night soil treatment has 4 night soil digestion tanks with a unit capacity of 1130m^3 and a sludge extraction system by means of a submersible mobile pump at a flow rate of 20m^3/h at 15m total head.

ii. Supernatant pumping station

Supernatant pumping station equipped with a single vane impeller pump (submersible type) of a flow rate of 54m^3/h at 6.5m total head.

iii. Drying beds

The night soil treatment has also a sludge drying area comprising 26 beds of 7.5m by 20.06m (Area = 3,912 m^2).

3.4 Effluent Guidelines and Standards

A significant element in wastewater disposal is the potential environmental impact associated with it. Environmental standards are developed to ensure that the impacts of treated wastewater discharges into ambient waters are acceptable. Standards play a fundamental role in the determination of the level of wastewater treatment required and in the selection of the discharge location and outfall structures.

Regulations and procedures vary from one country to another and are continuously reviewed and updated to reflect growing concern for the protection of ambient waters. The Environmental Protection Agency of Ethiopia developed general national pollutant discharge limit to control water pollution by regulating point sources that discharge pollutants into waters. The guidelines developed by Ethiopian EPA for general effluent standards are depicted in the following table.

Table 3.2: Guideline for general effluent discharge standards set by EPA of Ethiopia

S.No	**Parameters**	**Limit**
1	pH	6-9
2	Temperature	40˚c
3	TSS	100mg/l
4	TDS	3000mg/l
5	EC	1000 µs/cm(@20˚c)
6	Total Ammonia	30mg/l
7	Total Nitrogen	80mg/l
8	Total Phosphate	10mg/l
9	BOD_5	80mg/l
10	COD	250mg/l
11	Total coliform	400cfu/100ml

Chapter 4- Materials and Methods

4.1 Materials

4.1.1 Historical Data

The historical data used in this work were obtained from a biological wastewater treatment plant, namely the Kaliti wastewater treatment plant, found in the Addis Ababa district, Ethiopia. Available Laboratory records of Biochemical Oxygen Demand (BOD_5) ,Total Suspended Solids (TSS), pH, Total Dissolved Solids (TDS), Electrical Conductivity (EC), Dissolved Oxygen (DO), Chemical Oxygen Demand (COD), Total Volatile Solids (TVS), Ammonia (NH_3), Nitrite (NO_2^-), Nitrate (NO_3^-), Sulphate (SO_4^{2-}) and Phosphate(PO_4^{3-}) for over 30 months ,from September 2003 to April 2006, were obtained from the plant laboratory.

The available data for the Kaliti WWTP were carefully investigated. After considering the available options for modelling the treatment plant performance, it was decided to relate the outputs of the polishing pond II effluent stream to the inputs of the distribution cell stream. This is because of the unavailability of sufficient data at each stages of the biological treatment plant. The measurements of the 13 operational variables collected over the two-year period was satisfactory as it covers all probable seasonal variations in the studied variables. The measurements were performed in the plant almost every 15 days. Table 4.1 below shows the various input parameters for the prediction of the treatment plant performance.

Table 4.1: Selected operational variables of Kaliti WWTP

No	Parameters	Nomen clature	Unit	No	Parameters	Nomen clature	Unit
1	pH	pH	-	8	Total Volatile Solids	TVS	mg/l
2	Total Dissolved Solids	TDS	mg/l	9	Ammonia	NH_3	mg/l
3	Electrical Conductivity	EC	µs/cm	10	Nitrate	NO_2	mg/l
4	Dissolved Oxygen	DO	mg/l	11	Nitrite	NO_3	mg/l
5	Chemical Oxygen Demand	COD	mg/l	12	Sulphate	SO_4^{2-}	mg/l
6	Biochemical Oxygen Demand	BOD_5	mg/l	13	Phosphate	PO_4^{3-}	mg/l
7	Total Suspended Solids	TSS	mg/l				

4.1.2 Software

NeuralWorks Predict® software will be used for designing, building, training, testing and deploying neural networks to solve the prediction problem raised in this work. Neuralworks Predict® software is selected because it is an application that integrates all the components needed to apply neural computing to a wide variety of problems. It is different from other neural network software applications in that it automates much of the painstaking manipulation, selection, and data pruning that monopolizes most of the time in building a real world neural network application. This powerful system combines neural network technology with fuzzy logic, statistics and genetic algorithms. The algorithms used in NeuralWorks Predict® are grounded in statistics. Key concepts such as ridge regression, maximum likelihood and cross-validation are seamlessly integrated with neural techniques.

In addition to NeuralWorks Predict®, a high-performance interactive software package for scientific and engineering computation, MATLAB® software package with optimization and neural network toolbox (Version 6.0.3), will be used for development of neural network models. In addition, MATLAB® will be used to perform statistical analysis and representation of historical data graphically. Both written matlab script and neural network tool box (nntool) will be used for development of ANN models to predict the performance of the treatment plant.

Visual basic for Applications (VBA) will be used in this work for developing an application software. The application software will be used for non-linear input variable selection using the concept of information theory called mutual information.

JMP PRO® software and Microsoft excel will be used in this work mainly for all data cleaning and integration tasks that will be required for the development of the ANN models.

4.2 Methods

4.2.1 Selection of Appropriate Model Outputs

In a wastewater treatment plant, there are certain key parameters which can be used to assess the plant performance. These parameters could include Biological Oxygen Demand (BOD), pH, temperature, chemical dosages, Suspended Solid (SS) and Chemical Oxygen Demand (COD). And based on a priori knowledge and the available data obtained, the following operational and quality parameters were selected to measure the effectiveness (performance) of the wastewater treatment plant in this work.

1. pH
2. BOD_5
3. COD
4. NH_3
5. TDS

These parameters are considered as good indicators of the plant performance as they represent the organic matter and nutrient-removal capabilities of wastewater treatment processes.

4.2.2 Data Pre-processing

The raw plant data available for training and testing the ANN have been examined for completeness. The missing values have been estimated by interpolation. Data refining was performed on the raw experimental data by excluding all outliers which were unusual points for the outlier removed data based experiment. The data refining was accomplished by removing measurements that were not within the range of $\pm 3\sigma$, standard deviations around the group or design cell mean.

So as to ensure the statistical distribution of the values for each net input and output is roughly uniform, a standard procedure for data preparation called data scaling was carried out. The data sets are scaled so that they fall within a specified range of -1 to +1.

4.2.3 Input Selection

The methodology followed in the selection of inputs in this work both considered the significance of the inputs as well as the independency of the inputs. Both analytical model free approach to input selection, in which a non-linear statistical dependence measure of significance is used to assess the strength of the relationship between potential model inputs and outputs ,and model based global approach, where a global optimization algorithm, genetic

algorithm, is used to select the combination of inputs that maximizes model performance approach was basically applied in this work.

Because of the fact that redundant inputs increase the likelihood of over fitting (over training) and also the inclusion of redundant model inputs introduces additional local minima in the error surface in weight space, input independence was considered in addition to input significance. Dimensionality reduction approaches were followed to cater for input independence (redundancy) for the model based input selection approaches described above. The aim of the technique used is to reduce the dimensionality of the input space by eliminating correlated candidate inputs. For the second case, constructive stepwise model building process, i.e. the partial mutual information input variable selection algorithms, which combine a stepwise partial modelling approach that caters for input independence (redundancy) with an analytical measure of statistical dependence that caters for input significance was applied in this work.

4.2.4 Data Division

As part of the ANN model development process, the available data are generally divided into training, testing and validation subsets. The training set is used to estimate the unknown connection weights, the testing set is used to decide when to stop training in order to avoid over fitting and/or which network structure is optimal, and the validation set is used to assess the generalization ability of the trained model. Supervised data division approaches viz. trial and error, and optimization based approach using genetic algorithm are used in this work. The explicit goal of supervised data division methods is to ensure that the statistical properties of the various subsets are similar. This was achieved by using a trial-and-error approach, as part of which manual adjustments are made to the composition of the various subsets until an arbitrarily satisfactory level of agreement between the statistical properties of the various data subsets were reached, and by using a formal optimization approach to minimize a measure of difference between the statistical properties of the data subsets.

4.2.5 Model Architecture Selection

Model (network) architecture determines the overall structure and information flow in ANN models. Consequently, it has a significant impact on the functional form of the relationship between model inputs and output(s), f (•).Traditionally, ANN architectures have been divided into feed forward and recurrent networks. In feed forward networks, the information propagation is only in one direction, i.e. from input layer to the output layer. Multilayer

Perceptrons (MLPs) are the most common form of feed forward model architecture and are the architectures used in this work. An MLP uses three or more layers of artificial neurons with linear aggregation functions and linear and/or non linear activation functions.

4.2.6 Model Structure Selection

Model (network) structure, together with model (network) architecture, defines the functional form of the relationship between model inputs and output(s), f(•). Determination of an appropriate network structure involves the selection of a suitable number of hidden nodes, how they are arranged and how they process incoming signals. The optimal network structure generally strikes a balance between generalization ability and network complexity (e.g. network size and the number of free parameters). So as to simultaneously optimize network parameters and structure and to result in the best ANN structure and/or parameters; global methods based on competitive evolution found in nature ,genetic algorithm, was used in this work.

In developing a neural network model for the application, the performance of the models developed (speed of convergence and accuracy of prediction) were maximized by investigating the following network characteristics before experimenting with any future tests.

i. Selecting the Proper Transfer Function

Generally, the hyperbolic tangent and sigmoid functions are appropriate for most types of networks, especially for prediction problems. The hyperbolic tangent function was preferred over the sigmoid function in this work for the following reasons:

1. The output varying from -1 to +1 for the hyperbolic tangent and only 0 to 1 for the sigmoid function. This means that the hyperbolic tangent function has a negative response for a negative input value and a positive response for a positive input value, while the sigmoid function always has a positive response.

2. The slope of the hyperbolic tangent is much greater than the slope of the sigmoid function. Which means the hyperbolic tangent function is more sensitive to small changes in input.

ii. Network Configuration

In this work both Multiple-Input-Single-Output (MISO) and Multiple-Input-Multiple-Output (MIMO) configurations, using the same number of training and testing example sets, for both raw and outlier removed operational data were tested. In training a separate MISO network

for each of the five outputs (pH, BOD_5, COD, NH_3 and TDS), networks with inputs selected based on both the PMIS and GA algorithm were used. Tables 4.2 and 4.3 show the prediction network specification used for training the MISO and MIMO networks.

Table 4.2: Network specification used for PMIS based input selection

S.No.	Output Variable	No. of Input Variables		Configuration
		For raw operational Data	For outlier removed	
1	pH	6	6	MISO
2	TDS	6	5	MISO
3	COD	7	7	MISO
4	BOD_5	6	6	MISO
5	NH_3	7	6	MISO

Table 4.3: Network specification used for GA based input selection

S.No.	Output Variable	No. of Input Variables	Configuration
1	pH	13*	MISO
2	TDS	13*	MISO
3	COD	13*	MISO
4	BOD_5	13*	MISO
5	NH_3	13*	MISO
6	pH,TDS,COD,BOD_5 and NH_3	13*	MIMO

* The potential model input variables supplied to genetic algorithm before data transformation and variable selection.

iii. Learning rule selection

Learning rules that are supported in NeuralWorks Predict®, adaptive gradient and kalman filter will be used in developing the predictive models. The adaptive gradient learning rule uses back propagated gradient architecture to guide an iterative line search algorithm. The adaptive gradient process is repeated until a local minimum of the objective function has been found. The kalman filter learning rule considers the weight to be states and the desired output to be the observations within a discrete state space to search along that direction for a minimum of the objective function. For the data found to be very noisy, the kalman learning rule was used and for the clean and moderate noisy data, the adaptive gradient was used.

iv. Initial weight-factor distribution

NeuralWare's NeuralWorks Predict® software package sets the initial weight-factor distribution to a fairly narrow range, and will allow it to broaden using high learning rates and high momentum coefficients in the early stages of the training process.

4.2.7 Model Training

Computationally efficient deterministic approach, first-order Gradient method (back-propagation) and global optimization methods, genetic algorithm, because of their increased ability to find global optima in the error surface, were used to conduct the ANN training. The aim of model calibration (ANN training) is to find a set of model parameters that enables a model with a given functional form to best represent the desired input/output relationship.

4.2.8 Model Evaluation

In order to determine which network structure is optimal, the performance of a calibrated model was evaluated against one or more criteria. In this work, the ANN model performance will be assessed using a quantitative error metric, squared error. Squared errors are based on the squares of the differences between actual and modeled output values. The employed metrics belonging to this category include average absolute error, maximum absolute error, root mean square error and R Correlation.

Chapter 5- Results and Discussions

5.1 Data Pre-Processing

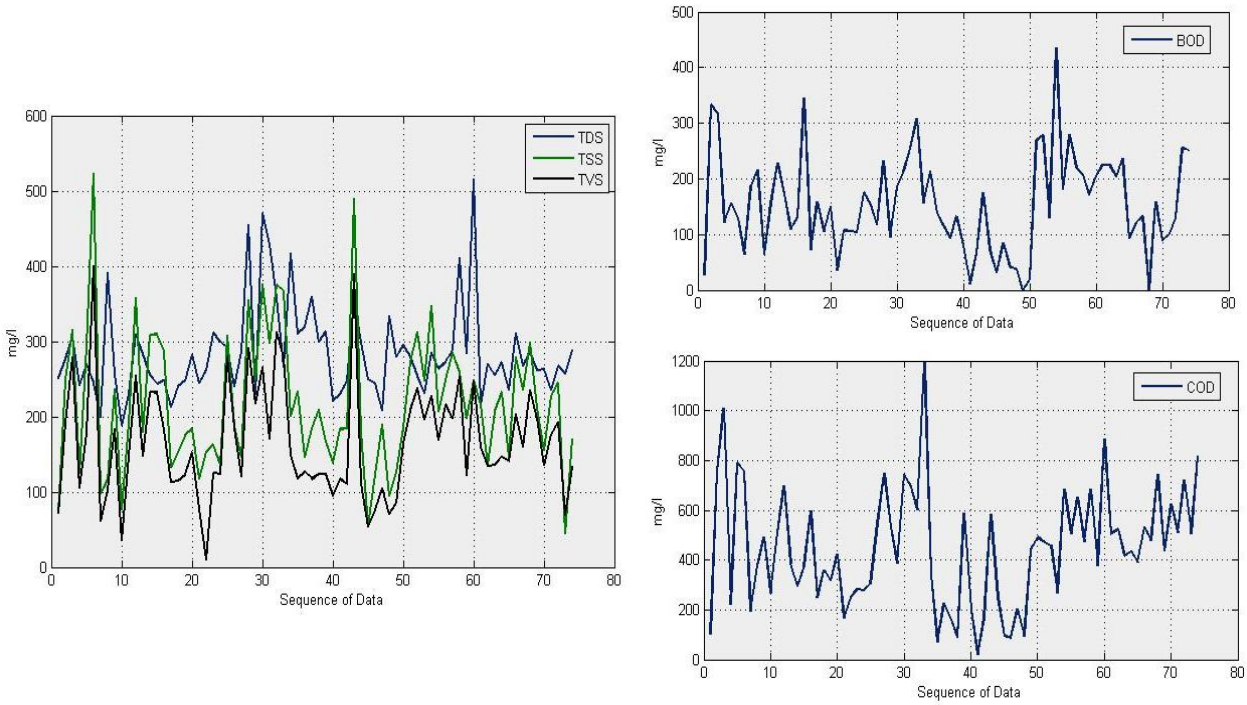
The object of data pre-processing was to produce the training set of the neural network, which represents the relationship of network inputs and outputs to make the problem much more suitable for the network. The major tasks carried out in data preprocessing were: Data cleaning, Data integration and Data transformation.

To handle the data imperfections, data cleaning algorithms was applied which can fill in missing values, identify outliers and smooth out noisy data, and correct inconsistent data. Careful integration of the data was also done to reduce/avoid redundancies and inconsistencies and improve data quality. There were more than 18 measurements collected but reduced to 13 operational and wastewater quality parameters by applying the above articulated data preprocessing techniques.

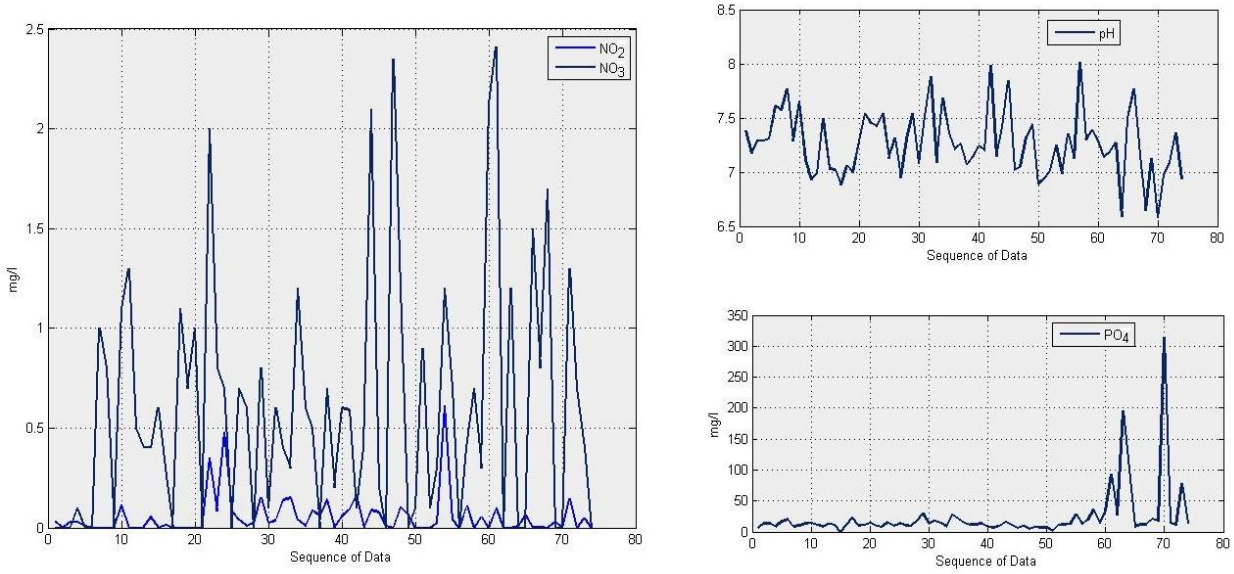
The raw laboratory data of Kaliti wastewater treatment plant, for the 13 operational and wastewater quality parameters, are directly plotted using MATLAB® software for both the potential input variables and target variables as seen in the following graphs.

The values for each data field were examined to determine the type of field data. And all the data fields were encoded in ways that are appropriate for neural network processing. Various transforms was applied using NeuralWorks Predict® in order to get a more uniform distribution of records over each field. This helped the neural network learn from small differences in the records.

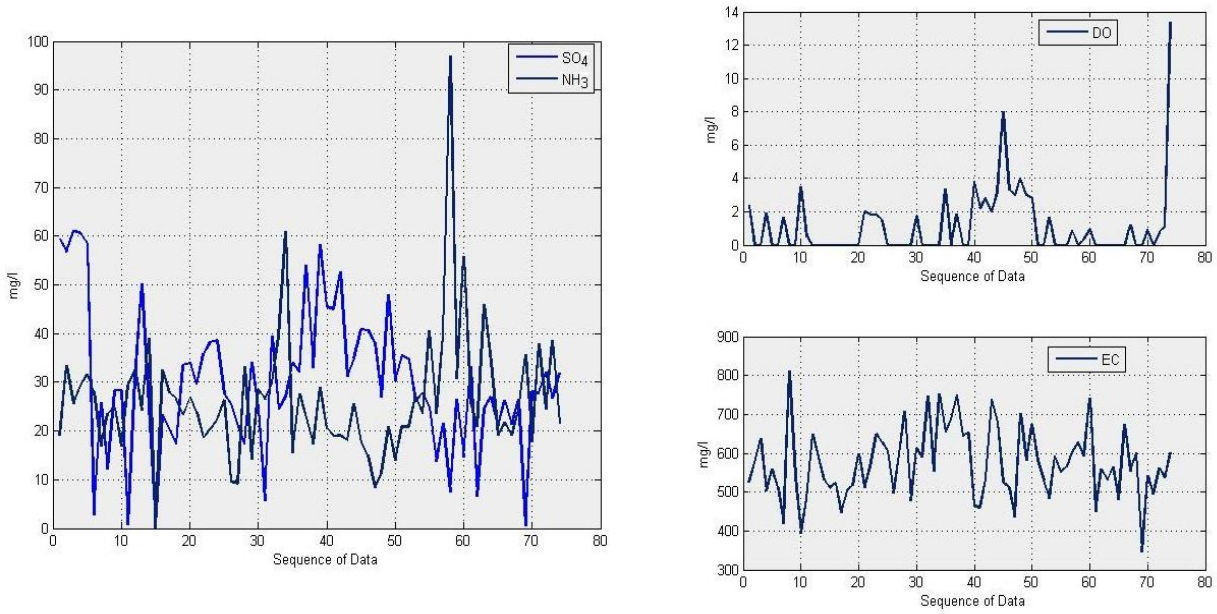
The ranges to which the input and output should be scaled, for presentation to the network is set between limits of -1 and 1, having the average value set at 0. NeuralWorks Predict® then computed the proper scale and offset for each data field. Real world values are then scaled to network ranges for presentation to the network. After the network has produced a network scaled results, the result was de-scaled to real world units.


TDS
TSS
TVS
mg/l
Sequence of Data
BOD
mg/l
Sequence of Data
COD
mg/l
Sequence of Data

(a)

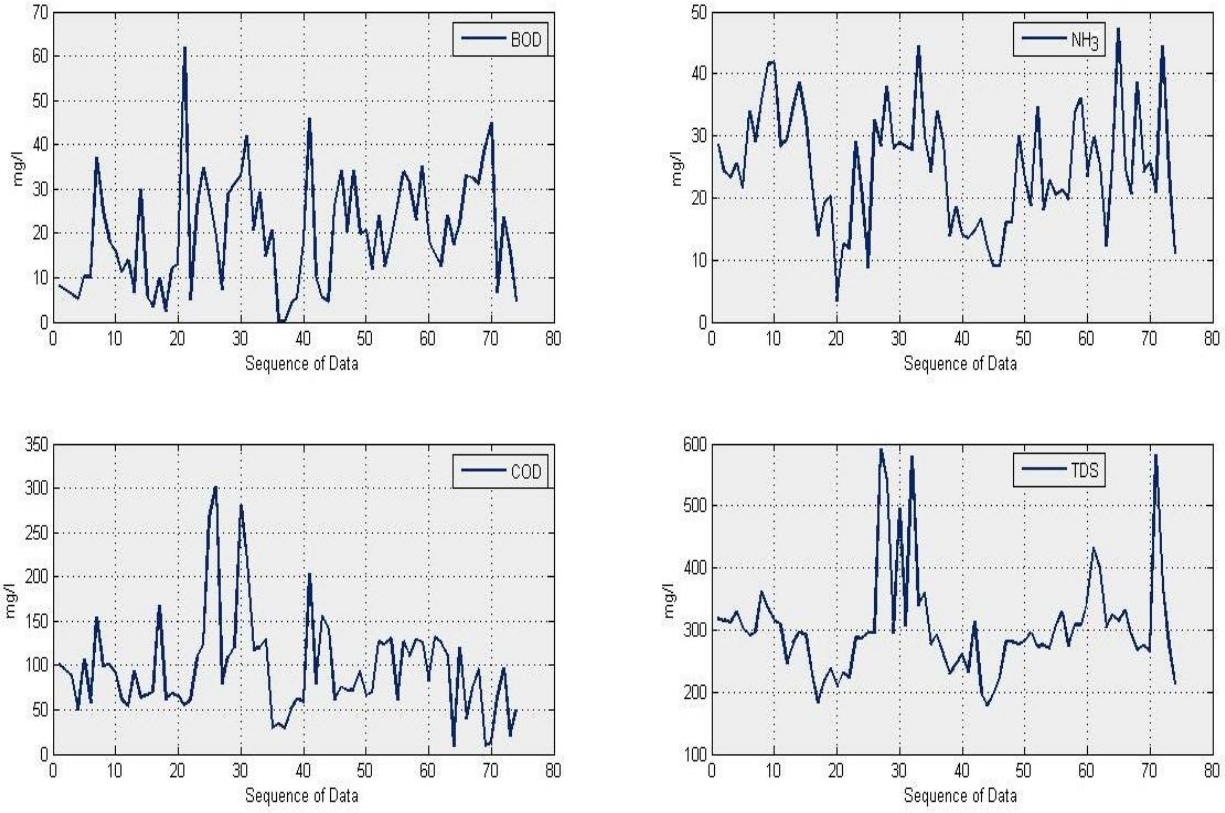

NO2
NO3
mg/l
Sequence of Data
pH
Sequence of Data
PO4
mg/l
Sequence of Data

(b)

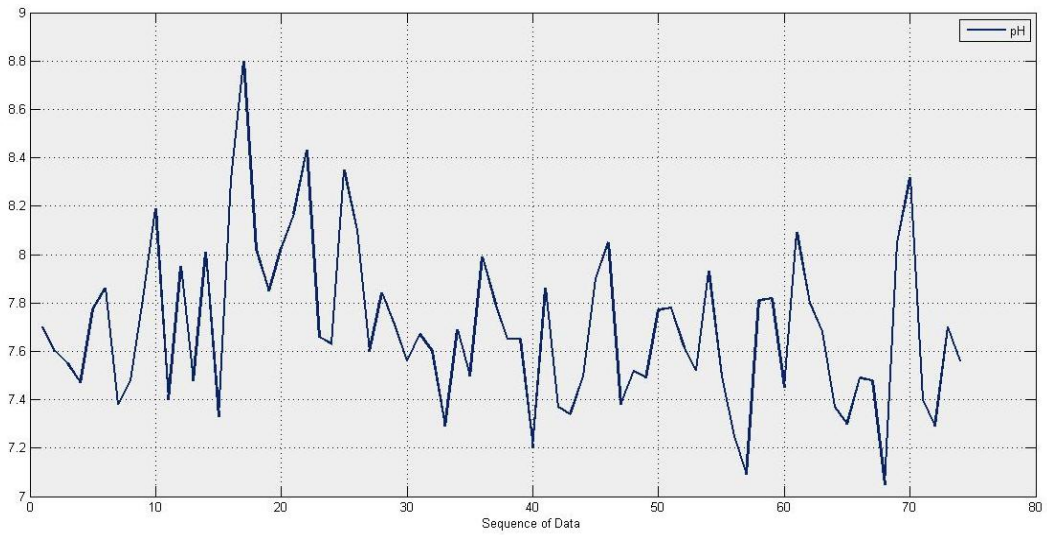


(c)

Figure 5.1: Potential input variables for the raw operational data of Kaliti wastewater treatment plant (a) TDS,TVS,TSS,BOD and COD,(b) NO_2,NO_3,pH and PO_4,(c) SO_4,NH_3,DO and EC.



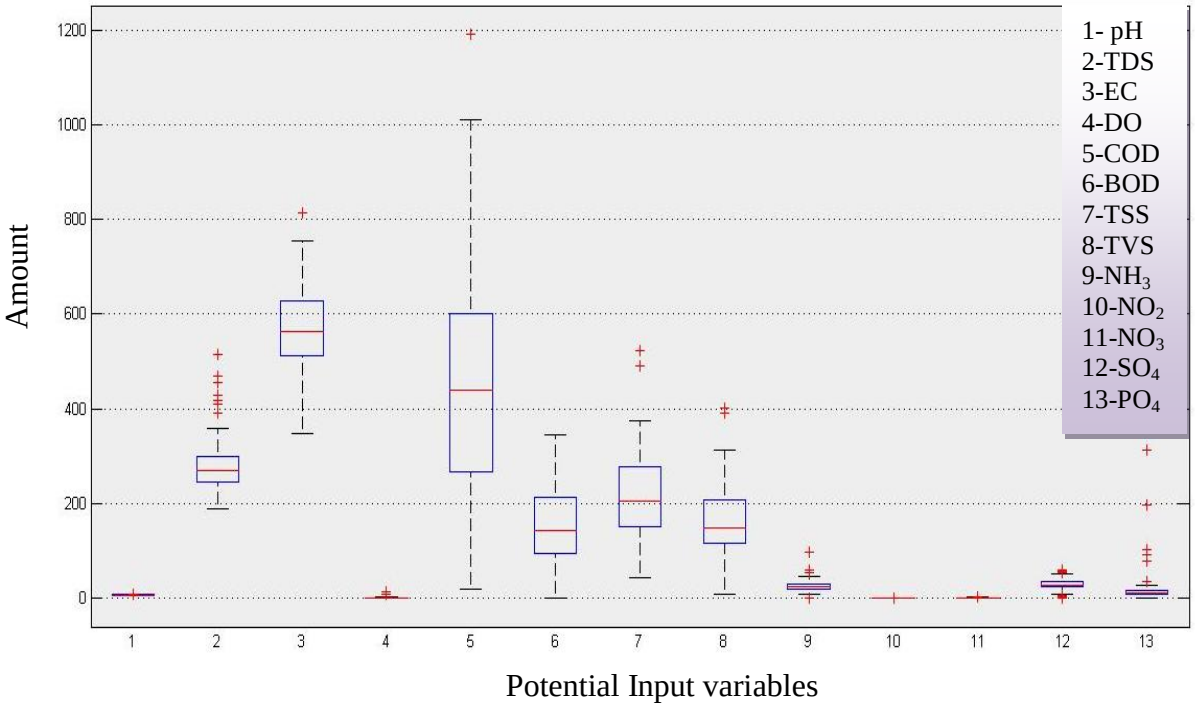
(a)



(b)

Figure 5.2: Appropriate Output variables for the raw operational data of Kaliti wastewater treatment plant.(a) BOD.NH_3,COD,and TDS.(b) pH.

Because of the likely effect of outliers have on the performance of a neural network model, an outlier removed data was also used for development of a robust ANN model in addition to the raw operational data for the sake of comparison. Data refining was performed on the raw experimental data by excluding all outliers which were unusual points. The data refining was accomplished by removing measurements that were not within the range of $\pm 3\sigma$, standard deviations around the group or design cell mean. The missing values have been estimated by interpolation using JMP PRO® statistical software. Outliers are removed by plotting and examining statistics. These plots summarize each variable by three components; a central line to indicate central tendency or location; a box to indicate variability around this central tendency and whiskers around the box to indicate the range of the variable. This is shown in Fig.5.3 for the input data and target data. The plots illustrate the extent of outlier density in each variable as indicated by the points extending beyond the whiskers. In addition, it shows the range of each variable and, consequently, the efficiency of the plant treatment.



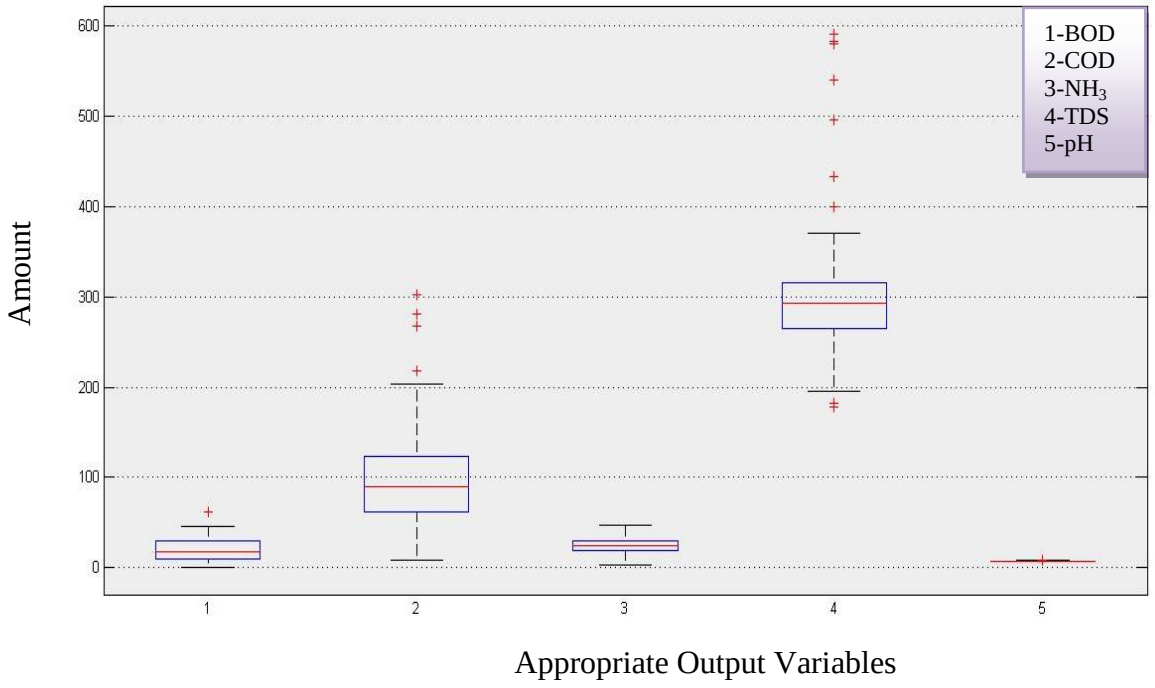


Figure 5.3: Box diagrams for the potential input data and appropriate target data

The Basic statistics descriptors for the appropriate model output selected and the potential model input variables for both the raw and outlier removed operational data are indicated in tables 5.1 to 5.4.

Table 5.1: Basic statistics descriptors for appropriate selected model outputs of raw operational data

S. No	Appropriate Model Output Variables	Nomenclature	Unit	Min	Max	Mean	Median	Mode	Standard Deviation
1	pH	pH	-	7.05	8.8	7.698	7.655	7.48	0.3293
2	Total dissolved solids	TDS	mg/l	178	591	303.6	292.5	277	83.87
3	Chemical Oxygen Demand	COD	mg/l	9	302	96.65	89.65	62	56.34
4	Biochemical Oxygen Demand	BOD_5	mg/l	0.061	62	19.8	18.13	12	12.67
5	Ammonia	NH_3	mg/l	3.3	47.3	24.77	24.2	29	9.529

Table 5.2: Basic statistics descriptors for potential model input variables of raw operational data

S. No	Potential Input Variables	Nomenclature	Unit	Min	Max	Mean	Median	Mode	Standard Deviation
1	pH	pH	-	6.58	8.01	7.26	7.255	7.29	0.295
2	Total dissolved solids	TDS	mg/l	189	516	284.9	270	244	62.74
3	Electrical Conductivity	EC	µs/cm	347	813	573.1	562.5	484	91.29
4	Dissolved Oxygen	DO	mg/l	0	13.4	1.126	0	0	2.054
5	Chemical Oxygen Demand	COD	mg/l	19	1192	447.9	439	380	237.6
6	Biochemical Oxygen Demand	BOD_5	mg/l	0	346	152.4	144	130	83.85
7	Total Suspended Solids	TSS	mg/l	44.2	523	218.2	204.1	184	93.04
8	Total Volatile Solids	TVS	mg/l	9.48	401	165.1	148.3	124	75.87
9	Ammonia	NH_3	mg/l	0	97	26.26	24.3	20.83	13
10	Nitrate	NO_2	mg/l	0	0.61	0.0551	0.0182	0	0.1019
11	Nitrite	NO_3	mg/l	0	2.41	0.593	0.45	0	0.6177
12	Sulphate	SO_4^{2-}	mg/l	0.47	61	30.16	28.2	17.4	14.32
13	Phosphate	PO_4^{3-}	mg/l	0	314	23.24	12.6	11.08	43.52

Table 5.3 **:** Basic statistics descriptors for appropriate selected model outputs of outlier removed data

S. No	Appropriate Model Output Variables	Nomenclature	Unit	Min	Max	Mean	Median	Mode	Standard Deviation
1	pH	pH	-	7.05	8.8	7.735	7.705	7.48	0.3498
2	Total dissolved solids	TDS	mg/l	182	591	302.4	290	295	89.71
3	Chemical Oxygen Demand	COD	mg/l	29.1	302	97.9	85.5	65	56.35
4	Biochemical Oxygen Demand	BOD_5	mg/l	0.061	62	20.7	20.33	12	13.18
5	Ammonia	NH_3	mg/l	3.3	47.3	25.4	25.03	29	10.28

Table 5.4: Basic statistics descriptors for potential model input variables of outlier removed data

S. No	Potential Input Variables	Nomenclature	Unit	Min	Max	Mean	Median	Mode	Standard Deviation
1	pH	pH	-	6.65	7.88	7.233	7.225	7.02	0.2603
2	Total Dissolved Solids	TDS	mg/l	189	360	268.9	268	240	38.84
3	Electrical Conductivity	EC	µs/cm	394	750	564.1	557	674	84.84
4	Dissolved Oxygen	DO	mg/l	0	4	0.9464	0	0	1.34
5	Chemical Oxygen Demand	COD	mg/l	19	747	379.7	387.5	266	199
6	Biochemical Oxygen Demand	BOD_5	mg/l	0	346	132.4	119	0	82.65
7	Total Suspended Solids	TSS	mg/l	73	375	202.2	186.5	184	75.35
8	Total Volatile Solids	TVS	mg/l	36	312.5	154.4	144.3	117.3	62.61
9	Ammonia	NH_3	mg/l	9.21	40.6	22.86	22.15	20.83	7.282
10	Nitrate	NO_2	mg/l	0	0.15	0.03976	0.01115	0	0.0486
11	Nitrite	NO_3	mg/l	0	1.7	0.5331	0.544	0	0.4588
12	Sulphate	SO_4^{2-}	mg/l	6.6	59.3	31.08	29	28.4	10.48
13	Phosphate	PO_4^{3-}	mg/l	1.279	28.9	12.89	11.75	1.279	5.86

5.2 Input Selection

Generally, the data sets obtained from the municipal WWTP are high dimension and these variables are often highly correlated. These high-correlations can be an obstacle for modelling. In addition, increasing number of input variables increases the complexity of the model and it takes a longer time to train and forecast effluent quality, and it may also introduce unwanted noise. In this study two approaches were followed in order to develop robust ANN models. The first is the selection of the input variables using recently developed non linear input selection algorithm called the Partial Mutual Information-based Input Selection (PMIS) algorithm. A separate software program was successfully developed, using Visual basic for Applications (VBA) that works in an excel interface ,for estimation of the partial mutual information based on an algorithm modified by May et.al (2008) and originally developed by Sharma (2000).

Secondly, a genetic algorithm was employed to identify the best set of input variables from the set of all input variables and transformations of input for the models developed using a function incorporated in NeuralWorks Predict® software. The input variables are preprocessed, using the genetic algorithm, before they are fed to the back propagated ANN. A genetic algorithm was used since it efficiently explores the large space of subsets of possible input variables.

In this study, the GA-PMIS based models of the WWTP are developed for comparison and provides a reasonable method for reducing any redundant attributes and determining the proper inputs for the model. Table 5.5 lists the partial mutual information score. This summary table reports the partial mutual information score for the best input at each iteration, and the corresponding critical values of mutual information using a bootstrap approach, and the Akaike Information Criterion (AIC) score. Because the distributions of the data may not hold the assumption of Gaussian data, the AIC criteria was considered to determine the optimum number of variables to use as inputs for a neural network model. The AIC method provides a general measure of the trade-off between information gain and the complexity introduced to the modelling domain by the addition of input variables. This criterion lends itself to clear and simple interpretation and is expected to provide consistent and reliable selection for any dataset. The input variables selected and their corresponding AIC value are indicated in bold in the PMIS summary table for each of output variables. The optimum cut-off point ,for inclusion of variables as inputs, was identified using Akaike test .i.e. the inputs are selected until $AIC(p) < AIC(k)$.

Table 5.5: Partial mutual information score for pH output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**EC**	0.12969	0.182995	0.21892	**8.15724**	**13.6045**
2	**pH**	0.10634	0.182995	0.21892	**19.6667**	**40.3745**
3	**PO_4^{3-}**	0.10021	0.182995	0.21892	**15.9086**	**41.8744**
4	**NO_3**	0.10042	0.182995	0.21892	**25.6733**	**68.7334**
5	**NO_2**	0.11132	0.182995	0.21892	**25.9166**	**78.3541**
6	**SO_4^{2-}**	0.11427	0.182995	0.21892	**40.0217**	**112.904**
7	DO	0.08884	0.182995	0.21892	123.783	42.6682
8	BOD_5	0.12563	0.182995	0.21892	142.501	48.2144
9	TSS	0.0902	0.182995	0.21892	155.532	52.9901
10	TVS	0.09781	0.182995	0.21892	158.707	54.7185
11	TDS	0.08183	0.182995	0.21892	161.976	57.8705
12	COD	0.09747	0.182995	0.21892	166.185	58.8754
13	NH_3	0.06387	0.182995	0.21892	171.075	61.637

Key:

I(x;y): Partial mutual information score between x input variable and y output variable.

MC-I*(95): Critical values of mutual information at each iteration using a bootstrap approach for 95 percentile

MC-I*(99): Critical values of mutual information at each iteration using a bootstrap approach for 99 percentile

AIC (k): Akaike information criterion where k is the number of variables

AIC(p): Akaike information criterion where p is the number of model parameters

Based on the AIC method of determining the optimum cut off point, EC, pH, PO_4^{3-}, NO_3, NO_2 and SO_4^{2-} were selected to build pH prediction model using the MISO configuration.

Table 5.6: Partial mutual information score for BOD_5 output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**SO_4^{2-}**	0.13421	0.182995	0.21892	**-0.284434**	**4.16065**
2	**TVS**	0.1184	0.182995	0.21892	**5.51212**	**25.4497**
3	**pH**	0.14562	0.182995	0.21892	**10.8223**	**51.8563**
4	**EC**	0.15121	0.182995	0.21892	**15.7893**	**78.8675**
5	**BOD_5**	0.13691	0.182995	0.21892	**21.0993**	**103.095**
6	**NO_3**	0.137	0.182995	0.21892	**27.6309**	**123.906**
7	TDS	0.10665	0.182995	0.21892	122.085	25.0926
8	TSS	0.11038	0.182995	0.21892	129.396	28.7152
9	NH_3	0.0972	0.182995	0.21892	133.327	27.703
10	COD	0.07968	0.182995	0.21892	141.234	31.0713
11	DO	0.06699	0.182995	0.21892	141.232	30.9582
12	PO_4^{3-}	0.04669	0.182995	0.21892	136.603	28.3612
13	NO_2	0.04336	0.182995	0.21892	149.525	40.0878

For development of BOD_5 prediction model using the MISO configuration, SO_4^{2-}, TVS, pH, EC, BOD_5, and NO_3 were selected based on their AIC scores.

Table 5.7: Partial mutual information score for COD output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**TDS**	0.10387	0.182995	0.21892	**9.13551**	**14.7384**
2	**EC**	0.18221	0.182995	0.21892	**-0.051866**	**12.6118**
3	**pH**	0.10879	0.182995	0.21892	**9.77131**	**38.8264**
4	**NH_3**	0.14233	0.182995	0.21892	**15.3158**	**58.0711**
5	**TVS**	0.14216	0.182995	0.21892	**28.4821**	**89.3261**
6	**COD**	0.12185	0.182995	0.21892	**24.0294**	**98.3515**
7	**NO_3**	0.10669	0.182995	0.21892	**30.2356**	**119.994**
8	TSS	0.10239	0.182995	0.21892	122.015	28.1301
9	BOD_5	0.08283	0.182995	0.21892	138.192	36.3207
10	SO_4^{2-}	0.07794	0.182995	0.21892	147.746	37.5833
11	NO_2	0.07271	0.182995	0.21892	152.398	40.6877
12	DO	0.02233	0.182995	0.21892	157.085	45.7153
13	PO_4^{3-}	0.00677	0.182995	0.21892	158.254	48.8161

7 input variables are selected using the AIC termination criterion viz. TDS, EC, pH, NH3, TVS, COD and NO_3

Table 5.8: Partial mutual information score for NH_3 output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**COD**	0.1752	0.182995	0.21892	**0.922012**	**6.32646**
2	**SO_4^{2-}**	0.10982	0.182995	0.21892	**4.80959**	**23.3596**
3	**DO**	0.10877	0.182995	0.21892	**6.91923**	**32.7975**
4	**TDS**	0.11135	0.182995	0.21892	**3.6548**	**47.0311**
5	**TVS**	0.11408	0.182995	0.21892	**10.3583**	**66.8946**
6	**EC**	0.09478	0.182995	0.21892	**11.5969**	**76.8172**
7	pH	0.08304	0.182995	0.21892	97.2517	14.9184
8	NO_2	0.08407	0.182995	0.21892	107.664	19.4915
9	TSS	0.08775	0.182995	0.21892	122.75	29.8076
10	BOD_5	0.08579	0.182995	0.21892	137.935	36.3664
11	NH_3	0.06708	0.182995	0.21892	146.706	40.4589
12	NO_3	0.0626	0.182995	0.21892	148.482	37.1121
13	PO_4^{3-}	0.00873	0.182995	0.21892	149.831	40.3936

Candidates which have high mutual information score for the first six iterations and are selected to be the input variables for development of NH_3 Model using the AIC criterion are COD, SO_4^{2-} , DO, TDS, TVS, and EC. TVS, pH, TSS, COD, DO, TDS and NH_3 are the selected input variables for TDS prediction using the termination criterion.

Table 5.9: Partial mutual information score for TDS output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**TVS**	0.1219	0.182995	0.21892	**2.27361**	**3.04323**
2	**pH**	0.16379	0.182995	0.21892	**17.362**	**39.5234**
3	**TSS**	0.12583	0.182995	0.21892	**12.7359**	**40.7143**
4	**COD**	0.11744	0.182995	0.21892	**22.0606**	**66.9851**
5	**DO**	0.16419	0.182995	0.21892	**10.5477**	**62.4068**
6	**TDS**	0.0838	0.182995	0.21892	**3.17881**	**68.6641**
7	**NH_3**	0.07655	0.182995	0.21892	**9.82708**	**86.4689**
8	BOD_5	0.09335	0.182995	0.21892	104.59	18.2866
9	EC	0.11626	0.182995	0.21892	115.095	21.5153
10	SO_4^{2-}	0.10971	0.182995	0.21892	124.973	21.4137
11	NO_3	0.10884	0.182995	0.21892	127.235	16.962
12	PO_4^{3-}	0.05553	0.182995	0.21892	123.717	15.4753
13	NO_2	0.05607	0.182995	0.21892	127.967	18.5296

Applying the same partial mutual information-based input selection for the outlier removed data, the following variables are selected and used as input variables for the corresponding output variable as shown in the following table.

Table 5.10: Selected input variables using PMIS for outlier removed data

S. No	Output Variables	Input variables Selected	Stopping Criterion
1	pH	pH,PO_4^{3-},NO_3,COD,EC,NH_3	AIC
2	BOD_5	pH,COD,NO_2,EC,TSS,NH_3	AIC
3	COD	EC,TVS,NH_3,NO_3, NO_2TDS,BOD_5	AIC
4	NH_3	COD,pH, SO_4^{2-},NO_2,TVS,EC	AIC
5	TDS	COD,pH,NO_2,NH_3,TSS	AIC

5.3 Data partitioning

The train, test, and validation datasets from the available input data was automatically selected using NeuralWorks Predict®. The data was partitioned in such a way that the test set is statistically close to the training set. In this work, the default setting was manipulated so that the software uses 20% of the data for validation, 70% of the data remaining out of the validation data for training and remaining 30% for testing.

Table 5.11: Data partition set for both the raw operational and outlier removed data

		For Raw operational Data		For Outlier removed data	
S.No	Data partition set	Records	Percentage	Records	Percentage
1	Training set	42	57	24	57
2	Validation set	14	19	8	19
3	Test set	18	24	10	24
4	Ignored set	0	0	0	0
	Total	74	100	42	100

The training set is a part of the input dataset used for neural network training, i.e. for adjustment of network weights. The validation set is a part of the data used to tune network topology or network parameters other than weights. To choose the best network (i.e. by changing the number of units in the hidden layer) the validation set is used. The test set is a part of the input data set used to test how well the neural network will perform on new data. The test set was used after the network is ready (trained), to test what errors will occur during future network application.

5.4 Model Training and Testing

Because of the lack of theoretical foundations, training a neural network requires a long trial and error process, experimenting different combinations of learning rates, momentum terms, transfer functions, and network architectures. The determination of the learning rates and other network parameters is fundamental to train the network successfully. So as to overcome this difficulty, genetic algorithms are used in this work to reduce the arbitrary nature of the

determination of the training parameters, improving the training process and, therefore, the forecasting performance of the network.

Using the above mentioned neural network design and software setting, training of the models was performed. During training, the weights of the neural network was adjusted in order to minimize the error between the network output and the target value for all of the records in the training set. In addition to adjusting weights, new processing elements was added automatically using NeuralWorks Predict® in order to decrease the error of the network. To ensure that the network does not over-fit the training data (by learning patterns specific only to the training set), the performance of the network on the test set was periodically evaluated. When performance on the test set begins to degrade, training was stopped.

NeuralWorks Predict® constructs the actual neural network incrementally, using a technique known as cascade correlation, a method of adding processing elements incrementally. Hidden units are periodically added, usually one or two at a time. Each time a hidden unit or pair of hidden units is added, weights are trained from several different initialization values. Each initialization is referred to as a candidate. The best candidate is established in the network, and then all the weights to the output node(s) of the network are retrained.

The developed models using the above mentioned design procedures and/or setting was evaluated first using the similarity of the training set performance and test set performance, performance statistics, which is one of the sign of a good model. This is because it is always possible to get good performance on a training set, but the important thing is to have it perform well on new data.

Basic model performance statistics with respect to the train and test data sets are shown in each modeled indicators of the wastewater treatment plant performance. The performance statistics were automatically computed for each prediction models. Another way undertaken to evaluate model performance is to compare individual target values to the corresponding predicted values produced by the model. i.e. plotting the target against the predicted values.

5.4.3 Modelling Results of MISO Configuration

Below are the robust neural network models selected based on the above stated criteria for each of the four setting, for the MISO network configuration, using NeuralWorks Predict® software.

5.4.3.1 pH Prediction Model

Selected models based on their statistical performance and testing the model using new data for prediction of pH are shown below for the four setting used to experiment with in this work viz. PMIS based model prediction for both the raw and outlier removed data and the GA based model prediction, again for both raw and outlier removed historical plant data. Based on this selected models for each setting, final selection of the robust model was made to identify a single model that can be used to predict pH using the MISO configuration. Table 5.12 lists the statistical performance results of these four models.

Table 5.12: Performance Statistics of selected models for pH prediction

Parameters	pH^1		pH^2		pH^3		pH^4	
	R	RMS	R	RMS	R	RMS	R	RMS
ALL	0.809895	0.193003	0.754066	0.216361	0.924419	0.132753	0.971741	0.081817
Train	0.906851	0.143436	0.770862	0.217702	0.986809	0.059495	0.99996	0.011625
Test	0.587535	0.272448	0.769214	0.213357	0.924419	0.132753	0.971741	0.081817
Selected ANN structure	12-30-1		15-27-1		6-50-1		14-43-1	

pH^1: Selected PMIS based pH prediction model for the raw operational data
pH^2: Selected GA based pH prediction model for the raw operational data
pH^3: Selected PMIS based pH prediction model for the outlier removed data
pH^4: Selected GA based pH prediction model for the outlier removed data

Key:

R [R Correlation]: The linear correlation between predicted outputs and target outputs, in problem domain units.

RMS [Root Mean Square Error]: The root mean square error between the predicted outputs and the target outputs.

From the performance statistics of screened models, the R and RMS are key indicators of how well a model performs and are used in this work accordingly. The R values for the pH^4 model on the training and test sets are close to each other (0.99996 and 0.971741), which means the model generalizes well and is likely to make accurate predictions when new data (data that is not from the training or testing dataset) is provided when compared to pH^1, pH^2 and pH^3 models. Furthermore, the correlation values are close to 1.0, which is another indication that the model performs well. Furthermore, the RMS value of this model is minimum as compared to the RMS value of other models.

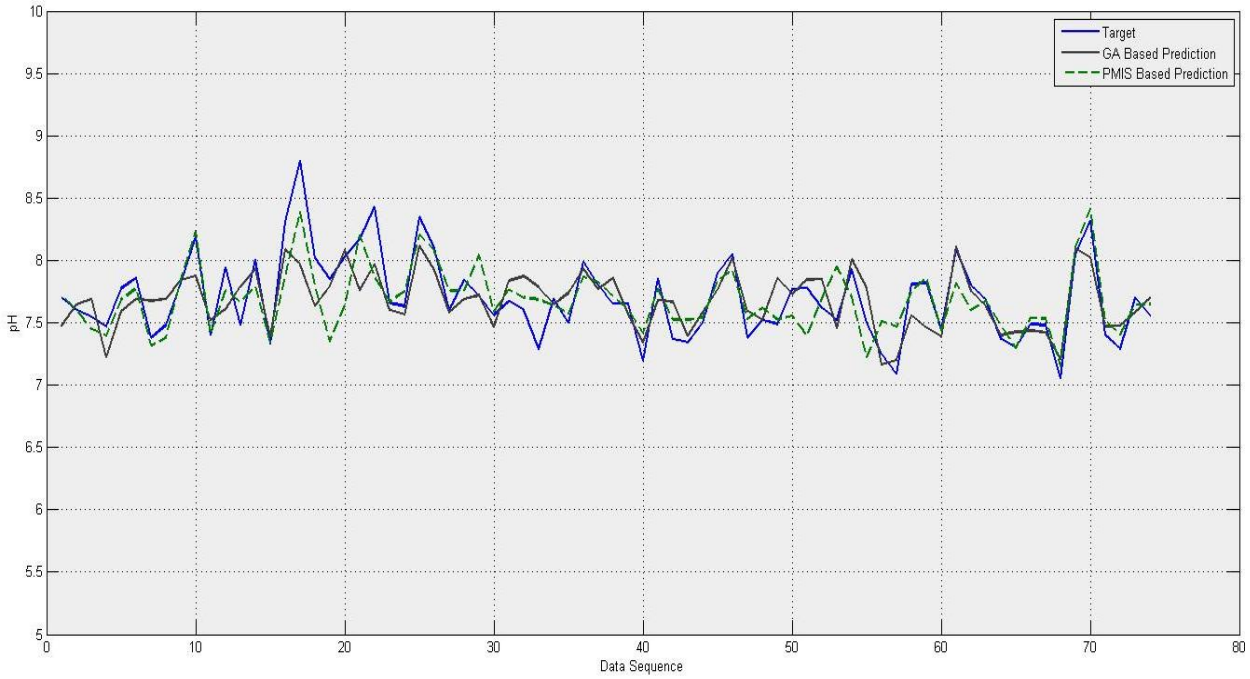


Figure 5.4: Prediction of pH based on the raw operational data.

The quality of match between the ANN-modeled and measured concentrations for the above mentioned four setting was plotted in figs.5.4 and 5.5. As can be seen in fig 5.5, Actual measured values of pH to predicted values in pH^4 indicate a good fit when compared to other models developed using the rest three setting. The suitable architecture for pH prediction was determined to consist of an input layer with 14 neurons, a hidden layer with 43 neurons and an output layer with one neuron (for predicted pH). To reach this result, different network structure/training rate combinations were evaluated.

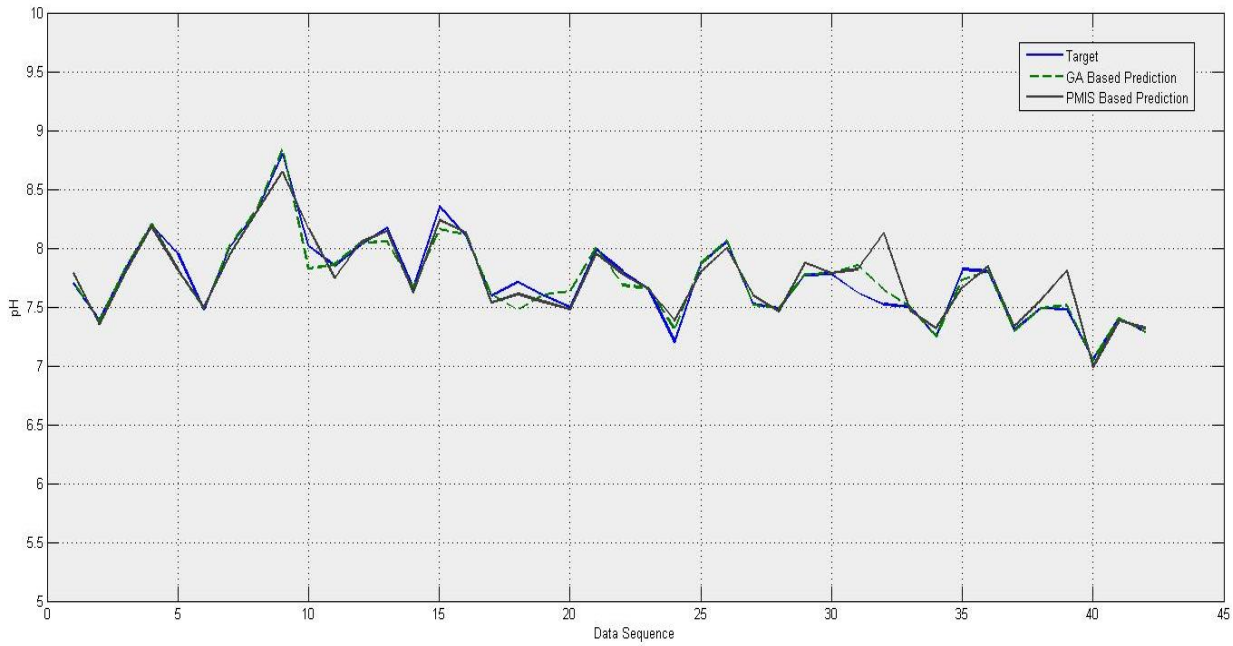


Figure 5.5: Prediction of pH based on outlier removed data.

5.4.3.2 BOD_5 Prediction Model

The testing datasets were compared to predicted values by the neural network models, to evaluate the models performance for the BOD_5 prediction ANN model. Fig 5.6 and 5.7 shows the measured versus ANN-modeled concentrations for the testing data sets for setting 1 – 4. Visual inspection indicates that the ANN models resulted in a good fit for the measured BOD_5 in configuration BOD_5^3 and BOD_5^4. R values of model BOD_5^3 and BOD_5^4 are also very close to each other as can be seen in the performance statistics table. Not only the values of R are close to each other in each of the setting, the R value is also close to 1 which is an indication of a performance of a good model.

Table 5.13: Performance Statistics of selected models for BOD_5 prediction

Parameters	BOD_5^1		BOD_5^2		BOD_5^3		BOD_5^4	
	R	RMS	R	RMS	R	RMS	R	RMS
ALL	0.839671	6.883232	0.737474	10.22612	0.941606	4.38657	0.942927	4.360813
Train	0.859275	6.619238	0.7869	10.59145	0.998487	0.754895	0.993631	2.001283
Test	0.808277	7.43525	0.66225	9.365356	0.941606	4.38657	0.942927	4.360813
Selected ANN structure	12-13-1		13-20-1		16-29-1		16-29-1	

BOD_5^1: Selected PMIS based BOD_5 prediction model for the raw operational data
BOD_5^2: Selected GA based BOD_5prediction model for the raw operational data
BOD_5^3: Selected PMIS based BOD_5 prediction model for the outlier removed data
BOD_5^4: Selected GA based BOD_5 prediction model for the outlier removed data

From the two candidate models described above; because of the reason that the R values are closer to each other for the train and test set, and RMS value is minimal in comparison, the GA based BOD_5 prediction model for the outlier removed data is selected as optimum model for predicting BOD_5.

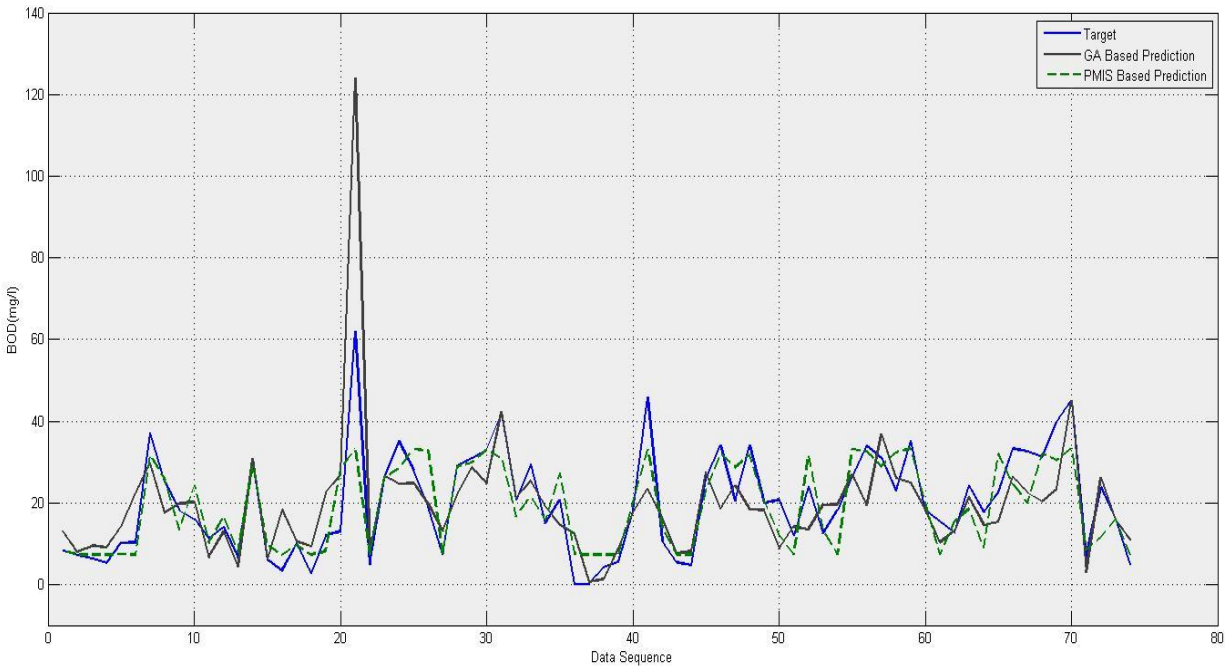


Figure 5.6 : Prediction of BOD_5 based on the raw operational data.

The analysis of the performance statistics is supported by plot of the measured values of BOD_5 against the predicted values for the four setting. Hence, the suitable optimum architecture for BOD_5 prediction was determined to consist of an input layer with 16 neurons, a hidden layer with 29 neurons and an output layer with one neuron.

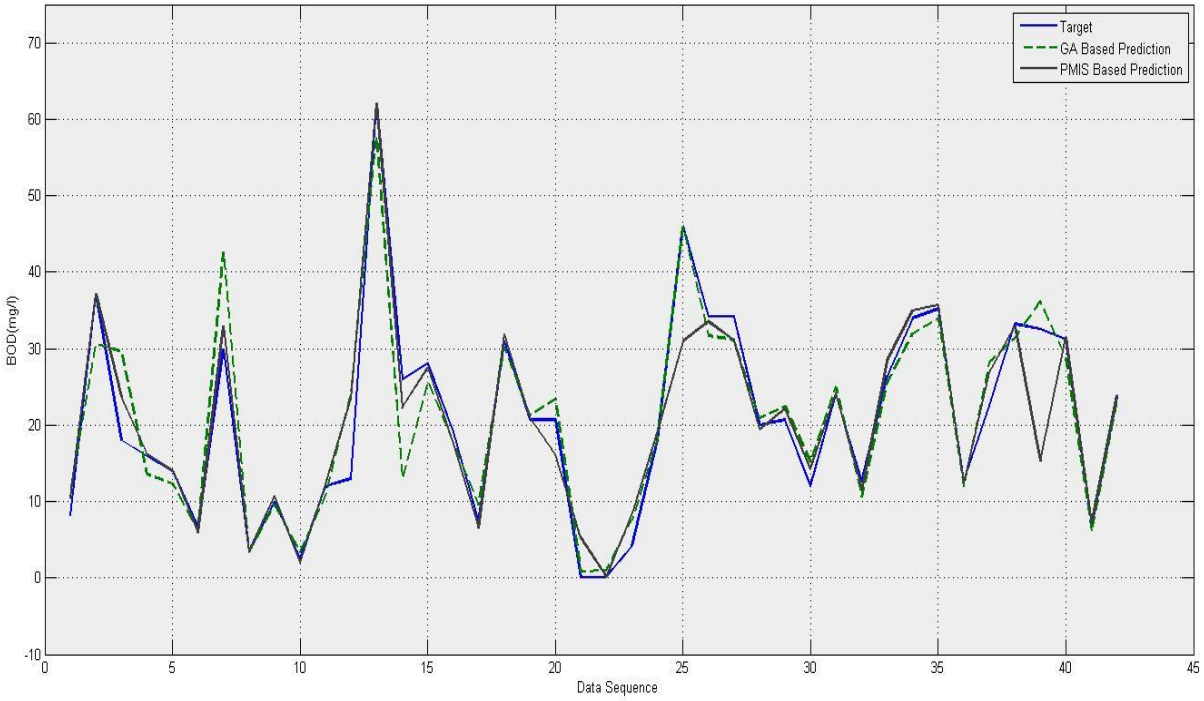


Figure 5.7: Prediction of BOD_5 based on outlier removed data.

5.4.3.3 COD Prediction Model

PMIS based COD prediction model for the outlier removed data are found to be the best and optimum models to predict COD for the MISO configuration with R values of 0.997739 and 0.979791 for training and test set respectively. The RMS value of this model is minimum as compared to other models selected for the rest of three configuration.

Table 5.14: Performance Statistics of selected models for COD prediction

Parameters	COD^1		COD^2		COD^3		COD^4	
	R	RMS	R	RMS	R	RMS	R	RMS
ALL	0.813688	32.52795	0.833831	30.89079	0.979791	11.15782	0.950419	18.1249
Train	0.888291	28.68833	0.897657	25.25064	0.997739	5.7497	0.982395	13.1907
Test	0.773882	39.73988	0.721641	40.69852	0.979791	11.15782	0.950419	18.1249
Selected ANN structure	7-20-1		13-18-1		14-50-1		13-20-1	

COD^1: Selected PMIS based COD prediction model for the raw operational data
COD^2: Selected GA based COD prediction model for the raw operational data
COD^3: Selected PMIS based COD prediction model for the outlier removed data
COD^4: Selected GA based COD prediction model for the outlier removed data

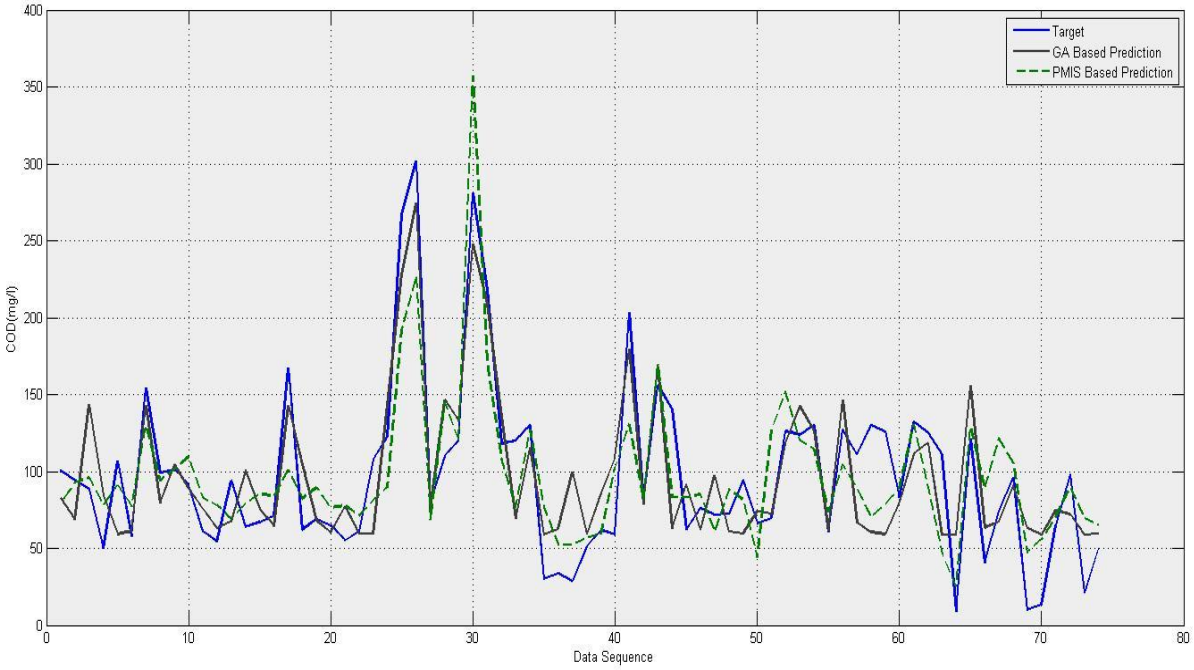


Figure 5.8: Prediction of COD based on the raw operational data.

As can be seen in fig 5.9, the plot of the target value against the predicted test for the PMIS based COD prediction model for the outlier removed data shows a quality match as compared to other settings against the measured value of COD. An input layer with 14 neurons, a hidden layer with 50 neurons and an output layer with one neuron was determined to be the optimum architecture for COD prediction.

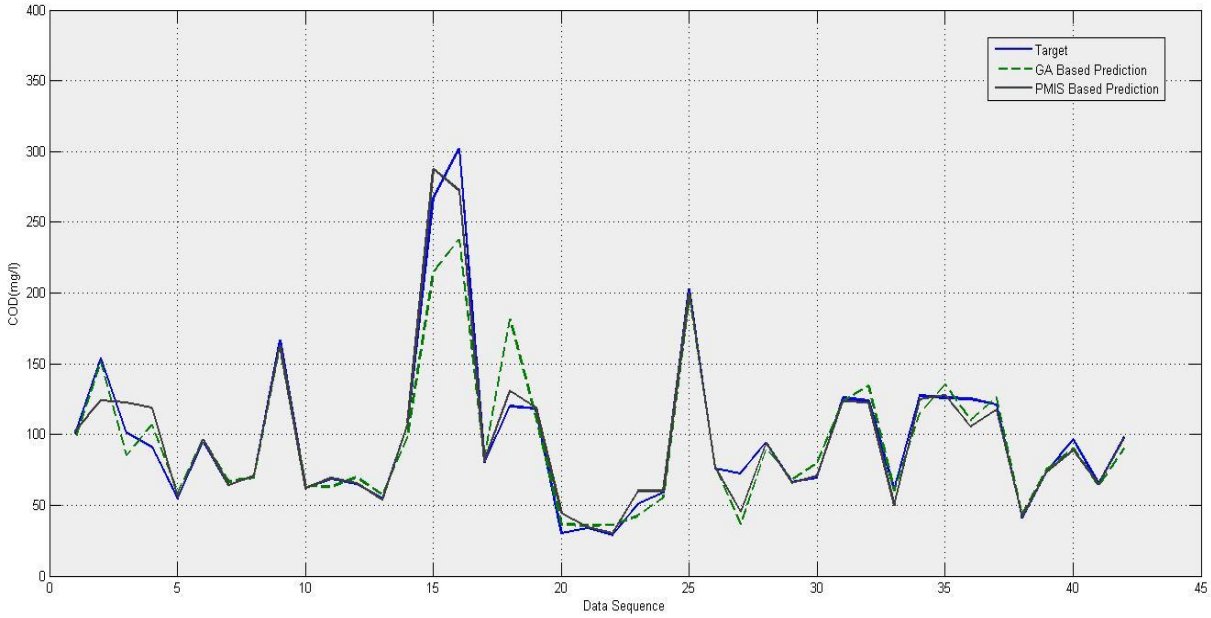


Figure 5.9: Prediction of COD based on outlier removed data.

5.4.3.4 NH_3 Prediction Model

Good values of RMS and R which is an indication of goodness-of-fit are displayed in NH_3^3 and NH_3^4 performance statistics in table 5.15 as compared to other setting for NH_3 prediction. In these two configurations the R value of the train and test set are relatively close to each other than the first two configurations. Though the RMS value of NH_3^3 and NH_3^4 is remarkably similar, the PMIS based NH_3 prediction model for the outlier removed data is selected to be the better one in relation to the fairly minimal RMS value than RMS value of GA based NH_3 prediction model for the outlier removed data**.**

Table 5.15: Performance Statistics of selected models for NH_3 prediction

Parameters	NH_3^1		NH_3^2		NH_3^3		NH_3^4	
	R	RMS	R	RMS	R	RMS	R	RMS
ALL	0.735055	6.417225	0.850131	4.985439	0.937415	3.5513	0.932651	3.66715
Train	0.797996	5.816507	0.924128	3.717764	0.988621	1.611054	0.994888	1.072096
Test	0.598545	7.581321	0.666004	7.022728	0.937415	3.5513	0.932651	3.66715
Selected ANN structure	12-24-1		18-23-1		6-28-1		12-21-1	

NH_3^1: Selected PMIS based NH_3 prediction model for the raw operational data
NH_3^2: Selected GA based NH_3 prediction model for the raw operational data
NH_3^3: Selected PMIS based NH_3 prediction model for the outlier removed data
NH_3^4: Selected GA based NH_3 prediction model for the outlier removed data

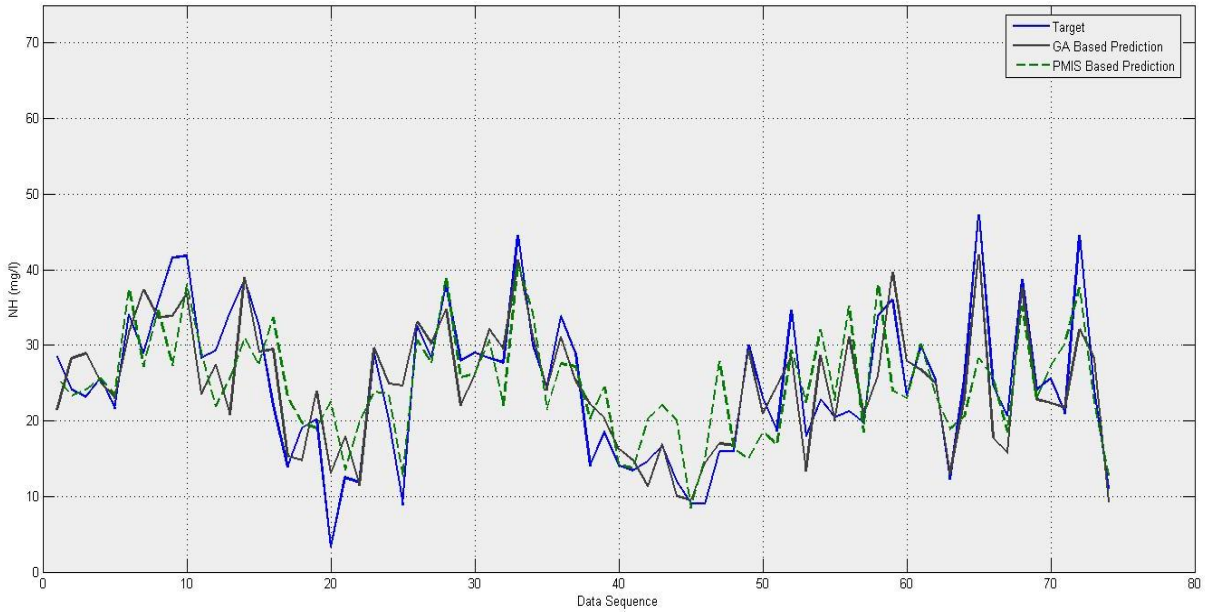


Figure 5.10: Prediction of NH_3 based on the raw operational data.

Fig 5.10 shows the plot of the measured value against that of the GA and PMIS based prediction for the raw operational data , and as can also be inferred from the performance statistics of these setting the model is not performing well. Whereas the plot of the target against the predicted values of GA and PMIS based prediction for the outlier removed data, the plot shows a very good fit between the measured and the predicted. Out of the two configurations, PMIS based prediction of the NH_3 for the outlier removed data is found to be so fit and an input layer with 6 neurons, a hidden layer with 28 neurons and an output layer with one neuron was determined to be the suitable architecture for NH_3 prediction.

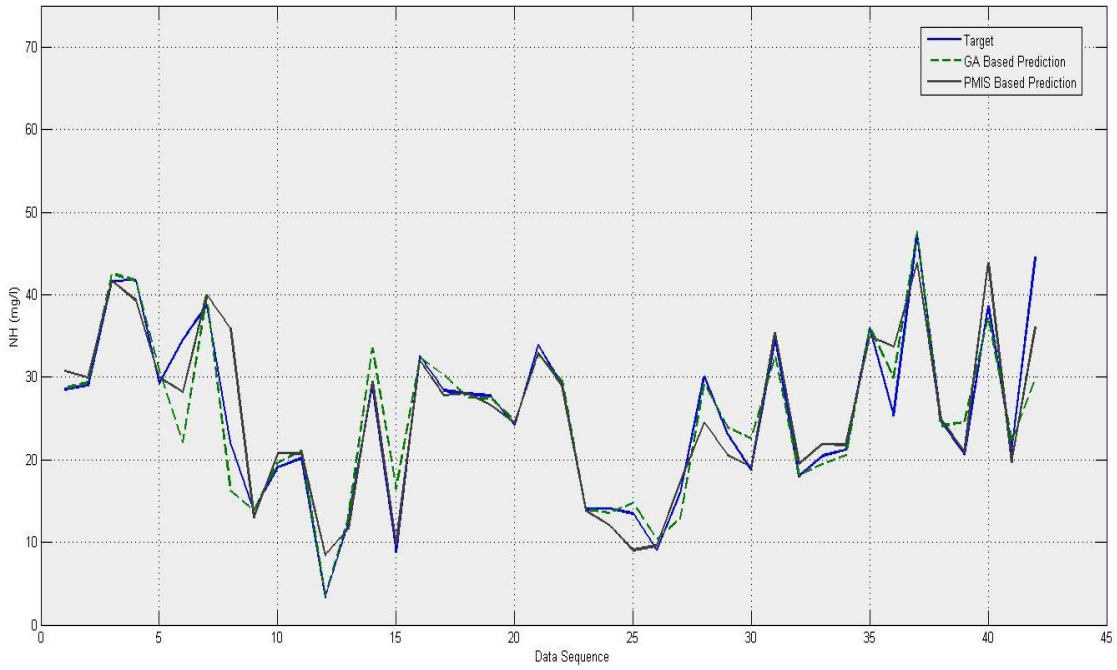


Figure 5.11: Prediction of NH_3based on outlier removed data.

5.4.3.5 TDS Prediction Model

Comparison of the R and RMS value for the four configuration used to develop TDS ANN prediction model again witnessed the superiority of the models developed using outlier removed data than with the raw operational data. GA based TDS prediction model for the outlier removed data showed a very close R values for the training and test sets (0.997679 and 0.983556), which means the model generalizes well and is likely to make accurate predictions with new data provided.

Table 5.16: Performance Statistics of selected models for TDS prediction

Parameters	TDS^1		TDS^2		TDS^3		TDS^4	
	R	RMS	R	RMS	R	RMS	R	RMS
ALL	0.875012	40.34379	0.879255	49.54824	0.939943	31.24594	0.983556	16.98332
Train	0.933669	30.0353	0.741543	68.37957	0.979528	18.80981	0.997679	6.789838
Test	0.745387	56.88886	0.744567	56.0826	0.939943	31.24594	0.983556	16.98332
Selected ANN structure	7/20/2001		39-29-1		14-25-1		10-43-1	

TDS^1: Selected PMIS based TDS prediction model for the raw operational data
TDS^2: Selected GA based TDS prediction model for the raw operational data
TDS^3: Selected PMIS based TDS prediction model for the outlier removed data
TDS^4: Selected GA based TDS prediction model for the outlier removed data

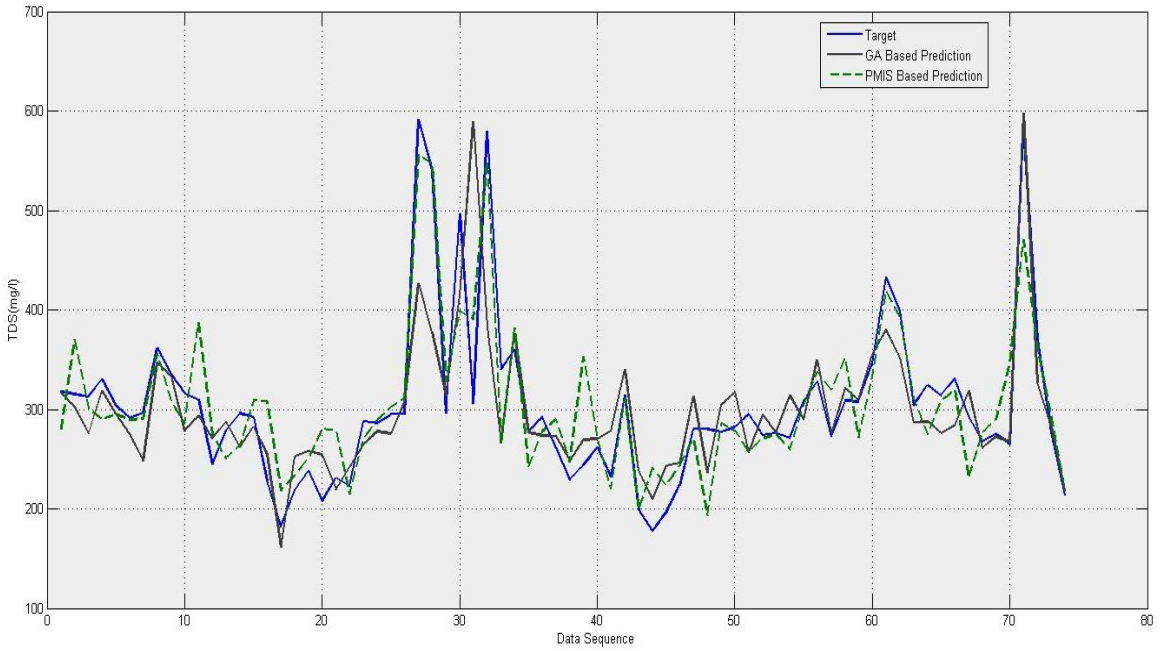


Figure 5.12: Prediction of TDS based on the raw operational data.

GA based prediction for outlier removed data with 10 input neurons in the input layer,43 neurons in the hidden layer and one neuron in the output layer (for predicted TDS) was determined to be the suitable architecture for prediction of TDS after both comparing the performance statistics and graphical plot of the measured against the predicted values.

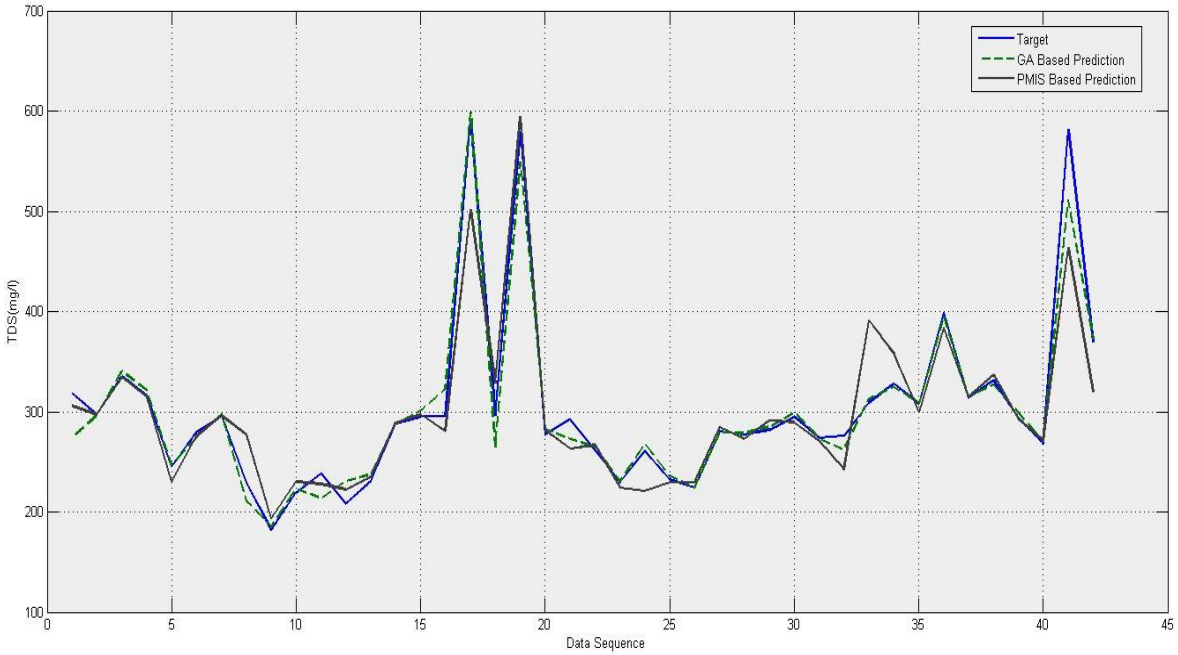


Figure 5.13: Prediction of TDS based on outlier removed data.

5.4.4 Modelling Results of MIMO Configuration

Below are depicted the automatically selected optimum models based on the R and RMS value which are good model performance indicators. For both the raw operational and outlier removed data, detail performance statistics is provided in table 5.17 and 5.18. The measured data against predicted values of the test data set are plotted for both the raw and outlier removed data.

Table 5.17: Performance Statistics of selected model for MIMO configuration for the raw operational data

BOD_5	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.665902	-0.6692	7.093496	37.31574	9.518576	0.851351	18.87429
Train	0.743264	-0.74355	6.764815	37.31574	9.607651	0.843137	19.22356
Test	0.407261	-0.41936	7.822309	18.68425	9.318023	0.869565	19.32258
COD	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.614722	-0.6176	34.16406	144.5612	44.91491	0.837838	89.06132
Train	0.622059	-0.62991	28.87189	108.7453	38.57597	0.843137	77.18511
Test	0.507061	-0.49996	45.89887	144.5612	56.48795	0.826087	117.1378
NH_3	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.481684	-0.48144	6.699217	19.75381	8.295756	0.702703	16.44957
Train	0.611055	-0.6105	5.924035	18.74619	7.28407	0.745098	14.5744
Test	0.058299	-0.05742	8.418099	19.75381	10.18675	0.608696	21.12404
TDS	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.577925	-0.58366	46.89051	290.9774	69.17676	0.878378	137.1699
Train	0.423156	-0.44246	38.49882	290.9774	56.92988	0.941177	113.9087
Test	0.534736	-0.53086	65.49816	218.5096	90.60902	0.73913	187.8939
pH	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.477297	-0.4747	0.236679	1.004416	0.289366	0.797297	0.57378
Train	0.519527	-0.51499	0.243205	1.004416	0.304651	0.784314	0.609563
Test	0.401116	-0.4051	0.222209	0.450713	0.252189	0.826087	0.522959

Key:

R [R Correlation]: The linear correlation between predicted outputs and target outputs, in problem domain units.

Net-R: The linear correlation between the target values and the raw network output values (before they are transformed into the measurement units of the problem).

Avg Abs [Average Absolute Error]: The average absolute difference between predicted output values and target output values.

Max Abs [Maximum Absolute Error]: The maximum absolute difference between a predicted output value and a target output value.

RMS [Root Mean Square Error]: The root mean square error between the predicted outputs and the target outputs.

Accuracy: The percent of predicted output values that lie within 20% of their corresponding target output values.

Conf Interval [Confidence Intervals]: 95% of the model predictions lie within the range around target output values bounded by the confidence intervals.

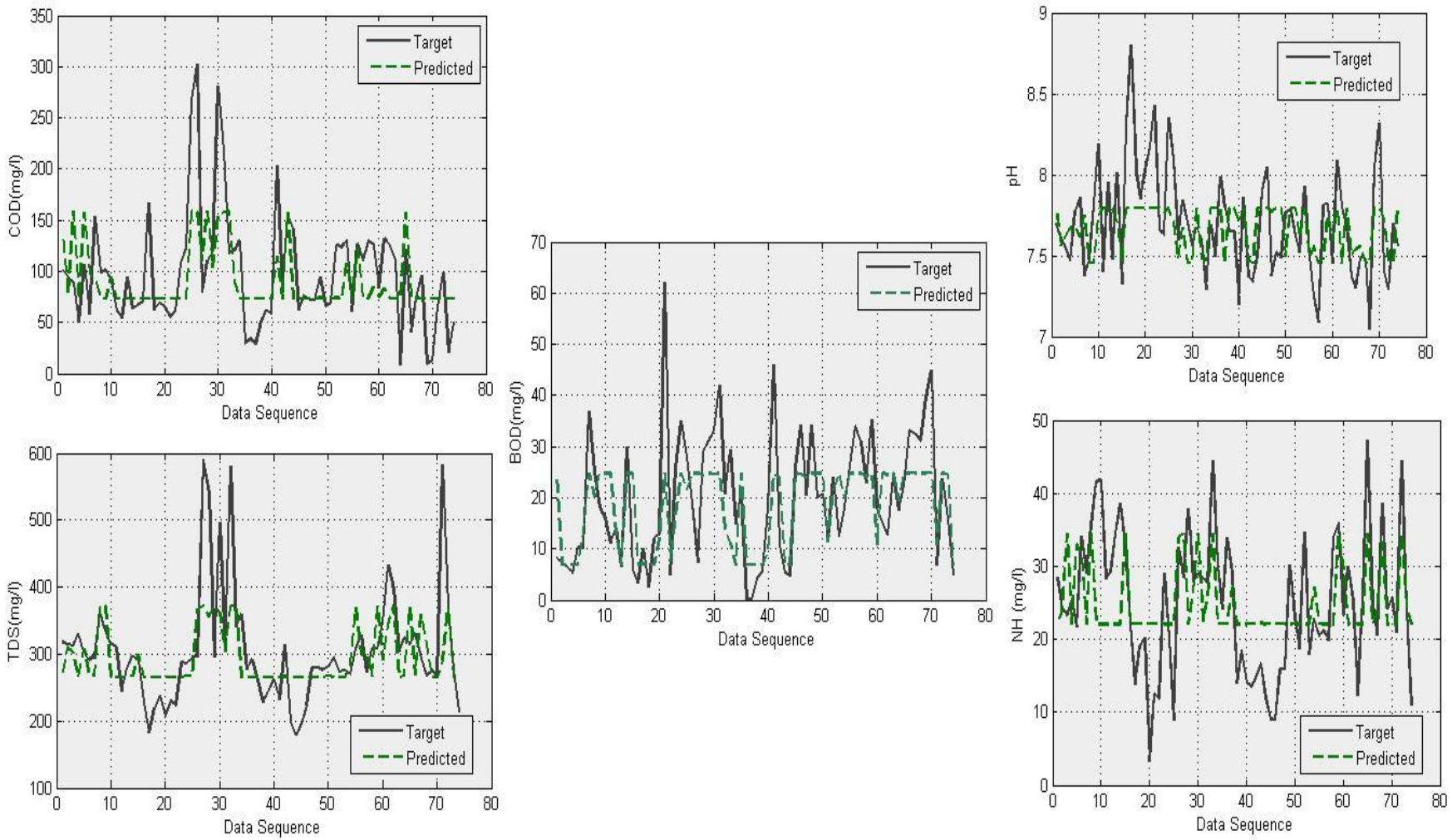


Figure 5.14: Prediction of performance indicators of the WWTP based on raw operational data

Table 5.18: Performance Statistics of selected models for MIMO configuration for outlier removed data.

BOD_5	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.691221	-0.69109	6.543926	30.31692	9.62135	0.857143	19.37148
Train	0.94574	-0.93051	4.168068	25.33458	6.397295	0.965517	13.08722
Test	0.691221	-0.69109	6.543926	30.31692	9.62135	0.857143	19.37148
COD	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.70919	-0.69107	24.93085	132.9553	39.84171	0.857143	80.21669
Train	0.817227	-0.8143	19.73663	119.8602	32.12383	0.896552	65.71708
Test	0.70919	-0.69107	24.93085	132.9553	39.84171	0.857143	80.21669
NH_3	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.750951	0.751282	5.08588	23.99922	6.782681	0.857143	13.65615
Train	0.893523	0.894454	3.658505	12.01299	4.752661	0.931035	9.72272
Test	0.750951	0.751282	5.08588	23.99922	6.782681	0.857143	13.65615
TDS	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.768323	-0.73134	35.54494	174.5506	58.08655	0.857143	116.9506
Train	0.89614	-0.82155	24.83604	174.5506	48.68373	0.931035	99.59437
Test	0.768323	-0.73134	35.54494	174.5506	58.08655	0.857143	116.9506
pH	**R**	**Net-R**	**Avg. Abs.**	**Max. Abs.**	**RMS**	**Accuracy (20%)**	**Conf. Interval (95%)**
All	0.70989	-0.69026	0.170091	0.802259	0.249426	0.880952	0.502191
Train	0.922989	-0.91434	0.123419	0.597432	0.175067	0.965517	0.358142
Test	0.70989	-0.69026	0.170091	0.802259	0.249426	0.880952	0.502191

As shown in Table 5.17 and 5.18, and Figures 5.14 and 5.15, the prediction of MIMO configuration for the selected performance indicators (outputs) is poor as compared to the performance of the MISO configuration.

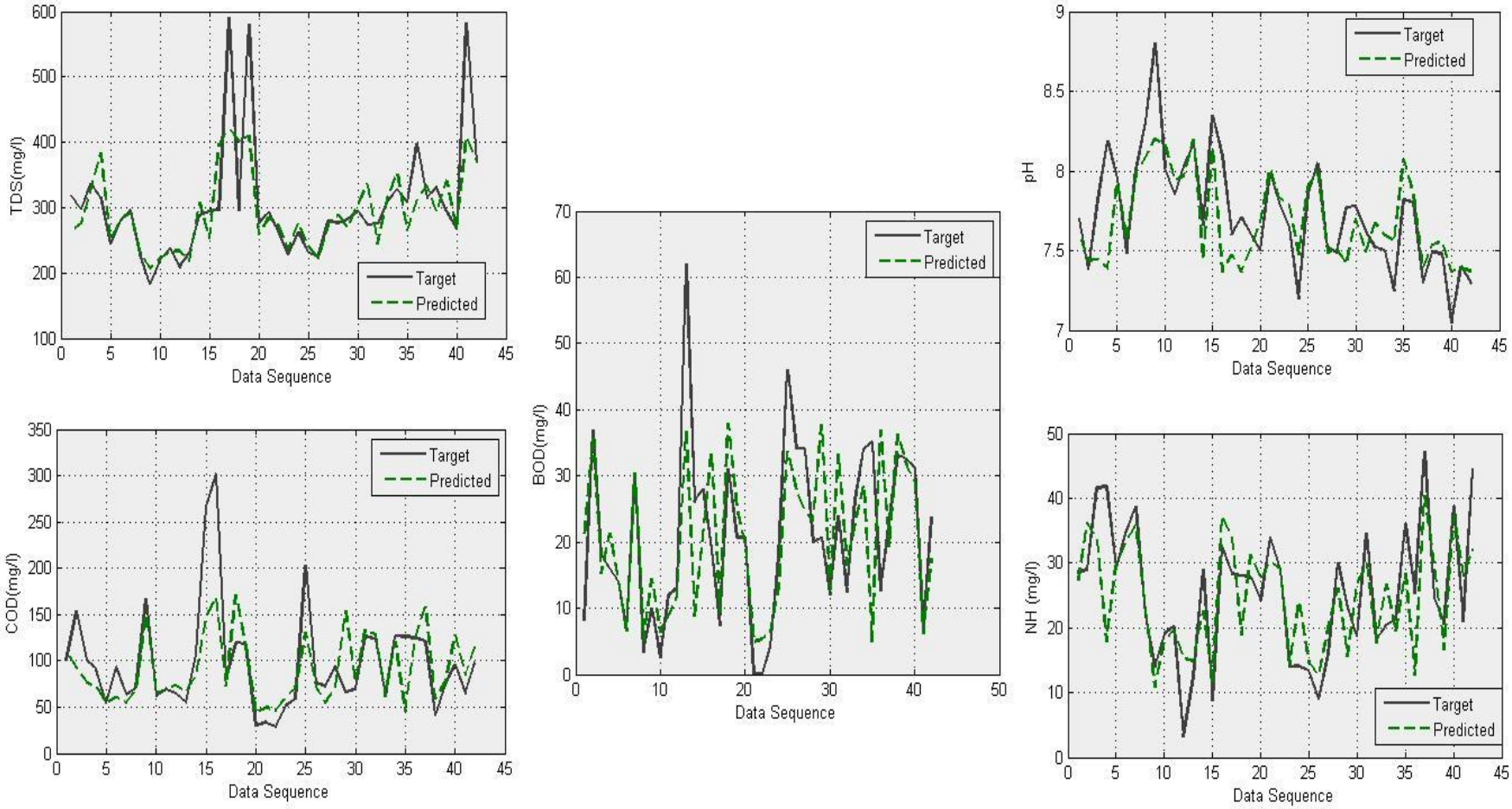


Figure 5.15: Prediction of performance indicators of the WWTP based on outlier removed data

The RMS errors in the MISO network, for both the raw and outlier removed data are more stable than that for the MIMO network. There is a significant difference in the prediction accuracy between the MISO and MIMO networks. While the MIMO network may be easier to develop and implement, the MISO networks showed better modelling ability. Not only the statistical analysis shows this , but also the graph in fig 5.4 to 5.13 illustrates how closely the ANN model tracks to actual experience using MISO configuration. The test performance of selected models for the MISO configuration, for all the five performance indicators selected is summarized in table 5.19 below.

Table 5.19: Test performance of selected models

Effluent	Architecture Selected	RMS	R
pH	14-43-1	0.081817	0.971741
BOD_5	16-29-1	4.360813	0.942927
COD	14-50-1	11.15782	0.979791
NH_3	6-28-1	3.5513	0.937415
TDS	10-48-1	16.98332	0.983556

Chapter 6- Conclusion and Recommendations

6.1 Conclusion

The control and prediction of WWTP is important in order to keep the system stable under a wide range of circumstances and avoid disturbing the environmental balance. Investigating the availability of models characterizing WWTP behavior as a dynamic system is thus a necessary first step. Though the complexity of the involved processes are high and the WWTP data are heterogeneous, incomplete and imprecise, which makes finding suitable models substantial problems, four separate model structures were developed for predicting the performance of the wastewater treatment plant using the MISO configuration and another two separate using the MIMO configuration. Using MISO configuration, based on the raw and outlier removed historical data, experiments were carried out using the PMIS and GA based input selection algorithms for each five output variables selected to indicate the performance of the wastewater treatment plant. Moreover, for the raw operational and outlier removed historical plant data, two experiments were carried out using GA based input selection techniques. Based on the test results of developed models obtained experimenting with the two configurations indicated above, the MISO configuration yielded a better quality and utility models than using MIMO configuration for predicting plant performance.

Optimum model architecture of 14-43-1 for pH, 16-29-1 for BOD_5, 14-50-1 for COD, 6-28-1 for NH_3, and 10-48-1 for TDS were selected for predicting the performance of Kaliti wastewater treatment plant using MISO configuration. The linear correlation between predicted outputs and target outputs for the optimum model architecture described above are 0.97 for pH, 0.94 for BOD_5, 0.98 for COD, 0.94 for NH_3, and 0.98 for TDS. GA based input selection algorithm applied to the outlier removed data showed remarkable performance in predicting pH, BOD_5 and TDS. And Partial mutual information input selection algorithm applied to outlier removed historical plant data was found to be successful in predicting COD and NH_3 for predicting the performance of the plant.

Among ANN several features, their ability to recognize and learn the underlying relations between input and output without explicit physical consideration, regardless of the problem's dimensionality and system nonlinearity, and the high tolerance to data containing noise and measurement errors are substantiated by the undertakings of this thesis work. Moreover, this work clearly indicates the effectiveness and the reliability of the proposed approach, for variable selection, from input data to increase the resulting network's ability to predict.

In conclusion, the experimental result showed that the trained and tested ANN models developed, after optimizing the input parameters, for each of the five performance indicators could potentially be employed for predicting the performance of Kaliti wastewater treatment plant. The developed ANN models allow operators to react quickly to changing conditions while at the same time minimizing costs and maximizing performance.

6.2 Recommendations

Considering results obtained in this thesis work, the following recommendations are suggested.

1. So as to overcome the limitations of using artificial neural network like lack of clear rules or fixed guidelines for optimal ANN architecture design, lack of physical concepts and relations, and the inability to explain in a comprehensible form the process through which a given decision (answer) was made by the ANN; Hybridizing ANNs with conventional approaches such as expert systems should be practiced and implemented to yield a stronger computational paradigms for solving complex and computationally expensive problems like this.
2. The present work can be extended by collecting long term data, for all stages of the biological wastewater treatment, and developing intelligent predictive models for predicting the performance of the plant at each stages of the treatment plant.
3. Generally, I do recommend the following network parameters and functions to be used as a reference for modelling a neural network for a domestic wastewater treatment plant: (1) zero-mean normalization method for input variables; (2) Gaussian weigh-factor distribution for initial values; (3) hyperbolic tangent transfer function; (4) genetic algorithm to automate much of the painstaking manipulation, selection, and data pruning that monopolizes most of the time in building a real world neural network application.
4. Given the universal function approximation capability of ANNs, research on the areas of process optimization of wastewater treatment, resource optimization and management, and constructing robust ANN application for the wastewater treatment process control system should be done to utilize their comparative advantage over other traditional analytical methods like the capability of ANNs to solve problems that are unsolvable with traditional tools, the accuracy of ANN result, the ability of ANN models to identify erroneous instrument readings and notify operators that values are outside of normal parameters, and their characteristics of being fast and objective compared to mechanical models.

References

ASCE,(2000). Artificial neural networks in hydrology. I. Preliminary concepts. Journal of Hydro. Eng. ASCE 5, 115–123.

AAWSA, (2002). Wastewater Master plan Volume III, Addis Ababa Water and Sewerage Authority, Ethiopia.

Basheer, I.A., M. Hajmeer,(2000), Artificial neural networks: fundamentals, computing, design, and Application, Journal of Microbiological Methods 43, 3–31.

Fu, L.,(1995). Neural Networks in Computer Intelligence. McGraw-Hill, New York.

Hassoun,M.H.,(1995). Fundamentals of Artificial Neural Networks. MIT Press, Cambridge, MA.

Haykin, S.,(1994). Neural Networks: A Comprehensive Foundation. Macmillan, New York.

Hecht-Nielsen, R., (1988). Applications of counter propagation networks. Neural Networks 1,131–139.

Hong, Y-S.T., Rosen, M.R., Bhamidimarri, R., (2003).Analysis of a municipal wastewater treatment plant using a neural network-based pattern analysis. Water Research 37, 1608–1618.

I.A. Basheer, M. Hajmeer, (2000). Artificial neural networks: fundamentals, computing, design, and application, Journal of Microbiological Methods 43,3–31.

Jain, A.K., Mao, J., Mohiuddin, K.M.,(1996).Artificial neural networks: a tutorial. IEEE March,31–44.

Jakeman,A.J.,Letcher,R.A.,Norton,J.P.,(2006).Ten iterative steps in development and evaluation of environmental models, Journal of Environmental Modelling and Software 21(5) ,602-614.

Khawla A. Al-Shayjl,(1998). Modelling, Simulation of desalination plants, Ph.D. Dissertation, 39-115, Blacksburg, Virginia.

Maier, H.R., Dandy, G.C., (2000). Neural networks for the prediction and forecasting of Water resources variables: a review of modeling issues and applications, Journal of Environmental Modeling and Software 15(1),101-124.

May, R. J., Maier, H.R., Dandy, G.C., Fernando, T.M.K.,(2008). Non-linear variable selection for artificial neural networks using partial mutual information, Journal of Environmental Modelling and Software 23, 1312–1326.

McCulloh, W.S., Pitts, W.,(1943).A logical calculus of the ideas immanent in nervous activity. Biophys.5,115–133.

Moreno-Alfonso, N., Redondo, C.F., (2001).Intelligent waste-water treatment with neural networks, Water Policy3, 267–271.

Pham, D.T.,(1994).Applications of Artificial Intelligence in Engineering, Proceedings of the 9th International Conference. Computational Mechanics Publications,Southampton,pp.3–36.

Rumelhart,D.E.,Durbin,R.,Golden,R.,Chauvin,Y.,(1995).Backpropagation:Theory,Architecture,and Applications. Lawrence Erlbaum,NJ,pp.1–34.

Schalkoff, R.J.,(1997).Artificial Neural Networks. McGraw-Hill,. New York.

Sharma, A.,(2000). Seasonal to inter annual rainfall probabilistic forecasts for improved water supply management: part 1–a strategy for system predictor identification, Journal of Hydrology 239.

Wythoff, B.J.,(1993).Back propagation neural networks: a tutorial.Intell.Lab.Syst.18,115-155.

Zhang, Q., Stanley, S.J., (1999).Real-time water treatment process control with artificial neural networks, Journal of Environmental Engineering125 (2).

Zupan, J., Gasteiger, J.,(1993).Neural Networks For Chemists: An Introduction. VCH, New York.

Appendix

Appendix-A: Estimation of Partial Mutual Information

Given a random output variable Y, there will be some uncertainty surrounding an observation y ∈ Y, which can be defined according to the Shannon entropy, H. The reduced uncertainties surrounding X and Y are denoted by the conditional entropies H (X | Y) and H (Y | X), respectively. Mutual information can be determined directly using

$$I(X;Y) = \iint p(x,y)\ \log \frac{p(x,y)}{p(x)p(y)}\,dx\,dy \quad \text{.....(1)}$$

Where p(y) and p(x) are the marginal probability density functions (pdfs) of X and Y, respectively; and p(x, y) is the joint pdf. However, within a practical context, the true functional forms of the pdfs in (1) are typically unknown. Hence, estimates of the densities are used instead. Substitution of density estimates are used instead. Substitution of density estimates into a numerical approximation of the integral in (1) gives

$$I(X;Y) \approx \frac{1}{n}\sum_{i=1}^{n} \log\left[\frac{f(x_i,y_i)}{f(x_i)f(y_i)}\right] \quad \text{.....(2)}$$

Where f denotes the estimated density based on a sample of n observations of (x, y).note that the base of the logarithm varies within the literature, and use of either 2 or e is often reported, although the natural algorithm is assumed in this study, unless otherwise stated.

Given the form of (2), it follows that efficient and accurate estimation of MI is largely dependent on the techniques employed to estimate the marginal and joint pdfs. Non parametric density estimation techniques are typically considered suitable and accurate. In particular kernel density estimation (KDE) is used, alternatives, such as the histogram. The simple parzen window forms the basis for this approach, in which an estimator for f is given by

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) \quad \text{.....(3)}$$

Where f(x) denotes the estimate of the pdf at x; x_i {i=1,……n} denote sample observations of x;and K_h is some kernel function for which h denotes the kernel bandwidth (or,smoothing parameter). Acommon choice for K_h is the gaussian kernel

$$K_h = \frac{1}{\left(\sqrt{2\pi}h\right)^d \sqrt{|\sum|}} exp\left(\frac{-\|x-x_i\|}{2h^2}\right) \quad \text{.....(4)}$$

Here, d denotes the number of dimensions of x, $\sum$ is the sample covariance matrix, and $\|x - x_i\|$ is the Mahalanobis distance metric, which is given by

$$\|x - x_i\| = (x - x_i)^T \sum^{-1} (x - x_i) \quad \text{.....(5)}$$

Substituting the expression for the kernel into (3), the estimator for f becomes

$$\hat{f}(x) = \frac{1}{n(\sqrt{2\pi}h)^d \sqrt{|\sum|}} \sum_{i=1}^{n} exp\left(\frac{-\|x-x_i\|}{2h^2}\right) \quad \text{.....(6)}$$

Based on the KDE approach, an estimator for the regression of Y on X is written as

$$\hat{m}_Y(x) = E[y|X = x] = \frac{1}{n} \frac{\sum_{i=1}^{n} y_i K_h(x-x_i)}{\sum_{i=1}^{n} K_h(x-x_i)} \quad \text{.....(7)}$$

Where $\hat{m}_Y(x)$ denotes the regression estimator; n is the number of observed values (y_i , x_i); K_h is as given in (5) and E[y | X=x] denotes the conditional expectation of y given an observed x. An estimator $\hat{m}_z(x)$ can be similarly constructed, and the residuals u and v can be subsequently obtained using the expressions

$$u = Y - \hat{m}_Y(X) \quad \text{.....(8)}$$

And

$$v = Z - \hat{m}_Z(X) \quad \text{.....(9)}$$

Using the residuals obtained in (8) and (9), the PMI is then calculated as

$$I'_{ZY\,.\,X} = I(v; u) \quad \text{.....(10)}$$

Where the subscript notation $I'_{ZY\,.\,X}$ is used to denote the PMI, otherwise written as I (Z; Y | X). This notion of PMI allows for the evaluation of the dependence between variables that takes into consideration any information already provided by a given variable X.

The PMI-based input selection (PMIS) algorithm used in this study was originally developed by Sharma (2000) for the identification of inputs for hydrological models. Given a candidate set, C, and output variable, Y, the PMIS algorithm proceeds at each iteration by finding the candidate C_s that maximizes the PMI with respect to the output variable, conditional on the inputs that have been previously selected, the statistical significance of the PMI estimated for C_s is assessed based on confidence bounds drawn from the distribution generated by a bootstrap loop. If the input is significant, C_s is added to S and the selection continues; otherwise, there are no more significant candidates remaining and the algorithm is subsequently terminated.

The details of the algorithm are as follows:

Step 1: Let $S \rightarrow \Phi$ (Initialization)

Step 2: While $C \neq \Phi$ (forward selection)

Step 3: Construct kernel regression estimator $\hat{m}_Y(S)$

Step 4: Calculate residual output $u = Y - \hat{m}_Y(S)$

Step 5: For each $C_j \in C$

Step 6: Construct kernel regression estimator $\hat{m}_{C_j}(S)$

Step 7: Calculate residual candidate $v = C_j - \hat{m}_{C_j}(S)$

Step 8: Estimate $I(v; u)$

Step 9: Find candidate C_s (V_s) that maximizes $I(v; u)$

Step 10: For b=1 to B (Bootstrap)

Step 11: Randomly shuffle V_s to obtain V_s^*

Step 12: Estimate $I_b = I(v_s^*; u)$

Step 13: Find confidence bound $I_b^{(95)}$

Step 14: If $I(v_s; u) > I_b^{(95)}$ (selection/termination)

Step 15: Move Cs to S

Step 16: Else

Step 17: Break

Step 18: Return selected input set S.

Here, B is the bootstrap size; and $I_b^{(95)}$ denotes the 95th percentile bootstrap estimate of the randomized PMI, I_b.

Appendix-B: VBA Code For Estimation of Partial Mutual Information (Interface only)

……...

```
Option Explicit

Public Function BinPath() As String

   BinPath = ThisWorkbook.Path & "\bin"

End Function

Public Function TempPath() As String

   TempPath = ThisWorkbook.Path & "\temp"

End Function

Public Sub ReadAsNewSheet(strFilename As String, strName As String, wbBook As
Workbook)

      Dim fs
      Dim wsDataset As Worksheet
      Set fs = CreateObject("Scripting.FileSystemObject")
      If fs.FileExists(strFilename) Then

         ' Remove any sheet with same name before creating new sheet
         Dim ws As Worksheet
         For Each ws In wbBook.Worksheets
            If ws.Name = strName Then
               ws.Delete
               Exit For
            End If
         Next ws

         Set wsDataset = wbBook.Worksheets.Add
         wsDataset.Name = strName

         On Error GoTo HandleException
         Call ReadDataset(strFilename, wsDataset)

      Else
         Exception.Throw ("Specified file does not exist")

      End If

      Exit Sub
```

```
HandleException:
    ' Clean up worksheet added to book
    wsDataset.Delete
    Exception.Throw ("Error opening dataset " & strFilename)

End Sub

Public Sub ReadDataset(strFilename As String, wsDataset As Worksheet)

    ' Create new text-file import querytable
    Dim qtFileImport As QueryTable
    Set qtFileImport = wsDataset.QueryTables.Add("TEXT;" & strFilename,
wsDataset.Range("A1"))

    With qtFileImport

        ' Specify file format
        .TextFileStartRow = 1
        .TextFileParseType = xlDelimited
        .TextFileTabDelimiter = True
        .TextFileCommaDelimiter = True
        .TextFileSpaceDelimiter = False
        .TextFileSemicolonDelimiter = False
        .TextFileConsecutiveDelimiter = True
        .TextFileTextQualifier = xlTextQualifierNone
        .RefreshStyle = xlOverwriteCells

        ' Refresh query to import text-file
        On Error GoTo HandleException
        .Refresh BackgroundQuery:=False

        ' Remove querytable
        .Delete

    End With

    Exit Sub

HandleException:
    qtFileImport.Delete
    Call Exception.Throw("Error opening dataset file " & strFilename)

End Sub

Public Sub WriteDataset(strFilename As String, wsDataset As Worksheet)

    Application.ScreenUpdating = False

    ' Create temporary workbook
    Dim wbTemp As Workbook
```

```
    Set wbTemp = Workbooks.Add

    ' Copy dataset to new workbook and save
    Call wsDataset.Cells.Copy(wbTemp.Worksheets(1).Cells(1, 1))
    On Error GoTo HandleException
    Call wbTemp.Worksheets(1).SaveAs(Filename:=strFilename, FileFormat:=xlTextMSDOS)

    ' Clean up and activate original workbook
    wbTemp.Close SaveChanges:=False

    Application.ScreenUpdating = True

    wsDataset.Activate

    Exit Sub

HandleException:
    ' Clean up and throw an exception
    wbTemp.Close SaveChanges:=False
    Application.ScreenUpdating = True
    wsDataset.Activate
    Call Exception.Throw("Error saving file " & strFilename)

End Sub
……...

Option Explicit

Private Sub btnOK_BeforeDropOrPaste(ByVal Cancel As MSForms.ReturnBoolean, ByVal
Action As MSForms.fmAction, ByVal Data As MSForms.DataObject, ByVal X As Single,
ByVal Y As Single, ByVal Effect As MSForms.ReturnEffect, ByVal Shift As Integer)

End Sub

Private Sub btnOK_Click()

    ' Check that three worksheets are specified
    If cbDataset.Value = "" Then
        MsgBox "A worksheet must be specified", vbExclamation, "Missing data"
        Exit Sub
    End If

    ' Create list of temporary input and output data filenames
    Dim Filenames(1 To 2) As String
    Filenames(1) = TempPath & "\" & "PMISSource.dat"
    Filenames(2) = TempPath & "\" & "PMISSummary.txt"

    ' Write temporary data files
    Application.DisplayAlerts = False
    Call WriteDataset(Filenames(1), ActiveWorkbook.Worksheets(cbDataset.Value))
```

```
    Application.DisplayAlerts = True

    ' Generate argument list
    Dim Args As String
    If chkDoSetMaxIterations.Value = True Then
        Args = Args & " -k" & tbMaxIterations
    End If
    If chkDoBootstrap.Value = True Then
        Args = Args & " -b" & tbBootstrapSize
    End If

     ' Generate commmand string
    Dim strCommand As String
    strCommand = BuildCommand("pmis.exe", Filenames, Args)

    ' Run the command
    Call System.Execute(strCommand)

    Call ReadAsNewSheet(Filenames(2), cbDataset.Value & ".pmi", ActiveWorkbook)

    ' Clean up temporary files
    Call System.RemoveFiles(Filenames)

    ' Close dialog box
    Unload Me

End Sub

Private Sub btnCancel_Click()

    Unload Me

End Sub

Private Sub cbDataset_Change()

End Sub

Private Sub chkDoBootstrap_AfterUpdate()

End Sub

Private Sub chkDoBootstrap_BeforeDragOver(ByVal Cancel As MSForms.ReturnBoolean,
ByVal Data As MSForms.DataObject, ByVal X As Single, ByVal Y As Single, ByVal
DragState As MSForms.fmDragState, ByVal Effect As MSForms.ReturnEffect, ByVal Shift
As Integer)

End Sub

Private Sub chkDoBootstrap_Click()
```

```
End Sub

Private Sub chkDoBootstrap_Error(ByVal Number As Integer, ByVal Description As
MSForms.ReturnString, ByVal SCode As Long, ByVal Source As String, ByVal HelpFile As
String, ByVal HelpContext As Long, ByVal CancelDisplay As MSForms.ReturnBoolean)

End Sub

Private Sub chkDoBootstrap_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal
Shift As Integer)

End Sub

Private Sub chkDoBootstrap_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

End Sub

Private Sub chkDoBootstrap_MouseDown(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)

End Sub

Private Sub chkDoBootstrap_MouseMove(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)

End Sub

Private Sub chkDoBootstrap_MouseUp(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)

End Sub

Private Sub chkDoSetMaxIterations_Click()

End Sub

Private Sub chkDoSetMaxIterations_Error(ByVal Number As Integer, ByVal Description As
MSForms.ReturnString, ByVal SCode As Long, ByVal Source As String, ByVal HelpFile As
String, ByVal HelpContext As Long, ByVal CancelDisplay As MSForms.ReturnBoolean)

End Sub

Private Sub Frame1_Click()

End Sub

Private Sub Frame2_Click()

End Sub
```

```
Private Sub Label1_Click()

End Sub

Private Sub Label2_Click()

End Sub

Private Sub tbBootstrapSize_AfterUpdate()

    ' Check that value specified is numeric
    If Not IsNumeric(tbBootstrapSize.Value) Then
        tbBootstrapSize = 20
        MsgBox "Bootstrap size must be a whole number", vbExclamation, "Invalid parameter"

    ' Check at least one iteration is selected
    ElseIf Int(tbBootstrapSize) < 20 Then
        tbBootstrapSize.Value = 20
        MsgBox "Bootstrap size must be at least 20", vbExclamation, "Invalid parameter"

    ' Force specified number to nearest integer value
    Else
       tbBootstrapSize.Value = Int(tbBootstrapSize.Value)

    End If

End Sub

Private Sub tbMaxIterations_AfterUpdate()

    ' Check that value specified is numeric
    If Not IsNumeric(tbMaxIterations.Value) Then
        tbMaxIterations = 1
        MsgBox "Number of iterations must be a whole number", vbExclamation, "Invalid
parameter"

    ' Check at least one iteration is selected
    ElseIf tbMaxIterations < 1 Then
        tbMaxIterations.Value = 1
        MsgBox "Number of iterations must be at least 1", vbExclamation, "Invalid parameter"

    ' Force specified number to nearest integer value
    Else
       tbMaxIterations.Value = Int(tbMaxIterations.Value)
    End If


End Sub
```

```
Private Sub UserForm_Initialize()

    UpdateComboBoxes

End Sub

Private Sub UpdateComboBoxes()

    Dim ws As Worksheet
    For Each ws In ActiveWorkbook.Worksheets
        cbDataset.AddItem ws.Name
    Next ws

End Sub
```

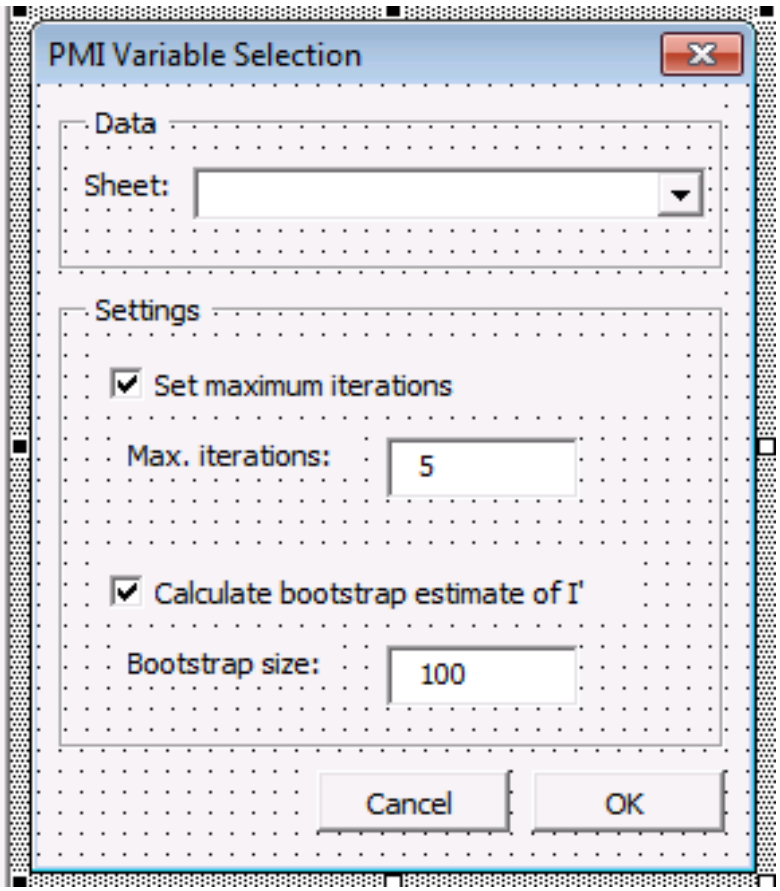
……...

```
Option Explicit

Private Sub btnOK_Click()

    ' Check that three worksheets are specified
    If cbDataset.Value = "" Then
        MsgBox "A worksheet must be specified", vbExclamation, "Missing data"
        Exit Sub
    End If

    ' Create list of temporary input and output data filenames
    Dim Filenames(1 To 2) As String
    Filenames(1) = TempPath & "\" & "PMISSource.dat"
    Filenames(2) = TempPath & "\" & "PMISSummary.txt"

    ' Write temporary data files
    Application.DisplayAlerts = False
    Call WriteDataset(Filenames(1), ActiveWorkbook.Worksheets(cbDataset.Value))
    Application.DisplayAlerts = True

    ' Generate argument list
    Dim Args As String
    If chkDoSetMaxIterations.Value = True Then
        Args = Args & " -k" & tbMaxIterations
    End If
    If chkDoBootstrap.Value = True Then
        Args = Args & " -b" & tbBootstrapSize
    End If

     ' Generate commmand string
    Dim strCommand As String
    strCommand = BuildCommand("pmis.exe", Filenames, Args)

    ' Run the command
```

```
    Call System.Execute(strCommand)

    Call ReadAsNewSheet(Filenames(2), cbDataset.Value & ".pmi", ActiveWorkbook)

    ' Clean up temporary files
    Call System.RemoveFiles(Filenames)

    ' Close dialog box
    Unload Me

End Sub

Private Sub btnCancel_Click()

    Unload Me

End Sub

Private Sub cbDataset_Change()

End Sub

Private Sub chkDoBootstrap_Click()

End Sub

Private Sub chkDoSetMaxIterations_Click()

End Sub

Private Sub Frame1_Click()

End Sub

Private Sub Frame2_Click()

End Sub

Private Sub Label1_Click()

End Sub

Private Sub Label3_Click()

End Sub

Private Sub tbBootstrapSize_AfterUpdate()

    ' Check that value specified is numeric
    If Not IsNumeric(tbBootstrapSize.Value) Then
```

```
        tbBootstrapSize = 20
        MsgBox "Bootstrap size must be a whole number", vbExclamation, "Invalid parameter"

    ' Check at least one iteration is selected
    ElseIf Int(tbBootstrapSize) < 20 Then
        tbBootstrapSize.Value = 20
        MsgBox "Bootstrap size must be at least 20", vbExclamation, "Invalid parameter"

    ' Force specified number to nearest integer value
    Else
      tbBootstrapSize.Value = Int(tbBootstrapSize.Value)

    End If

End Sub

Private Sub tbMaxIterations_AfterUpdate()

    ' Check that value specified is numeric
    If Not IsNumeric(tbMaxIterations.Value) Then
        tbMaxIterations = 1
        MsgBox "Number of iterations must be a whole number", vbExclamation, "Invalid
parameter"

    ' Check at least one iteration is selected
    ElseIf tbMaxIterations < 1 Then
        tbMaxIterations.Value = 1
        MsgBox "Number of iterations must be at least 1", vbExclamation, "Invalid parameter"

    ' Force specified number to nearest integer value
    Else
      tbMaxIterations.Value = Int(tbMaxIterations.Value)
    End If

End Sub

Private Sub UserForm_Initialize()

    UpdateComboBoxes

End Sub

Private Sub UpdateComboBoxes()

    Dim ws As Worksheet
    For Each ws In ActiveWorkbook.Worksheets
        cbDataset.AddItem ws.Name
    Next ws

End Sub
```

The Graphical User Interface (GUI) of the partial mutual information-based input variable selection developed looks like the following.



A software that implements PMI variable selection program from within MS Excel on a dataset along with different tools that can be used for development of a neural network particularly for developing MLP and Generalized regression networks is attached with this work.

Appendix-C: PMI Score for outlier removed data

1. Partial mutual information score for pH output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**pH**	0.1302	-8.28E+06	-1.08E+07	**11.8781**	**16.0501**
2	**PO_4^{3-}**	0.15461	-8.28E+06	-1.08E+07	**31.689**	**45.3585**
3	**NO_3**	0.24998	-8.28E+06	-1.08E+07	**19.8015**	**45.2942**
4	**COD**	0.15422	-8.28E+06	-1.08E+07	**18.0219**	**56.7352**
5	**EC**	0.13151	-8.28E+06	-1.08E+07	**28.317**	**76.0626**
6	**NH_3**	0.12232	-8.28E+06	-1.08E+07	**28.4186**	**84.6105**
7	TDS	0.14399	-8.28E+06	-1.08E+07	90.3132	34.5323
8	NO_2	0.10831	-8.28E+06	-1.08E+07	90.5568	34.0486
9	TSS	0.10574	-8.28E+06	-1.08E+07	90.172	32.9934
10	TVS	0.09827	-8.28E+06	-1.08E+07	92.1203	36.2918
11	SO_4^{2-}	0.10043	-8.28E+06	-1.08E+07	103.909	46.3884
12	BOD_5	0.08406	-8.28E+06	-1.08E+07	104.079	47.7933
13	DO	0.06117	-8.28E+06	-1.08E+07	103.767	48.8691

2. Partial mutual information score for TDS output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**COD**	0.14711	-8.28E+06	-1.08E+07	**2.30458**	**5.19217**
2	**pH**	0.22192	-8.28E+06	-1.08E+07	**10.9562**	**22.8987**
3	**NO_2**	0.22593	-8.28E+06	-1.08E+07	**10.1729**	**32.7806**
4	**NH_3**	0.19601	-8.28E+06	-1.08E+07	**6.99669**	**43.3029**
5	**TSS**	0.09704	-8.28E+06	-1.08E+07	**10.2454**	**55.5718**
6	BOD_5	0.08232	-8.28E+06	-1.08E+07	63.8247	13.0198
7	TVS	0.09178	-8.28E+06	-1.08E+07	65.8124	15.2624
8	SO_4^{2-}	0.08748	-8.28E+06	-1.08E+07	76.1151	18.3657
9	NO_3	0.07843	-8.28E+06	-1.08E+07	78.8437	18.1694
10	PO_4^{3-}	0.04035	-8.28E+06	-1.08E+07	85.9564	26.6869
11	TDS	0.08727	-8.28E+06	-1.08E+07	88.6587	30.5854
12	EC	0.12475	-8.28E+06	-1.08E+07	92.6316	36.3455
13	DO	0.08869	-8.28E+06	-1.08E+07	86.0678	31.1695

3. Partial mutual information score for NH_3output variable

Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**COD**	0.185	-8.28E+06	-1.08E+07	**3.05885**	**5.94644**
2	**pH**	0.17765	-8.28E+06	-1.08E+07	**1.95998**	**13.9024**
3	**SO_4^{2-}**	0.10576	-8.28E+06	-1.08E+07	**4.43986**	**31.5683**
4	**NO_2**	0.12165	-8.28E+06	-1.08E+07	**15.099**	**52.1742**
5	**TVS**	0.10298	-8.28E+06	-1.08E+07	**23.3468**	**68.847**
6	**EC**	0.12972	-8.28E+06	-1.08E+07	**24.5065**	**76.8725**
7	TDS	0.08763	-8.28E+06	-1.08E+07	80.6191	27.9993
8	NH_3	0.08047	-8.28E+06	-1.08E+07	90.496	32.6715
9	BOD_5	0.11053	-8.28E+06	-1.08E+07	96.6917	37.7935
10	DO	0.10953	-8.28E+06	-1.08E+07	100.601	42.7777
11	NO_3	0.10259	-8.28E+06	-1.08E+07	94.1883	35.9007
12	TSS	0.10907	-8.28E+06	-1.08E+07	96.8231	40.0692
13	PO_4^{3-}	0.07136	-8.28E+06	-1.08E+07	97.9353	43.0369

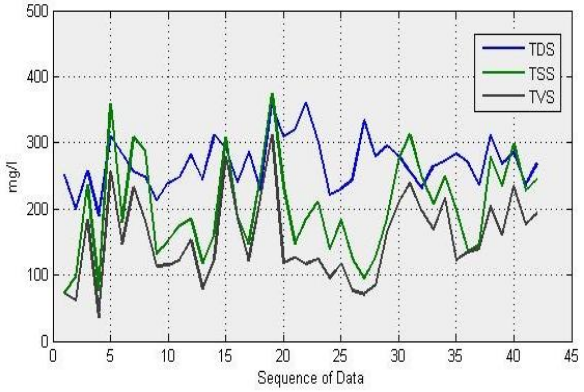
4. Partial mutual information score for COD output variable

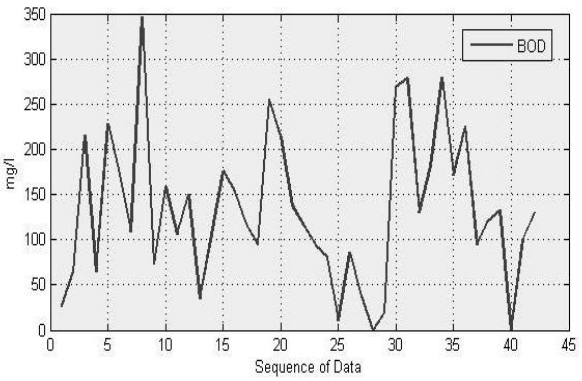
Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**EC**	0.19114	-8.28E+06	-1.08E+07	**4.04258**	**7.54175**
2	**TVS**	0.15142	-8.28E+06	-1.08E+07	**8.77546**	**21.454**
3	**NH_3**	0.19019	-8.28E+06	-1.08E+07	**13.4922**	**40.3507**
4	**NO_3**	0.16565	-8.28E+06	-1.08E+07	**16.1786**	**56.2123**
5	**NO_2**	0.16822	-8.28E+06	-1.08E+07	**26.3533**	**74.2286**
6	**TDS**	0.23192	-8.28E+06	-1.08E+07	**27.4733**	**75.7804**
7	**BOD_5**	0.20205	-8.28E+06	-1.08E+07	**32.2284**	**85.4622**
8	COD	0.1742	-8.28E+06	-1.08E+07	89.0466	33.41
9	pH	0.14738	-8.28E+06	-1.08E+07	88.4449	31.3967
10	TSS	0.11401	-8.28E+06	-1.08E+07	93.7615	36.9401
11	SO_4^{2-}	0.11951	-8.28E+06	-1.08E+07	99.525	41.3328
12	PO_4^{3-}	0.10674	-8.28E+06	-1.08E+07	94.6826	38.3965
13	DO	0.09072	-8.28E+06	-1.08E+07	99.1211	44.2228

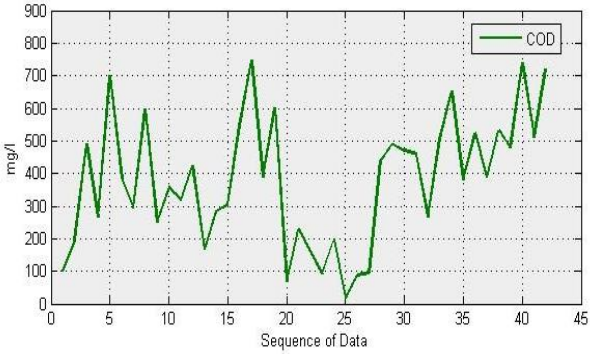
5. Partial mutual information score for BOD output variable

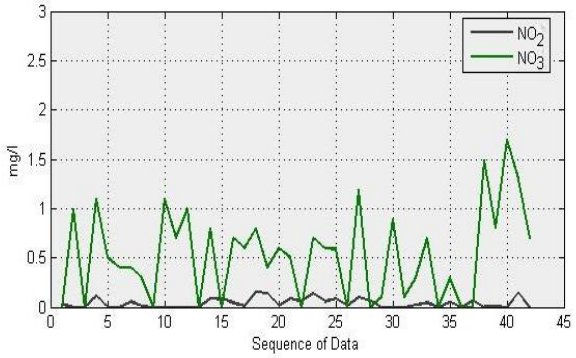
Iteration	Variable	I(x;y)	MC-I*(95)	MC-I*(99)	AIC(k)	AIC(p)
1	**pH**	0.16701	-8.28E+06	-1.08E+07	**1.73629**	**5.90827**
2	**COD**	0.13794	-8.28E+06	-1.08E+07	**8.78457**	**20.727**
3	**NO_2**	0.14516	-8.28E+06	-1.08E+07	**12.1397**	**34.7474**
4	**EC**	0.13929	-8.28E+06	-1.08E+07	**21.7923**	**56.0073**
5	**TSS**	0.14461	-8.28E+06	-1.08E+07	**25.1048**	**69.794**
6	**NH_3**	0.19322	-8.28E+06	-1.08E+07	**20.7451**	**73.0613**
7	SO_4^{2-}	0.21503	-8.28E+06	-1.08E+07	78.4068	19.198
8	BOD_5	0.14356	-8.28E+06	-1.08E+07	85.1011	24.921
9	TVS	0.16202	-8.28E+06	-1.08E+07	88.2894	29.3419
10	DO	0.10919	-8.28E+06	-1.08E+07	87.1974	29.5452
11	TDS	0.10635	-8.28E+06	-1.08E+07	88.1733	31.6306
12	PO_4^{3-}	0.08947	-8.28E+06	-1.08E+07	92.5331	36.8418
13	NO_3	0.06229	-8.28E+06	-1.08E+07	94.7197	39.8214

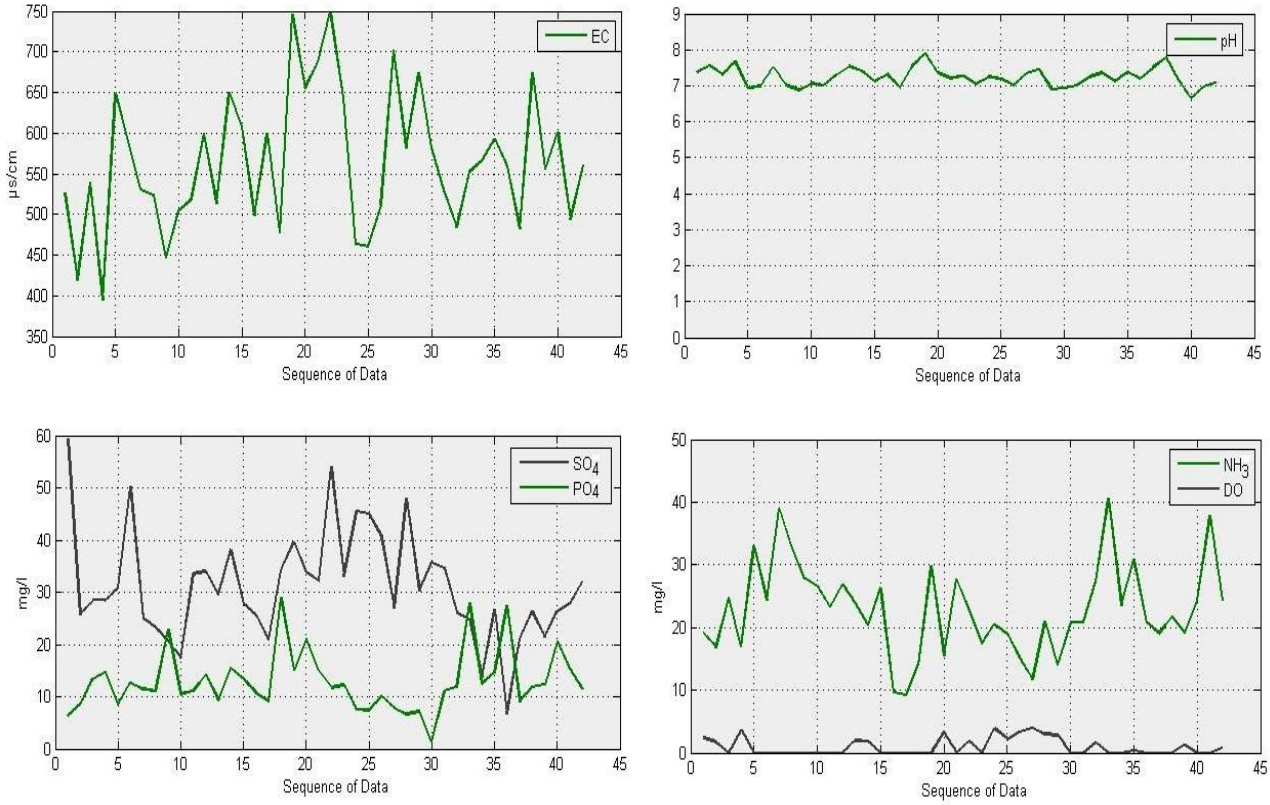
Appendix-D: Graphical plot of potential input variables and target variables for outlier removed data



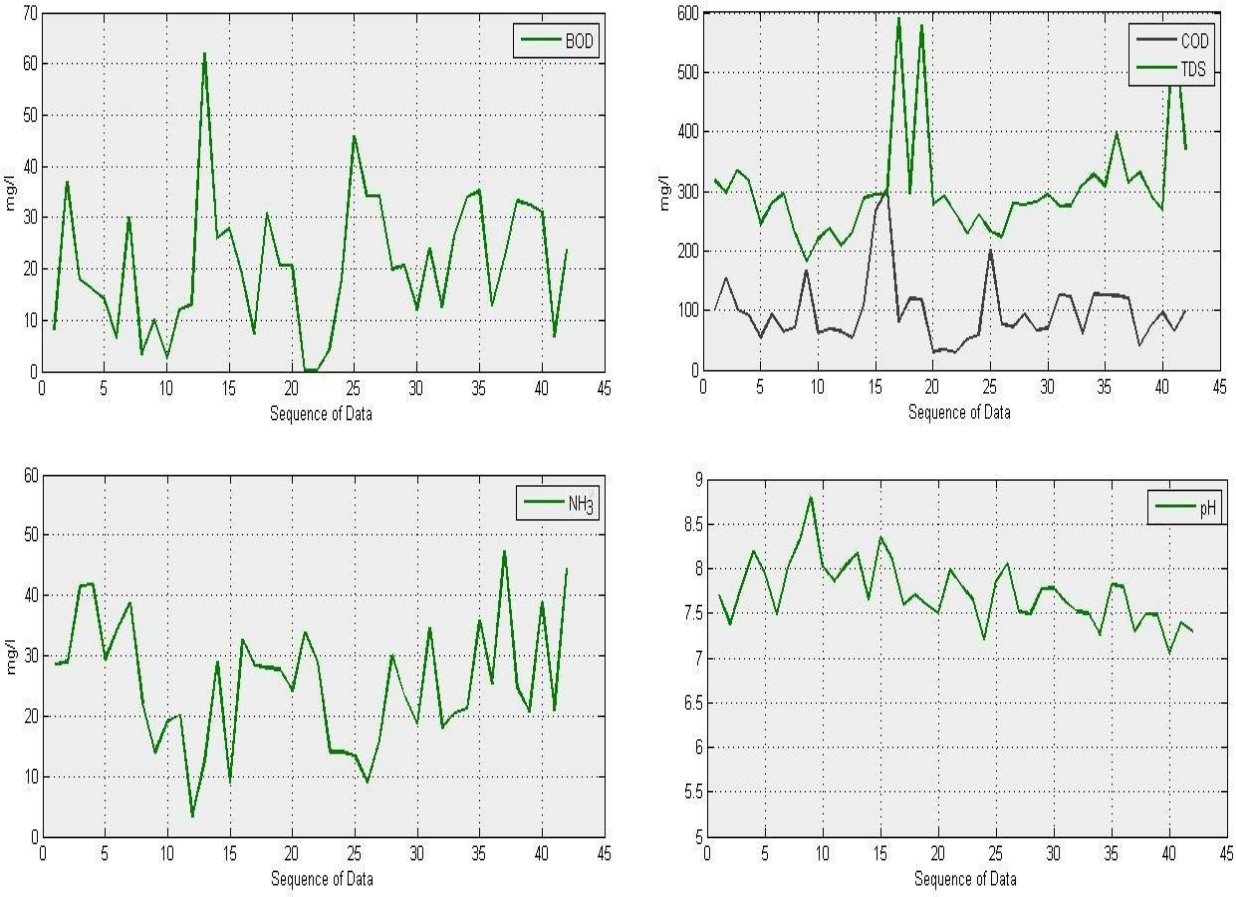








(a) Potentials input Variables



(b) Output Variables

Appendix-E: Network design and software settings

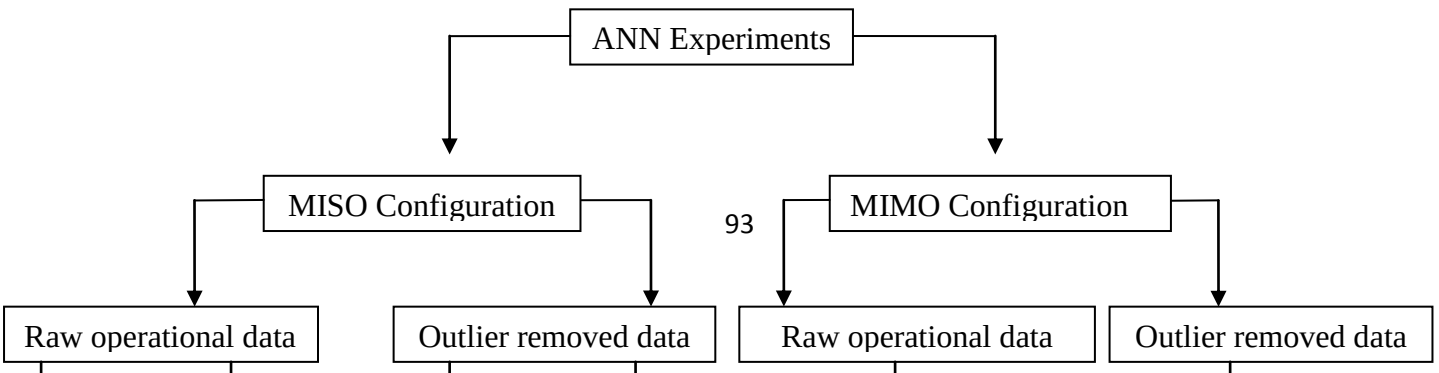
This study utilized a prediction problem type. Predict supports two learning rules: adaptive gradient and kalman filter. The adaptive gradient learning rule uses back propagated gradient architecture to guide an iterative line search algorithm. Brent's algorithm is used to search along that direction for a minimum of the objective function, the adaptive gradient process is repeated until a local minimum of the objective function has been found. The kalman filter learning rule considers the weight to be states and the desired output to be the observations within a discrete state space to search along that direction for a minimum of the objective function. This study used moderately noisy data and very noisy data and thus the adaptive gradient approach and the kalman filter respectively.

Scale data only, superficial data transformation, moderate data transformation, or comprehensive data transformation were used for data analysis and transformation. Predict automatically analyzes data and converts it into a form suitable for building an effective network. This may include converting string inputs to categorical inputs, scaling and transforming data, and removing outliers.

A genetic algorithm that is incorporated in the software is used for input variable selection. Predict utilizes a genetic algorithm to search for good sets of input variables as created by the data analysis and transformation component. For each possible set, a network is developed, and the performance of the network is used to rank the subset of inputs. The no variable selection was used to develop the model using the PMIS. And all the settings: no variable selection, moderate variable selection, comprehensive variable selection, or exhaustive variable selections were used in this study. Predict automatically selects input variables (or transformations of input variables) that are the most influential in predicting target values. A genetic algorithm is used to search through the space of combinations of input variables.

Predict allows the option of training several networks rather than just one. This is important because the first network trained is not necessarily the best model. To identify the best model, different combinations of the number of input processing elements or the number of hidden processing elements are examined. By trial and error, it was determined at least five networks but no more than ten networks would be trained. Predict includes two features referred to as patience and tolerance. The patience level in predict refers to the improvement of fitness within the tolerance for this number of iterations. The tolerance refers to the meaningful improvement in fitness of the model. The maximum number of iterations may not be achieved if the test performance of the network does not improve by more than the tolerance value at the patience level specified for successive networks. The best performing network is retrained at the end.

Another important step in model development within predict is the selection of training, testing, and validation sets. The purpose of developing a neural network model is to produce a formula that captures essential relationships in data. Once developed, this formula is used to interpolate from a new set of inputs to corresponding outputs, in neural networks, this is called generalization. The training set is the set of data points that are used to fit the parameters of the model. The test set measures how well the model interpolates. It is used as part of the model building process to prevent over fitting. The validation set is used to estimate model performance in a deployed environment. The Setup of the ANN experiment using Neuralworks Predict® software is indicated schematically as follows.



Theoretically, MIMO PMI could be done, but in practice multivariate MI estimation using kernel density-based techniques is highly inaccurate due to the curse of dimensionality. Also, consider that one variable might be a good predictor of output A, but not output B. The overall relevance PMI might be reduced based on estimation of I (X; A, B).

Also, in my opinion, MIMO modelling should be avoided for several reasons including:

- Multiple outputs do not necessarily require the same input variables, resulting in a larger ANN architecture with insensitive links that can confuse or impair the training process.
- The training surface is more complex and harder to train, which might degrade model performance (the model is trying to learn two things at once).
- Larger architecture of MIMO models may take longer to train and require more data overall.
- Significance of input variables within a MIMO ANN, with respect to each output variable, is harder to determine, making it more difficult to interpret the trained ANN.

Appendix-F: Basic Statistics

i) The Pearson correlation coefficient

Correlation coefficients measure the strength of association between two variables. The most common correlation coefficient, called the Pearson product-moment correlation coefficient**,** measures the strength of the *linear association* between variables. The formula below uses population means and population standard deviations to compute a population correlation coefficient (ρ) from population data. The correlation ρ between two variables is:

$$\rho = [\ 1 / N\ ] * \Sigma\ \{\ [\ (X_i - \mu_X) / \sigma_x\ ] * [\ (Y_i - \mu_Y) / \sigma_y\ ]\ \} \quad \textbf{.....(1)}$$

where N is the number of observations in the population, Σ is the summation symbol, X_i is the X value for observation i, μ_X is the population mean for variable X, Y_i is the Y value for observation i, μ_Y is the population mean for variable Y, σ_x is the population standard deviation of X, and σ_y is the population standard deviation of Y.

ii) The Variance

In a population, variance is the average squared deviation from the population mean, as defined by the following formula:

$\sigma^2 = \Sigma (X_i - \mu)^2 / N$ **……..(2)**

Where σ^2 is the population variance, μ is the population mean, X_i is the *i*th element from the population, and N is the number of elements in the population.

iii) The Standard Deviation

The standard deviation is the square root of the variance. Thus, the standard deviation of a population is:

$\sigma = \text{sqrt} [\sigma^2] = \text{sqrt} [\Sigma (X_i - \mu)^2 / N]$ **……..(3)**

Where σ is the population standard deviation, σ^2 is the population variance, μ is the population mean, X_i is the *i*th element from the population, and N is the number of elements in the population. And the standard deviation of a sample is:

$s = \text{sqrt} [s^2] = \text{sqrt} [\Sigma (x_i - x)^2 / (n - 1)]$ **……(4)**

Where *s* is the sample standard deviation, s^2 is the sample variance, x is the sample mean, x_i is the *i*th element from the sample, and n is the number of elements in the sample.

iv) The Mean and the Median

The two most common measures of central tendency are the median and the mean.

- To find the median, we arrange the observations in order from smallest to largest value. If there is an odd number of observations, the median is the middle value. If there is an even number of observations, the median is the average of the two middle values.
- The mean of a sample or a population is computed by adding all of the observations and dividing by the number of observations. In the general case, the mean can be calculated, using the following equations:

Population mean = $\mu = \Sigma X / N$**……...(5)**

Where ΣX is the sum of all the population observations, N is the number of population observations, Σx is the sum of all the sample observations, and n is the number of sample observations.

v) The Confidence Interval

The confidence level describes the uncertainty associated with a *sampling method*. Suppose we used the same sampling method to select different samples and to compute a different interval estimate for each sample. Some interval estimates would include the true population parameter and some would not. A 90% confidence level means that we would expect 90% of the interval estimates to include the population parameter; A 95% confidence level means that 95% of the intervals would include the parameter; and so on. To express a confidence interval, we need three pieces of information.

- Confidence level
- Statistic
- Margin of error

Given these inputs, the range of the confidence interval is defined by the *sample statistic* $\pm$ *margin of error*. And the uncertainty associated with the confidence interval is specified by the confidence level. There are four steps to constructing a confidence interval.

- Identify a sample statistic. Choose the statistic (e.g, mean, standard deviation) that we will use to estimate a population parameter.
- Select a confidence level. As we noted in the previous section, the confidence level describes the uncertainty of a sampling method. Often, researchers choose 90%, 95%, or 99% confidence levels; but any percentage can be used.
- Find the margin of error. Often we will need to compute the margin of error, based on one of the following equations.

  Margin of error = Critical value * Standard error of statistic **……(6)**
- Specify the confidence interval. The uncertainty is denoted by the confidence level. And the range of the confidence interval is defined by the following equation.

  Confidence interval = sample statistic $\pm$ Margin of error**……..(7)**

Declaration Sheet

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

Getnet Sewnet Kassahun	_____	_____
Candidate	Signature	Date

Confirmed By:

Dr.-Ing Berhanu Assefa	_____	_____
Advisor	Signature	Date

Place: Addis Ababa University, Addis Ababa Institute of Technology (AAU-AAiT)