

**MODELING THE WATER QUALITY OF LAKE EYMİR USING
ARTIFICIAL NEURAL NETWORKS (ANN) AND ADAPTIVE NEURO
FUZZY INFERENCE SYSTEM (ANFIS)**

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY**

BY

MUHİTTİN ASLAN

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
ENVIRONMENTAL ENGINEERING**

NOVEMBER 2008

Approval of the thesis:

**MODELING THE WATER QUALITY OF LAKE EYMİR USING
ARTIFICIAL NEURAL NETWORKS (ANN) AND ADAPTIVE NEURO
FUZZY INFERENCE SYSTEM (ANFIS)**

**submitted by MUHİTTİN ASLAN in partial fulfillment of the requirements for
the degree of Master of Science in Environmental Engineering Department,
Middle East Technical University by,**

Prof. Dr. Canan Özgen
Dean, Graduate School of Natural and Applied Sciences

Prof. Dr. Göksel N. Demirer
Head of Department, Environmental Engineering

Assoc. Prof. Dr. Ayşegül AKSOY
Supervisor, Environmental Engineering, METU

Examining Committee Members:

Prof. Dr. Kahraman Ünlü
Environmental Engineering, METU

Assoc. Prof. Dr. Ayşegül Aksoy
Environmental Engineering, METU

Prof. Dr. Celal F. Gökçay
Environmental Engineering, METU

Assoc. Prof. Dr. İpek İmamoğlu
Environmental Engineering, METU

Dr. Elçin Kentel
Civil Engineering, METU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Muhittin ASLAN

Signature:

ABSTRACT

MODELING THE WATER QUALITY OF LAKE EYMİR USING ARTIFICIAL NEURAL NETWORKS (ANN) AND ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)

ASLAN, Muhittin

M.S., Department of Environmental Engineering

Supervisor: Assoc. Prof. Dr. Ayşegül AKSOY

November 2008, 122 pages

Lakes present in arid regions of Central Anatolia need further attention with regard to water quality. In most cases, mathematical modeling is a helpful tool that might be used to predict the DO concentration of a lake. Deterministic models are frequently used to describe the system behavior. However most ecological systems are so complex and unstable. In case, the deterministic models have high chance of failure due to absence of priori information. For such cases black box models might be essential. In this study DO in Eymir Lake located in Ankara was modeled by using both Artificial Neural Networks (ANN) and Adaptive Neuro Fuzzy Inference System (ANFIS). Phosphate, Orthophosphate, pH, Chlorophyll-a, Temperature, Alkalinity, Nitrate, Total Kjeldahl Nitrogen, Wind, Precipitation, Air Temperature were the input parameters of ANN and ANFIS. The aims of these modeling studies were: to develop models with ANN to predict DO concentration in Lake Eymir with high

fidelity to actual DO data, to compare the success (prediction capacity) of ANN and ANFIS on DO modeling, to determine the degree of dependence of different parameters on DO. For modeling studies “Matlab R 2007b” software was used.

The results indicated that ANN has high prediction capacity of DO and ANFIS has low with respect to ANN. Failure of ANFIS was due to low functionality of Matlab ANFIS Graphical User Interface. For ANN Modeling effect of meteorological data on DO data on surface of the lake was successfully described and summer month super saturation DO concentrations were successfully predicted.

Keywords: Lake Eymir, artificial intelligence, artificial neural network, modeling, fuzzy logic.

ÖZ

EYMİR GÖLÜ'NDE SU KALİTESİNİN YAPAY SINIR AĞLARI (YSA) VE ADAPTİF SİNİRSEL BULANIK İLİŞKİSEL SİSTEM (ASBİS) İLE MODELLENMESİ

ASLAN, Muhittin

Y. Lisans, Çevre Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Ayşegül AKSOY

Kasım 2008, 122 sayfa

İç Anadolu'nun karasal bölgelerinde bulunan göllere su kalitesi anlamında özel önem verilmesi gerekmektedir. Genel olarak matematiksel modelleme Çözünmüş Oksijen (ÇÖ) modellemesi için sıkça başvurulan bir yöntemdir. Bu kapsamda deterministik modeller sistem davranışının açıklanması için sıklıkla kullanılır. Fakat ekolojik sistemlerin karmaşık ve karmaşık yapıları deterministik modellerin başarısızlığına neden olabilmektedir. Bu gibi durumlarda teorik tabanlı modeller güvenli öncül bilgi eksikliğine bağlı olarak hata yapabilirler. Diğer taraftan, karmaşık algoritma yapıları ve düşük güvenli öncül bilgi gereksinimleri göz önüne alındığında kara-kutu modellerin kullanılması daha iyi sonuçlar verebilmektedir.

Bu çalışmada Ankara'da yer alan Eymir Gölü'nde ölçülen ÇÖ verileri YSA ve ASBİS modelleme araçları kullanılarak modellenmiştir. Elde edilen ve kullanılan giriş

verileri Fosfor, Fosfat, pH, Klorofil-a, Sıcaklık, Alkalinite, Nitrat, Toplam Kjeldhal Azotu, Rüzgar, Yağış ve Hava Sıcaklığı verileridir. Modelleme çalışmasının amaçları; göldeki ÇÖ verisinin mümkün mevcut ekosistem ile yüksek ölçüde doğruluk içerecek biçimde tahmin eden bir YSA modeli geliştirmek, ekolojik modelleme araçları olarak YSA ve ASBİS'in aynı şartlar altında karşılaştırılması ve ÇÖ üzerinde diğer su kalitesi parametrelerinin etki ölçüsünün ortaya konulması. Modelleme çalışmaları sırasında yazılım olarak Matlab R 2007b kullanılmıştır.

Çalışmanın sonucunda, YSA modellemesi ÇÖ tahmininde başarıya ulaşırken ASBİS modelleme çalışmasında aynı başarı gözlenmemiştir. ASBİS'in başarısız olmasının sebebi Matlab ASBİS Kullanıcı Arayüzünün fonksiyonellikten uzak olmasıdır. Ayrıca YSA Modelleme çalışmasında meteorolojik verilerin göl yüzeyindeki ÇÖ üzerine etkileri ve yaz aylarındaki aşırı ÇÖ yükselmesi açıklanabilmektedir.

Anahtar Kelimeler : Eymir Gölü, yapay zeka, yapay sinir ağları, modelleme, bulanık mantık.

To my father

To my mother

To my brother Mustafa Yasin

&

To Funda

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Assoc. Prof. Dr. Ayşegül Aksoy for her support, valuable criticism, and endless and forever patience throughout my study.

I would also like to thank my other committee members Prof. Dr. Kahraman ÜNLÜ, Prof. Dr. Celal F. Gökçay, Assoc. Prof. Dr. İpek İmamoğlu and Dr. Elçin Kentel for their precious suggestions and contributions to this study.

I would like to thank to my chief Hakan Moral for his direct support to this study both technically and conceptually.

I would like to thank to Gamze Güngör Demirci and Tarek El-Ahdab for their unique friendship and support while carrying out on site measurements and laboratory analysis.

I would like thank to my friend Ahmet Codal for his valuable contribution to this study on laboratory analysis.

I would like to thank to my Head of Department Mr Mustafa SATILMIŞ, for his direct support during my thesis study while my professional work in Ministry of Environment and Forestry.

Finally, I would like to express my special thanks to my family for their endless love and patience throughout my study.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ.....	vi
DEDICATION.....	viii
ACKNOWLEDGEMENTS	ix
TABLE OF CONTENTS	x
LIST OF TABLES	xii
LIST OF FIGURES	xiv
ABBREVIATIONS	xviii
CHAPTERS	
1.INTRODUCTION	1
2.LITERATURE REVIEW & THEORETICAL BACKGROUND	4
2.1 LIMNOLOGY AND DISSOLVED OXYGEN	4
2.1.1 <i>Dissolved Oxygen as an Ecological Parameter</i>	5
2.2 WATER QUALITY OF LAKES AND WATER QUALITY MODELING.....	7
2.2.1 <i>Turnover</i>	8
2.2.2 <i>Significance of Water Quality Modeling in Lakes</i>	9
2.3 ARTIFICIAL NEURAL NETWORKS	10
2.3.1 <i>General</i>	10
2.3.2 <i>Basic Concepts of ANNs</i>	11
2.3.3 <i>Learning in ANNs</i>	14
2.3.4 <i>ANNs as Function Predictor</i>	14
2.3.5 <i>Neural Networks in Lake or Reservoir Water Quality Modeling</i>	15
2.4 ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS).....	16
2.4.1 <i>General</i>	16
2.4.2 <i>Basics of Fuzzy Logic</i>	17
2.4.3 <i>Fuzzy Reasoning and Fuzzy Inference Systems (FIS)</i>	22
2.4.4 <i>Adaptive Neuro Fuzzy Inference System (ANFIS)</i>	25
3.MATERIALS AND METHODS.....	30

3.1 LAKE EYMIR -----	30
3.1.1 <i>Field Study</i> -----	32
3.1.2 <i>Laboratory Studies</i> -----	35
3.1.3 <i>Eutrophic Determination of Lake Eymir</i> -----	36
3.2 MODELING STUDIES WITH ARTIFICIAL NEURAL NETWORKS -----	37
3.2.1 <i>Back-propagation Algorithm and Over-fitting</i> -----	37
3.2.2 <i>Training with Artificial Neural Network</i> -----	42
3.2.3 <i>Modeling with ANFIS</i> -----	50
4.RESULTS AND DISCUSSION -----	52
4.1 RESULTS OF ANN MODELING -----	52
PERFORMANCE PARAMETERS -----	55
4.1.1 <i>Results for Reduced Input Data Array</i> -----	66
4.1.2 <i>Sensitivity Analysis</i> -----	75
4.2 RESULTS OF ANFIS MODELING -----	77
5.CONCLUSIONS AND RECOMMENDATIONS -----	90
5.1 CONCLUSIONS -----	90
REFERENCES -----	92
APPENDICES	
A -----	98
B -----	113
C -----	117

LIST OF TABLES

TABLE

3.1	Data Sampling Dates & Data Types.....	33
3.2	Parameter Analysis Methods.....	35
3.3	TSI Based Trophic State Classification	37
3.4	Linear Ranking Procure: Linear Correlation Coefficients with DO	44
3.5	Linear Ranking Procure: Summation and Ranking.....	44
3.6	Training Functions Used for ANN Training	47
3.7	Membership Functions Used for ANFIS Training	51
4.1	Best Results of ANN Modeling: Results for Test Array.....	53
4.2	Best Results of ANN Modeling: Results for Validation Array.....	54
4.3	Best Results of ANN Modeling: Results for Whole Training+Validation+Test) Array.....	55
4.4	Normalized Results of ANN Modeling: Results for Test Array.....	57
4.5	Normalized Results of ANN Modeling: Results for Validation Array.....	58
4.6	Success Order of Input Combinations for ANN Considering (F) Values.....	59
4.7	Comparison of Identical A and B Input Combinations on Correlation Coefficient (R) basis.....	66
4.8	Best Results of ANN Modeling: Results for Reduced Whole Array.....	67
4.9	Best Results of ANN Modeling: Results for Reduced Test Array.....	68
4.10	Mapped Results of ANN Modeling: Results for Reduced Whole Array.....	69

4.11	Mapped Results of ANN Modeling: Results for Reduced Test Array.....	69
4.12	Success Order of Reduced Input Combinations Considering (F) Values	70
4.13	Sensitivity Results of ANN Model for A6 Data Combination(average data)	75
4.14	Best Results of ANFIS Modeling Test Array.....	78
4.15	Best Results of ANFIS Modeling Validation Array.....	78
4.16	Best Results of ANFIS Modeling: Whole Array.....	79
4.17	Mapped Results of ANFIS Modeling: Test Array.....	79
4.18	Mapped Results of ANFIS Modeling: Whole Array.....	80
4.19	Success Order of Input Combinations Considering (F) Values.....	80
B.1	Surface Data.....	112
B.2	Average Data.....	113

LIST OF FIGURES

FIGURE

2.1	Outline of the Perceptron Model.....	12
2.2	Typical Model of a Feed-Forward Neural Network.....	12
2.3	Difference between Crisp and Fuzzy Set.....	19
2.4	Types of Membership Functions.....	20
2.5	Boolean Logic Operators.....	20
2.6	Fuzzy Logic Operators.....	21
2.7	Example Fuzzy Inference System (FIS).....	24
2.8	Basic ANFIS Structure.....	25
2.9	Graphical User Interface (GUI) of ANFIS.....	29
3.1	Satellite Photograph of Lake Eymir and Sample Points.....	32
3.2	YSI 6600 EDS Water Quality Sonde.....	34
3.3	Van Dorn Sampling Apparatus.....	35
3.4	Typical Backpropagation ANN Structure.....	38
3.5	Activation Functions “Logsig” and “Tansig”	48
4.1	“Predicted DO vs. Real DO” Graph of Surface A8 Data Combination	60
4.2	“Linear Regression” Graph of Surface A8 Data Combination.....	60
4.3	“Predicted DO vs. Real DO” Graph of Surface A9 Data Combination	61
4.4	“Linear Regression” Graph of Surface A9 Data Combination	61
4.5	“Predicted DO vs. Real DO” Graph of Average A5 Data Combination	64

4.6	“Linear Regression” Graph of Average A5 Data Combination	64
4.7	“Predicted DO vs. Real DO” Graph of Average A6 Data Combination	65
4.8	“Linear Regression” Graph of Average A6 Data Combination	65
4.9	“Predicted DO vs Real DO” Graph of Reduced Average B5 Data Comb.....	71
4.10	“Linear Regression” Graph of Reduced Average B5 Data Combination	71
4.11	“Predicted DO vs Real DO” Graph of Reduced Average A9 Data Comb.....	72
4.12	“Linear Regression” Graph of Reduced Average A9 Data Comb.....	72
4.13	“Predicted DO vs Real DO” Graph of Reduced Surface B8 Data Comb.....	73
4.14	“Linear Regression” Graph of Reduced Surface B8 Data Combination.....	73
4.15	“Predicted DO vs Real DO” Graph of Reduced Surface B9 Data Comb.....	74
4.16	“Linear Regression” Graph of Reduced Surface B9 Data Combination.....	74
4.17	“Predicted DO vs. Real DO” Graph of Average A3 Data Combination	81
4.18	“Linear Regression” Graph of Average A3 Data Combination.....	81
4.19	“Predicted DO vs. Real DO” Graph of Average A4 Data Combination.....	82
4.20	“Linear Regression” Graph of Average A4 Data Combination.....	82
4.21	“Predicted DO vs. Real DO” Graph of Average A6 Data Combination.....	83
4.22	“Linear Regression” Graph of Average A6 Data Combination.....	83
4.23	“Predicted DO vs. Real DO” Graph of Average A9 Data Combination.....	84
4.24	“Linear Regression” Graph of Average A9 Data Combination.....	84
4.25	“Predicted DO vs. Real DO” Graph of Average B3 Data Combination.....	85
4.26	“Linear Regression” Graph of Average B3 Data Combination.....	85
4.27	“Predicted DO vs. Real DO” Graph of Average B4 Data Combination.....	86
4.28	“Linear Regression” Graph of Average B4 Data Combination.....	86
4.29	“Predicted DO vs. Real DO” Graph of Average B6 Data Combination.....	87

4.30	“Linear Regression” Graph of Average B6 Data Combination.....	87
4.31	“Predicted DO vs. Real DO” Graph of Average B6 Data Combination.....	88
4.32	“Linear Regression” Graph of Average B6 Data Combination.....	88
A.1	“Predicted DO vs. Real DO” Graph of Surface A1 Data Combination	98
A.2	“Linear Regression” Graph of Surface A1 Data Combination	98
A.3	“Predicted DO vs. Real DO” Graph of Surface A2 Data Combination	99
A.4	“Linear Regression” Graph of Surface A2 Data Combination	99
A.5	“Predicted DO vs. Real DO” Graph of Surface A3 Data Combination	100
A.6	“Linear Regression” Graph of Surface A3 Data Combination	100
A.7	“Predicted DO vs. Real DO” Graph of Surface A4 Data Combination	101
A.8	“Linear Regression” Graph of Surface A4 Data Combination	101
A.9	“Predicted DO vs. Real DO” Graph of Surface A5 Data Combination	102
A.10	“Linear Regression” Graph of Surface A5 Data Combination	102
A.11	“Predicted DO vs. Real DO” Graph of Surface A6 Data Combination	103
A.12	“Linear Regression” Graph of Surface A6 Data Combination	103
A.13	“Predicted DO vs. Real DO” Graph of Surface B4 Data Combination	104
A.14	“Linear Regression” Graph of Surface B4 Data Combination	104
A.15	“Predicted DO vs. Real DO” Graph of Surface B5 Data Combination	105
A.16	“Linear Regression” Graph of Surface B5 Data Combination	105
A.17	“Predicted DO vs. Real DO” Graph of Surface B6 Data Combination	106
A.18	“Linear Regression” Graph of Surface B6 Data Combination	106
A.19	“Predicted DO vs. Real DO” Graph of Surface B7 Data Combination	107
A.20	“Linear Regression ” Graph of Surface B7 Data Combination	107
A.21	“Predicted DO vs. Real DO” Graph of Surface B8 Data Combination	108

A.22	“Linear Regression” Graph of Surface B8 Data Combination	108
A.23	“Predicted DO vs. Real DO” Graph of Surface B9 Data Combination	109
A.24	“Linear Regression” Graph of Surface B9 Data Combination.....	109
A.25	“Predicted DO vs. Real DO” Graph of Average A9 Data Combination	110
A.26	“Linear Regression” Graph of Average A9 Data Combination	110
A.27	“Predicted DO vs. Real DO” Graph of Average B5 Data Combination	111
A.28	“Linear Regression” Graph of Average B5 Data Combination	111
A.29	“Predicted DO vs. Real DO” Graph of Average B6 Data Combination	112
A.30	“Linear Regression” Graph of Average B6 Data Combination	112

LIST OF ABBREVIATIONS

ANN	: Artificial Neural Network
ANFIS	: Adaptive Neuro Fuzzy Inference System
TKN	: Total Kjehldal Nitrogen
Total P	: Total Phosphorus
DO	: Dissolved Oxygen
NO ₃	: Nitrate
PO ₄	: Phosphate
A1-A9	: A Type Input Combinations
B1-B9	: B Type Input Combinations
GUI	: Graphical User Interface

CHAPTER 1

INTRODUCTION

Lake Eymir is one of the few natural recreational lakes in Ankara. It has been taken under environmental protection by Environmental Protection Agency for Special Areas (EPASA) of Turkey since 22nd December 1990. The lake suffers from eutrophication and drought conditions (Elahdab, 2003; Tan & Beklioglu, 2005; Altınbilek et al., 1995). Water quality problems have been observed in many ways (Elahdab, 2003; Altınbilek et al, 1995, Karakoç et al, 2003, Tan & Beklioğlu, 2005). However, one of the most significant problems associated also with eutrophication is the declined dissolved oxygen (DO) concentrations in summer months. Several water quality monitoring and assessment studies have been conducted in the lake (Elahdab, 2003; Altınbilek et al, 1995, Karakoç et al, 2003, Tan & Beklioğlu, 2005). However, these studies were not an integrated part of a continuous long term monitoring and only a number of parameters were considered. May be as a result of lack of data, no water quality modeling studies has been performed.

Water quality modeling is not a new issue for the lake ecosystems. However many of these studies have been carried out with deterministic models (Ates et al, 2007). Deterministic models rely on a theoretical basis and several assumptions are used to define a system. Therefore, these models can have the pitfall of being just the abstraction of the real situation in a system. On the other hand, the assumptions used in a deterministic model can be far from representing the real situation.

Mathematical modeling problems are often classified into black box or white box models, according to how much a priori information is available for the system (MacKay, 2004). A black-box model is a system where there is no a priori

information available. A white-box model (also called glass box or clear box) is a system where all necessary information is available. Practically all systems are somewhere between the black-box and white-box models. Therefore, these terminologies are used only as an intuitive guide.

If there is no a priori information, we would try to use as general functions as possible to cover all different mechanisms in a system. An often used approach for black-box models are the neural networks which usually do not assume almost anything about the system and incoming data. The main advantage of black-box models is their ability to learn the system from system's own data. This ability makes the model unique for a system and this uniqueness minimizes the level of affection degree of assumptions.

Human intelligence uses existing knowledge to reduce the number of choices in order to raise the effectiveness of a simulation model manipulation (Chau, 2004). As in traditional model calibration, each time, merely one or two parameters are altered disregarding the synergistic effects of them. This is because if many parameters are altered at the same time, it is easy to get lost in the manipulation direction. However, artificial intelligence techniques are capable to mimic this behavior as well as to complement the deficiency.

In this study, two black-box modeling tools; Artificial Neural Networks (ANN) and Adaptive Neuro Fuzzy Inference System (ANFIS) are used to model the DO concentrations in Lake Eymir are useful in modeling the systems where the interrelationship between system components is not well known. ANNs are mathematical models that mimic the human brain. They make use of the massively parallel local processing and distributed representation that are believed to exist in the brain. Inherent of many specific disciplines (psychology, mathematics, neuroscience, physics, engineering, computer science, philosophy, linguistics and biology) neural network theory helps us to create intelligent systems that can learn any phenomena and create good solutions to problems that have never been introduced before (Karul, 1999).

ANFIS bases on the main idea of fuzzy control or modeling. Fuzzy modeling has proven to be very successful method in building a model that mimics a human control expert who is capable of controlling a plant without thinking in a mathematical model. The control expert specifies his control actions in the form of linguistic rules. These control rules are translated into the framework of fuzzy set theory providing a calculus which can simulate the behavior of the control expert (Nauck et al, 1993). In brief, the main purpose of usage of fuzzy logic is to create a mathematical decision making system that is more close to human decision making via linguistic rules or representative membership functions.

ANNs offer the possibility to solve the tuning problem. The combination of ANNs and fuzzy controllers in ANFIS assembles the advantages of both approaches and avoids the drawbacks of them individually. Although an ANN is able to learn from given data, the trained ANN is generally regarded as a black box. Neither it is possible to extract structural knowledge from the trained neural network, nor can we integrate special information about the problem into the ANN in order to simplify the learning procedure. On the other hand, a fuzzy controller is designed to work with knowledge in the form of rules. But there exist no formal framework for the choice of parameters and the optimization of parameters has to be done manually (Nauck et al, 1993).

Under the light of brief information given above and insufficient water quality data of Eymir for a traditional deterministic water quality model simulations, ANN and ANFIS were utilized to model the DO concentrations. The scope of this study focused on comparing of these black-box models in describing the variation in DO concentrations in Lake Eymir. For this purpose, a water quality monitoring program was conducted in Lake Eymir in March 2005 to November 2005. The data obtained from that study was used as the input to the ANN and ANFIS models.

CHAPTER 2

LITERATURE REVIEW & THEORETICAL BACKGROUND

2.1 Limnology and Dissolved Oxygen

Limnology is the study of lakes, rivers, and wetlands. More specifically has long been preeminent work on energy and material flow, in the manipulation of large enclosures and whole systems, as well as in the use of experimental stream channels and ecosystem modeling. As inferred from the definition limnology is the way science to assess the ecological/biological balance issues in surface freshwater systems. Historically, most biological limnologists have addressed themselves primarily to the study of competitive and predator-prey interactions. Even so, the energy and nutrients that make biological interactions possible are ultimately derived from beyond the shoreline. The light energy needed for photosynthesis and subsequent flow of energy and nutrients from plants to animals is, together with heat energy, obtained via atmosphere. The nutrients necessary for growth are ultimately derived from terrestrial drainage basins, with a sometimes important contribution from beyond delivered atmosphere. Much terrestrial-derived organic matter reaches aquatic systems and supplements that produced by aquatic plants. As a result biological/ecological properties are consequently the result not only of biological interactions below the waterline but by properties and attributes at higher levels in the hierarchy which increasingly modified by human impacts. Human activities increasingly impact both the stage and the actors themselves. Both fundamental and applied research complements this by measuring the impact of various degrees of human activity on aquatic systems that provides the basis of environmental management (Kalf, 2002).

2.1.1 Dissolved Oxygen as an Ecological Parameter

DO levels in aquatic systems probably reveal more about their metabolism than any other single measurement. Concentrations reflect the momentary balance between oxygen supply from the atmosphere and photosynthesis in one hand, and the metabolic processes that consume oxygen on the other (Kallf, 2002). Hutchinson (1957) says that:

“A skilled limnologist can probably learn more about the nature of lake from a series of oxygen determinations than from any other kind of chemical data.”

As inferred from the Hutchinson’s words, DO is one of the typical parameter that has crucial effect on a lake’s ecosystem and water quality. As given below, DO may have significant impact on the survival of fish species (Thomann and Mueller, 1987). In below, “warm waters” refer to temperatures greater than 20 °C and “cold waters” to temperature range between 15 and 20 °C. The DO values represent the minimum level required values.

1. Salmonid Waters (coldwater):
 - a. Embryo and larval stages.

No Production Impairment	= 11
Slight Production Impairment	= 9
Moderate Production Impairment	= 8
Severe Production Impairment	= 7
Acute Production Limit	= 6
 - b. Other life stages

No Production Impairment	= 8
Slight Production Impairment	= 6
Moderate Production Impairment	= 5
Severe Production Impairment	= 4
Acute Production Limit	= 3
2. Non – salmoid waters (warm waters).
 - a. Early Life Stages

No Production Impairment	= 6.5
--------------------------	-------

Slight Production Impairment	= 5.5
Moderate Production Impairment	= 5
Severe Production Impairment	= 4.5
Acute Production Limit	= 4
b. No Production Impairment	= 6
Slight Production Impairment	= 5
Moderate Production Impairment	= 4
Severe Production Impairment	= 3.5
Acute Production Limit	= 3

2.1.1.1 Solubility of Oxygen in Water

The solubility of DO in water is primarily determined by water temperature. At standard pressure, about 14.62 mg/L can be held in distilled water at 0 °C, declining to 11.29 mg/L at 10 °C. At 25 °C 8.26 mg/l of DO is attainable in equilibrium with water-saturated air at standard pressure (Kallf, 2002). As it could be inferred from the Kallf's analyses, the saturated DO in freshwaters would be inversely proportional with the temperature of water. As well as the temperature of water, pressure effects the DO concentration. The solubility of any gas in a volume of liquid is proportional to the pressure that the gas exerts (Henry's Law). Therefore, the solubility of DO declines with decreasing barometric pressure. Salinity also decreases the DO saturation value (Thomann and Mueller, 1987).

The amount of DO that can be held in water in equilibrium with the atmosphere at a particular temperature, pressure and salinity is known as the saturation or equilibrium concentration, the concentration observed in nature is often expressed as a percentage of this value (Kallf, 2002). In freshwaters, there are various sources and sinks of DO. Sources of DO are reaeration from atmosphere, photosynthetic oxygen production, DO in incoming tributaries or effluents, DO input from wind driven turbulence. The sinks of DO are oxidation of carbonaceous waste material, oxidation of nitrogenous waste material, oxygen demand of sediments, and use of oxygen for respiration by aquatic plants (Thoman and Mueller, 1987). In addition to these direct factors, nutrient concentrations may indirectly impact DO by enhancing algal growth. Especially, for an eutrophic lake in the summer time,

drastic increase in DO observed in the epilimnion is an occasional situation (Chapra, 1997)

2.2 Water Quality of Lakes and Water Quality Modeling

Water quality in lakes and reservoirs is subject to the natural degradation and the impacts of societal development. These factors, especially human activities give rise to eutrophication problem (Kallf, 2002). The water quality of a lake is often described by the trophic state. Lakes are classified as oligotrophic, mesotrophic or eutrophic, depending on the availability of nutrients and the productivity of the water in terms of living species. Oligotrophic lakes are usually more aesthetically pleasing than eutrophic lakes. Water quality is generally more suitable for a wider range of uses. Highly eutrophic bodies often are subject to algal blooms due to excess in nutrient concentration in the lake and DO deficiency (Chapra, 1997).

When we look at the hydraulic behavior of lakes, they behave as non-ideal (arbitrary flow) reactors with long hydraulic residence times. Even small lakes may have residence times of ten years. This means that the hydraulic aspects have little effect on such systems compared to river systems. Major perturbations are caused by wind and ambient air temperature (Tchobanoglous et al., 1987). Temperature variation and density variation may be considered as synonymous in most cases. Since temperature is simple to measure, it is used as a fundamental parameter rather than the density.

Lakes gain and lose energy through the surface because of shear forces from wind, solar heating, and radiant cooling. In warm weather, vertical convection currents are formed because of differential cooling and heating during the day and night. Gradually, the water at lower levels become significantly cooler and denser than that at the surface, and convective forces are damped out except in a surface layer (epilimnion), which may be a few meters deep. Although the epilimnion is well

mixed, the lower layer (hypolimnion) is weakly mixed and usually has distinct gradients in nutrient and oxygen concentrations (Tchobanoglous et al., 1987).

2.2.1 Turnover

A lake's vertical thermal regime has dual significance for the water quality modeler. Temperature has a strong influence on the rates of chemical and biological reactions. (Chapra, 1997) The thermal regime of temperate lakes is primarily the result of the interplay of two processes: (1) heat and momentum transfer across the lake's surface and (2) the force of gravity acting on density differences within lake (Chapra, 1997).

The epilimnion temperatures may approach to 4 °C, while the hypolimnion is lower than 4 °C. At this temperature the density of epilimnion become greater than lower part and an unstable situation occurs. Small perturbations, as from wind shear, result in a turnover of the lake contents, and for a period the lake is completely mixed. After mixing, the entire content of the lake will be less than 4 °C, and the lake will re-stratify with the colder water near the surface. This is called "fall turnover". A "spring turnover" also occurs as the water warms on the surface, the maximum density develops as temperatures approach 4 °C, and instability develops (Tchobanoglous et al., 1987).

Turnover affects the water quality; (1) by changing the nutrient and temperature distribution and (2) by supplying the mixing of the bottom materials with upper portions in the lake throughout the volume. Quite often, nutrients accumulate in the lower depths, either in the sediment or due to lower biological activity. When nutrients are brought to the surface as a result of turnover, with the availability of sunlight and higher temperatures, algal production can be increased. This, in turn, would enhance eutrophication (Tchobanoglous et al., 1987).

Stratification allows the reservoir to be operated with respect to the temperature, and destratification (turnover) destroys this ability to control the

system. Normally, stratification develops over a period of a few weeks, and the times of a year that instabilities develop are not critical (Tchobanoglous et al, 1987).

When stratification occurs in a relatively shallow lake, reservoir, or impoundment, it is often possible that the oxygen resources will be affected because of algal growth, bacterial activity, bacterial activity and the oxygen demands of bottom mud (Tchobanoglous et al, 1987).

2.2.2 Significance of Water Quality Modeling in Lakes

The extreme difficulty in lake modeling forces the question of why it must be done. As in other ecosystems, the reaction of an ecosystem to a change in conditions in many cases would not be instantaneous. Therefore, real-time water quality control is not possible. Response times of lakes are long; many variables are uncontrollable or hard to measure. Moreover, natural factors such as weather conditions play an important role in the water quality. Models might be used to answer three very important questions; (1) Why has the current situation developed? (2) What is likely to happen to lake water quality and organism populations in the future? (3) What will happen if the conditions are changed by a new discharge, removal of a present discharge or a change in discharge quality? The answers to these questions allow public to make plans or choices. Lakes are vulnerable resources and maintenance of the water quality in lakes is essential (Tchobanoglous et al, 1987). Modeling is a critical tool in achieving the better management and therefore preservation of the lakes.

2.3 Artificial Neural Networks

2.3.1 General

The essence of ANNs stemmed from the basics of a biological neuron. The biological neuron is the basic unit of the brain and works on the principle of a logical processing unit. The types of neurons are mainly: 1) interneuron cells, which process locally and have their input and output connections at about 100 microns; 2) output cells that connect different regions of the brain to each other, namely, connect the brain to muscles or connect other organs to brain (Beale et al, 1988).

McCulloch and Pitts (1943) designed systems that are generally regarded as the first ANNs (McCulloch et al., 1943). The weights on a McCulloch-Pitts neuron are set so that the neuron performs a particularly simple logic function while with different neurons perform different functions (Fausett, 1994). Then, Hebb (1949) designed the first learning law for ANNs. The *perceptron* which are large classes of ANNs, was firstly introduced by Rosenblatt (1958; 1959; 1962). In the era of 1970s, the associative neural nets were dealt and developed by Kohonen (1972) and Anderson (1968; 1972). The back-propagation algorithm, which was used in this study, was firstly introduced by Werbos (1972) and developed by Rumelhart et al. (1986) and McClelland et al., (1988).

An ANN is an information processing system that has certain performance characteristics in common with biological neural networks (Fausett, 1994). In addition to this Hetch-Nielsen defined a neural network as (Chitra, 1993):

“... a computing system made up of a number of simple highly interconnected processing elements, which processes information by its dynamic state response to external inputs”

An ANN has four common assumptions:

- Information processing occurs at many simple elements called *neurons*,
- Signals are passed between neurons over connection links,
- Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted,
- Each neuron applies an activation function (usually non-linear) to its net input (sum of weighed input signals) to determine its output signal (Fausett, 1994).

2.3.2 Basic Concepts of ANNs

The ability to learn is a fundamental trait of intelligence. Although a precise definition of learning process is difficult to formulate, a learning process in ANN context can be viewed as the problem of updating network architecture and connection weights so that a network can efficiently perform a task. The network usually must learn the connection weights from available training patterns and these patterns should be representative of the system to be modeled

The basic model, which is analogous to a biological neuron, is shown in Figure 2.1. This model, which is called a perceptron simply performs a weighted sum of inputs (a linear combination), compares this to a threshold value in the processing unit and turns on if this value is exceeded, otherwise it stays as off. Since the inputs are passed through the model neuron to produce the output once, the system is known as a feed forward one (Tapkin, 2004).

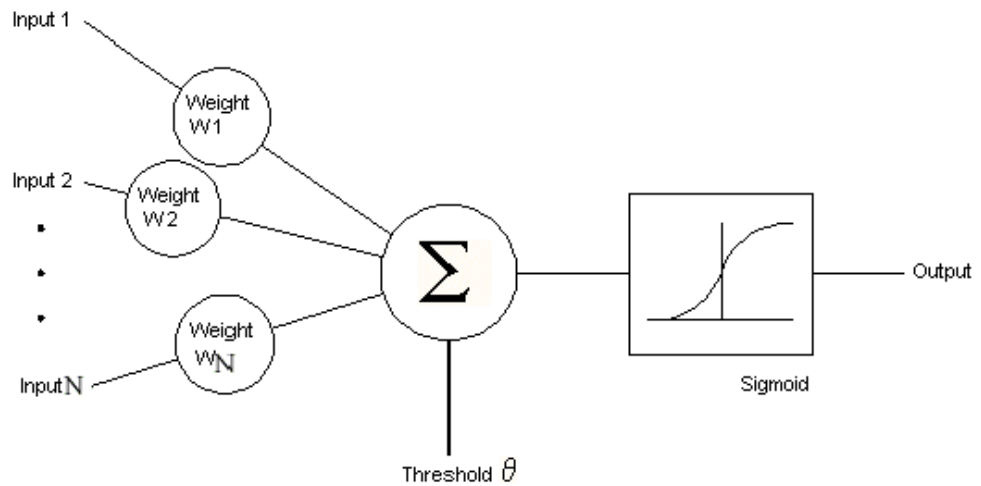


Figure 2.1 Outline of the Perceptron Model

The ANN architecture is a little bit complex from the perceptron model. Figure 2.2 illustrates a typical ANN. An ANN, whether it is feed-forward or feed-back, must have at least three layers. The first layer is the *input layer* in which the input data is introduced to the model. The second layer is the *hidden layer* which processes the input data in the input layer. In hidden layer, *activation function* is applied on neurons. The number of hidden layer might be more than one. Finally the output layer is the component of the network that contains the output data of the model.

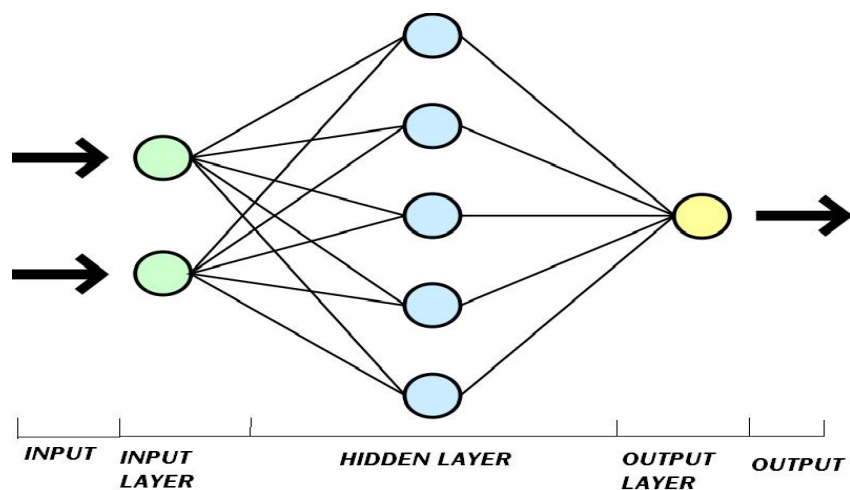


Figure 2.2 Typical Model of a Feed-Forward Neural Network

A connection between the nodes in different layers is called an interlayer connection. A connection between the nodes within the same layer is called an intralayer connection. A connection pointing from a node to itself is called a self-connection. In addition, a connection between nodes in distant (nonadjacent) layers is called a supralayer connection. The term connectivity refers to how nodes are connected. For example, full connectivity often means that every node in one layer is connected to every node in its adjacent layer. A high order connection is a connection that combines inputs from more than one node, often by multiplication. The number of the inputs determines the order of the connection (Simon, 1994).

An ANN is characterized by (1) its pattern of connections between the neurons (called its *architecture*), (2) its method of determining the weights on the connections (called its *training or learning algorithm*), and (3) its *activation function*.

Each neuron is connected to other neurons by means of directed communication links, each with an associated weight. The weights represent the information being used by the network to solve a problem (Fausett, 1994). Connection weights can be real numbers or integers. They can be confined to a range. They are adjustable during network training, but some can be fixed deliberately. When training is completed, all of them should be fixed (Tapkın, 2004).

In accordance with the structural classification there are ultimately two types of neural networks: *feed-forward* and *recurrent feed-back* networks. In the most common family of feed-forward networks, called multilayer perceptron, neurons are organized into layers that have unidirectional connections between them. Different connection patterns yield different network behaviors. Generally speaking, feed-forward networks are *static*, that is, they produce only one set of output values rather than a sequence of values from a given input. Different network architectures require appropriate learning algorithms. In recurrent feedback networks, the inputs of each layer can be affected by the outputs from previous layers. In addition, self feed-back is allowed. The inputs of the network

consist of both external inputs and the network outputs with some delays (Konar, 1999).

2.3.3 Learning in ANNs

The learning procedures of ANNs can be classified into supervised and unsupervised learning. Supervised learning requires an external teacher to control the learning and incorporate the global information. The teacher may be a training data set or an observer who grades the performance. Examples of supervised learning algorithms are the least mean square (LMS) algorithm, and radial basis function network (Fausett, 1994). In supervised learning, the ANN is trained to have the optimal agreement between the ANN output and the training data set. In environmental modeling applications, the training data set can be composed of environmental quality observations. In training of the ANN, the value of the weights in the connections between the neurons is modified according to the input/output samples.

When there is no external teacher, the system must organize itself by internal criteria and local information designed into the network. Unsupervised learning is sometimes referred to as the self-organized learning, i.e. learning to classify without being taught. In this category, only the samples are available and the network classifies the input patterns into different groups. Kohonen network is an example of unsupervised learning (Konar, 1999).

2.3.4 ANNs as Function Predictor

As it is known from mathematics, function is a mathematical expression that creates relationship between two or more variables. This relationship is constructed between dependent and independent variables by calibration of some constants. The number and type of constants are dependent on the type of function. In

nature, there are vast amounts of processes that are complex. Therefore, these processes cannot be explained by simple linear relationships.

Dependent variables, independent variables and calibration of a typical model correspond to input data, output data and training, respectively, in the ANN terminology. However, in ANN modeling approaches, the trained network should also be tested for its generalization (Tapkın, 2004). That is practically to check whether the ANN model established a functional relationship between the input and output data rather than memorizing the behavior.

ANNs are trained essentially by sequentially altering the connection weights between the neurons after each pattern is presented to the network. This process continues until an error term of the computed output value with the target output value is minimized. This error term is generally the average of the sum of the squared errors. ANNs become better predictor as more patterns are presented to the network during the training process (Tapkın, 2004).

2.3.5 Neural Networks in Lake or Reservoir Water Quality Modeling

ANNs have been applied in a number of studies focusing on the modeling of the water quality in lakes or reservoirs. Elhatip and Kömür (2008) modeled the water quality in the reservoir of Mamasin Dam. The results illustrated the ability of ANNs to predict the output values. The output was the water quality parameters of the recharge and discharge areas at the Mamasin Dam site.

Li et al. (2007), applied back-propagation ANN technique to Lake Dianchi, located in southwest of China. In that study, they tried to predict the *Microcystis* spp. population using the typical water quality parameters as inputs. They obtained a correlation coefficient (R^2) of 0.911 between the measured and predicted data of *Microcystis* spp..

Another study was conducted by Karul in 1999. In this study, Karul (1999) applied ANNs as lake management tool. An optimum neural network topology was

determined. Several water quality parameters obtained from Keban Dam, Mogan and Eymir Lakes were trained with the optimum topology. The output was chlorophyll-a concentration. TP, NO₃-N, NH₃-N, water temperature, electrical conductivity, pH, turbidity, secchi depth and suspended solids were used as the inputs. and Chlorophyll-a was the output of the system. A regression coefficient of 0.92 was obtained (Karul, 1999).

Talib and Recknagel (2008) studied the algal dynamics of two Dutch lakes using recurrent ANNs and hybrid evolutionary algorithm. The results indicated that 20 years of efficient nutrient limitation and 10 years of food web biomanipulation limited the population of algae.

2.4 Adaptive Neuro-Fuzzy Inference System (ANFIS)

2.4.1 General

The fuzzy system and neural networks are complementary technologies. The most important reason for combining fuzzy systems with neural networks is to use learning capability of neural network. While the learning capability is an advantage from the view point of a fuzzy system, from the viewpoint of a neural network there are additional advantages to a combined system. Because a neuron-fuzzy system is based on linguistic rules, we can easily integrate prior knowledge in to the system, and this can substantially shorten the learning process. One of the popular integrated systems is a ANFIS (Adaptive Neuro- Fuzzy Inference System), which is about taking a fuzzy inference system and backpropagation algorithm (Lin and Lee, 1996 and Jang.et.al, 1997).

Unlike the traditional hard computing, soft computing methods that utilize fuzzy logic is aimed at an accommodation with pervasive imprecision of the real world. The guiding principle of soft computing is to exploit the tolerance for

imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost. In coming years, soft computing will likely to play an increasingly important role in the conception and design of systems whose MIQ (Machine IQ) is much higher than that of systems designed by conventional methods. Among various combinations of methodologies in soft computing, the one that has the highest visibility is fuzzy logic and neurocomputing, leading to so called *neuro-fuzzy systems*. Neuro-fuzzy systems are multi-layer feed forward adaptive networks that realize the basic elements of fuzzy logic (Matlab Tutorial, 2006).

2.4.2 Basics of Fuzzy Logic

Fuzzy logic can be considered as a logical system that provides a model for modes of human reasoning that are approximations rather than exact” (Rutkowska, 2002). The concept of fuzziness was first proposed by Zadeh (1965). He aimed to describe complex and complicated systems using fuzzy approximation and introduced fuzzy sets. Zadeh (1971) says that as the complexity of a system increases, our ability to make precision and yet significant statements about the behavior diminishes until a threshold is reached beyond which precise and significance become almost mutually exclusive characteristics.

Fuzzy Logic starts with the concept of a fuzzy set. A *fuzzy set* is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. Fuzzy sets differ from classical sets by the acceptance level of an element to a set. A classical set is a set such that any given element is completely included or not. Therefore, *of any subject, one thing must be either asserted or denied*. It was Aristotle who first formulated the Law of the Excluded Middle, which says X must be either in set A or in set-not-A (Matlab Tutorial, 2006).

For example, let us consider a set A of tall person. In classical terms the set is considered crisp and may be defined as a number greater than 170 cm which can be expressed as:

$$A = \{x \mid x > 170\}, \quad (2.1)$$

According to the definition, if x is greater than 170, the x will belong to set A; otherwise x does not belong to the set. However, it is unsuitable that this method in the transition zone, like the heights of 169.9 cm and 170.1 cm, will be result in undesired behavior even almost identical properties. These difficulties can be avoided by generalizing the binary view to the notion of membership in the view of fuzzy sets. We allow a degree of membership which may assume all values between 0 and 1. The value 1 stands for complete membership of the set, while 0 means that an object does not belong to the set at all. (Jeon, 2007)

It should be noted that the example above represents a tricky situation. This is due to the fact that when we make decisions, individual perceptions and cultural background are taken into account. These will likely impact how we define which days constitute the weekend. But, this is exactly the point. Therefore, in such a situation yes-no logic may not be useful. Fuzzy reasoning becomes valuable exactly when we talk about how people really perceive the concept of weekend as opposed to a simple classification (Matlab Tutorial, 2006). In fuzzy logic, the truth of any statement becomes a matter of degree.

The above example is a linguistic description. However, fuzzy logic should be mapped into a mathematical form for computational purposes. The explanation of the fuzzy set and crisp set on mathematical grounds is as follows:

A fuzzy set is a simple extension of the definition of a classical set in which the characteristic function is permitted to have any values between 0 and 1 (Castillo and Melin, 2000). A fuzzy set A in X can be defined as a set of ordered pairs:

$$A = \left\{ \left((x, \mu_A(x)) \mid x \in X \right) \right\} \quad (2.2)$$

where $\mu_A(x)$ is called membership function for the fuzzy set A. It maps x to a membership grade between 0 and 1. In common, three types of membership functions are known which are triangular, trapezoidal and Gaussian. The formulations used for these membership functions are depicted in Figure 2.4

Triangular membership functions:

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (2.3)$$

Trapezoidal membership functions:

$$\text{trapezoidal}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (2.4)$$

Gaussian membership functions: MFs:

$$\text{gaussian}(x; c, \sigma) = e^{-\frac{(x-c)^2}{2\sigma}} \quad (2.5)$$

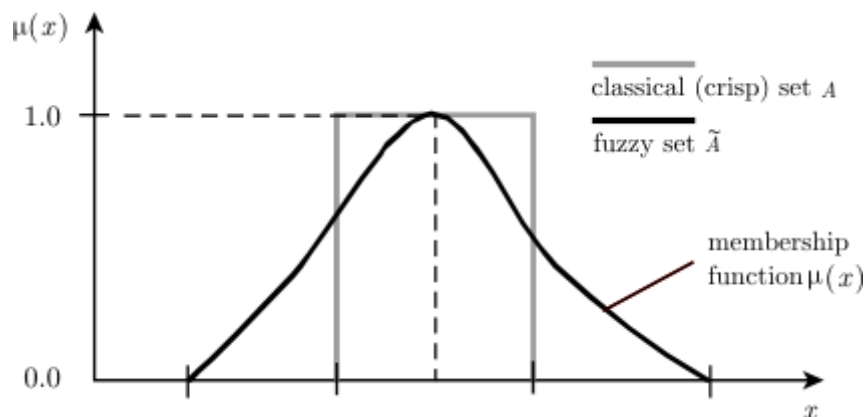


Figure 2.3. Difference between Crisp and Fuzzy Set (Matlab Tutorial, 2006)

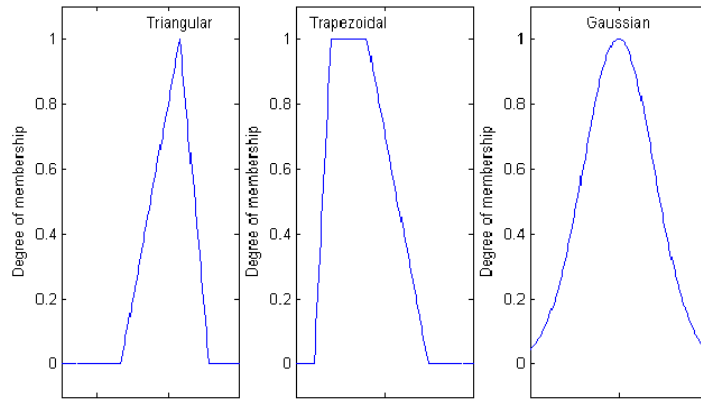


Figure 2.4 Types of Membership Functions (Yilmaz, 2003)

The most important thing to realize about fuzzy reasoning is the fact that it is a superset of standard Boolean logic. In other words, if we keep the fuzzy values at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold as given in Figure 2.5.

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	not A
0	1
1	0

NOT

Figure 2.5 Boolean Logic Operators (Matlab Tutorial, 2006)

Remembering that in fuzzy logic the truth of any statement is a matter of degree, how will these truth tables be altered? The input values can be real numbers between 0 and 1. What function will preserve the results of the AND truth table (for example) and also extend to all real numbers between 0 and 1? One answer is the *min* operation. That is, resolve the statement $A \text{ AND } B$, where A and B

are limited to the range (0,1), by using the function $\min(A,B)$. Using the same reasoning, we can replace the OR operation with the \max function, so that $A \text{ OR } B$ becomes equivalent to $\max(A,B)$. Finally, the operation NOT A becomes equivalent to the operation $1-A$. Notice how the truth table above is completely unchanged by this substitution.

A	B	$\min(A,B)$
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	$\max(A,B)$
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	$1 - A$
0	1
1	0

NOT

Figure 2.6 Fuzzy Logic Operators (Matlab Tutorial, 2006)

Another important concept for fuzzy logic is *if-then rules*. A fuzzy if-then rule (fuzzy rule, fuzzy implication, or fuzzy conditional statement) is expressed as follow:

If x is A then y is B

where A and B linguistic values defined by fuzzy sets. "x is A" is called "antecedent" or "premise", while "y is B" is called the "consequence" or "conclusion" (Castillo and Melin 2000). Some of the if-then rule examples can be given below:

- If pressure is high, then volume is small.
- If the speed is low AND the distance is small, then the force on brake should be small.

We can give a similar example for DO in a lake

- If temperature is high AND salinity is high, then DO should be small.

2.4.3 Fuzzy Reasoning and Fuzzy Inference Systems (FIS)

So many new concepts were described above but a system approach was not developed yet. The concept, *fuzzy reasoning* is the key concept that gives us the procedure of operational steps in fuzzy logic. Jang et al. (1993) explains the *fuzzy reasoning* steps as follows:

1. "Input variables are compared with the membership functions on the premise part to obtain the membership values of each linguistic label (fuzzification).
2. The membership values on the premise part are combined through specific fuzzy set operations such as: min, max, or multiplication to get firing strength (weight) of each rule.
3. The qualified consequent (either fuzzy or crisp) is generated depending on the firing strength.
4. The qualified consequents are aggregated to produce crisp output according to the defined methods such as: centroid of area, bisector of area, mean of maximum, smallest of maximum and largest of maximum (defuzification)".

Fuzzy systems (Fuzzy Inference Systems) are made of a knowledge base and reasoning mechanism is called the fuzzy inference engine. A fuzzy inference engine combines fuzzy if-then rules into a mapping from the inputs of the system into its outputs, using fuzzy reasoning methods. (Czogala et al, 2000) Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. Fuzzy Inference Systems (FIS) are popular computing frameworks based on the concepts of fuzzy set theory, which have been applied with success in many fields like control, decision support, system identification, etc.

Their success is mainly due to their closeness to human perception and reasoning, as well as their intuitive handling and simplicity, which are important factors for acceptance and usability of the systems (Matlab Tutorial, 2006). The most common and oldest FIS was developed by Mamdani et al. (1975). Mamdani type FIS is also called the Transparent Mamdani FIS. Another important type of FIS was developed by Takagi and Sugeno (1985).

The main steps followed in Mamdani FIS are described below:

1. *Fuzzification of Crisp Input*: In this initial step, the crisp input values are fired into membership functions and their respective membership degrees are determined. (As described before the membership degree is always between 0 and 1.)
2. *Application of Fuzzy Operator*: In this step the Fuzzy Operators (please recall AND, OR, NOT \rightarrow min, max, $1-A$ transformation) are applied to the fuzzified input values. It is important that for this operation at least two fuzzified input are needed. At the end of this operation only one value is obtained.
3. *Application of Implication Method (If Then Rules)*: After a single output is fired from the fuzzy operator by comparing at least two inputs, the if-then rule or implication is applied. At this stage the system that is created by the first two steps is called *antecedent*. After antecedent is created the *consequent* is formed with implication. The consequent is also a membership function that measures the linguistic membership degree of the final decision. Recall that ***if x is A then y is B***. The output of the implication method is a fuzzy set that is used in the 4th step.

In some systems the A decision might be weighed between 0 and 1. As noted earlier, fuzzy logic is the modeling of linguistic values and for an FIS it is common that there is more than one A decision with regard to number of

different linguistic inputs. At this point although we have a single value after the second step, this value might be desired to be more efficient or less efficient with respect to its assigned weight.

4. *Summation (Aggregation) of output Fuzzy Sets:* It is common that more one implication is applied and this means that more than one fuzzy set is obtained. In order to get a one crisp input, we need one fuzzy set. Because of this, the output fuzzy sets are geometrically aggregated.
5. *Defuzzification:* In fact this processes the opposite of the first one. Here, the output fuzzy set is transformed into a single crisp value. There are several methods to achieve this. The most convenient one is Centroid Calculation which returns the center of the area under the fuzzy set curve.

The following figure (Figure 2.7) depicts a simple FIS implementation. In this a restaurant customer is trying to decide on the tip that she/he should give while considering the service quality and taste of meal.

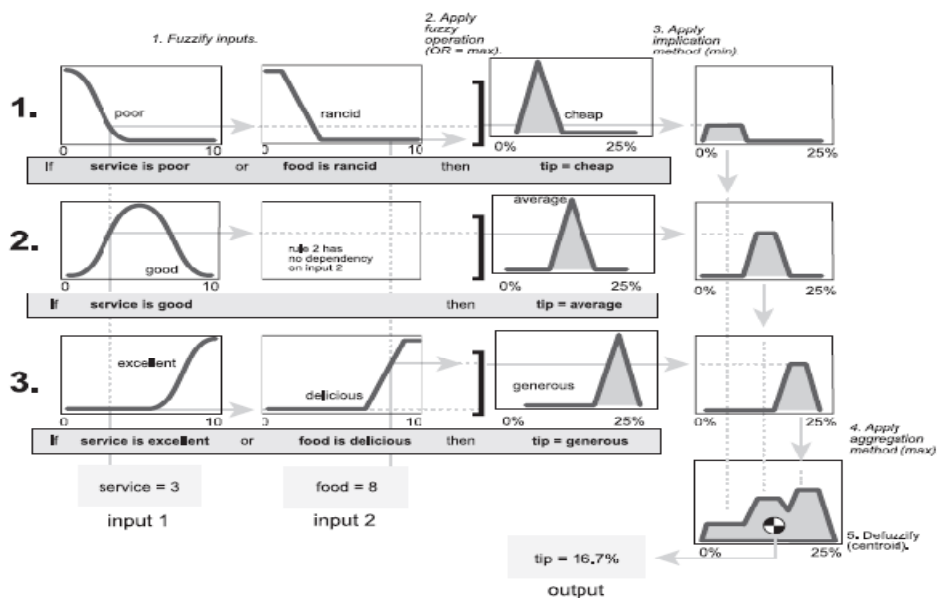


Figure 2.7 Example Fuzzy Inference System (FIS) (Matlab Tutorial, 2006)

2.4.4 Adaptive Neuro Fuzzy Inference System (ANFIS)

In the previous chapter the Mamdani FIS was described because of its simple character and it is easy to understand. On the other hand, ANFIS was built on Takagi – Sugeno FIS Model by Jang et al. (Jang et al., 1993; Takagi et al., 1985). The main difference between Mamdani FIS and ANFIS arises at the end of the third stage given for the Mamdani FIS. The difference of Takagi- Sugeno model is that each rule has a crisp output, and the overall output is determined as weighted average of the single rules output.

In ANFIS, the output of each rule can be a linear combination of input variables plus a constant term or can be only a constant term. The final output is the weighted average of each rule’s output. Basic ANFIS architecture that has two inputs x and y and one output z is shown in Figure 2.8. The rule base contains two Takagi-Sugeno if-then rules as follows:

Rule 1 \rightarrow If x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r_1$

Rule 2 \rightarrow If x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r_2$

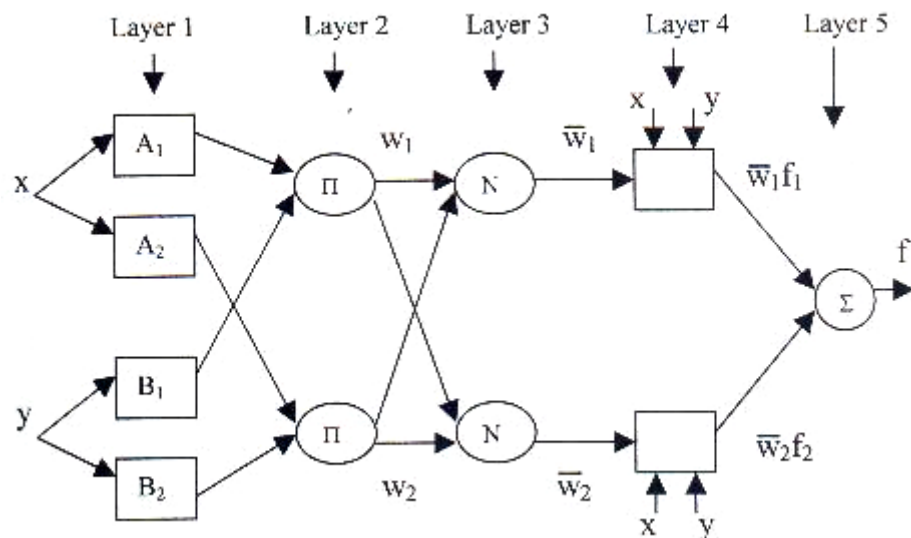


Figure 2.8 Basic ANFIS Structure

Yilmaz (2003) describes the typical ANFIS structure as follows:

The neural network structure contains 5 layers excluding the input layer (Layer 0):

1. *Layer 0*, the input layer, has n nodes where n is the number of inputs to the system.
2. *Layer 1* is the *fuzzification* layer in which each node represents a membership value to a linguistic term as a Gaussian function with the mean:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\frac{(x - c_i)^2}{a_i} \right]^2} \quad (2.6)$$

where a_i , b_i , c_i are parameters for the function. These are adaptive parameters. Their values are adapted by means of the back-propagation algorithm during the learning stage. As the values of the parameters change, the membership function of the linguistic term A_i changes. These parameters are called premise parameters.

In that layer there exist $n \times p$ nodes where n is the number of input variables and p is the number of membership functions. Also it is possible that for an input variable the number of membership functions can be modified. In addition, as well as the Gaussian membership function, other types of membership functions are applicable.

3. In *Layer 2*, each node provides the strength of the rule by means of multiplication operator. It performs *min (AND)* operation.

$$w_i = \mu_{A_i}(x_0) * \mu_{B_i}(x_1) \quad (2.7)$$

Every node in this layer computes the multiplication of the values and gives the product as the output as in Equation 1. The membership values represented by $\mu_{A_i}(x_0)$ and $\mu_{B_i}(x_1)$ are multiplied in order to find the firing

strength of a rule where the variable x_0 has linguistic value of A_i , and x_i has linguistic value of B_i , in the antecedent part of Rule i .

There are p^n nodes denoting the number of rules in Layer 2. Each node represents the antecedent part of rule (**if** part of an **if-then** rule).

4. Layer 3 is the normalization layer which normalizes the strength of all rules according to the below equation

$$\bar{w}_i = \frac{w_i}{\sum_{j=1}^R w_j} \quad (2.8)$$

where w_i is the firing strength of the i^{th} rule which was computed in Layer 2. Node i computes the ratio of the i^{th} rule's firing strength to the sum of all rule's firing strengths. There are p^n nodes in this layer.

5. Layer 4 is a layer of adaptive nodes. Every node in this layer computes a linear function where the function coefficients are adapted by using the error function of the multi-layer feed-forward neural network.

$$\bar{w}_i f_i = \bar{w}_i (p_0 x_0 + p_1 x_1 + p_2) \quad (2.9)$$

p_i 's are the parameters where $i = n + 1$ and n is the number of inputs to the system (i.e. number of nodes in Layer 0). This equation was given for two inputs. Finally \bar{w}_i is the output of Layer 3.

These parameters are updated by a learning step. The least squares approximation is used in ANFIS. In a temporal model, back-propagation algorithm is used for training.

6. Layer 5 is the output layer whose function is the summation of net outputs of the nodes in Layer 4. The output is computed as.

$$\sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.10)$$

where $\bar{w}_i f_i$ is the output of node i in Layer 4. It denotes the consequent part of rule i . The overall output of the neuro-fuzzy system is the summation of the rule consequents (Yilmaz, 2003).

ANFIS uses a hybrid learning algorithm in order to train the network. For the parameters in layer 1, back-propagation algorithm is used. For the training of the parameters in Layer 4, a variation of least squares approximation is used (Yilmaz, 2003).

In this study, MATLAB Fuzzy Logic Toolbox ANFIS GUI (Figure 2.9) was used as a modeling tool. ANFIS GUI is one of the most effective tools for ANFIS Modeling. However with respect theoretical explanation above, MATLAB ANFIS GUI has some limited constraints which are:

1. Be first or zeroth order Sugeno-type systems.
2. Have a single output, obtained using weighted average defuzzification. All output membership functions must be of the same type and either be linear or constant.
3. Have no rule sharing. Different rules cannot share the same output membership function, namely the number of output membership functions must be equal to the number of rules.
4. Have unity weight for each rule (Matlab Tutorial, 2006).

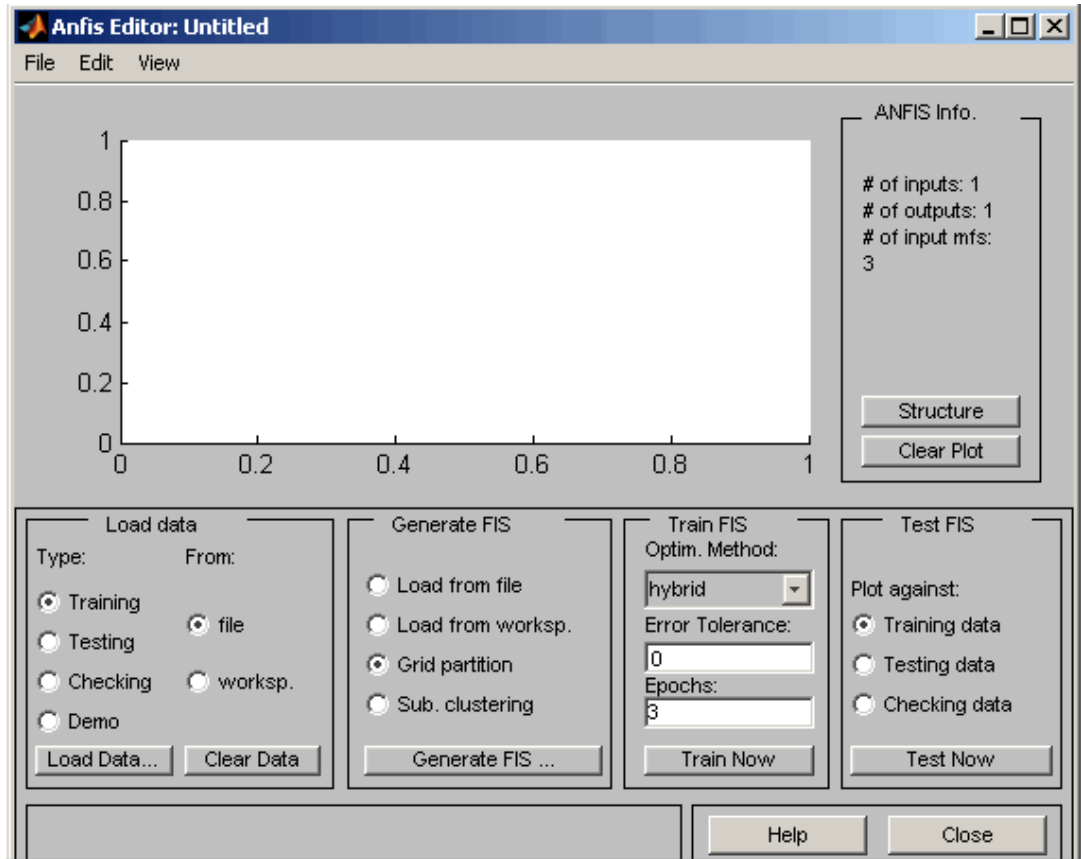


Figure 2.9 Graphical User Interface (GUI) of ANFIS

CHAPTER 3

MATERIALS AND METHODS

In this section the materials used and methods applied were described in a chronological order. The whole study that was conducted for this study employed three different steps. These are:

- Data Collection and Laboratory Studies
- Modeling of DO in Lake Eymir using ANN
- Modeling of DO in Lake Eymir using ANFIS

3.1 Lake Eymir

Lake Eymir is located 20 km south of Ankara. It has been declared as an environmentally protected area in 1990. The lake area changes between 1.05-1.25 km² depending on the depth of water. The lake area of Lake Eymir, with its average depth of about 3 m, is classified as a shallow lake (Tan and Beklioglu, 2005).

Eymir is hydrologically connected to Lake Mogan via concrete lined channel which is located in the southwest of Eymir. Lake Mogan, Kışlakçı Stream (a perennial stream at the northern end), and groundwater sources feed the lake. The excess water of the lake drains into İmrahor Creek at the north (Altınbilek et al., 1995).

There have been several pollution sources that impacted the water quality in the lake. The pollution of the Lake was primarily due to discharge of wastewaters of Gölbaşı district, the wastewaters coming from the sewage treatment plant of TEİAŞ (Turkish Electricity Transmission Company), and the Kışlakçı Creek. In order to improve the water quality of the lake some actions had been taken. In 1994, the slaughterhouse that was discharging to an area close to the lake was shut down.

The TEIAS residency was connected to the Golbasi sewerage system. In addition, in 1995, a 25833 m of bypass line was put in service by ASKI (Ankara Water and Sewage Directorate) in order to avoid the wastewaters of the Golbasi district from entering the lake. However, since the pumps were not operated adequately, Lake Eymir continued to be a receiving body for the Golbasi wastewaters (Altinbilek et al., 1995).

As a result of loading of the lake with pollutants, the lake exhibits eutrophic character (Altinbilek et al., 1995; Elahdab, 2006). The drought conditions observed in recent years resulted in severe deterioration of the water quality in the lake (Elahdab, 2006).

3.2 Data Collection & Laboratory Studies

The water quality data used in modeling of DO was collected within this study. Both in-situ measurements and laboratory analysis were conducted. The water quality parameters measured were: temperature ($^{\circ}\text{C}$), pH, electrical conductivity ($\mu\text{S}/\text{cm}$), alkalinity (mg/L as CaCO_3), PO_4 (mg/l), total phosphorus (TP) (mg/l), $\text{NO}_3\text{-N}$ (mg/l), TKN (mg/l), chlorophyll-a ($\mu\text{g}/\text{l}$), and DO (mg/l). In addition, meteorological data including maximum wind speed (knot), average wind speed (knot), maximum precipitation (mm), and Average precipitation (mm), ambient air temperature ($^{\circ}\text{C}$) were obtained from the meteorological station at Golbasi Police College operated by the General Directorate of Meteorological Works. Water temperature, pH, conductivity, chlorophyll-a ($\mu\text{g}/\text{L}$) and DO (mg/L) were measured at the site by YSI 6600 EDS Sonde Probe. $\text{NO}_3\text{-N}$, TP, $\text{PO}_4\text{-P}$, alkalinity, and TKN were determined by taking samples from the lake and analyzing the samples at the laboratory. Four observation points were selected based on previous works (Altinbilek et al., 1995). These points are illustrated on Figure 3.1. At each point samples were taken at three different depths. These depths are surface, middle and bottom. The exact depths of these points were determined according to the depth at the relevant point. As a result, the total number of measurement points was

twelve. The measurements and laboratory analysis were carried out in the year of 2005 between months of March and October. In this seven months time, the measurement and sample collection studies were carried out in fifteen different dates. These dates are 26th of March, 13th and 22nd of April, 6th and 26th of May, 8th of June, 1st , 13th and 25th of July, 8th and 22nd of August, 5th and 19th of September, 3rd and 19th of October. Although Secchi Depth, Photosynthetically Active Radiation (PAR) parameters were also measured on some dates, they were not included in the modeling studies due to missing data on some dates.

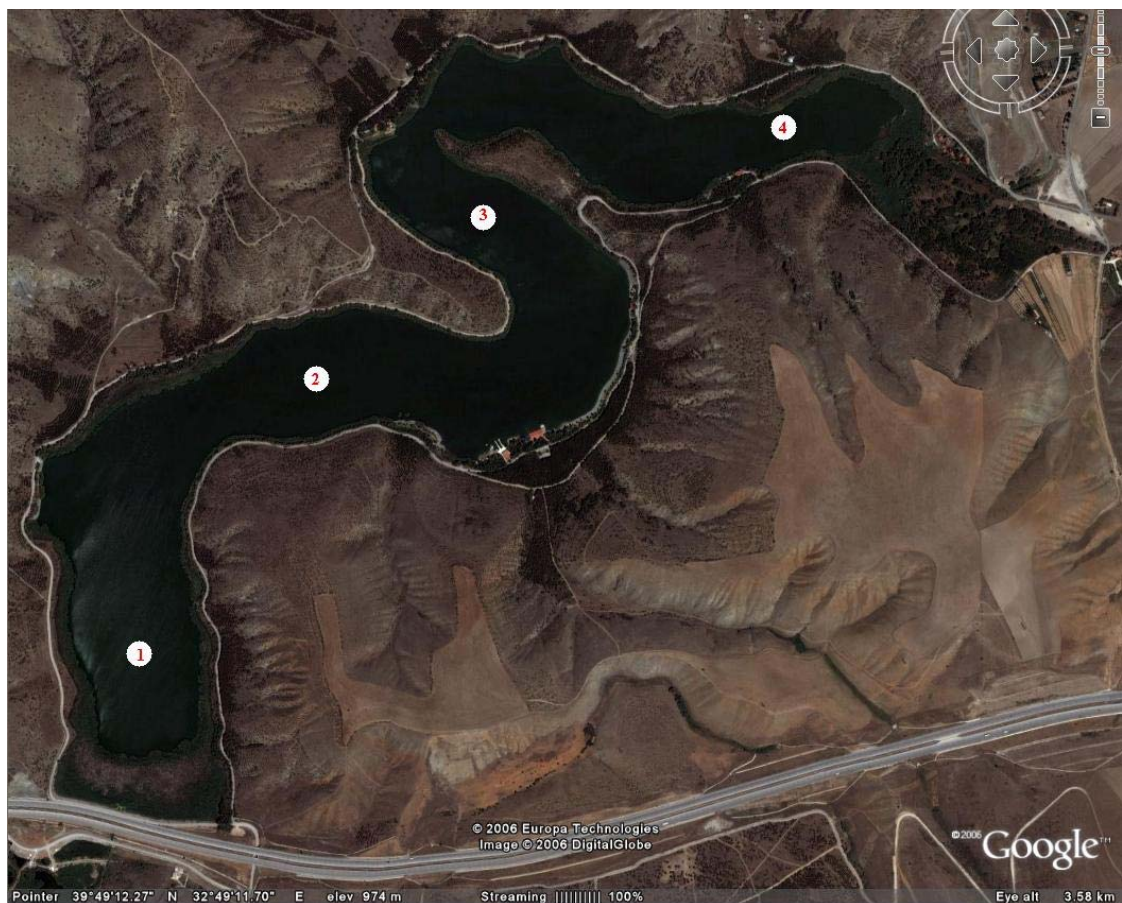


Figure 3.1 Satellite Photograph of Lake Eymir and Sample Points

3.1.1 Field Study

The field study was employed not only for the collection of samples for laboratory analysis, but also to conduct in-situ measurements. As stated before, temperature, pH, conductivity and DO were measured at the site by YSI 6600 EDS

(Figure 3.2), a multiparameter water quality sonde. This instrument can measure DO, turbidity, temperature, depth, Chlorophyll-a, oxidation-reduction potential, conductivity and pH in the water temperature range of -5 to +45 °C.

Table 3.1 Data Sampling Dates & Data Types

PARAMETER	IN-SITU MEASUREMENT	LABRATORY ANALYSIS	STATE METEROLOGICAL INSTITUTE
Alkalinity (mg/L as CaCO ₃)		X	
Chlorophyll (µg/L)	X		
Conductivity (ms/sec)	X		
NO ₃ - (mg/L)		X	
PO ₄ (mg/L)		X	
Precip._Ave (mm)			X
Precip._Max (mm)			X
TKN (mg/L)		X	
Temp (0C)	X		
Total-P (mg/L)		X	
Wind_ave (knot)			X
Wind_max (knot)			X
pH	X		
Dissolved Oxygen (mg/L)	X		

YSI 6600 automatically records data in user specified time intervals. The sonde was calibrated for conductivity and pH using the standard solutions before it was taken to the site. DO calibration was performed at the site. The time interval for data recording was set to two seconds, and the probe was suspended into the water very slowly and data was recorded until it hit to the bottom of the lake. Then, the data was transferred to a regular PC or a laptop by using Eco-Watch Software which is supplied by YSI with the sonde. Since measurements were obtained in a

very short time compared to laboratory analysis, it was possible to get in-situ measurements at all sampling points in a sampling study. It took about 5-10 minutes to obtain in-situ data at each sampling point after the sonde was released.

During in-situ measurements, the sonde recorded the depth of water. Then based on that instant information, the mid-depth was determined. Then samples were taken from the surface, mid-depth and bottom of the lake. Samples were taken using the Van Dorn sampling instrument (Figure 3.3). The apparatus was setup and lowered to the specified depth. Then, its messenger was dropped to close the caps of the cylindrical container, resulting in the entrapment of about 3 liters of lake water. Then, the apparatus was pulled out of water, and the sample was emptied into sample containers. These containers were immediately put in a cooler and kept away from direct sun. In the laboratory, sample containers were transferred into another large capacity cooler.



Figure 3.2 YSI 6600 EDS Water Quality Sonde

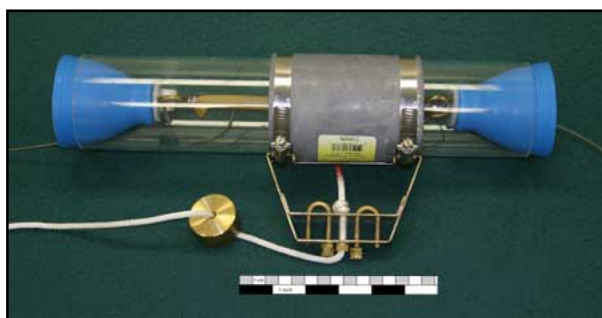


Figure 3.3 Van Dorn Sampling Apparatus

3.1.2 Laboratory Studies

On every sampling date, the twelve samples taken from the lake were brought to the METU Environmental Engineering Department Chemistry Laboratory in coolers approximately in the noon time. In determination of the water quality parameter values, experiments were conducted three times for each sample. In analysis of the samples, Standard Methods were employed. Table 3.2 presents the methods used for analysis. The results of the measurements are given in Appendix A.

Table 3.2 Parameter Analysis Methods

PARAMETER	METHOD
Alkalinity (mg/L as CaCO ₃)	Standard Methods: 2320
NO ₃ -N (mg/L)	Colorimetric Brucine Method (method 352.1) (U.S EPA, 1971)
PO ₄ -P (mg/L)	Standard Methods: 4500 - P (E ⁽¹⁾)
TKN (mg/L)	Standard Methods: 4500 – N _{org} Nitrogen (Organic) (B ⁽²⁾)
TP (mg/L)	Standard Methods: 4500 - P (E ⁽¹⁾)
NH ₃ -N (mg/L)	Standard Methods: 4500 – NH ₃ (B and C ⁽³⁾)

- (1) Ascorbic Acid Method.
 - (2) Macro-Kjeldahl Method
 - (3) Distillation and titrimetric method.
- Standard Methods 2004 Edition was used.

3.1.3 Eutrophic Determination of Lake Eymir

The trophic state of lakes are classified according to their biological productivities:

- Oligotrophic Lakes – Low Biological Productivity
- Mesotrophic Lakes – Moderate Level Biological Productivity
- Eutrophic Lakes – High Level Biological Productivity
- Hypereutrophic Lakes – Highest Level of Productivity.

In the context of this study the eutrophic determination of Lake Eymir was conducted according to the Carlson Trophic Index calculations. (Carlson et al, 1996) The following formulas were suggested by Carlson et al in order to evaluate the Carlson Trophic State Index (TSI);

$$TSI(SD) = 60 - 14.41 \ln(SD) \quad (3.1)$$

$$TSI(CHL) = 9.81 \ln(CHL) + 30.6 \quad (3.2)$$

$$TSI(TP) = 14.42 \ln(TP) + 4.15 \quad (3.3)$$

where;

SD = Secchi Depth (m), CHL: Chlorophyll-a Concentration ($\mu\text{g/L}$), TP: Total Phosphorus Concentration ($\mu\text{g/L}$) (Carlson, 1977)

Under the light of Carlson's approach TSI (CHL) and TSI (TP) was calculated by using annual average data of Lake Eymir. The annual average TP and Chl-a was calculated as: 510 $\mu\text{g/L}$ and 76 $\mu\text{g/L}$. According to this data TSI values were calculated as:

$$TSI(CHL) = 9.81 * \ln(76 \mu\text{g/L}) + 30.6 = 73.20$$

$$TSI(TP) = 14.42 * \ln(510 \mu\text{g/L}) + 4.15 = 94.05$$

According to TSI(CHL) Lake Eymir is hypereutrophic and with regard to TSI (TP) Lake Eymir is beyond hypereutrophic.

Table 3.3 TSI Based Trophic State Classification (Carlson, 1996)

TSI	Chl (ug/L)	SD (m)	TP (ug/L)	Attributes
<30	<0.95	>8	<6	Oligotrophy: Clear water, oxygen throughout the year in the hypolimnion
30-40	0.95-2.6	8-4	6-12	Hypolimnia of shallower lakes may become anoxic
40-50	2.6-7.3	4-2	12-24	Mesotrophy: Water moderately clear; increasing probability of hypolimnetic anoxia during summer
50-60	7.3-20	2-1	24-48	Eutrophy: Anoxic hypolimnia, macrophyte problems possible
60-70	20-56	0.5-1	48-96	Blue-green algae dominate, algal scums and macrophyte problems
70-80	56-155	0.25-0.5	96-192	Hypereutrophy: (light limited productivity). Dense algae and macrophytes
>80	>155	<0.25	192-384	Algal scums, few macrophytes

3.2 Modeling Studies with Artificial Neural Networks

3.2.1 Back-propagation Algorithm and Over-fitting

In this study, back-propagation algorithm was used while training the ANN. Back-propagation is a training method used for prediction problems associated with input/output pairs. Let us think that in a system there are various parameters that describe the system and we desire to construct a relationship structure in which we try to understand the relation of a unique parameter to other ones. For such a

problem a feed-forward neural net that is trained with back-propagation algorithm is the essential solution.

Why back-propagation is an essential solution of prediction of real data? Since, “the main additional specialty of back-propagation algorithm with respect to classical feed-forward networks is the continuous update of multiplier weights in order to minimize the absolute error between real and predicted output values. Figure 3.4 depicts a typical backpropagation ANN algorithm. In this study, MATLAB Neural Network Toolbox was used for network training and ANN buildup purposes. The output of the ANN was DO (mg/l). DO is the ultimate indicator of the water quality which can be easily monitored” (Chapra, 1997).

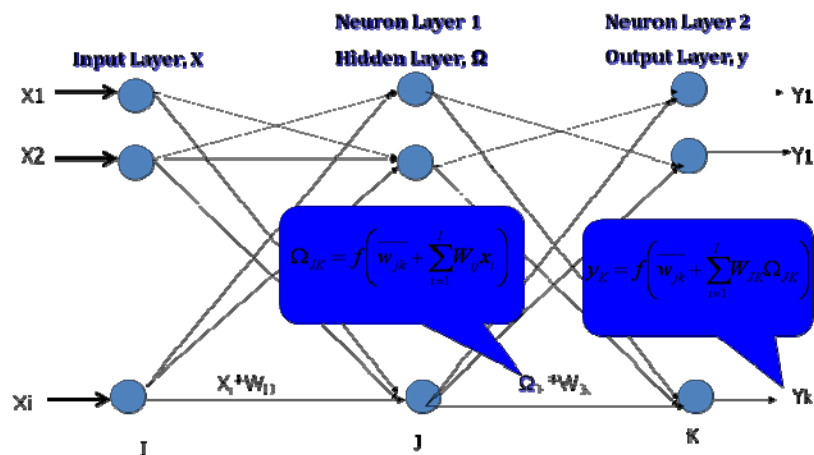


Figure 3.4 Typical Backpropagation ANN Structure

At this point in order to give more clarification regarding back-propagation algorithm the example that is used by Fausett will be given. The steps followed in *gradient descent back-propagation algorithm* is as follows: During feed-forward, each input unit (x_i) receives an input signal and broadcasts this signal to each of the hidden units Z_1, Z_2, \dots, Z_p . Each hidden unit, then, computes its activation and sends its signal (z_j) to each output unit. Each output unit (Y_k) computes its activation (y_k) to form the response of the net signal for the given input pattern.

During training, each output unit, compares its computed activation y_k with its target value (t_k) to determine the associated error for that pattern. Based on this error, the factor δ_k ($k = 1, 2 \dots m$) is computed. δ_k is used to distribute the error at the output unit Y_k back to all units in the previous layer (the hidden units that are connected to Y_k). It is also used (later) to update the weights between the output and the hidden layer. In a similar manner, the factor δ_j , ($j=1, 2 \dots p$) is computed for each hidden unit Z_j . It is not necessary to propagate the error back to the input layer, but δ_j is used to update the weights between the hidden layer and the input layer.

After all the δ factors have been determined, the weights for all layers are adjusted simultaneously. The adjustment to the weight w_{jk} (from hidden unit Z_j to output unit Y_k) is based on the factor δ_k and the activation z_j of the hidden unit Z_j . The adjustment to the weight v_{ij} (from input unit to X_i to hidden unit Z_j) is based on the factor δ_j and the activation x_i of the input unit.

The algorithm is as follows:

- Step 0. Initialize weights
(Set to small random values).
- Step 1. While stopping conditions is false, do steps 2 – 9.
- Step 2. For each training pair, do steps 3 – 8.

Feedforward

- Step 3. Each input unit (X_i , $i = 1, 2, \dots, n$) receives input signal x_i & broadcasts this signal to all units in the layer above (the hidden units).
- Step 4. Each hidden unit (Z_j , $j= 1, 2, \dots, p$) sums its weighed input signals,

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$
applies its activation function to compute its output signal,

$$z_j = f(z_in_j)$$

and sends this signal to all units in the layer above (output units).

Step 5. Each output unit (Y_k , $k = 1, 2, \dots, m$) sums its weighed input signals,

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

and applies its activation function to complete its output signal,

$$y_k = f(y_in_k)$$

Back-propagation of Error

Step 6. Each output unit (Y_k , $k = 1, \dots, m$) receives a target pattern corresponding to the input training pattern, computes its error information term,

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

calculates its weight correction term (used to update w_{kj} later),

$$\Delta w_{jk} = \alpha \delta_k z_j$$

calculates its bias correction term (used to update w_{0k} later.)

$$\Delta w_{0k} = \alpha \delta_k$$

Step 7 Each hidden unit (Z_j , $j = 1, 2, \dots, p$) sums its delta inputs (from in the layers above),

$$\delta_in_j = \sum_{k=1}^m \delta_k w_{jk}$$

multiplies by the derivative of its activation function to calculate its error information term,

$$\delta_j = \delta_{in_j} f'(z_{in_j}),$$

calculates its weight correction term (used to update v_{ij} later)

$$\Delta v_{ij} = \alpha \delta_j x_i$$

and calculates its bias correction term (used to update v_{0j} later),

$$\Delta v_{0j} = \alpha \delta_j$$

Update of weight and biases

Step 8 Each output unit ($Y_k, k = 1, \dots, m$) updates its bias and weights ($j = 0, \dots, p$);

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

Each hidden unit ($Z_j, j = 1, \dots, p$) updates its bias and weights ($i = 0, \dots, n$);

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

Step 9 Test stopping conditions (Fausett, 1994)

Only a single run cycle of this algorithm from Step 1 to 9 is called an *epoch*. While training ANN, the stopping conditions must be determined. For most cases, reaching the desired level of error is the common stopping criteria and for most cases zero (0) error is desired. Another stopping criterion arises from one drawback of ANN structure. This drawback is the danger of *data memorization* or *over fitting* of the neural network.

Over fitting often occurs during neural network training (Tzafestas, 1996). The error on the training set is driven to a very small value, but when new data is presented to the network the error is large (Li, 2007). This is called data

memorization such that network loses its generalization ability. In order to overcome this drawback *Cross-Validation* (Amari, 1997) and *regularization* (Girosi, 1995) are used. In this study, cross-validation was actively used. In the cross-validation technique the available sample data is divided into three subsets, *the training subset, the validation subset, and the testing subset*. The training set is used for computing the gradient and updating the network weights. The validation set is used to monitor the generalization error. The error on the validation set is monitored during the training process. When the network begins to overfit the data, the error on the validation set will typically begin to rise (Li, 2007). As validation error increases, the training will be stopped. Finally the *testing subset* is used to verify the performance of the network.

3.2.2 Training with Artificial Neural Network

3.2.2.1 Data Grouping

In order to model DO in Lake Eymir, the measured water quality and meteorological data was used as the input to the ANN model. Prediction of DO was employed for the surface and the average DO (arithmetic averages of DO measurements for the surface, middle and bottom depths). Modeling of the surface data is important since DO at surface can be distinctively different due to algal activity. Lake Eymir is a eutrophic lake and algal blooms are also observed on the surface of the water. None of the deterministic models can specially deal with surface algal blooms or predict the impact of turnover on DO. The likelihood of turnover increases with high winds, sudden changes in ambient air temperature that would change or break the temperature or density stratification in the lake. However, no general rules can be set for the occurrence of turnover (Chapra, 1997). For example, there is no information in literature which states the absolute temperature difference in the epilimnion and hypolimnion or wind speed which results in turnover. This can be specific for a lake. Actually, this situation makes ANNs and ANFIS interesting for DO prediction at the surface or in the lake. As well

as the surface DO, the average DO in the lake was modeled. The average DO is more representative of the general state of the lake.

For surface and average DO models, the input and output data arrays were prepared. For each measured parameter, the measurements were listed for four observation points in the chronological order of sampling. As a result, for each input array, 60 data points were created. As a result, 14 input vectors (x_i) and one output vector (t_k) were generated. Grouped data table is given in Appendix A.

3.2.2.2 Input Parameter Selection

In prediction of DO using ANNs different parameter sets were used as the input to the model. It is expected that the sensitivity of DO to different input parameters will be different. It is natural that while some parameters have strong correlation with DO some have weak. In this context different combination of input parameters were tested for prediction of DO. In order to do so prior study was carried. Simply a prior “*linear ranking*” procedure was done. First of all the chronological averages of every single data measured or analyzed from different sampling points and depths were calculated. After that *linear correlation coefficients* between DO and every other parameter was calculated for each depth for each sampling point. Then identical correlation coefficients were summed up for each parameter. According to the summations a prior prediction was tried to be made on determination on input combination structures. The results of Linear Ranking Procedure is depicted in Tables 3.4 and 3.5.

Table 3.4 Linear Ranking Procure: Linear Correlation Coefficients with DO (mg/l)

	P1_S	P1_M	P1_B	P2_S	P2_M	P2_B	P3_S	P3_M	P3_B	P4_S	P4_M	P4_B
Alk.	0.619	0.213	0.064	0.672	0.451	0.086	0.633	0.569	0.094	0.555	0.114	0.114
Chl-a	0.419	0.163	0.277	0.283	0.167	0.314	0.164	0.447	0.092	0.211	0.122	0.122
Cond.	0.108	0.264	0.344	0.335	0.264	0.159	0.236	0.362	0.066	0.093	0.117	0.117
NO3	0.176	0.050	0.382	0.220	0.072	0.522	0.077	0.126	0.390	0.011	0.189	0.189
PO4	0.583	0.375	0.109	0.572	0.546	0.033	0.422	0.598	0.051	0.390	0.152	0.152
Precip._Ave	0.057	0.125	0.371	0.166	0.051	0.173	0.244	0.069	0.265	0.172	0.013	0.013
Precip._Max	0.057	0.125	0.371	0.166	0.051	0.173	0.244	0.069	0.265	0.172	0.013	0.013
TKN	0.062	0.077	0.304	0.451	0.201	0.366	0.420	0.390	0.302	0.207	0.049	0.049
Temp	0.405	0.229	0.369	0.371	0.056	0.701	0.335	0.060	0.763	0.257	0.212	0.212
TotalP	0.416	0.328	0.144	0.324	0.406	0.192	0.312	0.414	0.176	0.213	0.091	0.091
Wind_ave	0.272	0.091	0.122	0.334	0.360	0.080	0.323	0.408	0.037	0.355	0.167	0.167
Wind_max	0.360	0.078	0.166	0.287	0.451	0.091	0.272	0.536	0.124	0.317	0.083	0.083
pH	0.445	0.037	0.227	0.553	0.355	0.042	0.359	0.505	0.105	0.356	0.119	0.119

Table 3.5 Linear Ranking Procure: Summation and Ranking

Parameter	Summation	Final Ranking
Alkalinity	4.184	Alkalinity
Chllorophyl	2.778	PO4
Conductivity	2.464	Temp
NO3	2.408	Chlorophyl
PO4	3.982	Total_P
Precip._Ave	1.719	pH
Precip._Max	1.719	TKN
TKN	2.877	Wind_max
Temp	3.972	Wind_ave
TotalP	3.108	NO3
Wind_ave	2.715	Conductivity
Wind_max	2.847	Precip_max
pH	3.221	Precip_ave

While creating the input combinations, it was assumed that nutrient data (NO₃-N, PO₄-P, TP, and TKN) may impact algal activity therefore DO concentrations. In the second assumption only the forms of nutrients (NO₃-N and PO₄-P) the algae can utilize are assumed to be the major input set. The major inputs of the first assumption was called as Core A, and the second assumption Core B. Then other eight parameter arrays, including other parameter measurements and meteorological data, were added to these core input groups one by one starting from the parameter of highest rank.. While doing this addition first water quality data (alkalinity, chlorophyll-a, temperature, pH and conductivity) and

meteorological data (daily maximum precipitation, daily maximum wind speed, ambient air temperature) were added to the input of ANN. As a result following input data groups were constructed:

- A.1 TKN, NO₃-N, TP, PO₄-P(Core A)
- A.2 Core A + alkalinity
- A.3 Core A + alkalinity + temperature
- A.4 Core A + alkalinity + temperature + pH
- A.5 Core A + alkalinity + temperature + pH + chlorophyll-a
- A.6 Core A + alkalinity + temperature + pH + chlorophyll-a + conductivity
- A.7 Core A + alkalinity + temperature + pH + chlorophyll + conductivity + precipitation
- A.8 Core A + alkalinity + temperature + pH + chlorophyll + conductivity + precipitation + wind
- A.9 Core A + alkalinity + temperature + pH + chlorophyll + conductivity + precipitation + wind + ambient air temp

- B.1 NO₃-N, PO₄-P (CoreB)
- B.2 Core B + alkalinity
- B.3 Core B + alkalinity + temperature
- B.4 Core B + alkalinity + temperature + pH
- B.5 Core B + alkalinity + temperature + pH + chlorophyll-a
- B.6 Core B + alkalinity + temperature + pH + chlorophyll-a + conductivity
- B.7 Core B + alkalinity + temperature + pH + chlorophyll-a + conductivity + precipitation
- B.8 Core B + alkalinity + temperature + pH + chlorophyll-a + conductivity + precipitation + wind
- B.9 Core B + alkalinity + temperature + pH + chlorophyll-a + conductivity + precipitation + wind + ambient air temp

3.2.2.3 Data Initialization

After creating the input vector and output vectors, the *data representation for ANN* was carried out. The data representation was made by normalizing the input data and back normalizing the output data. This method was applied in order to convert the input and output values into representative range of values for the computation using ANNs. In order to do this “*norm01.m*” and “*unnorm01.m*” m-files that work under MATLAB was used. In order to generate these files, similar files created by Moral (2004) were utilized. These files are the integrated parts of the network training matrix. These MATLAB files normalizes the data into either [0 1] or [-1 +1] range. “*norm01.m*” script converts the data into [0 1] range, while the built-in *premnmx* function converts the data into the [-1 +1] range.

3.2.2.4 Training

When the normalized input vectors were created, the training was carried out with the script developed by Moral (2004). A script is a series of computerized commands. Therefore, using a script it is possible to conduct a series of actions automatically instead of using one at-a-time manual actions. The script for automating training of ANNs (Moral, 2004; Moral et al., 2008) works under MATLAB Neural Network Toolbox (MNNT). For training of ANNs a three multilayer network (one input, one hidden and one output layers) was used. While training there were two stopping conditions. These are zero (0) error condition between the predicted and observed DO, and data over fitting (memorization) condition. In order to activate the over fitting condition, the input arrays each consisting of 60 elements were divided into three parts chronologically. The first 30 data was used for *training*, following 15 for *validation* and last 15 for *testing*. The validation set was used to check the over fitting condition. Twelve different back-propagation algorithms that was supplied by MNNT was used during network training. The names of these algorithms are given in Table 3.3 (Matlab Tutorial, 2006). After one

of the stopping conditions is fulfilled the output data was back normalized to obtain the DO concentrations and performance analyses were carried out.

Table 3.6 Training Functions Used for ANN Training

Training Functions	Brief Explanation
trainbfg	BFGS Quasi-Newton backpropagation
trainbr	Bayesian Regularization backpropagation
traincgb	Conjugate Gradient backpropagation with Powell-Bale restarts
traincgf	Conjugate Gradient backpropagation with Fletcher-Reeves updates
traincgp	Conjugate Gradient backpropagation with Polak-Ribiere updates
traingd	Gradient – descent backpropagation
traingda	Gradient descent with adaptive learning rate backpropagation
traingdm	Gradient descent with momentum backpropagation
traingdx	Gradient descent with adaptive learning rate and momentum backpropagation
trainlm	Levenberg – Marquardt backpropagation
trainoss	One step secant backpropagation
traainscg	Scaled conjugate gradient backpropagation

The other important point is the selection of transfer functions. For network training two different transfer function were tested. These were *tansig* (Hyperbolic tangent sigmoid transfer function – Figure 3.5) and *logsig* (Log-sigmoid transfer function Figure 3.5).

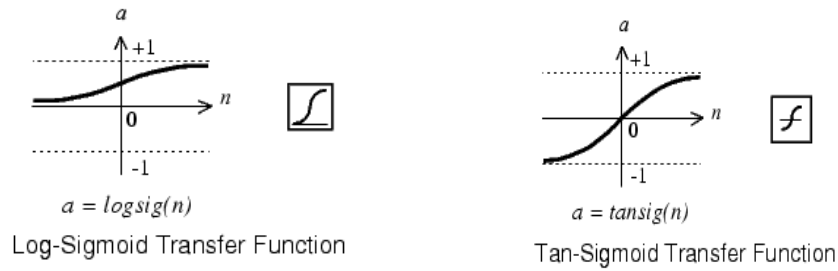


Figure 3.5 Activation Functions “Logsig” and “Tansig” (Matlab ANN Tutorial)

Another ANN parameter selected to construct the ANN DO model was the number of neurons in the hidden layer. 1 to 99 hidden neurons were tested to construct the ANN models. In summary, the following combinations were tested to generate the best ANN architecture for DO modeling.

1. Input parameter combinations = 18 combination
2. Back-propagation algorithms for training of ANNs = 11 combinations
3. Transfer function = 2 transfer function (*tansig* and *logsig*)
4. Number of hidden neurons in the hidden layer = 1 to 99

For each combination three runs were conducted. Procedure was repeated for surface and average DO models. As a result, for ANN modeling part of this study $18 \times 11 \times 2 \times 99 \times 3 \times 2 = 235,224$ ANN training runs were employed in search for the ANN model with the highest performance.

3.2.2.5 Testing Performance and Sensitivity Analysis

The script used for training of ANN models of different architectures applied an automatic performance analysis of the networks based on the linear correlation coefficient (R). R measures the strength and direction of a linear relationship between two variables. The script used for ANN model training checked the R value

for each ANN model and stored the information of the network that exhibited an R value greater than 0.75. Therefore, an initial screening of different ANN models were conducted. Then, the screened ANN models were further examined to decide which one is the best. For this purpose, visual inspection of time-series plots of measured and predicted DO, plots of observed versus predicted DO, and statistical analysis were employed. For the performance analysis, the following parameters were calculated for each ANN model.

1. Absolute Max Error for test, validation and whole arrays $\rightarrow |\max (t_k - y_k)|$
2. Arithmetic average of absolute errors test, validation and whole array $\rightarrow \frac{1}{60} * \sum_{k=1}^{60} |(t_k - y_k)|$
3. R^2 value from curve fitting of predicted versus real DO values for test, validation and whole arrays .
4. Mean Absolute Error for test, validation and whole arrays \rightarrow While $E_k = \frac{y_k - t_k}{t_k} * 100$ then $MAE = \frac{1}{60} \sum_{k=1}^{60} |E_k|$
5. Mean Square Error test, validation and whole arrays $\rightarrow MSE = \frac{1}{60} \sum_{k=1}^{60} (t_k - y_k)^2$

After calculation of five performance parameters, the determination of the best input combination for surface and average data was conducted. In order to do this, the following methodology was followed:

1. First of all each performance parameter calculated for the average and surface DO data was mapped in the range of zero (0) to one (1) by using the following equation:

$$N = \frac{A - \min}{\max - \min} \quad (3.4)$$

where N : Mapped value of parameter A between 0 to 1
 A : Actual Performance parameter in array
 max : max value of respective performance parameter

min : min value of respective performance parameter

2. Then, for each input combination, the equation below was applied on performance parameters mapped to the range of 0 to 1:

$$F: \text{Corr. Coeff}(R) - \text{Max. Error} - \text{Average Error} - \text{MSE} \quad (3.5)$$

where F : Total Performance Parameter

MAE and R^2 were not included in the total performance parameter since they did not supply additional information in determination of F.

In addition, in order to analyze the dominance of the individual parameters on DO, a series of sensitivity analysis were carried out. In sensitivity analysis, the network structure that predicted the real DO data successfully was used. In a network that exhibited good performance according to statistical analysis, the values of each input array was increased by %30, and %50. Similarly, input values were decreased by %30, and %50. Then, the results (output DO) was compared with the DO values obtained without the modification of the input array to see the level of deviation. If a change in one of the parameter arrays resulted in a meaningful deviation in the average DO values, then it was concluded that the model was sensitive to that parameter.

3.2.3 Modeling with ANFIS

As it was stated earlier, one of the ultimate aims of this study is to compare the performances of ANN and ANFIS for modeling of DO in Lake Eymir. In modeling of the DO using ANFIS, only the average DO modeling approach was utilized. Not all input array combinations but selected ones were tested for modeling. using ANFIS. In DO modeling with ANFIS, the MATLAB Fuzzy Logic Toolbox and ANFIS Graphical User Interface was used. While modeling the following procedure was applied. For the mathematical details of the ANFIS, please refer to Chapter 2.

1. Some successful input combinations an ANN Model was determined.
2. Then the input data arrays were divided into three as it was done in ANN Models in order to facilitate the *training*, *validation* and *testing*. Chronologically first 30 data were used for *training*, following 15 were used for *checking (validation)* and final 15 were used for *testing* purposes.
3. The type of membership functions to be tested was determined. The membership function types used by ANFIS are given in Table 3.4.
4. The hybrid learning algorithm was selected as the learning algorithm of ANFIS. The hybrid algorithm was the combination of simple back-propagation and Levenberg-Marquad. Subtractive clustering opportunity of ANFIS GUI was used as the ANFIS structure.
5. While applying subtractive clustering for each input parameter, ANFIS software can apply 1 to 30 membership functions in one cluster. In modeling, all of the number of membership functions were tested.
6. After the outputs were obtained, the performance of modeling was reviewed as it was done for ANN Models (See Section 3.2.2.5).
7. At the end, a comparison table was prepared in order to illustrate the comparison between ANN and ANFIS.

Table 3.7 Membership Functions Used for ANFIS Training (Matlab Tutorial, 2006)

Membership Functions	Brief Explanation
gbellmf	Generalized bell-shaped built-in membership function
gaussmf	Gaussian curve built-in membership function
gauss2mf	Gaussian combination membership function
dsigmf	Built-in membership function composed of difference between two sigmoidal membership functions
psigmf	Built-in membership function composed of product of two sigmoidally shaped membership functions

CHAPTER 4

RESULTS AND DISCUSSION

As stated in the Chapter 3; the modeling studies were conducted under two phases: ANN Phase and ANFIS Phase.

4.1 Results of ANN Modeling

The ANN models that had an R value of 0.75 or higher were screened out both for *surface* and *average DO* models based on R for the whole data. For all of these screened ANN models, five performance determination parameters (See Section 3.2.2.5) were calculated to check the DO prediction capability of ANN models that used different input data sets. Although five different performance parameters were calculated, the threshold for screening was selected based on R. This was due to the fact that the script used for ANN model screening was making the selection based on this parameter. As it might be recalled, while running the ANN models, the input data arrays were divided into three as *training*, *validation* and *test sets*. Although model screening was performed taking the R value of the whole data set (training, validation and test sets), an important performance indicator can be only the performance for the test data (the last 15 chronological data) since this set is never used for training of the ANN. Therefore, if the model predictions for the test set is successful, then performance of the ANN models can be attributed as successful. As a result, in this study whole data set and test data set were used for screening and prediction performance, respectively. The performance parameter values for the validation set are given just as information. The ANN model structures that had the best performance with respect to *test set* are given in Table 4.1. The performance for the *validation set* and *whole (training+validation+test)* set are depicted in Tables 4.2 and 4.3, respectively.

Table 4.1 Best Results of ANN Modeling: Results for Test Array

TEST ARRAY										
SURFACE					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE
	A1	TRAINRP	Logsig	63	8.448	3.818	0.772	0.879	0.27	19.058
	A2	TRAINRP	Logsig	10	8.473	2.786	0.848	0.921	0.166	12.318
	A3	TRAINBFG	Tansig	2	12.697	3.988	0.666	0.816	0.216	28.047
	A4	TRAINCGP	Logsig	62	9.866	3.978	0.757	0.87	0.254	21.388
	A5	TRAINOSS	Logsig	19	9.827	2.385	0.899	0.948	0.155	11.892
	A6	TRAINGDM	Logsig	35	8.553	2.491	0.866	0.93	0.132	12.826
	A7	TRAINCGB	Logsig	85	7.484	2.623	0.881	0.939	0.174	11.696
	A8	TRAINCGB	Logsig	34	7.48	2.778	0.898	0.948	0.169	12.521
	A9	TRAINGDX	Logsig	20	7.496	2.582	0.895	0.946	0.208	10.86
	B4	TRAINSCG	Logsig	19	12.561	3.813	0.691	0.831	0.27	25.288
	B5	TRAINCGP	Logsig	26	7.42	3.186	0.832	0.912	0.205	13.915
	B6	TRAINCGF	Logsig	59	10.107	3.194	0.881	0.938	0.178	18.373
	B7	TRAINCGP	Logsig	80	8.732	2.989	0.878	0.937	0.182	14.801
	B8	TRAINBFG	Logsig	88	9.115	2.674	0.829	0.91	0.137	16.144
B9	TRAINCGB	Logsig	52	10.647	3.796	0.804	0.897	0.219	24.328	
AVERAGE		Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS					
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE
	A5	TRAINLM	Logsig	38	5.396	1.597	0.818	0.904	0.16	13.577
	A6	TRAINLM	Logsig	57	3.876	1.85	0.66	0.813	0.187	11.001
	A9	TRAINLM	Logsig	94	14.739	4.68	0.679	0.824	0.411	62.482
	B5	TRAINLM	Logsig	19	4.97	2.415	0.683	0.826	0.258	7.533
B6	TRAINSCG	Tansig	28	5.264	2.809	0.598	0.774	0.31	9.497	

Table 4.2 Best Results of ANN Modeling: Results for Validation Array

	Input Combination	Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS					
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE
SURFACE	A1	TRAINRP	Logsig	63	11.255	3.676	0.011	0.104	0.294	22.711
	A2	TRAINRP	Logsig	10	8.359	3.833	0.188	0.433	0.292	22.675
	A3	TRAINBFG	Tansig	2	8.587	3.418	0.122	0.349	0.256	18.615
	A4	TRAINCGP	Logsig	62	7.751	2.414	0.462	0.680	0.204	11.109
	A5	TRAINOSS	Logsig	19	14.143	4.036	0.011	0.104	0.318	29.667
	A6	TRAINGDM	Logsig	35	15.479	5.036	0.294	0.543	0.364	39.292
	A7	TRAINCGB	Logsig	85	9.870	4.013	0.165	0.406	0.306	22.541
	A8	TRAINCGB	Logsig	34	9.060	3.856	0.408	0.639	0.242	21.362
	A9	TRAINIDX	Logsig	20	11.561	5.452	0.660	0.812	0.352	46.533
	B4	TRAINSCG	Logsig	19	11.455	4.061	0.150	0.388	0.285	27.106
	B5	TRAINCGP	Logsig	26	10.234	4.396	0.270	0.520	0.405	27.750
	B6	TRAINCGF	Logsig	59	9.437	3.192	0.474	0.688	0.324	17.994
	B7	TRAINCGP	Logsig	80	14.889	6.089	0.509	0.714	0.670	54.061
	B8	TRAINBFG	Logsig	88	17.586	4.954	0.160	0.400	0.458	44.524
	B9	TRAINCGB	Logsig	52	9.852	3.392	0.164	0.405	0.269	19.595
	AVERAGE					PERFORMANCE PARAMETERS				
		Training Algorithm	Activation Function	Number of Hidden Neurons	Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE
A5		TRAINLM	Logsig	38	3.723	1.772	0.699	0.836	0.144	4.426
A6		TRAINLM	Logsig	57	7.262	2.705	0.482	0.694	0.266	10.953
A9		TRAINLM	Logsig	94	4.553	2.220	0.564	0.751	0.238	7.360
B5		TRAINLM	Logsig	19	15.246	4.736	0.007	0.083	0.579	35.353
B6	TRAINSCG	Tansig	28	11.920	4.378	0.327	0.572	0.376	29.250	

Table 4.3 Best Results of ANN Modeling: Results for Whole (Training+Validation+Test) Array

WHOLE ARRAY										
	Input Combination	Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS					
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE
SURFACE	A1	TRAINRP	Logsig	63	11.255	2.530	0.602	0.776	0.210	12.509
	A2	TRAINRP	Logsig	10	8.473	3.591	0.604	0.777	0.275	18.231
	A3	TRAINBFG	Tansig	2	12.697	2.837	0.576	0.759	0.196	14.205
	A4	TRAINCGP	Logsig	62	9.866	2.835	0.573	0.757	0.215	13.618
	A5	TRAINOSS	Logsig	19	14.143	2.335	0.613	0.783	0.182	12.239
	A6	TRAINGDM	Logsig	35	15.479	2.980	0.577	0.760	0.210	17.507
	A7	TRAINCGB	Logsig	85	9.870	3.129	0.656	0.810	0.236	15.725
	A8	TRAINCGB	Logsig	34	9.135	2.487	0.660	0.812	0.173	11.321
	A9	TRAINGDY	Logsig	20	11.561	2.448	0.704	0.839	0.185	15.163
	B4	TRAINSCG	Logsig	19	12.561	2.611	0.576	0.759	0.192	14.319
	B5	TRAINCGP	Logsig	26	10.234	3.201	0.624	0.790	0.227	15.466
	B6	TRAINCGF	Logsig	59	10.107	2.871	0.610	0.781	0.197	14.370
	B7	TRAINCGP	Logsig	80	14.889	3.310	0.603	0.776	0.229	20.193
	B8	TRAINBFG	Logsig	88	17.586	2.400	0.566	0.752	0.166	15.838
	B9	TRAINCGB	Logsig	52	10.647	2.641	0.584	0.764	0.189	13.896
	AVERAGE		Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS				
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE
A5		TRAINLM	Logsig	38	7.262	1.499	0.644	0.802	0.151	10.738
A6		TRAINLM	Logsig	57	4.553	1.092	0.705	0.840	0.111	8.217
A9		TRAINLM	Logsig	94	14.739	3.245	0.500	0.707	0.311	33.597
B5		TRAINLM	Logsig	19	7.126	1.547	0.600	0.774	0.159	4.978
	B6	TRAINSCG	Tansig	28	6.328	1.689	0.578	0.760	0.172	4.828

When we examine the results obtained for the whole data (training, validation and test sets, Table 4.3), surface DO modeling resulted in more ANN models that exhibited an R greater than 0.75 compared to average DO models. This behavior may also be seen when the ANN models were examined based on the success of simulation in terms of R for the test data only (Table 4.1). It must be emphasized that the observation data was used as is without applying any type of prior filtering for removal of inconsistent measurements. In addition, it is possible that particulates entrapped in the sampler when sampling from the bottom of the lake might have caused deviation from the actual concentrations in the water column at some sampling days or points. This might be the case for middle sampling points in the lake. Any bias introduced in terms of depth of sampling would be insignificant for surface data compared to average data, where three different depths are used for calculations. When Tables 4.1, 4.2, and 4.3 were examined, it was seen that use of R or similarly R^2 to determine the best ANN surface and average DO models were not sufficient. For example, the ANN model with the minimum error values (maximum error, average error, MAE and MSE) did not necessarily associated with the ANN model exhibiting the highest correlation. As a result, F values were used to find out the best ANN model to predict the surface and average DO values. For this purpose, first the performance parameter values were mapped into the range of 1 to 0 as discussed in the previous chapter to minimize the impact and dominance of order of magnitude difference in different parameters on the F value. The mapped performance parameter and F values are shown in Tables 4.4, and 4.5, for the test, and whole data sets, respectively. As it can be inferred from Equation 3.5, the errors had a minus sign and R had a plus sign. So the rule was set as “the ANN model with the greater F is better”. The best ANN models determined based on the F values are summarized in Table 4.6 for the average and surface DO predictions.

Table 4.4 Mapped Results of ANN Modeling: Results for Test Array

	Input Combination	Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS						F
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE	
SURFACE	A1	TRAINRP	Logsig	63	0,195	0,894	0,455	0,477	1,000	0,477	-1,088
	A2	TRAINRP	Logsig	10	0,200	0,250	0,781	0,795	0,246	0,085	0,261
	A3	TRAINBFG	Tansig	2	1,000	1,000	0,000	0,000	0,609	1,000	-3,000
	A4	TRAINCGP	Logsig	62	0,464	0,994	0,391	0,409	0,884	0,613	-1,661
	A5	TRAINOSS	Logsig	19	0,456	0,000	1,000	1,000	0,167	0,060	0,484
	A6	TRAINGDM	Logsig	35	0,215	0,066	0,858	0,864	0,000	0,114	0,468
	A7	TRAINCGB	Logsig	85	0,012	0,148	0,923	0,932	0,304	0,049	0,723
	A8	TRAINCGB	Logsig	34	0,011	0,245	0,996	1,000	0,268	0,097	0,647
	A9	TRAINGDX	Logsig	20	0,014	0,123	0,983	0,985	0,551	0,000	0,848
	B4	TRAINSCG	Logsig	19	0,974	0,891	0,107	0,114	1,000	0,839	-2,591
	B5	TRAINCGP	Logsig	26	0,000	0,500	0,712	0,727	0,529	0,178	0,050
	B6	TRAINCGF	Logsig	59	0,509	0,505	0,923	0,924	0,333	0,437	-0,527
	B7	TRAINCGP	Logsig	80	0,249	0,377	0,910	0,917	0,362	0,229	0,062
B8	TRAINBFG	Logsig	88	0,321	0,180	0,700	0,712	0,036	0,307	-0,097	
B9	TRAINCGB	Logsig	52	0,612	0,880	0,592	0,614	0,630	0,784	-1,662	
AVERAGE		Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS						F
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE	RESULTS
	A5	TRAINLM	Logsig	38	0,140	0,000	1,000	1,000	0,000	0,110	0,750
	A6	TRAINLM	Logsig	57	0,000	0,082	0,282	0,300	0,108	0,063	0,155
	A9	TRAINLM	Logsig	94	1,000	1,000	0,368	0,385	1,000	1,000	-2,615
	B5	TRAINLM	Logsig	19	0,101	0,265	0,386	0,400	0,390	0,000	0,034
B6	TRAINSCG	Tansig	28	0,128	0,393	0,000	0,000	0,598	0,036	-0,557	

Table 4.5 Mapped Results of ANN Modeling: Results for Whole (Training+Validation+Test) Array

NORMALIZED WHOLE ARRAY											
	Input Combination	Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS						F
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE	
SURFACE	A1	TRAINRP	Logsig	63	0.305	0.155	0.261	0.272	0.402	0.134	-0,323
	A2	TRAINCGP	Logsig	62	0.000	1.000	0.274	0.285	1.000	0.779	-1,494
	A3	TRAINBFG	Tansig	2	0.464	0.400	0.071	0.075	0.274	0.325	-1,113
	A4	TRAINLM	Logsig	83	0.153	0.398	0.051	0.054	0.447	0.259	-0,756
	A5	TRAINBR	Logsig	18	0.622	0.000	0.338	0.350	0.143	0.103	-0,376
	A6	TRAINGDM	Logsig	71	0.769	0.513	0.081	0.085	0.401	0.697	-1,895
	A7	TRAINLM	Logsig	68	0.153	0.632	0.649	0.661	0.645	0.496	-0,621
	A8	TRAINLM	Logsig	58	0.073	0.121	0.679	0.691	0.064	0.000	0,497
	A9	TRAINGDX	Logsig	20	0.339	0.090	1.000	1.000	0.173	0.433	0,138
	B4	TRAINRP	Logsig	79	0.449	0.220	0.073	0.077	0.235	0.338	-0,930
	B5	TRAINBR	Logsig	54	0.193	0.690	0.420	0.433	0.561	0.467	-0,917
	B6	TRAINCGB	Logsig	44	0.179	0.427	0.316	0.328	0.280	0.344	-0,622
	B7	TRAINOSS	Logsig	72	0.704	0.776	0.265	0.276	0.574	1.000	-2,204
	B8	TRAINCGP	Logsig	12	1.000	0.052	0.000	0.000	0.000	0.509	-1,561
B9	TRAINOSS	Logsig	67	0.239	0.244	0.126	0.132	0.210	0.290	-0,641	
AVERAGE		Training Algorithm	Activation Function	Number of Hidden Neurons	PERFORMANCE PARAMETERS						F
					Max Error	Average Error	R2	Correlation Coefficient	MAE	MSE	
	A5	TRAINLM	Logsig	38	0.266	0.189	0.702	0.714	0.200	0.205	0,054
	A6	TRAINLM	Logsig	57	0.000	0.000	1.000	1.000	0.000	0.118	0,882
	A9	TRAINLM	Logsig	94	1.000	1.000	0.000	0.000	1.000	1.000	-3,000
B5	TRAINBR	Logsig	93	0.253	0.211	0.486	0.507	0.242	0.005	0,037	
B6	TRAINSCG	Tansig	28	0.174	0.277	0.380	0.398	0.305	0.000	-0,053	

Table 4.6 Success Order of Input Combinations for ANN Considering (F) Values

	AVERAGE		SURFACE	
Success Order	1st	2nd	1st	2nd
TEST	A5	A6	A9	A7
WHOLE	A6	A5	A8	A9

Examination of Tables 4.4 and 4.5 reveals that for surface DO modeling, input combinations A9 and A8 were better among other input combinations based on the F values for the test and whole data sets, respectively. For input data sets A8 and A9, the common input parameters were TKN, NO₃-N, TP, PO₄-P, alkalinity, temperature, pH, Chl-a, conductivity, precipitation, and wind were common. A9 had an additional input of ambient air temperature. Performance parameter values for A9 for the test set was such that R was 0.946 (the highest value), and error parameters had comparably less quantities compared to other input combinations (Table 4.1). When whole data set was considered, R was 0.812, with relatively low error quantities for ANN model that used the input combination given by A8.

Figure 4.1 depicts the time-series plots of observed and predicted DO at the surface of the lake for A8. Figure 4.2 shows the linear regression plot of ANN model A8 for surface DO. Figures 4.3 and 4.4 depict the time series and linear regression plots of surface DO, respectively, for ANN model A9. When the linear regression plots for A8 and A9 are compared, the best line for A9 is closer to the theoretical best line.

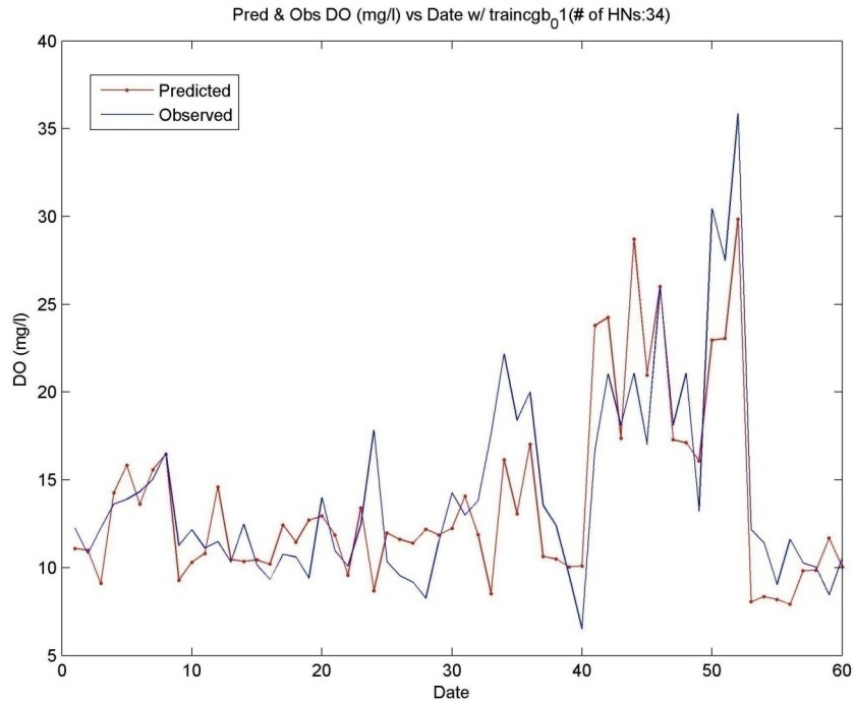


Figure 4.1 “Predicted DO vs. Real DO” Graph of **Surface A8** Data Combination (Training Algorithm: *traincgb*; Actv. Func.: *logsig + logsig*; No. of HD: 34)

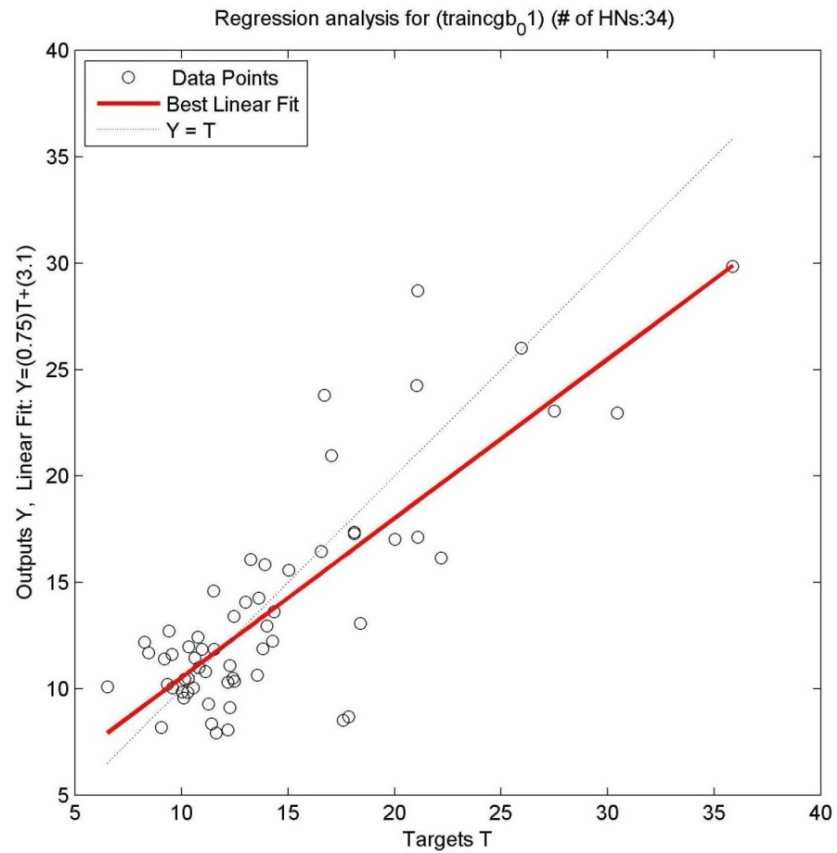


Figure 4.2 “Linear Regression” Graph of **Surface A8** Data Combination (Training Algorithm: *traincgb*; Actv. Func.: *logsig + logsig*; No. of HD: 34)

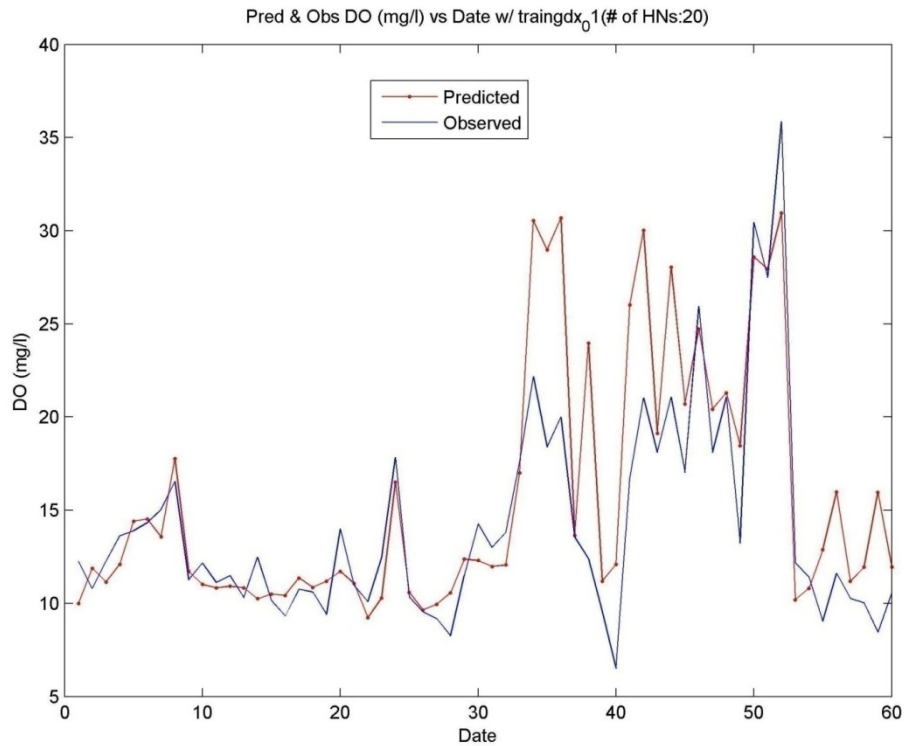


Figure 4.3 “Predicted DO vs. Real DO” Graph of **Surface A9** Data Combination (Training Algorithm: traingdx; Actv. Func.: logsig + logsig; No. of HD: 20)

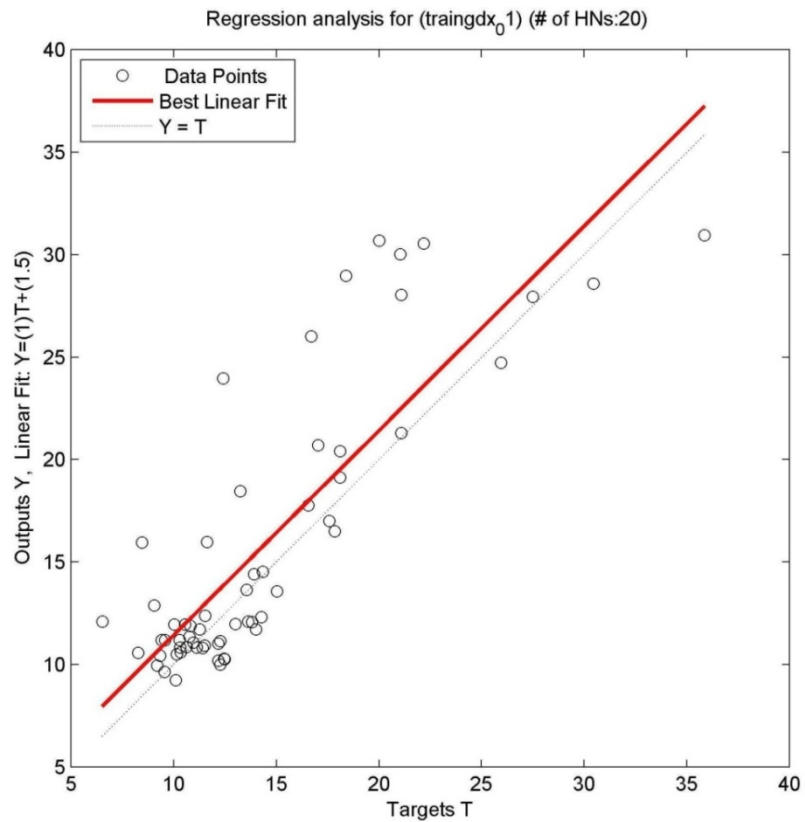


Figure 4.4 “Linear Regression” Graph of **Surface A9** Data Combination (Training Algorithm: traingdx; Actv. Func.: logsig + logsig; No. of HD: 20)

In overall, ANN models that used increasing number of input parameters exhibited high R and R² values in modeling of the surface DO. Especially, addition of meteorological inputs such as precipitation, wind speed and ambient temperature had positive impact on surface DO predictions. It is known that shallow lakes can be impacted by weather conditions more than the deep lakes. In most shallow lakes wind is a driving force for turnover and also reaeration (Chapra, 1997). ANN modeling results indicated that for Lake Eymir, these parameters can be as important as the other water quality parameters in determination of the DO concentrations. It must be noted that inclusion of Chl-a as an input significantly improved the DO predictions at the surface, which is expected.

The visual inspection of Figures 4.1 and 4.3 reveal that, ANN models were successful in predicting the supersaturation conditions covered in the period representing the time frame for the test data. In time-series plots, data after the 45th day represents the test data. Although the ANN models were trained using the data of the first 30 days, ANN models were able to capture the supersaturation conditions at the surface in the period considered for the testing of ANN models. Therefore, ANN models were able to simulate extreme conditions and able to capture the behavior in the lake using a limited number of input parameters. This was the case not only for the best ANN models but also for the others (Appendix A).

Extreme conditions are hard to simulate in deterministic models unless a high number of parameters and calibrated reaction rate constants are used. In Lake Eymir, extremely high DO concentrations are observed at the surface of the lake in summer months due to algal blooms (Elahdab, 2004). Results obtained in this study showed that ANN models were responsive to seasonal behavior of DO concentrations. In addition, ANN was able to establish the non-linear relationship between DO and given input parameters.

ANN modeling was employed for the prediction of average DO in the Lake Eymir as well. In ANN modeling of the average DO in Lake Eymir, the number of

successful ANN models that exhibited an R greater than 0.75 was less compared to surface DO models. As stated in Table 4.6, the best ANN input combinations for the average DO predictions were A5 and A6, when test and whole data sets were considered, respectively, based on the F values. For the whole array, A6 exhibited the highest R as 0.840 with A6. Compared to other input combinations given in Table 4.3, A6 had the best maximum error, average error, R^2 , R, and mean absolute error values. On the other hand, if test data was considered, the highest R was 0.904 obtained for A5. ANN model A5 had the best average error, mean absolute error, R^2 and R values compared to other ANN models that had an R value greater than 0.75. The time series and linear regression plots for A5 are given in Figures 4.5 and 4.6, respectively. Similar plots for A6 are depicted in Figures 4.7 and 4.8, respectively.

A5 input combination includes Core A nutrients, alkalinity, pH, temperature and Chl-a. A6 additionally includes the conductivity parameter. Conductivity is a measure of ions present in the water that can conduct electricity. It is known that algal activity may impact conductivity. In this regard, average DO is highly dependent on input data related with algal growth. This is an expected result due to the hypereutrophic state of the Lake Eymir. For average DO, there was no significant improvement in DO predictions with the addition of meteorological input. Therefore, for average DO, the parameters related with the water column were adequate for prediction of DO. Therefore, in ANN modeling, it is important to use the input data that may have major impact on the mechanism that is being simulated.

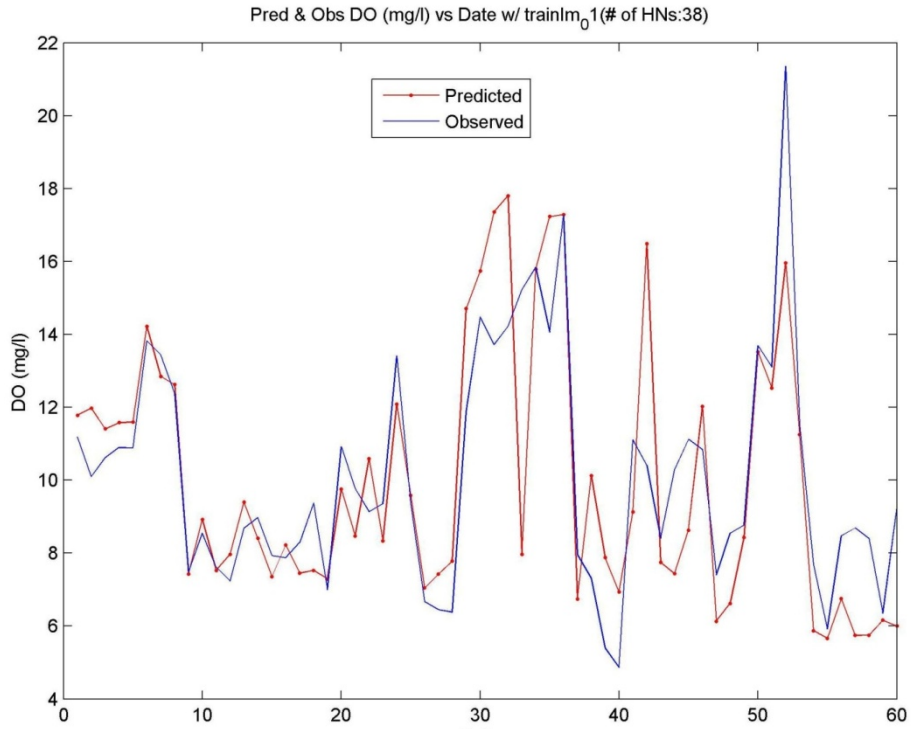


Figure 4.5 “Predicted DO vs. Real DO” Graph of **Average A5** Data Combination (*Training Algorithm: trainlm; Actv. Func.: logsig + logsig; No. of HD: 38*)

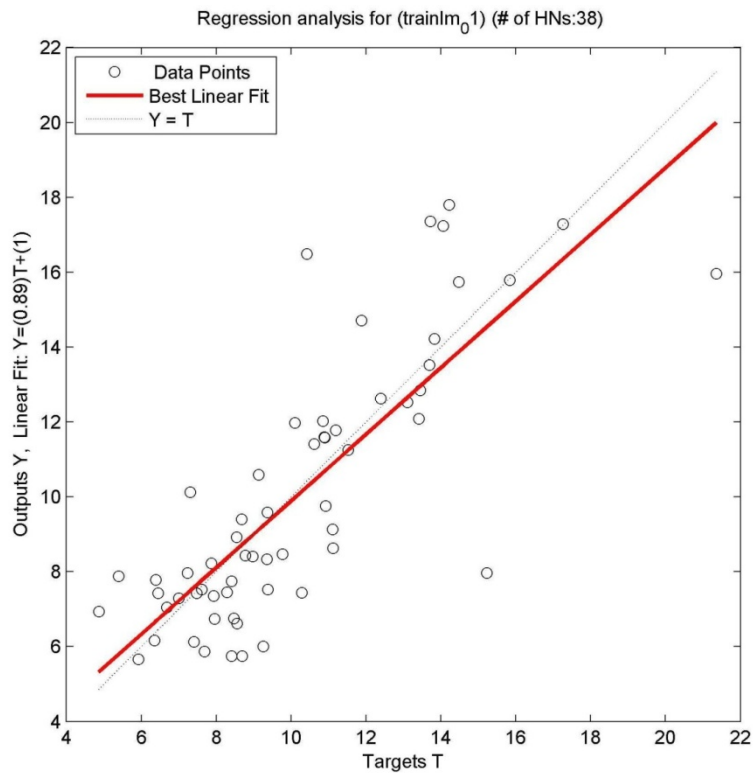


Figure 4.6 “Linear Regression” Graph of **Average A5** Data Combination (*Training Algorithm: trainlm; Actv. Func.: logsig + logsig; No. of HD: 38*)

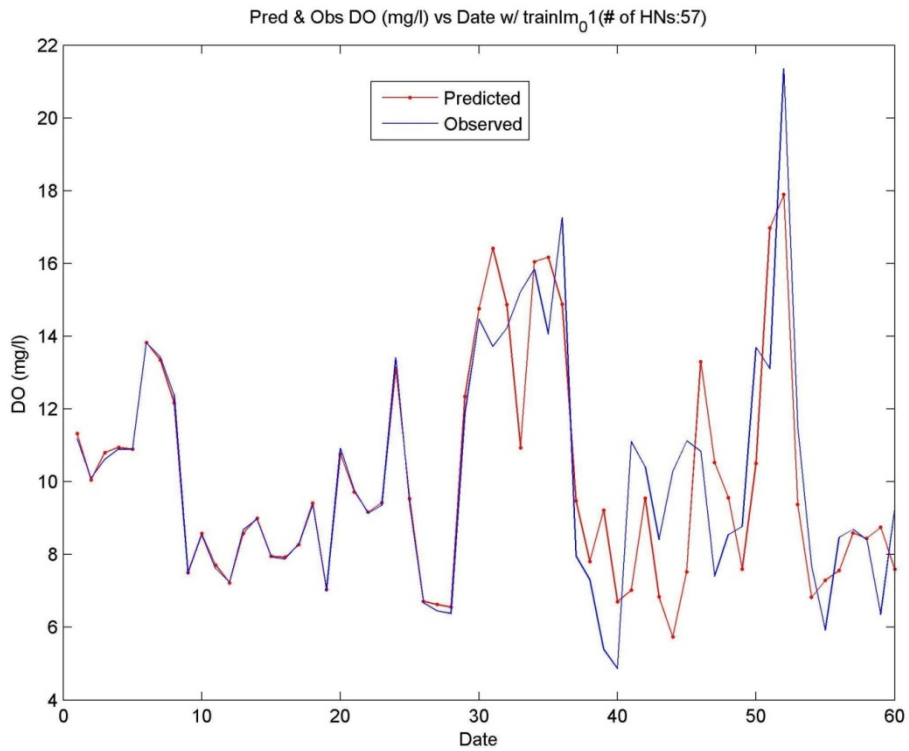


Figure 4.7 “Predicted DO vs. Real DO” Graph of **Average A6** Data Combination (Training Algorithm: *trainlm*; Actv. Func.: *logsig + logsig*; No. of HD: 57)

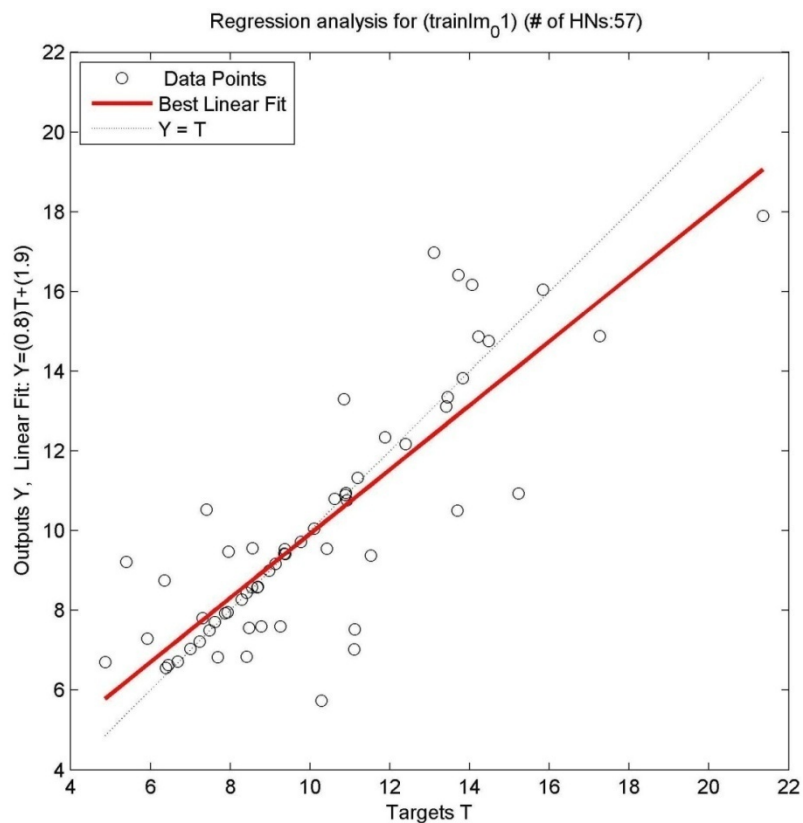


Figure 4.8 “Linear Regression” Graph of **Average A6** Data Combination (Training Algorithm: *trainlm*; Actv. Func.: *logsig + logsig*; No. of HD: 57)

When screened ANN models for surface and average DO modeling were examined, it was seen that Core A input data combinations exhibited better performance compared to Core B input data combinations. In Core B, only NO₃-N and PO₄-P are used as the nutrient data. Algae can readily utilize NO₃-N, NH₄-N and PO₄-P for growth (Chapra, 1997). TKN parameter used in Core A includes NH₄-N. In addition organic P and N would eventually be converted into inorganic forms the algae can utilize for growth. These organic portions are included in the TKN and TP parameters to an extent. Therefore, in general, inclusion of these parameters in Core A groups positively impacted the outcome of ANN models for predicting the surface DO. In Table 4.7, the impact of different core parameter groups on the outcome in terms of R values for different ANN input combinations are shown for the test and whole data sets for surface and average DO modeling.

Table 4.7 Comparison of Identical A and B Input Combinations on Correlation Coefficient (R) Basis

	<i>Correlation Coefficients for Whole Array</i>				<i>Correlation Coefficients for Test Array</i>			
SURFACE	A4	0.757	0.759	B4	A4	0.87	0.831	B4
	A5	0.783	0.790	B5	A5	0.948	0.912	B5
	A6	0.760	0.781	B6	A6	0.93	0.938	B6
	A7	0.810	0.776	B7	A7	0.939	0.937	B7
	A8	0.812	0.752	B8	A8	0.948	0.91	B8
	A9	0.839	0.764	B9	A9	0.946	0.897	B9
AVERAGE	A5	0.802	0.774	B5	A5	0.904	0.826	B5
	A6	0.840	0.760	B6	A6	0.813	0.774	B6

4.1.1 Results for Reduced Input Data Array

All parameters in data combinations are made up of arrays that had 60 elements. This 60 data were originated from sequential order of data of 4 sampling data of 15 different dates. This approach was used in order to increase the data used for training. In order to observe the performance of ANN modeling in predicting the average and surface DO for the average concentrations in the lake,

averages of measurements at the four sampling points for a sampling date were calculated. Therefore, the total number of data was reduced to 15. Runs were repeated for 15 data points in total. The best result for whole array and test array for input combinations whose overall R values was over 0.75 are given in Tables 4.8 and 4.9.

Table 4.8 Best Results of ANN Modeling: Results for Reduced Whole Array

	Input Combination	Training Algorithm	Activation Function	REDUCED WHOLE ARRAY					
				R2	Correlation Coefficient	Max Error	Average Error	MAE	MSE
SURFACE	A1	RP	Logsig	N/A	N/A	N/A	N/A	N/A	N/A
	A2	CGP	Logsig	0,680	0,824	7,583	2,370	0,145	12,307
	A3	BFG	Tansig	0,627	0,792	7,409	2,041	0,131	8,917
	A4	LM	Logsig	0,840	0,917	5,504	1,214	0,080	3,807
	A5	BR	Logsig	0,725	0,851	7,005	2,098	0,135	8,298
	A6	GDM	Logsig	0,659	0,812	6,138	2,513	0,195	8,999
	A7	LM	Logsig	0,815	0,903	4,412	1,270	0,100	4,611
	A8	LM	Logsig	0,846	0,920	5,630	1,492	0,099	5,069
	A9	GDX	Logsig	0,600	0,775	5,480	2,545	0,203	9,669
	B4	RP	Logsig	0,640	0,800	7,953	1,904	0,129	8,523
	B5	BR	Logsig	0,772	0,879	10,156	2,693	0,174	13,537
	B6	CGB	Logsig	0,827	0,909	3,463	1,642	0,133	4,428
	B7	OSS	Logsig	0,584	0,764	5,838	2,775	0,215	10,363
	B8	CGP	Logsig	0,854	0,924	6,660	1,322	0,097	4,288
	B9	OSS	Logsig	0,863	0,929	2,842	1,636	0,120	3,249
AVERAGE		Training Algorithm	Activation Function						
				R2	Correlation Coefficient	Max Error	Average Error	MAE	MSE
	A3	CGB	Logsig	0,694	0,833	3,508	0,923	0,090	2,234
	A5	LM	Logsig	0,583	0,764	3,613	1,409	0,158	3,236
	A6	LM	Logsig	0,833	0,912	2,987	0,643	0,069	1,348
	A9	LM	Logsig	0,831	0,911	2,318	0,807	0,078	1,359
	B5	BR	Logsig	0,751	0,866	5,118	1,016	0,111	2,911
B6	SCG	Tansig	0,777	0,881	3,091	0,889	0,091	1,710	

Table 4.9 Best Results of ANN Modeling: Results for Reduced Test Array

		REDUCED TEST ARRAY							
	Input Combination	Training Algorithm	Activation Function	R ²	Correlation Coefficient	Max Error	Average Error	MAE	MSE
				SURFACE	A1	RP	Logsig	N/A	N/A
A2	CGP	Logsig	0,656		0,810	7,583	4,899	0,278	30,060
A3	BFG	Tansig	0,716		0,846	6,131	3,008	0,157	13,907
A4	LM	Logsig	0,862		0,928	5,504	2,186	0,115	8,931
A5	BR	Logsig	0,723		0,850	7,005	4,395	0,256	23,397
A6	GDM	Logsig	0,798		0,893	6,138	3,604	0,295	16,527
A7	LM	Logsig	0,761		0,873	4,412	3,321	0,261	13,860
A8	LM	Logsig	0,993		0,996	5,630	2,667	0,131	11,318
A9	GDX	Logsig	0,677		0,823	5,480	3,983	0,287	17,928
B4	RP	Logsig	0,781		0,884	5,614	2,896	0,196	13,545
B5	BR	Logsig	0,991		0,996	10,156	4,969	0,261	36,393
B6	CGB	Logsig	0,895		0,946	3,368	2,570	0,201	7,888
B7	OSS	Logsig	0,677		0,823	5,838	3,758	0,296	17,792
B8	CGP	Logsig	0,968		0,984	1,921	1,298	0,100	1,845
B9	OSS	Logsig	0,990		0,995	2,334	1,716	0,123	3,099
AVERAGE		Training Algorithm	Activation Function	R ²	Correlation Coefficient	Max Error	Average Error	MAE	MSE
	A3	CGB	Logsig	0,425	0,652	2,963	1,808	0,181	3,880
	A5	LM	Logsig	0,463	0,680	3,255	2,213	0,237	5,569
	A6	LM	Logsig	0,877	0,936	1,172	0,891	0,091	0,887
	A9	LM	Logsig	0,823	0,907	2,318	1,275	0,122	2,443
	B5	BR	Logsig	0,970	0,985	1,327	0,825	0,085	0,896
	B6	SCG	Tansig	0,706	0,840	2,630	1,559	0,162	2,855

As it was conducted for the non-reduced arrays the mapping procedure (calculation of F values) was also carried out for results of reduced input combinations. The followed procedure was exactly same with the procedure that was applied to the non-reduced data. (See Section 3.2.2.5) The results of mapped data and F values for test set and whole set are depicted in Tables 4.10 and 4.11 respectively. Moreover when the Tables 4.10 and 4.12 are examined, according to F values Table 4.# depicts the success ranking of input combinations.

Table 4.10 Mapped Results of ANN Modeling: Results for Reduced Whole Array

	Input Combination	Training Algorithm	Activation Function	MAPPED WHOLE ARRAY						F
				R2	Correlation Coefficient	Max Error	Average Error	MAE	MSE	
SURFACE	A1	RP	Logsig	N/A	N/A	N/A	N/A	N/A	N/A	-1,905
	A2	CGP	Logsig	0,344	0,364	0,648	0,740	0,481	0,880	-1,905
	A3	BFG	Tansig	0,154	0,170	0,624	0,529	0,378	0,551	-1,535
	A4	LM	Logsig	0,918	0,927	0,364	0,000	0,000	0,054	0,509
	A5	BR	Logsig	0,505	0,527	0,569	0,566	0,407	0,491	-1,099
	A6	GDM	Logsig	0,269	0,291	0,451	0,832	0,852	0,559	-1,550
	A7	LM	Logsig	0,828	0,842	0,215	0,036	0,148	0,132	0,459
	A8	LM	Logsig	0,939	0,945	0,381	0,178	0,141	0,177	0,209
	A9	GDX	Logsig	0,057	0,067	0,361	0,853	0,911	0,624	-1,771
	B4	RP	Logsig	0,201	0,218	0,699	0,442	0,363	0,513	-1,435
	B5	BR	Logsig	0,674	0,697	1,000	0,947	0,696	1,000	-2,250
	B6	CGB	Logsig	0,871	0,879	0,085	0,274	0,393	0,115	0,405
	B7	OSS	Logsig	0,000	0,000	0,410	1,000	1,000	0,691	-2,101
	B8	CGP	Logsig	0,968	0,970	0,522	0,069	0,126	0,101	0,278
	B9	OSS	Logsig	1,000	1,000	0,000	0,270	0,296	0,000	0,730
	AVERAGE		Training Algorithm	Activation Function	R2	Correlation Coefficient	Max Error	Average Error	MAE	MSE
A3		CGB	Logsig	0,444	0,466	0,425	0,366	0,236	0,469	-0,794
A5		LM	Logsig	0,000	0,000	0,462	1,000	1,000	1,000	-2,462
A6		LM	Logsig	1,000	1,000	0,239	0,000	0,000	0,000	0,761
A9		LM	Logsig	0,992	0,993	0,000	0,213	0,101	0,006	0,774
B5		BR	Logsig	0,672	0,689	1,000	0,487	0,472	0,828	-1,625
B6		SCG	Tansig	0,776	0,791	0,276	0,320	0,247	0,192	0,003

Table 4.11 Mapped Results of ANN Modeling: Results for Reduced Test Array

	Input Combination	Training Algorithm	Activation Function	Mapped Test ARRAY						F
				R2	Correlation Coefficient	Max Error	Average Error	MAE	MSE	
SURFACE	A1	RP	Logsig	N/A	N/A	N/A	N/A	N/A	N/A	
	A2	CGP	Logsig	0,000	0,000	0,688	0,981	0,911	0,817	-2,485
	A3	BFG	Tansig	0,178	0,194	0,511	0,466	0,292	0,349	-1,133
	A4	LM	Logsig	0,611	0,634	0,435	0,242	0,075	0,205	-0,248
	A5	BR	Logsig	0,199	0,215	0,617	0,844	0,798	0,624	-1,870
	A6	GDM	Logsig	0,421	0,446	0,512	0,628	0,997	0,425	-1,119
	A7	LM	Logsig	0,312	0,339	0,302	0,551	0,821	0,348	-0,863
	A8	LM	Logsig	1,000	1,000	0,450	0,373	0,160	0,274	-0,098
	A9	GDX	Logsig	0,062	0,070	0,432	0,731	0,956	0,466	-1,559
	B4	RP	Logsig	0,371	0,398	0,448	0,435	0,490	0,339	-0,824
	B5	BR	Logsig	0,994	1,000	1,000	1,000	0,822	1,000	-2,000
	B6	CGB	Logsig	0,709	0,731	0,176	0,347	0,517	0,175	0,034
	B7	OSS	Logsig	0,062	0,070	0,476	0,670	1,000	0,462	-1,538
	B8	CGP	Logsig	0,926	0,935	0,000	0,000	0,000	0,000	0,935
	B9	OSS	Logsig	0,991	0,995	0,050	0,114	0,117	0,036	0,794
	AVERAGE		Training Algorithm	Activation Function	R2	Correlation Coefficient	Max Error	Average Error	MAE	MSE
A3		CGB	Logsig	0,000	0,000	0,860	0,708	0,632	0,639	-2,207
A5		LM	Logsig	0,070	0,084	1,000	1,000	1,000	1,000	-2,916
A6		LM	Logsig	0,829	0,853	0,000	0,047	0,038	0,000	0,806
A9		LM	Logsig	0,730	0,766	0,550	0,324	0,245	0,332	-0,441
B5		BR	Logsig	1,000	1,000	0,074	0,000	0,000	0,002	0,924
B6	SCG	Tansig	0,516	0,565	0,700	0,529	0,510	0,420	-1,085	

Table 4.12 Success Order of Reduced Input Combinations Considering (F) Values

	AVERAGE		SURFACE	
Success Order	1st	2nd	1st	2nd
TEST	B5	A6	B8	B9
WHOLE	A9	A6	B9	A4

After conducting the required analysis the reduced data sets gave successful results when the results of Tables 4.8 and 4.9 are examined. Especially when the results of test set is assessed the correlation coefficients are in the order of 0.8 and 0.9, so this situation shows consistency with the results of non-reduced data arrays. It is known that small amount of data creates challenge during training of a neural net. However obtaining such high results for reduced data arrays ,as it was obtained in non-reduced ones, shows the consistency of ANN system. Since, in this case the data was reduced but the chronological data behavior was not altered and the sets of training, validation and test sets were specified in the same quarterly manner as it was done in non-reduced case.

From the comparison perspective between input combinations considering F values calculated, the results of test set gave very reasonable results that successful input combinations of surface data (B8 and B9) includes meteorological data and successful input combinations of average data (B5 and A6 is not. This result again shows that meteorological factor are dominantly effective on surface DO but not so directly effective on interior dynamics of lake from DO perspective.

In order to support the evaluation expressed in the previous paragraphs the plots of best input combinations are given between Figures of 4.9 and 4.16

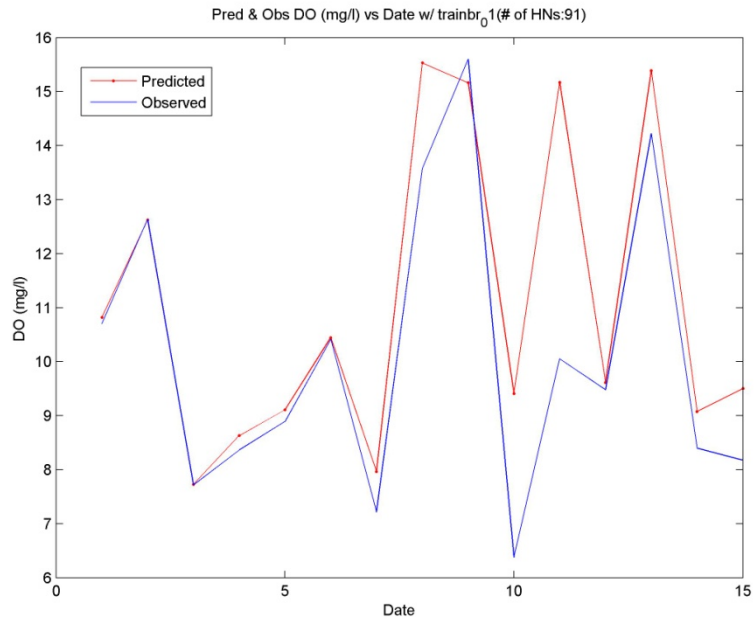


Figure 4.9 “Predicted DO vs Real DO” Graph of **Reduced Average B5** Data Combination (*Training Algorithm: trainrp; Actv. Func.: logsig + logsig; No. of HD: 91*)

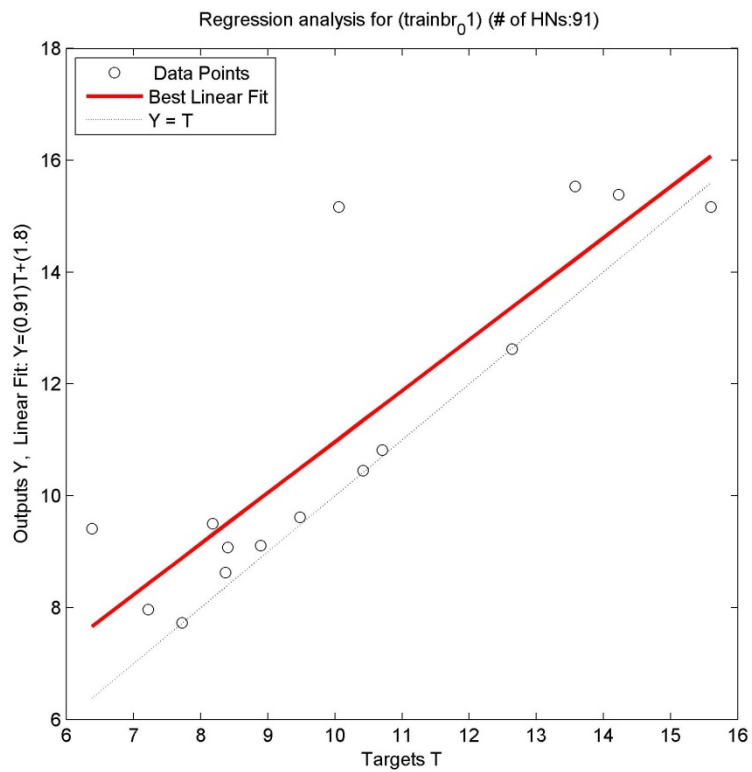


Figure 4.10 “Linear Regression” Graph of **Reduced Average B5** Data Combination (*Training Algorithm: trainrp; Actv. Func.: logsig + logsig; No. of HD: 91*)

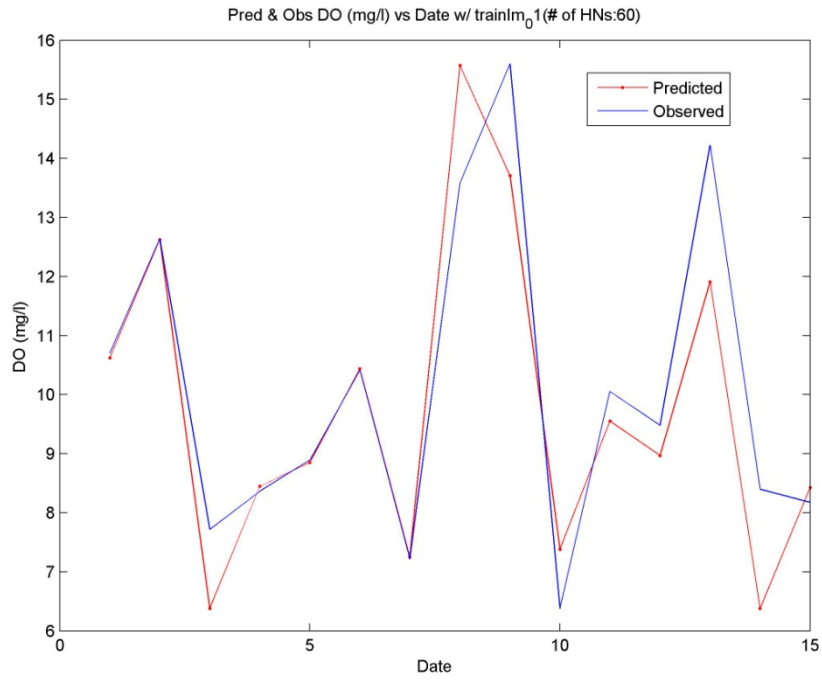


Figure 4.11 “Predicted DO vs Real DO” Graph of **Reduced Average A9** Data Combination (Training Algorithm: *trainlm*; Actv. Func.: *logsig + logsig*; No. of HD: 60)

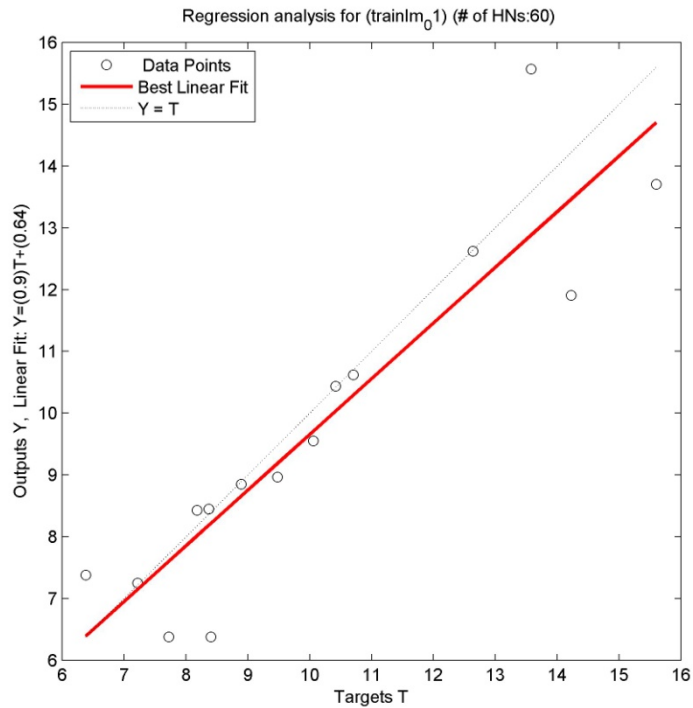


Figure 4.12 “Linear Regression” Graph of **Reduced Average A9** Data Combination (Training Algorithm: *trainlm*; Actv. Func.: *logsig + logsig*; No. of HD: 60)

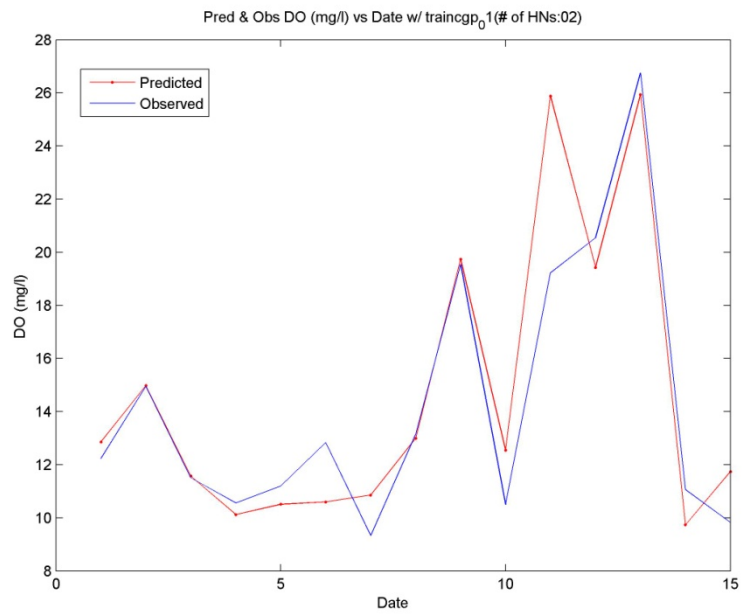


Figure 4.13 “Predicted DO vs Real DO” Graph of **Reduced Surface B8** Data Combination (Training Algorithm: *traincgp*; Actv. Func.: *logsig + logsig*; No. of HD: 02)

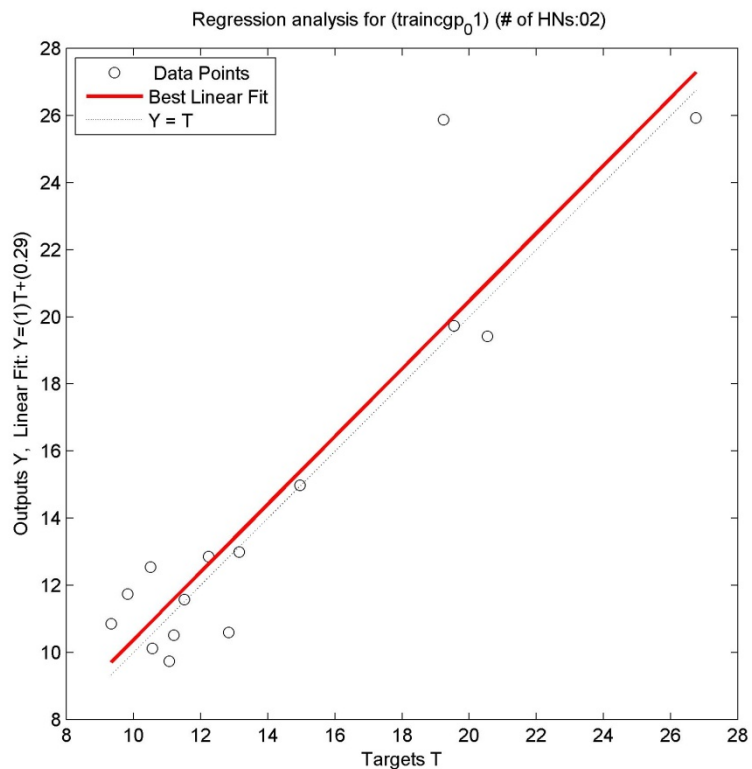


Figure 4.14 “Linear Regression” Graph of **Reduced Surface B8** Data Combination (Training Algorithm: *traincgp*; Actv. Func.: *logsig + logsig*; No. of HD: 02)

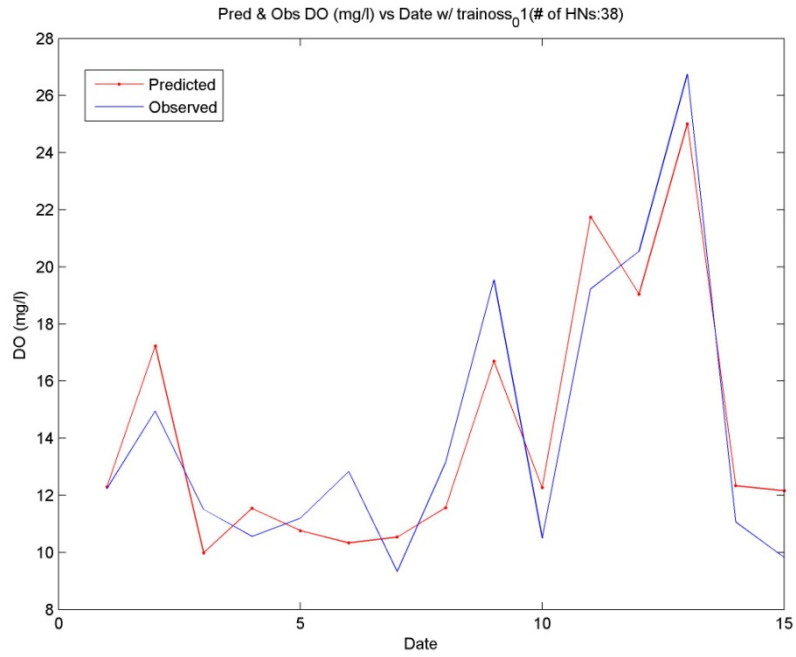


Figure 4.15 “Predicted DO vs Real DO” Graph of **Reduced Surface B9** Data Combination (Training Algorithm: trainoss; Actv. Func.: logsig + logsig; No. of HD: 38)

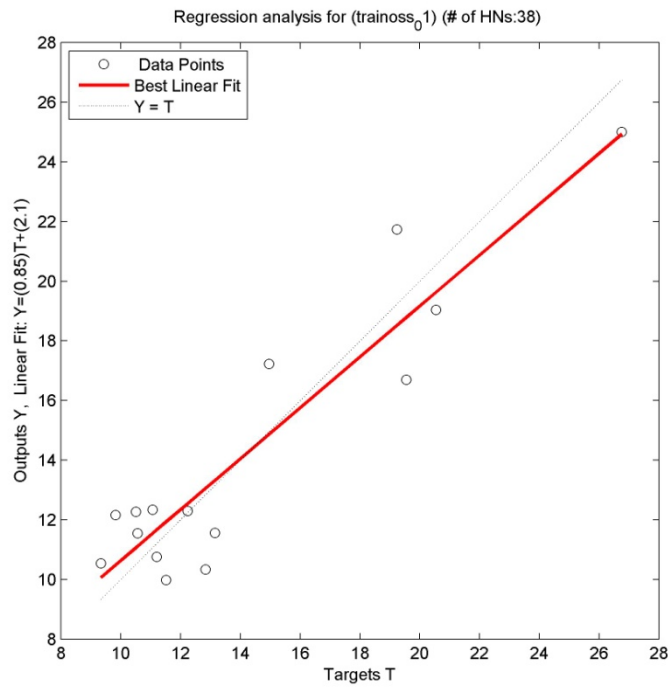


Figure 4.16 “Linear Regression” Graph of **Reduced Surface B9** Data Combination (Training Algorithm: trainoss; Actv. Func.: logsig + logsig; No. of HD: 38)

4.1.2 Sensitivity Analysis

After these general considerations a detailed consideration on individual parameters was made. By this way the parameters which are controlling the eutrophication period tried to be determined with Sensitivity Analysis. In sensitivity analysis average A6 input combination was used since that it was one of the most successful one in ANN Modeling.

A simple method for sensitivity analysis was chosen. The important parameters in A6 data combination were increased by 30% and 50% and they were declined by %30 and %50. After that the newly established data structures were modeled with exactly the same structure with the original data set. The determination on individual parameters was carried out by comparing the overall averages of the DO data. The chronological average of observed DO data is 10.122 mg/l. The averages and deviations are depicted in Table 4.13.

Table 4.13 Sensitivity Results of ANN Model for A6 Data Combination (average data)

Parameter	Change	Average DO	Percent Deviation
CHL-A	+30%	N/A	N/A
	-30%	10.322	1.97%
	+50%	9.832	-2.88%
	-50%	9.832	-2.88%
NO3	+30%	10.245	1.20%
	-30%	10.024	-0.98%
	+50%	10.058	-0.64%
	-50%	11.166	10.31%
pH	+30%	9.615	-5.01%
	-30%	9.498	-6.17%
	+50%	10.409	2.83%
	-50%	10.983	8.50%
PO4	+30%	10.554	4.26%
	-30%	9.835	-2.84%
	+50%	10.605	4.77%
	-50%	10.423	2.96%
Temperature	+30%	9.502	-6.13%
	-30%	10.234	1.10%
	+50%	9.651	-4.66%
	-50%	10.059	-0.63%
TKN	+30%	10.007	-1.14%
	-30%	10.107	-0.16%
	+50%	9.676	-4.41%
	-50%	10.578	4.49%
Total P	+30%	10.681	5.51%
	-30%	9.754	-3.64%
	+50%	10.506	3.79%
	-50%	10.058	-0.64%

When we look at the Table 4.5 the following interpretations are derived. Generally when we look at the “Percent Deviation” the rows that have 3% deviation was depicted as bold and the interpretations were made by using them. Because in such a complex system deviation below 3% might cause wrongly directed interpretations.

First of all we have + 10,31% deviation when the NO_3 is decreased as 50%. Similarly as the TKN was declined as 50%, the average DO concentration of lake had tendency of 4,49 percent increase. In brief the DO Concentration is inversely proportional with Nitrogen containing species. Normally this situation is an unexpected occurrence that shows low sensitivity of model on Nitrogen containing parameters.

On the other hand, the phosphorus included species had a direct proportionality with DO. For instance when PO_4 is increased 30% and 50%, the DO was also increased 4.26% and 4.77% respectively. Similarly when Total P values were increased by 30% and 50%, average DO concentrations were also increased as 5.51% and 3.79%. This situation is an expected event that the model that was developed is sensitive to phosphorus including parameters.

Apart from nutrients the alterations on DO with temperature is also reasonable. According to results DO is decreasing by 6.13% when temp is increased as by 30% and similarly average DO is decreasing by 4.66% as temperature is increased as 50%. By considering the direct chemical relationship with DO and temp in lake ecosystem and knowing their inverse proportional character, this result shows how the ANN model sensitive to temp and phosphorus containing structures.

For other parameter no concrete interpretation could be produced. To summarize when phosphorus containing parameters and temperature parameter are taken into account; their change and effect of this change on DO the eutrophication process is limited to phosphorus concentration and effect of

temperature is undeniable as normally. Also lack of other interpretations on other parameters shows the inadequacy of the model on its realistic ability.

4.2 Results of ANFIS Modeling

For modeling with ANFIS input combinations of A3, A4, A6, A9 and B3, B4, B6, B9 were selected. The aim behind this selection raised from trail of input combinations that have different characteristics. As it can be inferred A9 and B9 are the input combinations that have meteorological parameters. On the other hand A3 and A4 has nutrients and temperature. Besides A6 has Chlorophyll-a with respect to A3 and A4. Theoretically effect of Chlorophyll-a on DO is undeniable and by selection of A6-B6 with A3-B3 and A4-B4 was thought to determine the effect of Chlorophyll-a on DO with nutrients and temperature. Also selection of A4-B4 with respect to A3-B3 was conducted in order to evaluate the effect of pH on DO. Finally A9-B9 were selected in order to assess the effect of weather conditions on DO.

60 data in the input data arrays were used in ANFIS modeling. The data was divided in to three sections (training+validation+test) in the same manner similar to ANNs (30+15+15). This consistent data separation with ANN modeling provided a base for comparison of the results obtained with ANN and ANFIS models. MATLAB R2007b Fuzzy Logic Toolbox ANFIS GUI was the software that executed the model. As stated earlier, the execution of the ANFIS model was not as easy as ANN Modeling due to the lack of a script such as the one used for ANN model development. The script used for ANN model development (Moral et al., 2008) provided an automatic screening of ANNs with different structures efficiently. However, ANFIS GUI of MATLAB Fuzzy Logic Toolbox was not designed to create models in an automated way, rather the trial-and-error approach was used to find the best model. Therefore, each time the ANFIS parameters were changed manually, one-by-one. Besides the choices of ANFIS structure types, membership function types, training algorithm alternatives of ANFIS GUI were limited compared to ANN.

The modeling methodology of ANFIS has been depicted in Chapter 3. (See Section 3.2.3). The results of ANFIS Modeling are given in Tables 4.10 to 4.10. The normalization method that was applied to ANN Modeling results for comparison between input combinations was also applied to ANFIS Model in same manner described in “Section 4.1” and the results are depicted in Table 4.14. to 4.18.

Table 4.14 Best Results of ANFIS Modeling Test Array

TEST ARRAY								
Input Combination	Membership Function	# of MF In One Cluster	Correlation Coefficient	Max Error	Average Error	R2	MAE	MSE
A3	GBELL	30	0.252	3.596	1.248	0.064	0.256	13.028
A4	GAUSS	30	-0.206	5.714	3.612	0.042	0.274	25.634
A6	GAUSS	27	-0.500	4.873	1.637	0.250	0.333	24.848
A9	DSIG	28	0.129	13.813	1.551	0.017	0.378	26.246
B3	GAUSS	30	-0.537	2.754	1.069	0.288	0.348	20.567
B4	GAUSS	30	0.537	4.283	1.860	0.288	0.208	10.354
B6	GBELL	23	0.024	4.690	1.179	0.001	0.306	16.228
B9	DSIG	29	0.225	2.922	0.365	0.051	0.312	13.722

Table 4.15 Best Results of ANFIS Modeling Validation Array

VALIDATION ARRAY								
Input Combination	Membership Function	# of MF In One Cluster	Correlation Coefficient	Max Error	Average Error	R2	MAE	MSE
A3	GBELL	30	0.734	5.901	3.737	0.538	0.260	8.603
A4	GAUSS	30	0.764	6.708	4.067	0.584	0.257	8.367
A6	GAUSS	27	0.077	5.454	2.517	0.006	0.385	21.053
A9	DSIG	28	0.177	5.415	2.300	0.031	0.371	18.648
B3	GAUSS	30	0.632	3.402	1.638	0.399	0.261	8.723
B4	GAUSS	30	0.381	2.922	0.942	0.145	0.327	12.301
B6	GBELL	23	0.228	5.447	2.308	0.052	0.372	17.976
B9	DSIG	29	0.018	5.417	2.613	0.000	0.405	22.690

Table 4.16 Best Results of ANFIS Modeling: Whole Array

WHOLE ARRAY								
Input Combination	Membership Function	# of MF In One Cluster	Correlation Coefficient	Max Error	Average Error	R2	MAE	MSE
A3	GBELL	30	0.690	5.901	2.416	0.475	0.129	5.408
A4	GAUSS	30	0.581	6.708	3.090	0.338	0.133	8.500
A6	GAUSS	27	0.250	5.454	2.209	0.062	0.180	11.475
A9	DSIG	28	0.370	13.813	2.132	0.137	0.188	11.223
B3	GAUSS	30	0.533	4.809	1.846	0.284	0.152	7.323
B4	GAUSS	30	0.652	4.810	1.870	0.426	0.134	5.664
B6	GBELL	23	0.450	5.447	2.042	0.203	0.170	8.551
B9	DSIG	29	0.395	5.417	1.914	0.156	0.179	9.103

Table 4.17 Mapped Results of ANFIS Modeling: Test Array

PERFORMANCE PARAMETERS (MAPPED VERSION)									
			TEST ARRAY						
Input Combination	Membership Function	# of MF In One Cluster	Correlation Coefficient	Max Error	Average Error	R2	MAE	MSE	F
A3	GBELL	30	0.219	0.076	0.272	0.219	0.284	0.168	-0,297
A4	GAUSS	30	0.145	0.268	1.000	0.145	0.387	0.62	-2,084
A6	GAUSS	27	0.867	0.192	0.392	0.867	0.733	0.912	-0,629
A9	DSIG	28	0.056	1.000	0.365	0.056	1.000	1.000	-2,309
B3	GAUSS	30	1.000	0.000	0.217	1.000	0.824	0.643	0,141
B4	GAUSS	30	1.000	0.138	0.461	1.000	0.000	0.000	0,401
B6	GBELL	23	0.000	0.175	0.251	0.000	0.576	0.370	-0,795
B9	DSIG	29	0.174	0.015	0.000	0.174	0.611	0.212	-0,053

Table 4.18 Mapped Results of ANFIS Modeling: Whole Array

PERFORMANCE PARAMETERS (NORMALIZED VERSION)									
Input Combination	Membership Function	# of MF In One Cluster	VALIDATION ARRAY						F
			Correlation Coefficient	Max Error	Average Error	R2	MAE	MSE	
A3	GBELL	30	1.000	0.121	0.458	1.000	0.000	0.000	0,421
A4	GAUSS	30	0.754	0.211	1.000	0.667	0.065	0.510	-0,967
A6	GAUSS	27	0.000	0.072	0.292	0.000	0.866	1.000	-1,363
A9	DSIG	28	0.272	1.000	0.230	0.180	1.000	0.959	-1,916
B3	GAUSS	30	0.644	0.000	0.000	0.537	0.400	0.316	0,328
B4	GAUSS	30	0.915	0.000	0.019	0.879	0.082	0.042	0,854
B6	GBELL	23	0.456	0.071	0.157	0.340	0.695	0.518	-0,290
B9	DSIG	29	0.330	0.068	0.054	0.227	0.859	0.609	-0,401

After calculation of (F) values and comparison of them for test, validation and whole arrays gave the results that are depicted in Table 4.19:

Table 4.19 Success Order of Input Combinations Considering (F) Values

Success Order	AVERAGE		
	1st	2nd	3rd
TEST	B4	B3	A6
WHOLE	B4	B3	B9

As it can be inferred from the performance parameters that depicted in Tables 4.14 to 4.16, the performance of ANFIS was so low in terms of prediction performance of DO compared to ANNs. Within the Core A group input sets, A3 and A4 resulted in R values of 0.79 and 0.64, respectively. R for B3 and B4 were 0.58 and 0.71, respectively. The reasons behind pure DO predictions with ANFIS may be due to software limitations of ANFIS GUI, lack of a script that may constrain the efforts to develop a successful model, and user errors due to low degree of user-friendliness of ANFIS GUI. Figures 4.17 to 4.31 depict the linear regression and time-series for DO profiles obtained for ANFIS models.

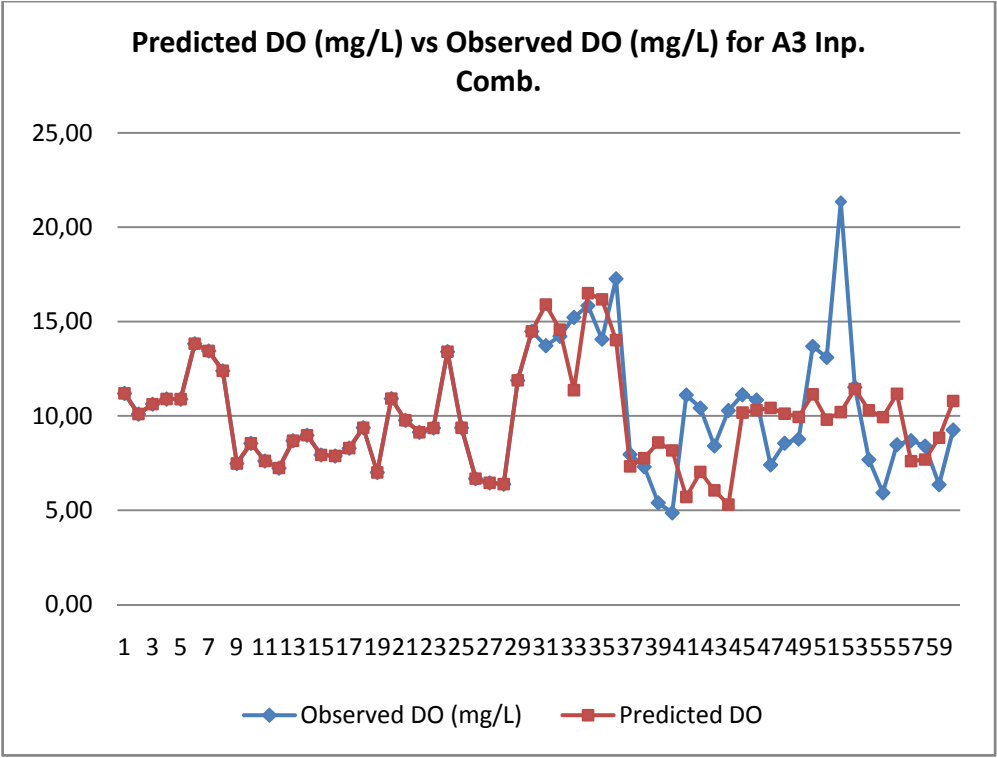


Figure 4.17 “Predicted DO vs. Real DO” Graph of **Average A3** Data Combination

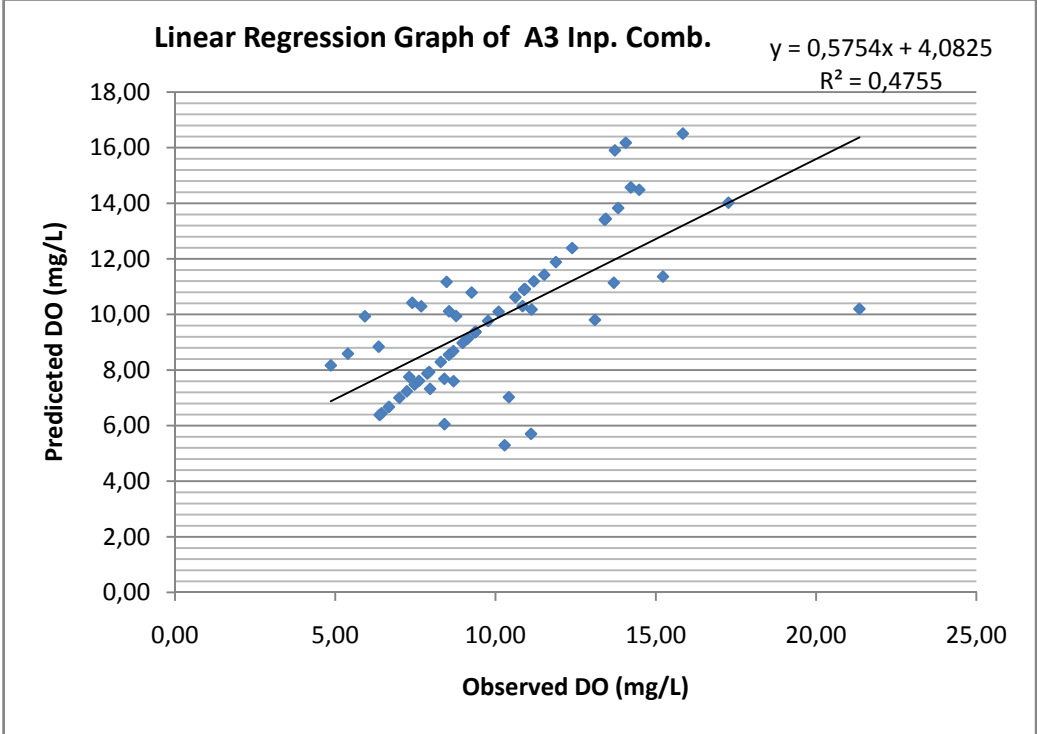


Figure 4.18 “Linear Regression” Graph of **Average A3** Data Combination

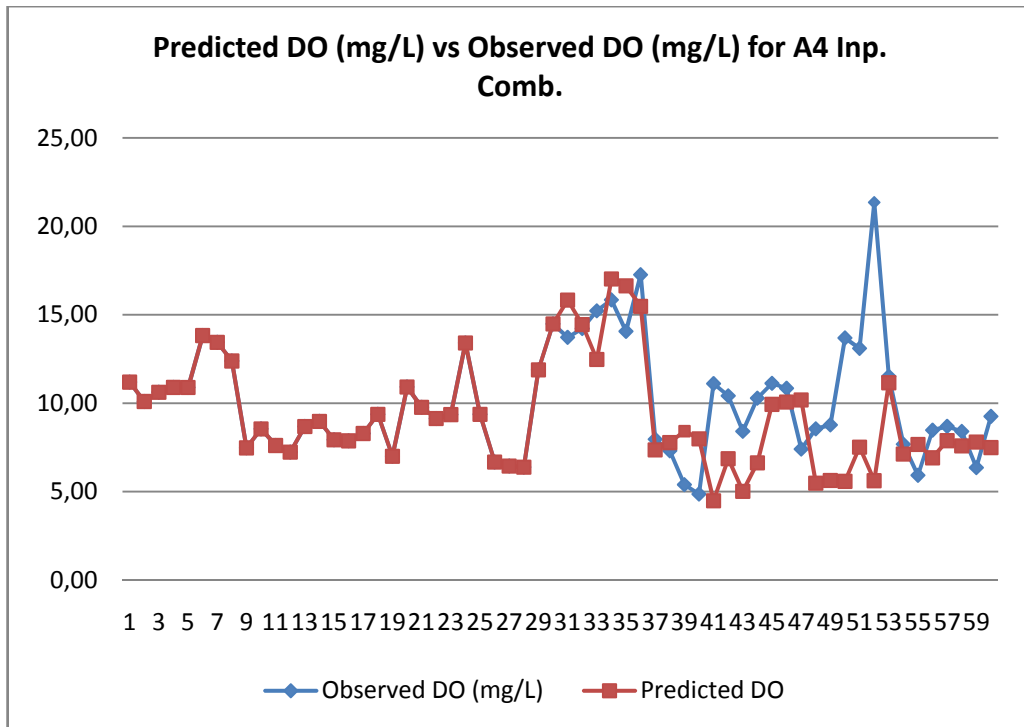


Figure 4.19 "Predicted DO vs. Real DO" Graph of **Average A4** Data Combination

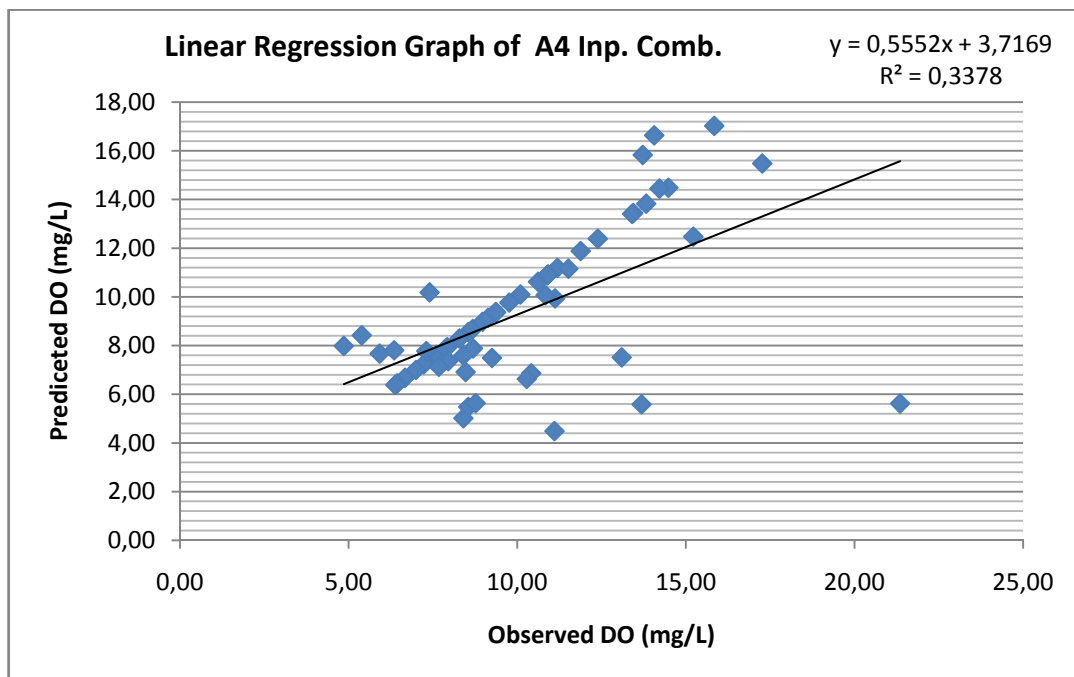


Figure 4.20 "Linear Regression" Graph of **Average A4** Data Combination

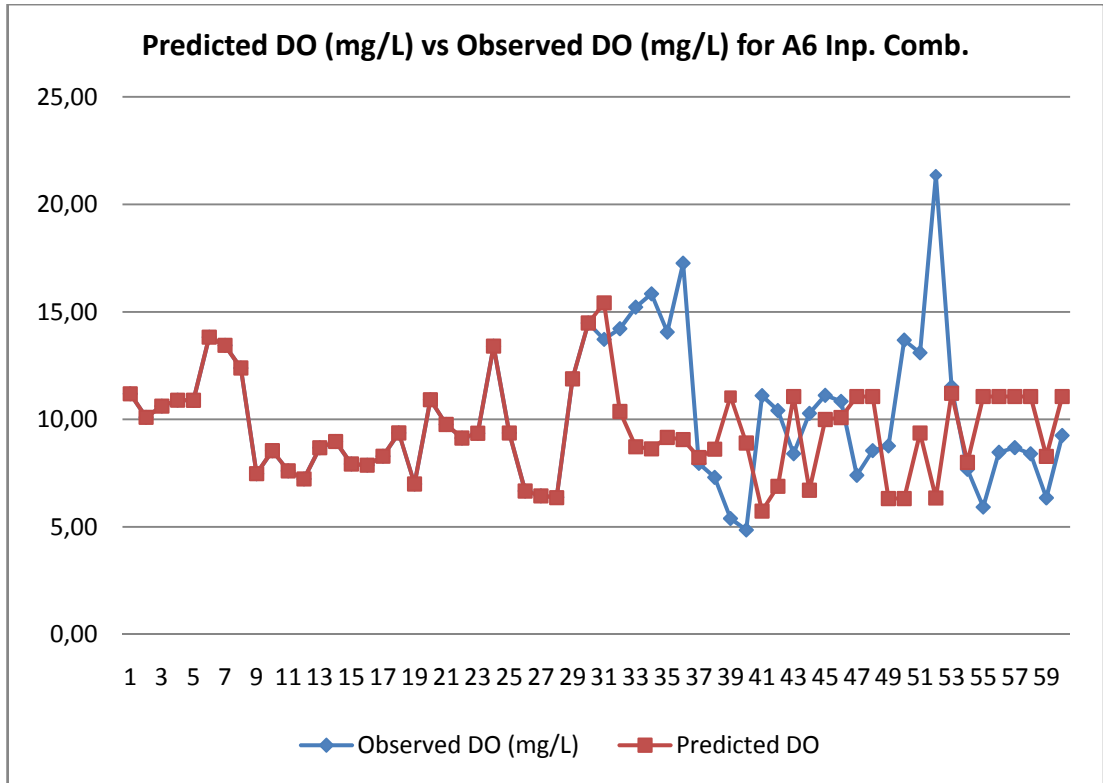


Figure 4.21 “Predicted DO vs. Real DO” Graph of **Average A6** Data Combination

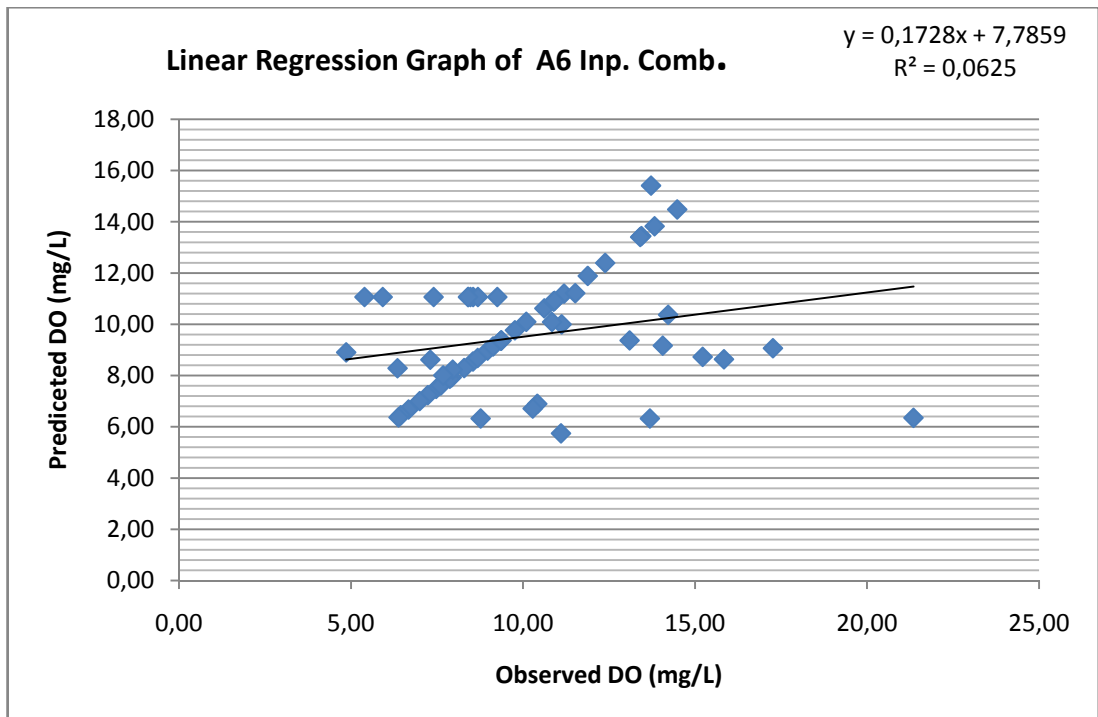


Figure 4.22 “Linear Regression” Graph of **Average A6** Data Combination

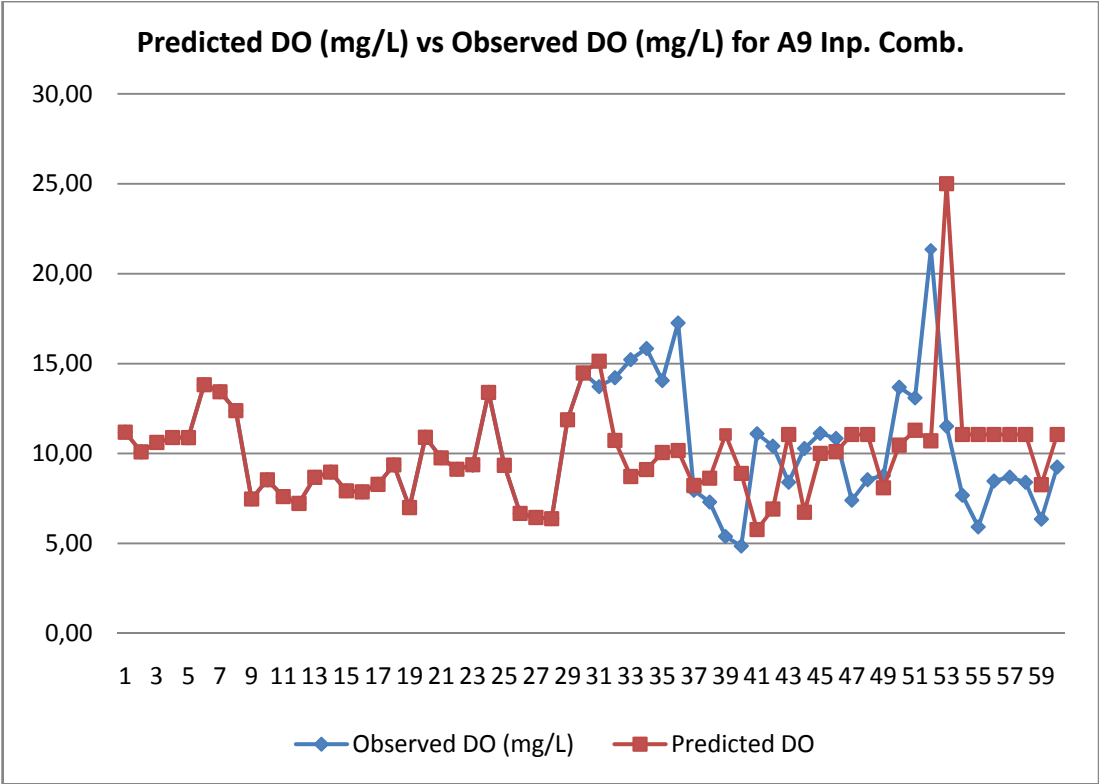


Figure 4.23 “Predicted DO vs. Real DO” Graph of **Average A9** Data Combination

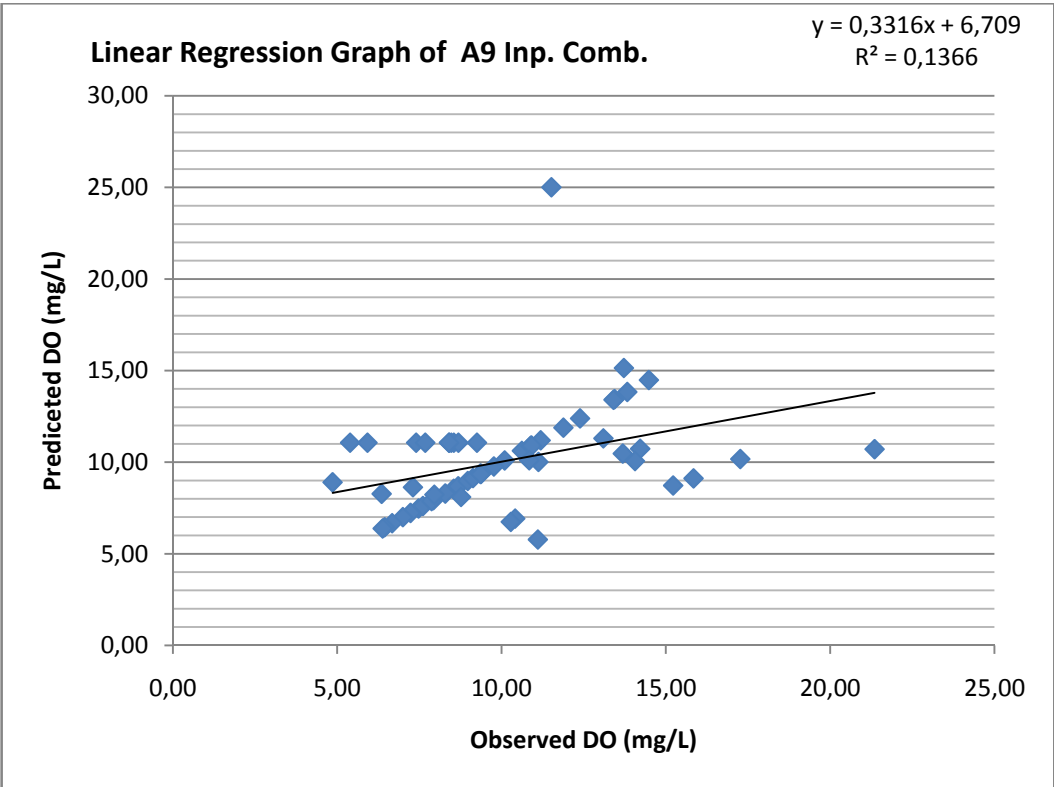


Figure 4.24 “Linear Regression” Graph of **Average A9** Data Combination

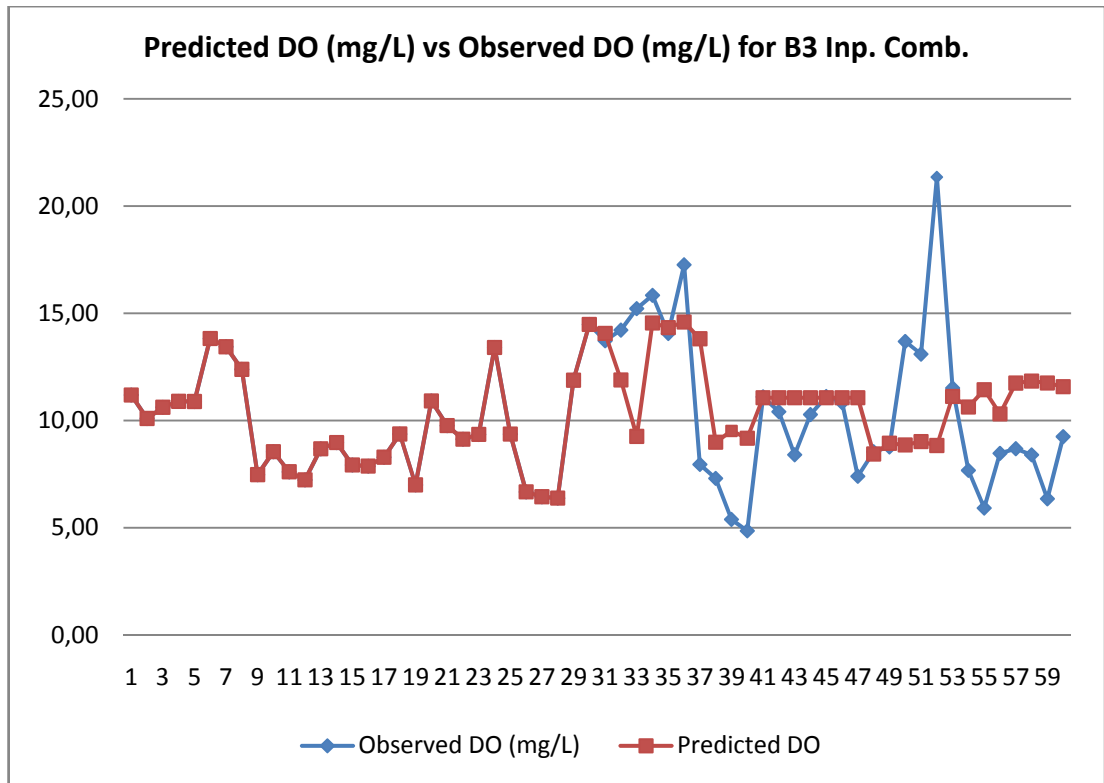


Figure 4.25 "Predicted DO vs. Real DO" Graph of **Average B3** Data Combination

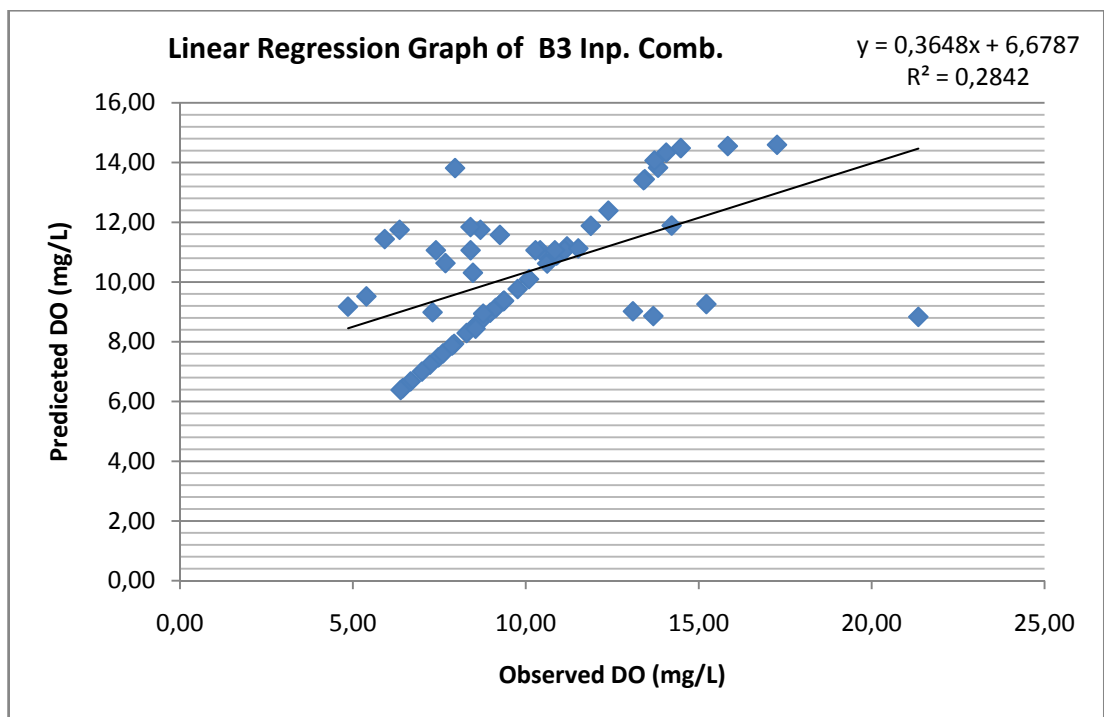


Figure 4.26 "Linear Regression" Graph of **Average B3** Data Combination

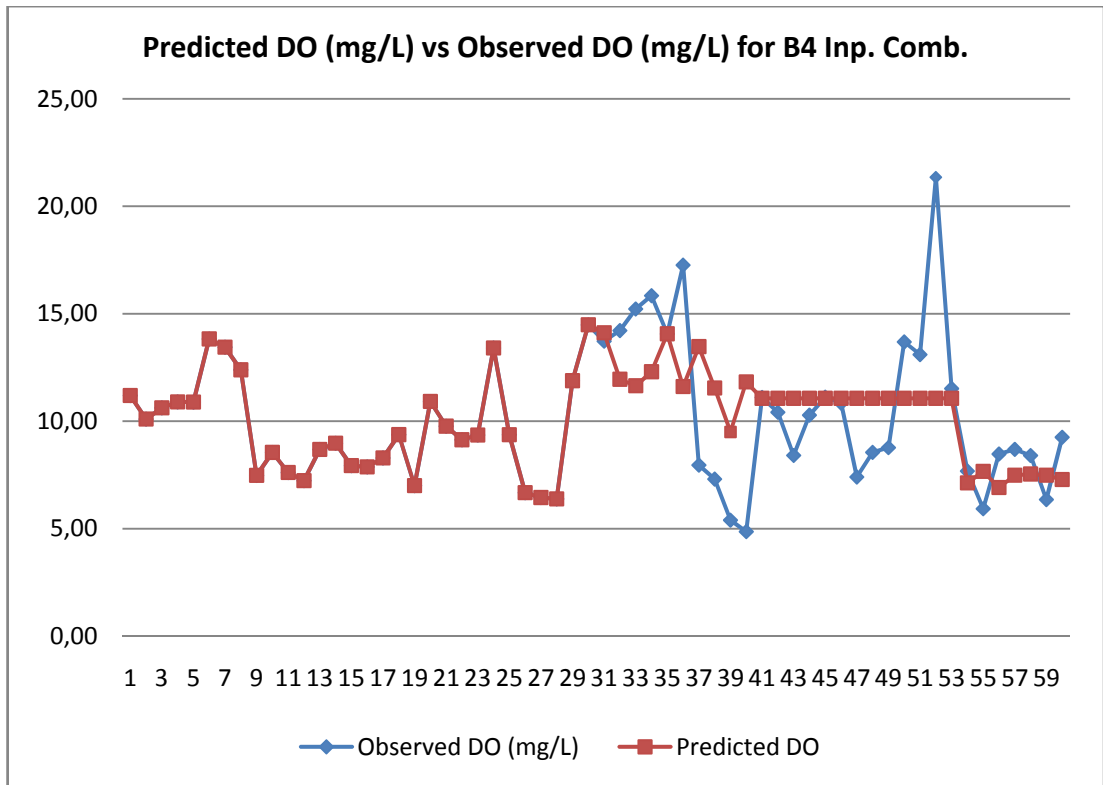


Figure 4.27 "Predicted DO vs. Real DO" Graph of **Average B4** Data Combination

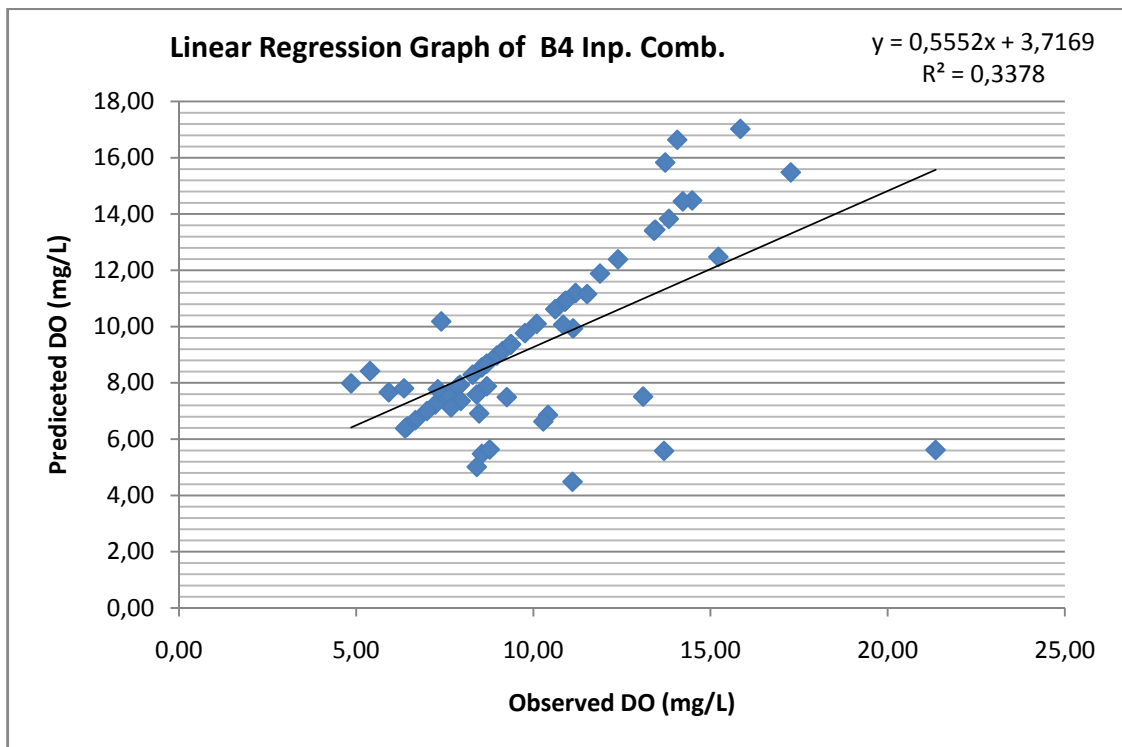


Figure 4.28 "Linear Regression" Graph of **Average B4** Data Combination

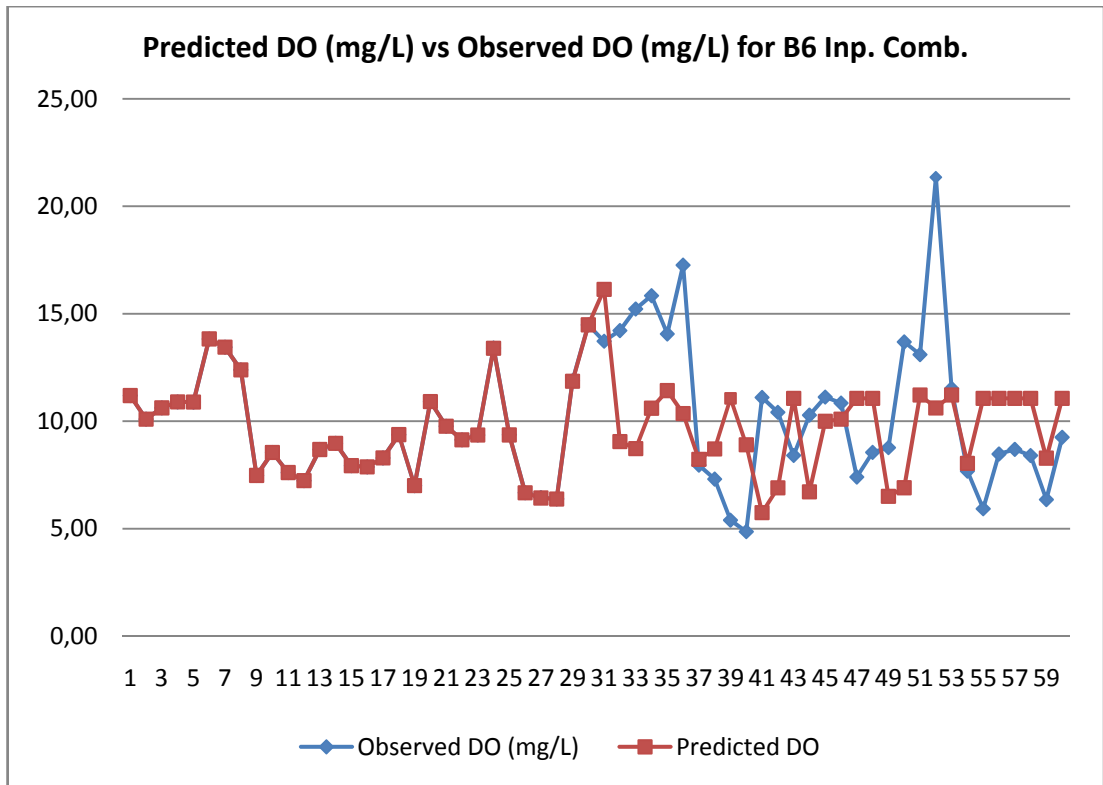


Figure 4.29 “Predicted DO vs. Real DO” Graph of **Average B6** Data Combination

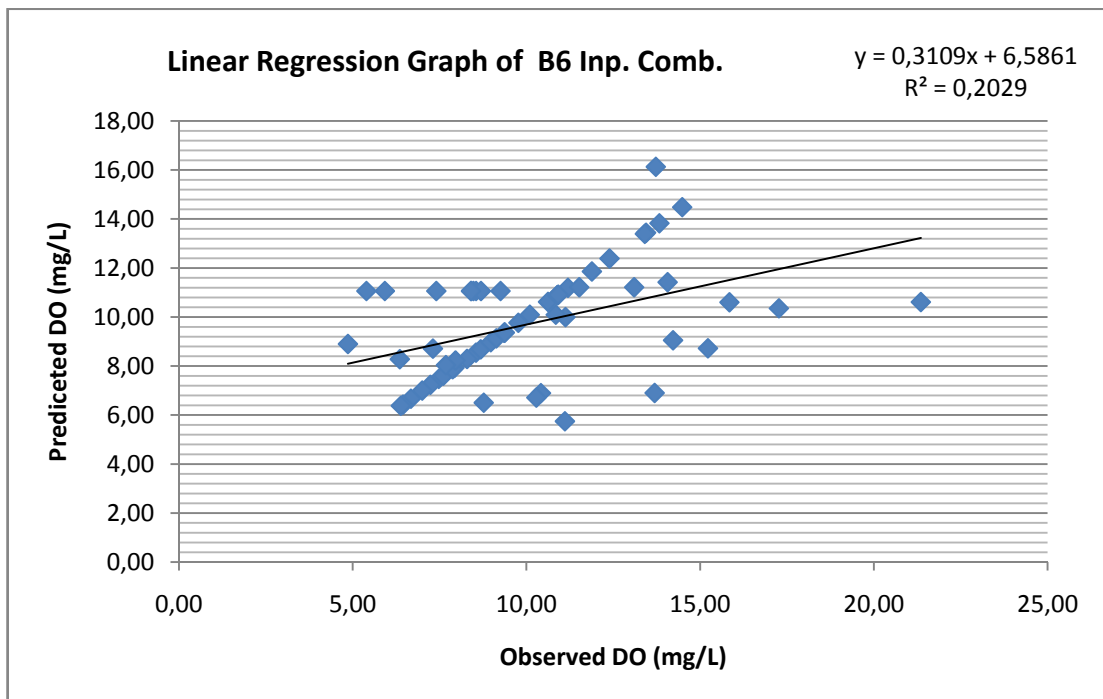


Figure 4.30 “Linear Regression” Graph of **Average B6** Data Combination

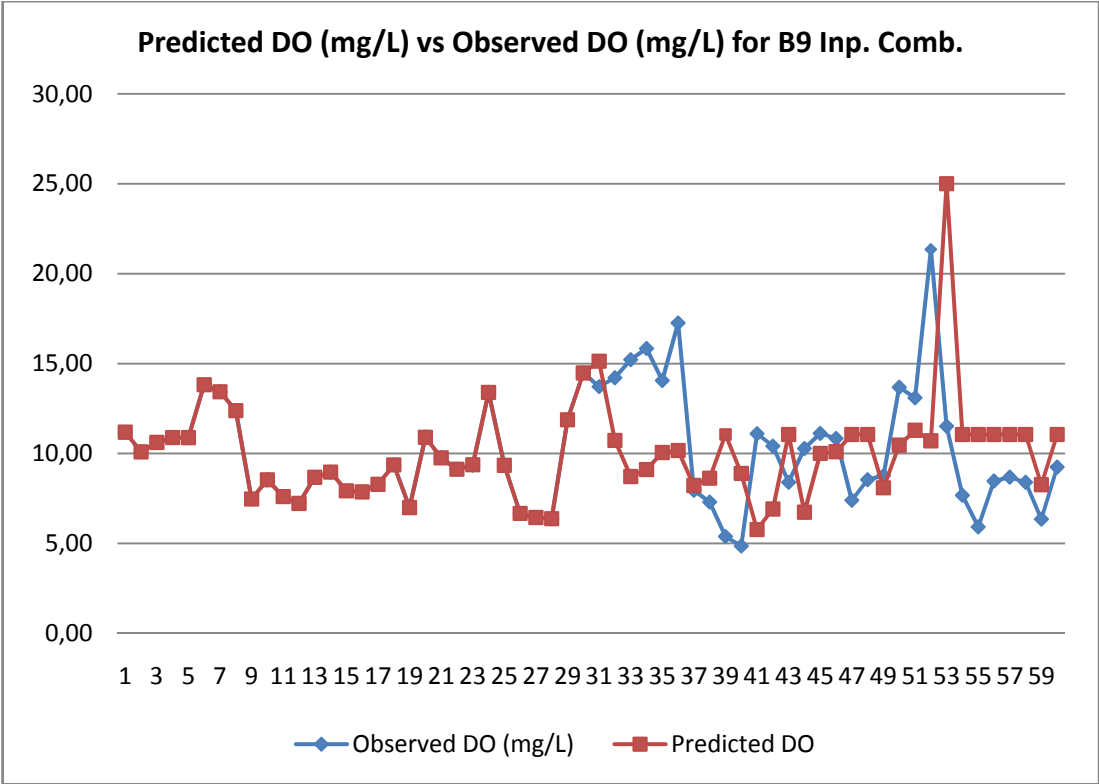


Figure 4.31 "Predicted DO vs. Real DO" Graph of **Average B6** Data Combination

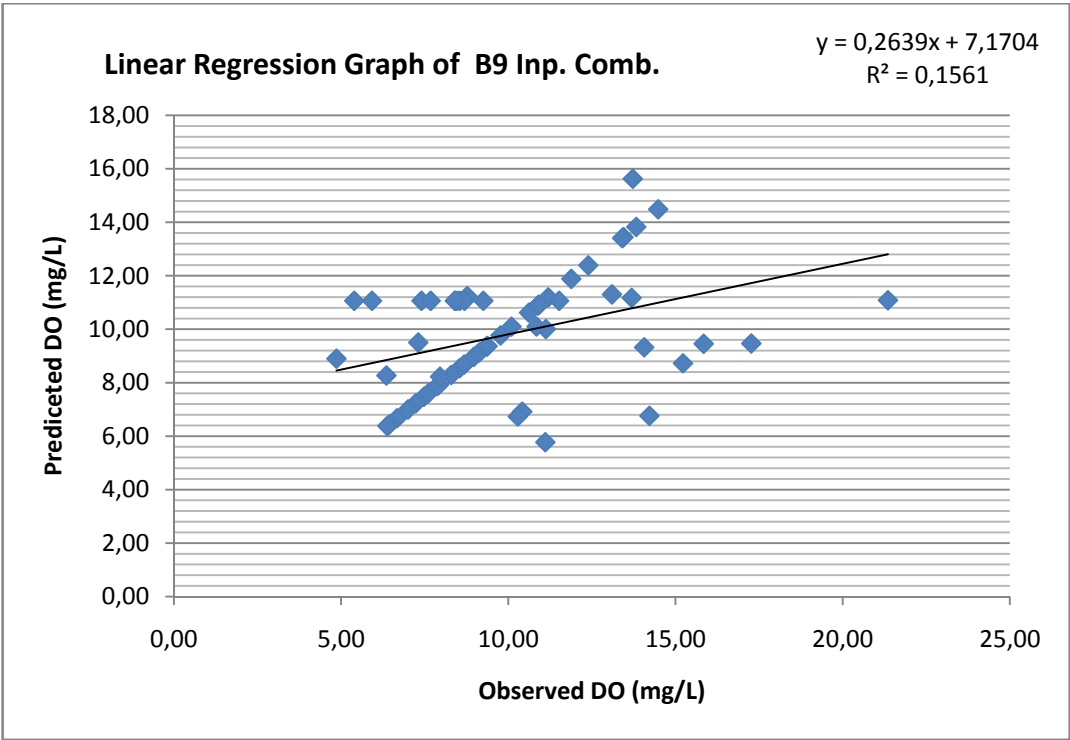


Figure 4.32 "Linear Regression" Graph of **Average B6** Data Combination

As it could be informed from the graphs of “Predicted DO vs. Observed DO” the prediction of training set (first 30) is fully overlapping with observed data. This situation is an expected case but this high performance overlapping does not say something about the overall performance of the model. When we look at the performances of validation and test set visually from graphs, so many times the real trend of DO could not be caught by prediction data except A9 and B9 input combination. Although the correlation coefficients of test data or whole data is so low the summer month super saturation DO concentration was predicted with a little latency. This particular result shows that ANFIS or ANFIS similar modeling tools that are capable of combining neural nets and fuzzy logic could give very high prediction performance. But a further study is needed on development of a script that automatically carries out the epochs. This automation enables trying high number of epochs in a short time with a low degree supervision.

From the (F) values perspective (normalization of performance parameter for comparative purpose); B3, B4 and A3 data become successful in terms of prediction performance with respect to other input combinations. From this point although the general prediction performance of ANFIS is so low, effect of nutrients, pH and temperature have a strong effect on DO concentrations in lake.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The results of this study indicated that it is possible to predict the DO levels in Lake Eymir using limited number of input parameters. ANN models were successful in capturing the extreme DO super saturation values observed in summer months. The typical deterministic models used for DO predictions may not handle these extreme situations or may require extensive calibration and verification work. However, it is possible to develop a DO model specific to the conditions in Lake Eymir. Probably, this is one of the biggest advantages of ANNs over traditional modeling techniques. R values obtained for DO predictions in this study are comparable with the values stated in literature for studies performed to predict different parameters in a lake.

The results of ANFIS DO models were not as successful as expected. ANN appeared to be more successful than ANFIS when correlation coefficients and other error showing parameters were considered. Unfortunately the correlation coefficients of for the whole array and correlations for the test array were very poor. These results may be due to insufficient data and use of the same membership function for all input parameters.

5.2 Recommendations for future study

The ANN models generated in this study can be used to develop an artificial measurement instrument. For example, the ANN model can be integrated to the

code of an in-situ measurement device to predict DO based on other parameters. This may especially be useful for prediction of parameters other than DO that is hard to measure.

There are parameters that have a role on the DO balance of a lake. Therefore, in future studies, inclusion of more input parameters may result in better representation of the system in the lake.

The results obtained in this study for the ANFIS models were not conclusive. DO modeling using ANFIS should be studied further to make a solid comparison between the outcomes of the ANN and ANFIS models. In addition, similar to the one used for ANN model development, an automatic architecture building script for ANFIS can be written to test more ANFIS parameter combinations rather than using the trial-and-error approach.

REFERENCES

Altınbilek D. (1995). "Gölbaşı Mogan Eymir Gölleri İçin Su Kaynakları Ve Çevre Yönetimi Plan Projesi." Ankara Büyükşehir Belediye Başkanlığı - Ankara Su ve Kanalizasyon İdaresi Genel Müdürlüğü (ASKİ).

Amari S., Murata N.K.R., Finke M. and Yang H.H. (1997). Asymptotic statistical theory of overtraining and cross-validation. IEEE Transactions on Neural Networks, 8(5), 985-986.

Anderson J. A. (1972). A Simple Neural Network Generating an Interactive Memory. Mathematical Biosciences, 14, 197-220. Reprinted in Anderson and Roenfeld (1988), pp. 181-192.

Anderson J. A. (1968). A Memory Storage Model Utilizing Spatial Correlation Functions. Kybernetics, (5), 113-119. Reprinted in Anderson, Pellionisz & Rosenfeld (1990), pp. 79-86.

Ates, N., "Disinfection By-Products Formation In Low Bromide and Low SUVA Waters", Ph.D. Thesis, Co-Supervised By Dr. Mehmet Kitis, METU, 2004

Beale R. and Jackson T. (1988). "Neural Computing: An Introduction", Department of Computer Science, University of York.

Carlson, R.E. (1977). A trophic state index for lakes. Limnology and Oceanography. 22:361-369.

Carlson, R.E. and J. Simpson. (1996). A Coordinator's Guide to Volunteer Lake Monitoring Methods. North American Lake Management Society. 96 pp.

Chau K.W. (2004). Intelligent Manipulation of Calibration Parameters in Numerical Modeling. *Advances in Environmental Research*, 8(3-4), 467-476.

Chapra S. C. "Surface Water Quality Modeling." McGraw Hill Series in Water Resources and Environmental Engineering, McGraw Hill, New York, 1997.

Czogala E. and Leski J. "Fuzzy and Neuro-Fuzzy Intelligent Systems", Physica-Verlag Heidelberg, New York, 2000.

Elhatip H. and Kömür M. A. (2008). Evaluation of water quality parameters for the Mamasin Dam in Aksaray City in the Central Anatolian Part of Turkey by means of artificial neural Networks. *Environ. Geol.*, 53, 1157-1164.

Elahdab T., "Investigation of algae distribution in Eymir Lake using site measurements and remotely sensed data", Master of Science, Thesis submitted to the Department of Environmental Engineering, METU, Ankara, 2006.

Fausett L. "Fundamentals of Neural Networks, Architectures, Algorithms, and Applications." 1994.

Girosi F., Jones M. and Poggio T. (1995). Regularization Theory and neural network architectures. *Neural Computation*, 7, 219-269.

Hebb D. O. (1949). "The Organization of Behavior". New York: John Wiley and Sons. Introduction and Chapter 4 reprinted in Anderson & Rosenfeld (1988), pp. 45-46.

Hutchinson G. E. "A Treatise on Limnology", Vol 1: Geography, Physics and Chemistry. New York: John Wiley and Sons, 1957.

Jang J. S. R., Sun. C. T., and Mizutani E. "Neuro-Fuzzy and soft computing – A computational approach to learning and machine intelligence". Prentice Hall, Inc. 1997.

Jang R. J. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Man, and Cybernetics, 3(3), 5-683.

Jeon J., "Fuzzy and Neural Network Models for Analyses of Piles", Doctor of Philosophy Thesis, Thesis submitted to the Department of Civil Engineering, North Carolina State University, Raleigh – North Carolina , 2007.

Kalff J. "Limnology : inland water ecosystems", Prentice Hall, 2002.

Karakoç G., Erkoç F. Ü. and Katırcıoğlu H. (2003). Water quality and impacts of pollution sources for Eymir and Mogan Lakes (Turkey). Environment International, 29(1), 21-27.

Karul C., "Development of an artificial neural network model for the estimation of Chlorophyll-a in Lakes, Doctor of Philosophy Thesis, Thesis submitted to the Department of Environmental Engineering, METU, Ankara, 1999.

Kohonen T. (1972). "Correlation Matrix Memories." IEEE Transactions on Computers, C-21:353-359. Reprinted in Anderson & Rosenfeld (1988), pp. 92-114.

Konar A. "Artificial Intelligence and Soft Computing, Behavioral and Cognitive Modeling of the Human Brain." 1999.

Li H., Hou G., Dakui F., Xiao B., Song, L. and Liu Y. (2007). Prediction and Elucidation of the Population Dynamics of *Mycrocystis* spp. in Lake Dianchi (China) by means of artificial neural networks. *Ecological Informatics* 2, 184-192.

Lin C. T. and Lee C. S. G. "Neural Fuzzy Systems: A Neuro-fuzzy synergism to intelligent system." Prentice-Hall Inc., Upper Saddle River, N.J. 1996.

MacKay D. J. "Information Theory, Inference, and Learning Algorithms." Cambridge, 2003-2004.

Mamdani E. H. and Assilian S. (1975). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7,1-13.

Matlab Fuzzy Logic Toolbox User's Guide (Release 2006a – www.mathworks.com).

Matlab Neural Network Toolbox Tutorial (Version 5 - www.mathworks.com).

McClelland J. L. and Rumelhart D. E. "Explorations in Parallel Distributed Processing." Cambridge, MA: MIT Press., 1988.

McCulloch W. S. and Pitts W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 115-133. Reprinted in Anderson & Rosenfeld (1988), pp. 18-28.

Moral H., "Modeling of activated sludge process by using artificial neural networks", Master of Science, Thesis submitted to the Department of Environmental Engineering, METU, Ankara, 2004.

Nauck D., Klawonn F., Kruse R., "Combining Neural Networks and Fuzzy Controllers", Technical University of Braunschweig, Bueltenweg 74-75, Braunschweig, Germany, 1993.

Rosenblatt F. (1958). The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65, 386-408. Reprinted in Anderson and Rosenfeld (1988), pp. 92-114.

Rosenblatt F. (1959). "Two Theorems of Statistical Separability in the Perceptron." *Mechanization of Thought Process: Proceeding of a Symposium Held at National Physical Laboratory, November 1958*. London: HM Stationery Office, pp 421-456.

Rosenblatt F. "Principles of Neurodynamics". New York: Spartan, 1962.

Rumelhart D. E., Hinton G. E. and Williams R.J. (1986a). "Learning Internal Representations by Error Propagation." In Rumelhart D. E. and McClelland J. L. eds., *Parallel Distributed Processing*, 1(8). Reprinted in Anderson & Rosenfeld (1988), pp. 675-695.

Rutkowska D. "Neuro-Fuzzy Architectures and Hybrid Learning", Physica-Verlag Heidelberg, New York, 2002.

Simon H. "Neural Networks, A Comprehensive Foundation." 1994.

Talib A., Recknagel F., Cao H. and van der Molen D.T. (2008). Forecasting and explanation of algal dynamics in two shallow lakes by recurrent artificial neural network and hybrid evolutionary algorithm. *Mathematics and Computers in Simulation*, 78(2-3), 424-434.

Takagi, T., Sugeno, M., (1985), "Fuzzy Identification of Systems and its Applications to Modeling and Control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, pp. 116-132

Tan C.O. and Beklioglu M. (2005). Catastrophic-Like Shifts In Shallow Turkish Lakes: A Modeling Approach. *Ecological Modelling*, 183 (4), 425-434.

Tapkin S., "A recommended neural trip distribution model", Master of Science, Thesis submitted to the Department of Civil Engineering, METU, Ankara, 2004.

Tchobanoglous G. and Schroeder E.D., "Water Quality, Characteristics, Modeling, Modification", Addison-Wesley Publishing Company, 1987.

Thomann R.V., and Mueller J.A., "Principles of surface water quality modeling and control", Harper Collins Publishers Inc., NewYork, 1987.

Tzafestas S.G., Dalianis P.J. and Anthopoulos, G. (1996). On the overtraining phenomenon of backpropagation neural networks. *Mathematics and Computers in Simulation*, 40, 507-521.

Werbos P. "Beyond Regression: New Tools For Prediction and Analysis in the Behavioral Sciences" Doctor of Philosophy, Thesis submitted to the Department of Harvard U. Committee on Applied Mathematics, Cambridge MA,1974.

Yilmaz N.A.S., "A Temporal Neuro-Fuzzy Approach for Time Series Analysis", Doctor of Philosophy, Thesis submitted to the Department of Computer Engineering, METU, Ankara, 2003.

Zadeh L.A. (1965). Fuzzy Sets. *Journal of Information Sciences*, 8, 338-353.

Zadeh L.A. (1971). Similarity Relations and Fuzzy Ordering. *Journal of Information Sciences*, 3, 177-206.

APPENDIX A

(RESULTANT GRAPHS OF ANN MODELING)

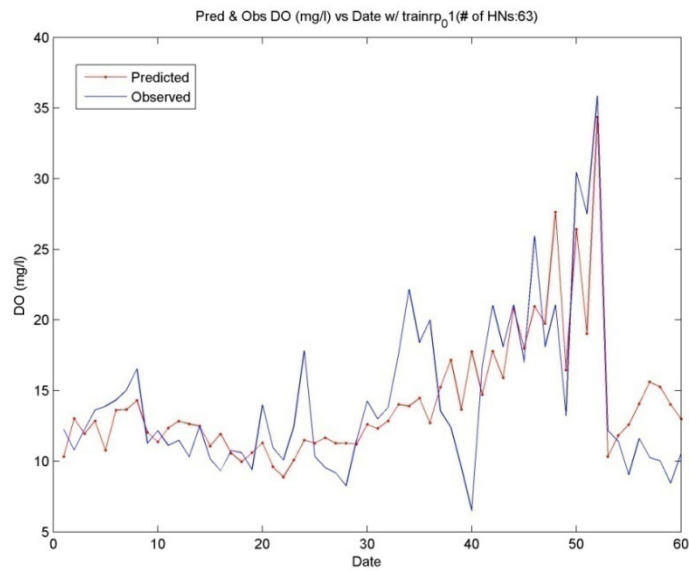


Figure A.1 “Predicted DO vs. Real DO” Graph of **Surface A1** Data Combination (*Training Algorithm: trainrp; Act. Func.: logsig + logsig; No. of HD: 63*)

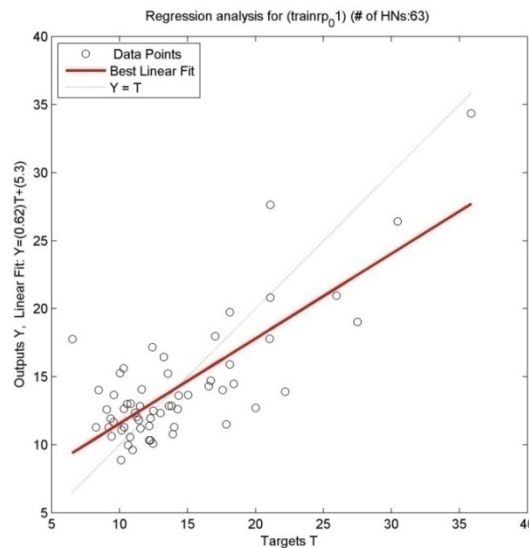


Figure A.2 “Linear Regression” Graph of **Surface A1** Data Combination (*Training Algorithm: trainrp; Actv. Func.: logsig + logsig; No. of HD: 63*)

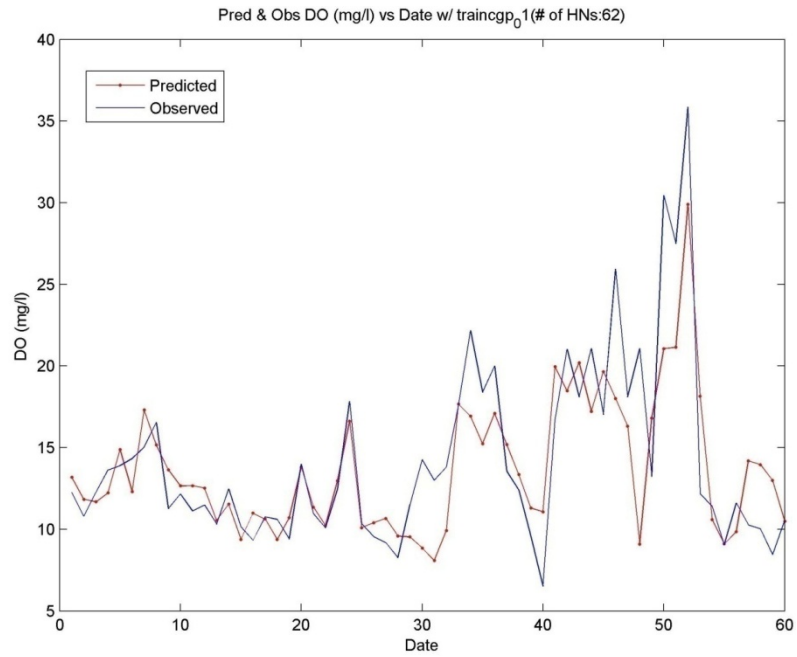


Figure A.3 “Predicted DO vs. Real DO” Graph of **Surface A2** Data Combination (*Training Algorithm: traincgp; Actv. Func.: logsig + logsig; No. of HD: 62*)

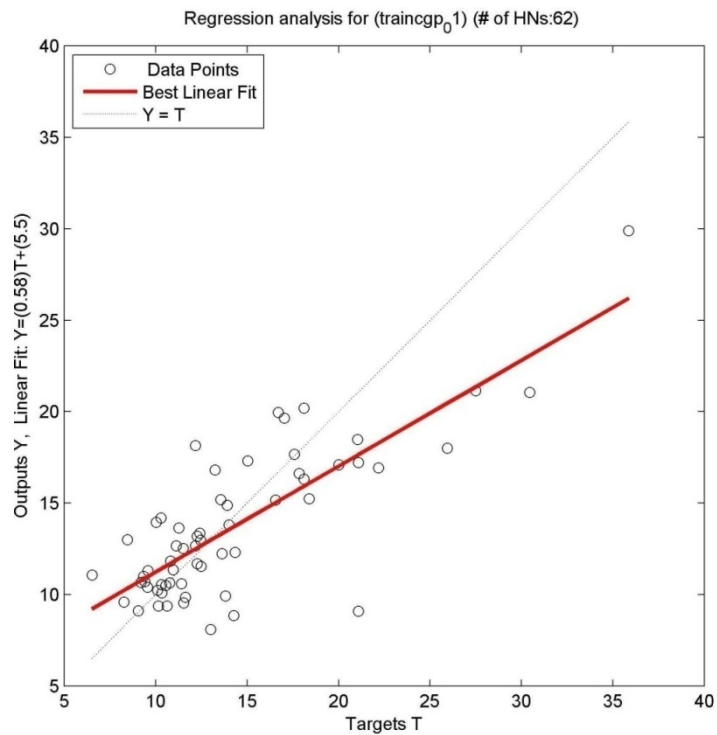


Figure A.4 “Linear Regression” Graph of **Surface A2** Data Combination (*Training Algorithm: traincgp; Actv. Func.: logsig + logsig; No. of HD: 62*)

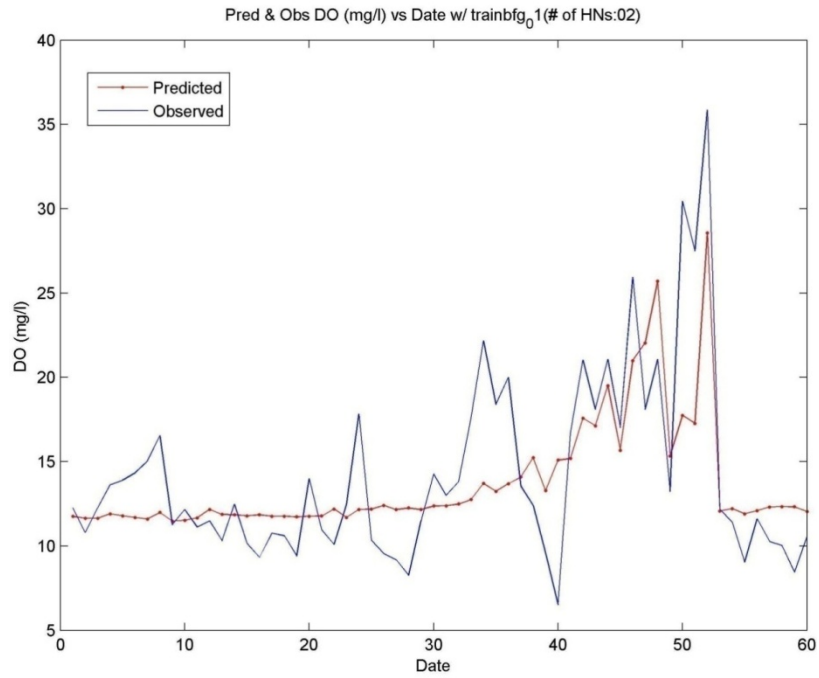


Figure A.5 “Predicted DO vs. Real DO” Graph of **Surface A3** Data Combination (*Training Algorithm: trainbfg; Actv. Func.: tansig + tansig; No. of HD: 2*)

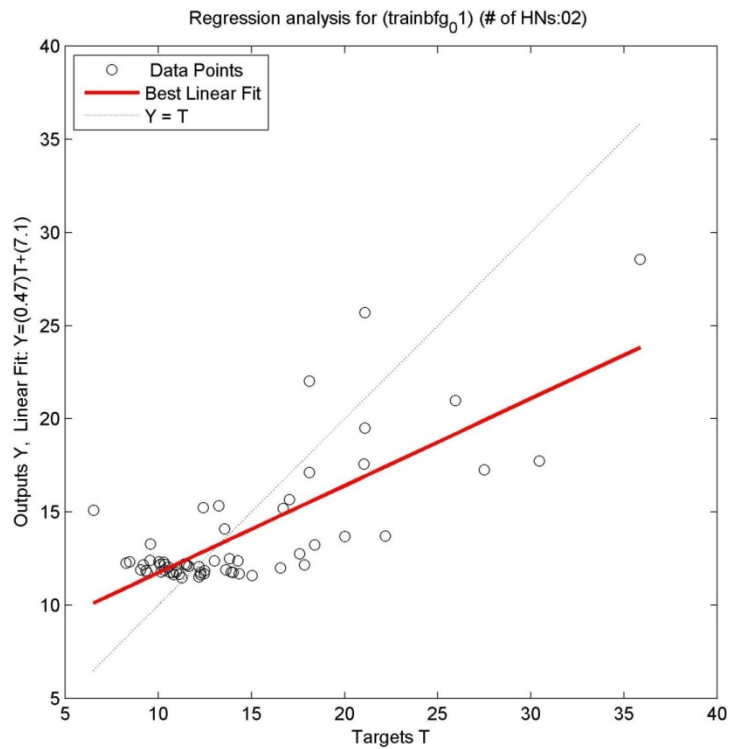


Figure A.6 “Linear Regression” Graph of **Surface A3** Data Combination (*Training Algorithm: trainbfg; Actv. Func.: tansig + tansig; No. of HD: 2*)

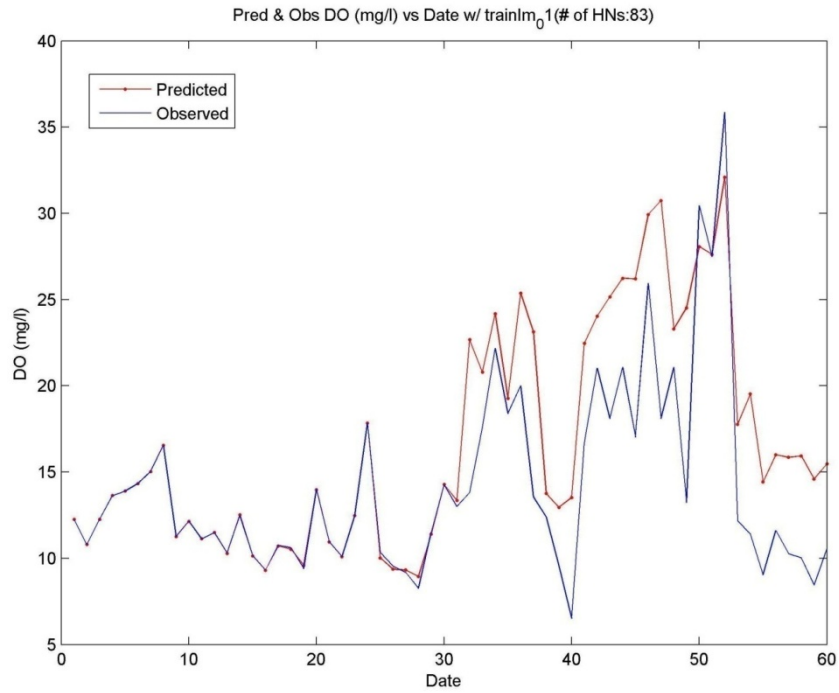


Figure A.7 “Predicted DO vs. Real DO” Graph of **Surface A4** Data Combination (*Training Algorithm: trainlm; Actv. Func.: logsig + logsig; No. of HD: 83*)

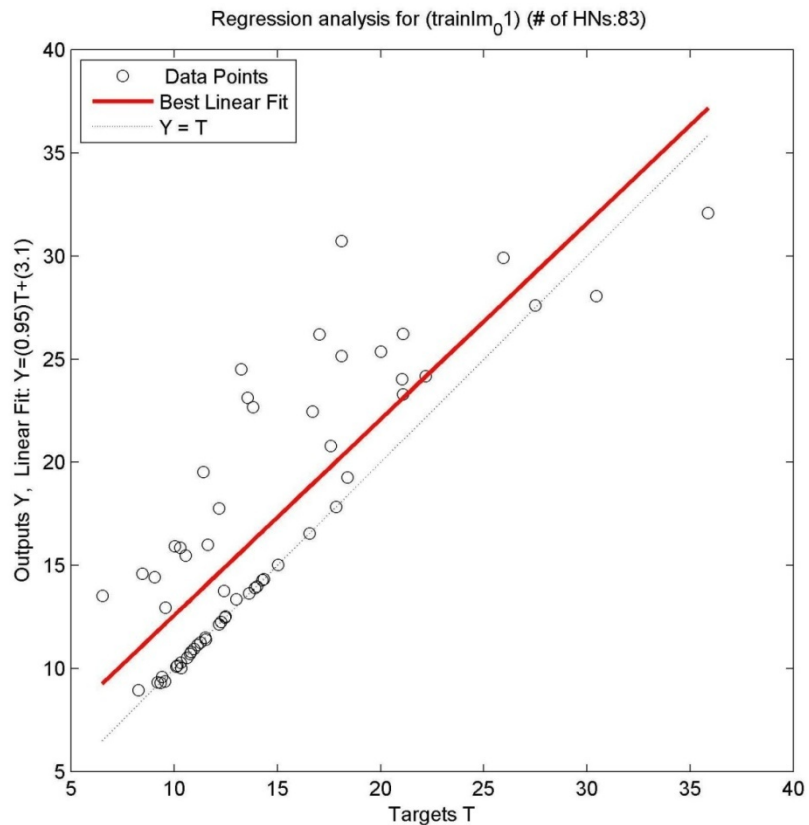


Figure A.8 “Linear Regression” Graph of **Surface A4** Data Combination (*Training Algorithm: trainlm; Actv. Func.: logsig + logsig; No. of HD: 83*)

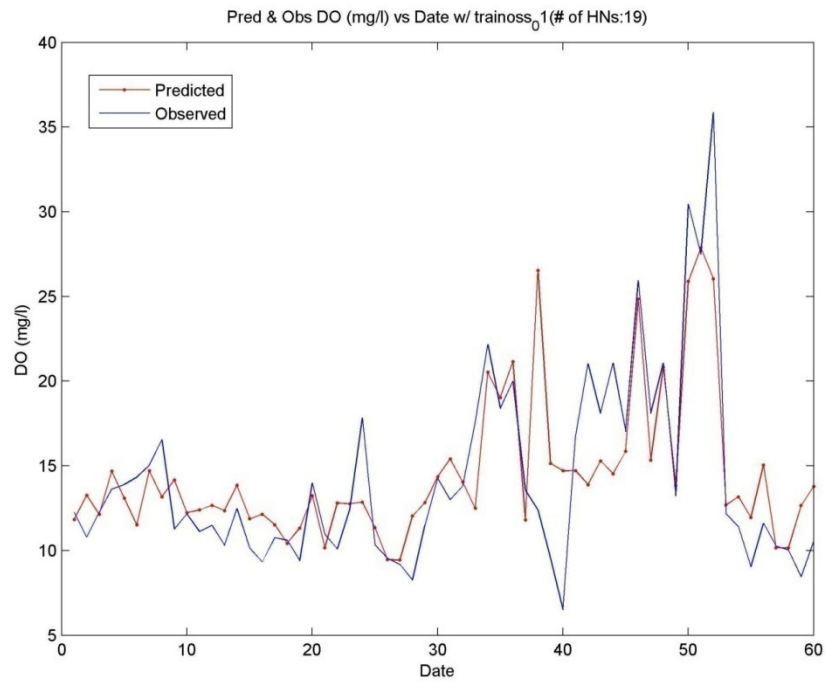


Figure A.9 “Predicted DO vs. Real DO” Graph of **Surface A5** Data Combination (*Training Algorithm: trainoss; Actv. Func.: logsig + logsig; No. of HD: 19*)

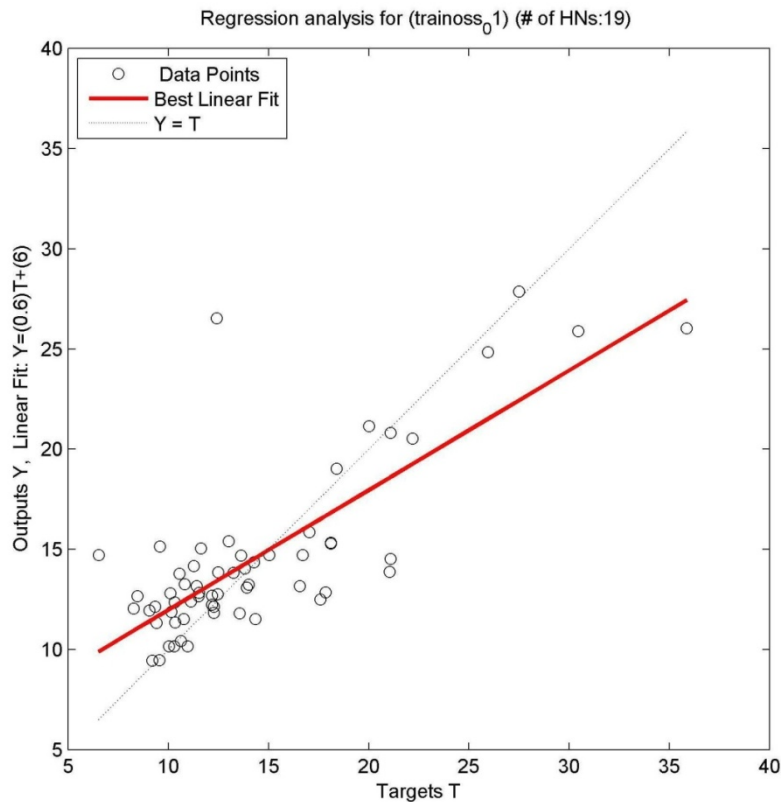


Figure A.10 “Linear Regression” Graph of **Surface A5** Data Combination (*Training Algorithm: trainoss; Actv. Func.: logsig + logsig; No. of HD: 19*)

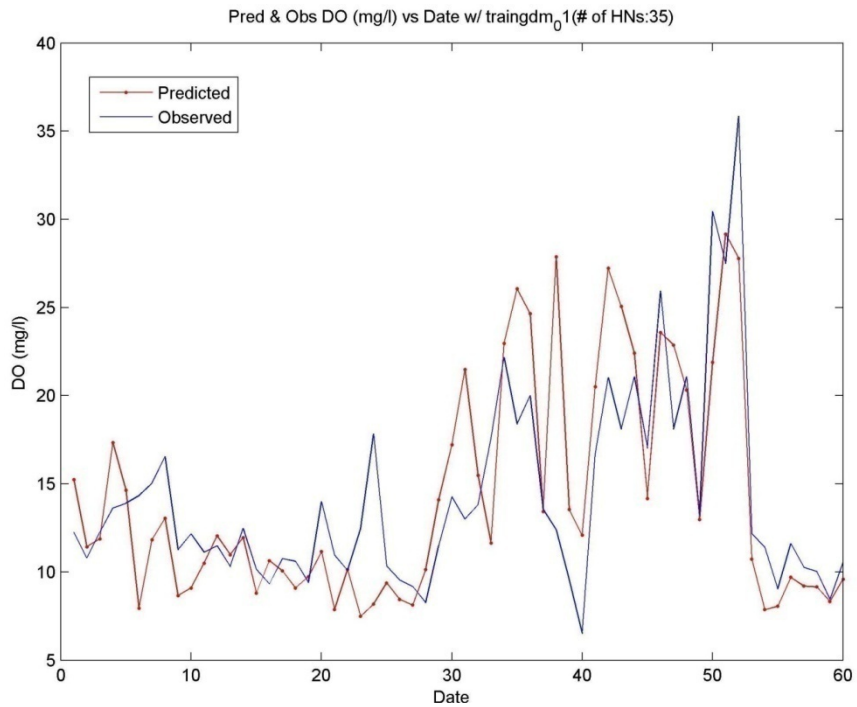


Figure A.11 “Predicted DO vs. Real DO” Graph of **Surface A6** Data Combination (*Training Algorithm: traingdm; Actv. Func.: logsig + logsig; No. of HD: 35*)

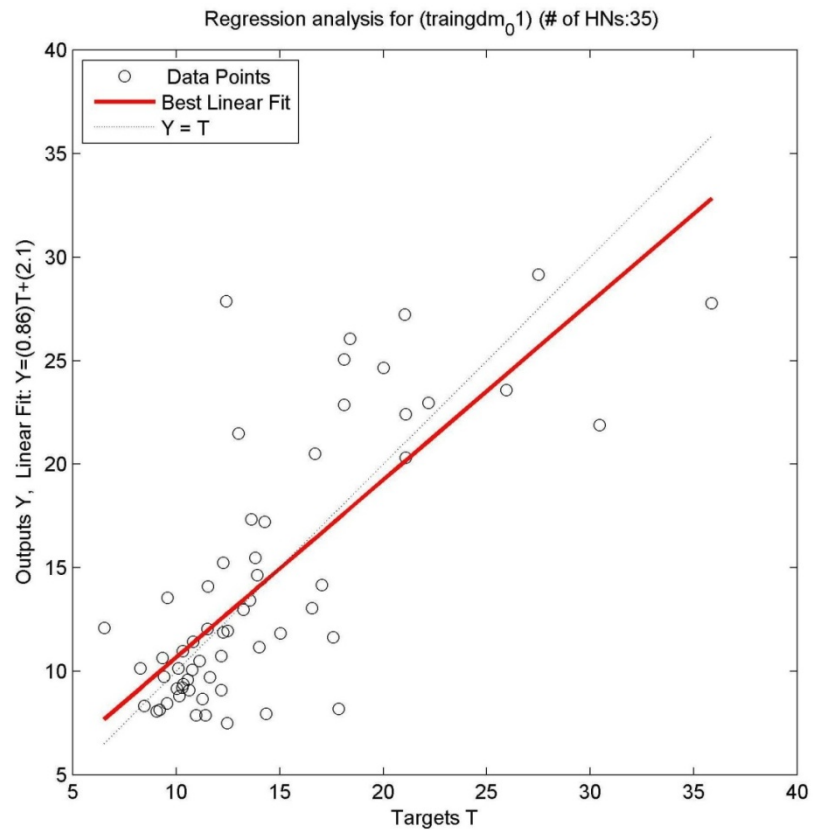


Figure A.12 “Linear Regression” Graph of **Surface A6** Data Combination (*Training Algorithm: traingdm; Actv. Func.: logsig + logsig; No. of HD: 35*)

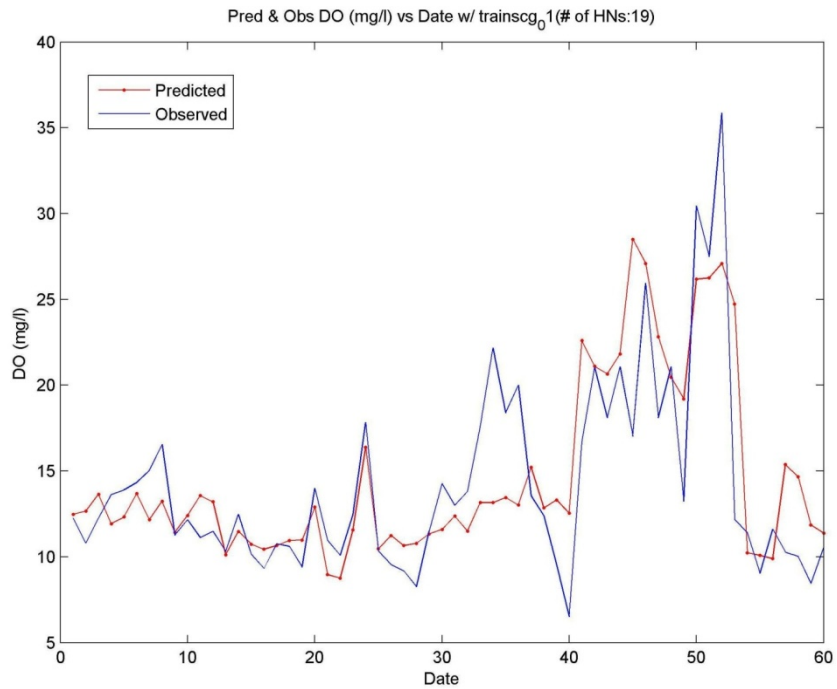


Figure A.13 “Predicted DO vs. Real DO” Graph of **Surface B4** Data Combination (*Training Algorithm: trainscg; Actv. Func.: logsig + logsig; No. of HD: 19*)

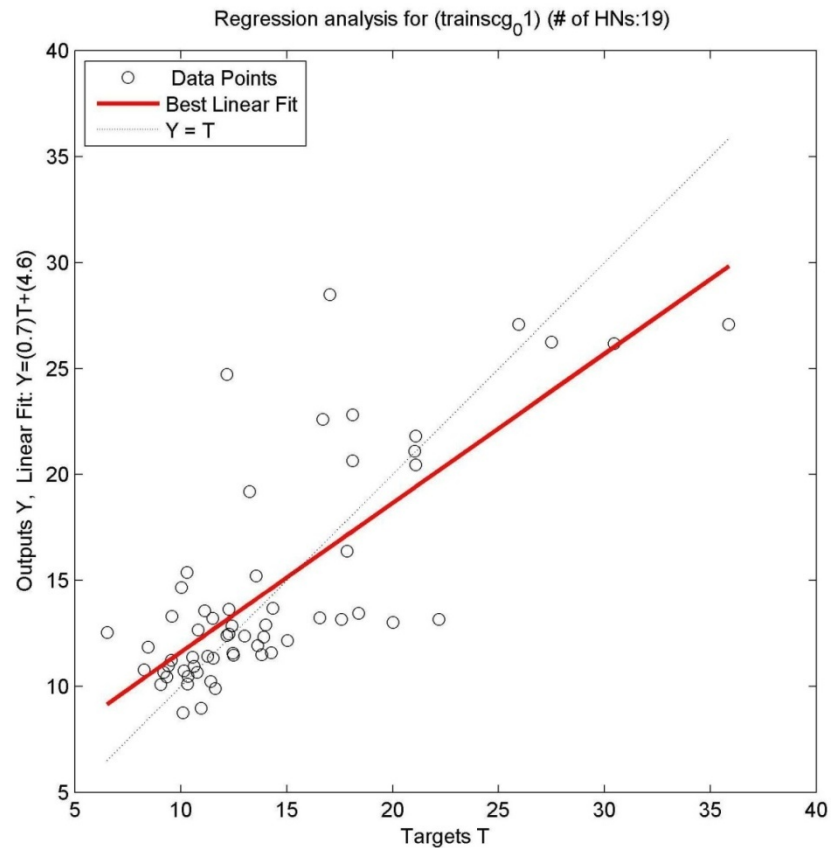


Figure A.14 “Linear Regression” Graph of **Surface B4** Data Combination (*Training Algorithm: trainscg; Actv. Func.: logsig + logsig; No. of HD: 19*)

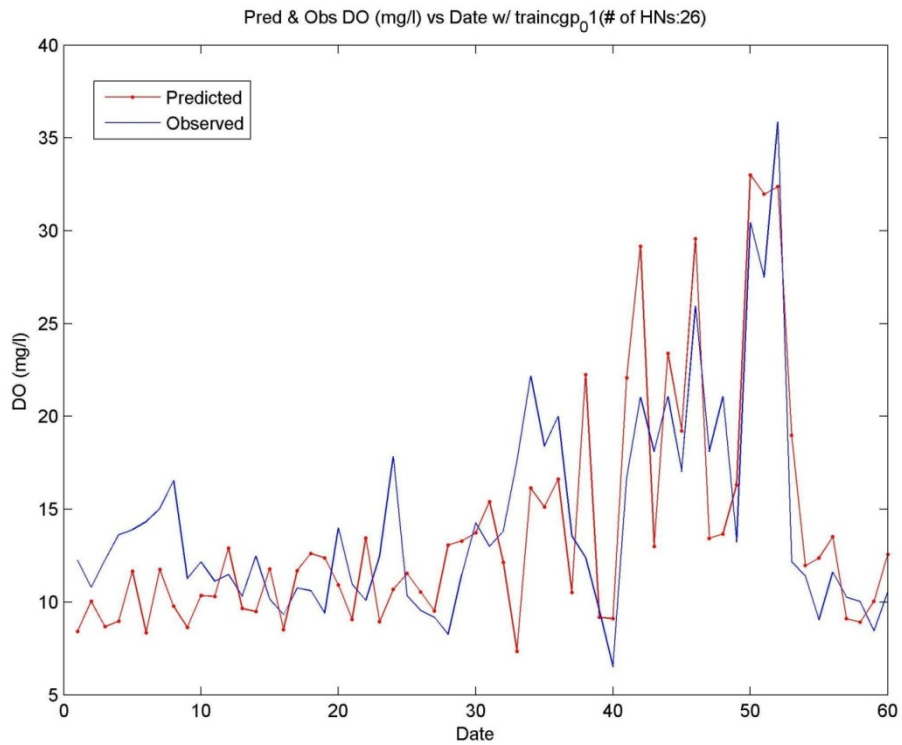


Figure A.15 “Predicted DO vs. Real DO” Graph of **Surface B5** Data Combination (*Training Algorithm: traincgp; Actv. Func.: logsig + logsig; No. of HD: 26*)

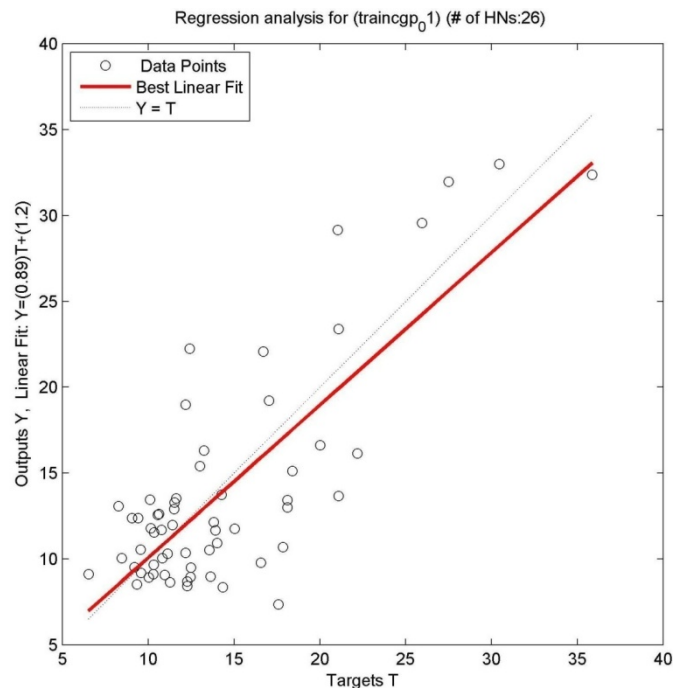


Figure A.16 “Linear Regression” Graph of **Surface B5** Data Combination (*Training Algorithm: traincgp; Actv. Func.: logsig + logsig; No. of HD: 26*)

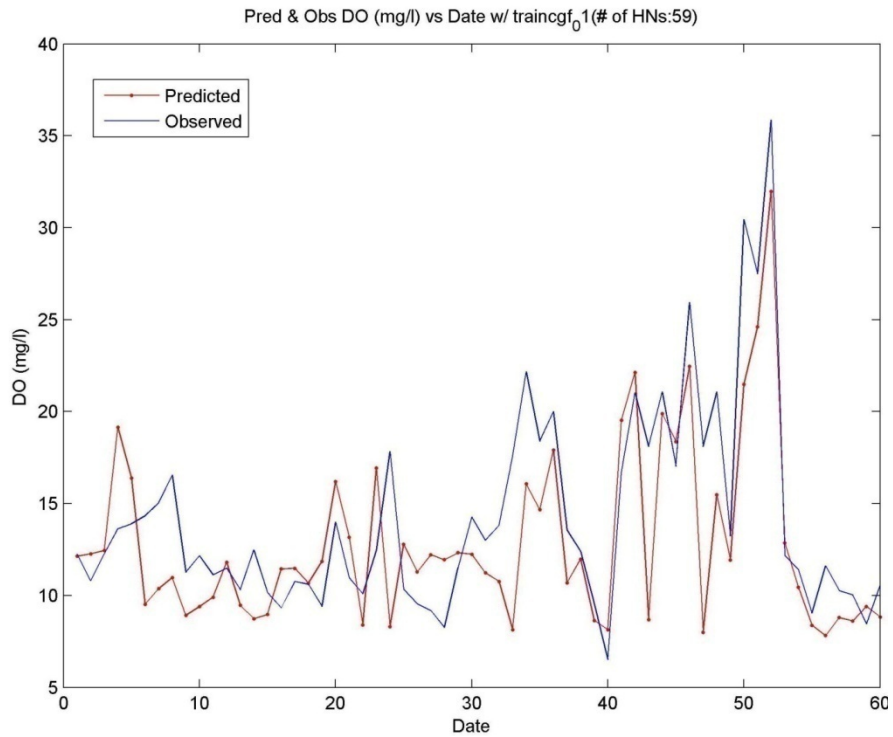


Figure A.17 “Predicted DO vs. Real DO” Graph of **Surface B6** Data Combination (*Training Algorithm: traincgf; Actv. Func.: logsig + logsig; No. of HD: 59*)

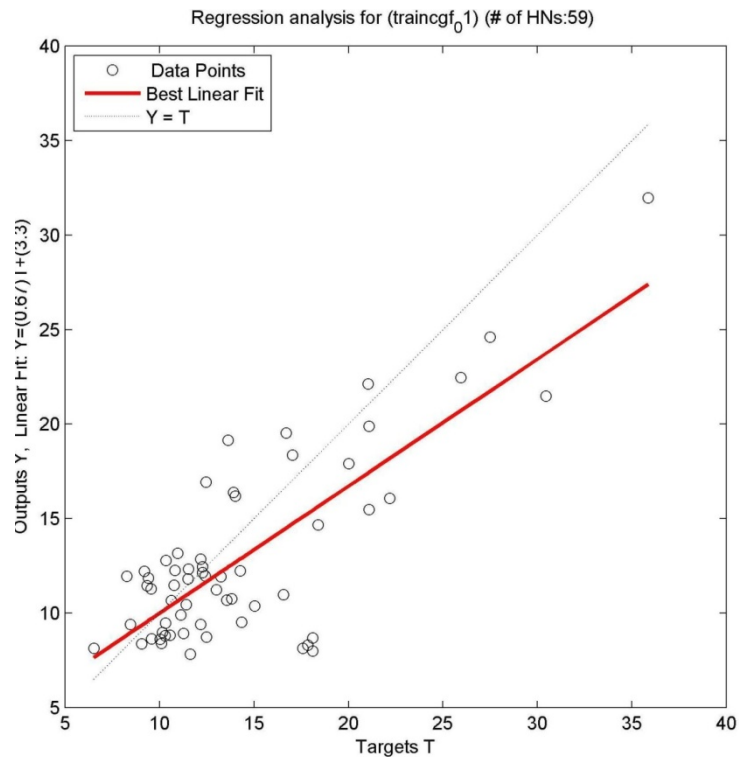


Figure A.18 “Linear Regression” Graph of **Surface B6** Data Combination (*Training Algorithm: traincgf; Actv. Func.: logsig + logsig; No. of HD: 59*)

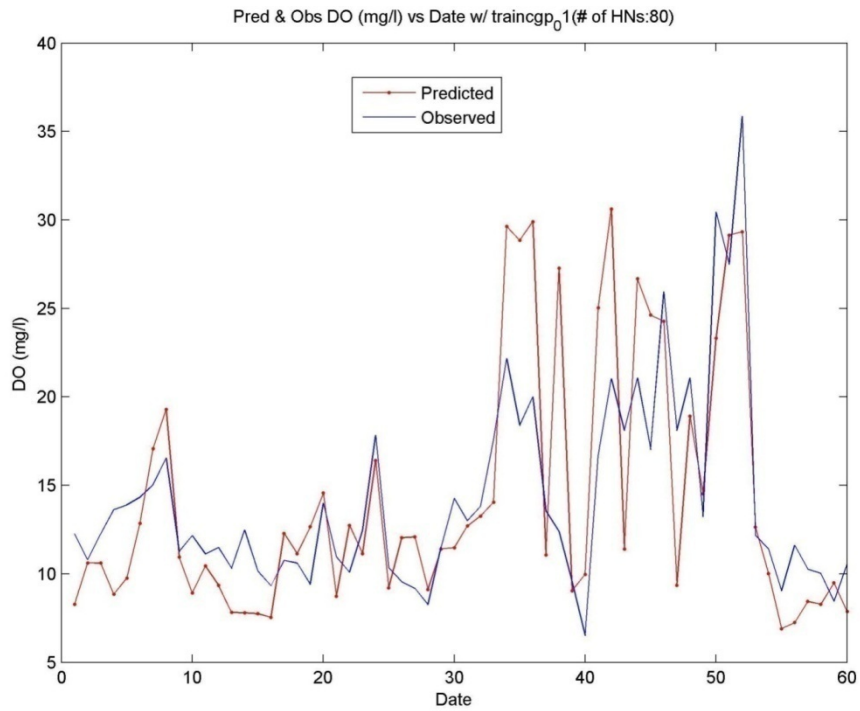


Figure A.19 “Predicted DO vs. Real DO” Graph of **Surface B7** Data Combination (*Training Algorithm: traincgp; Actv. Func.: logsig + logsig; No. of HD: 80*)

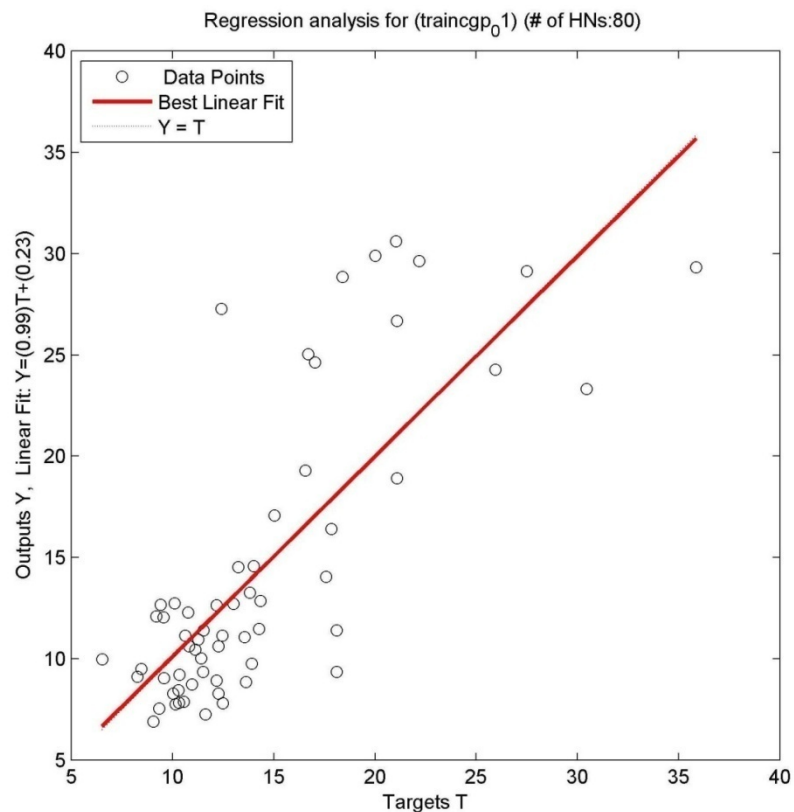


Figure A.20 “Linear Regression” Graph of **Surface B7** Data Combination (*Training Algorithm: traincgp; Actv. Func.: logsig + logsig; No. of HD: 80*)

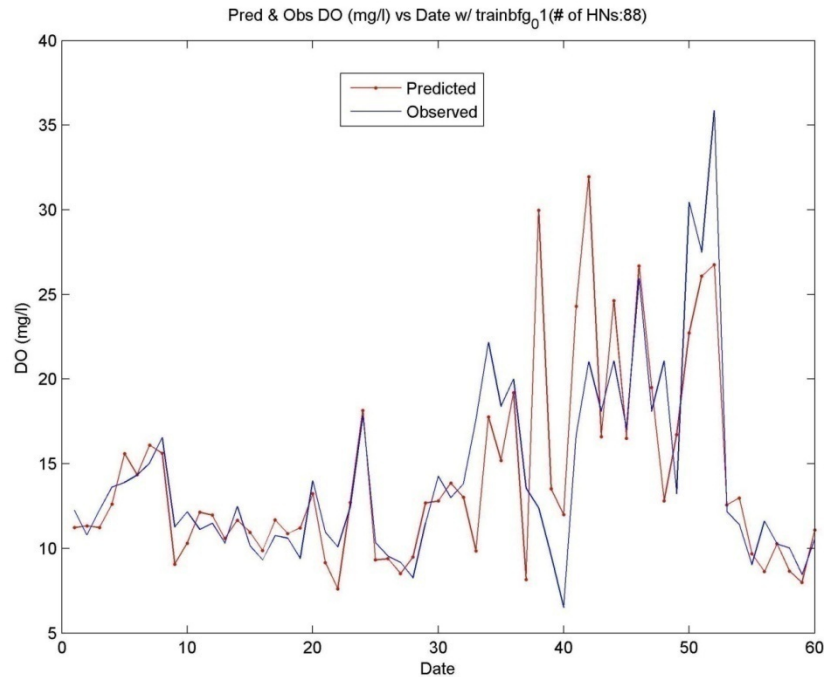


Figure A.21 “Predicted DO vs. Real DO” Graph of **Surface B8** Data Combination (*Training Algorithm: trainbfg; Actv. Func.: logsig + logsig; No. of HD: 88*)

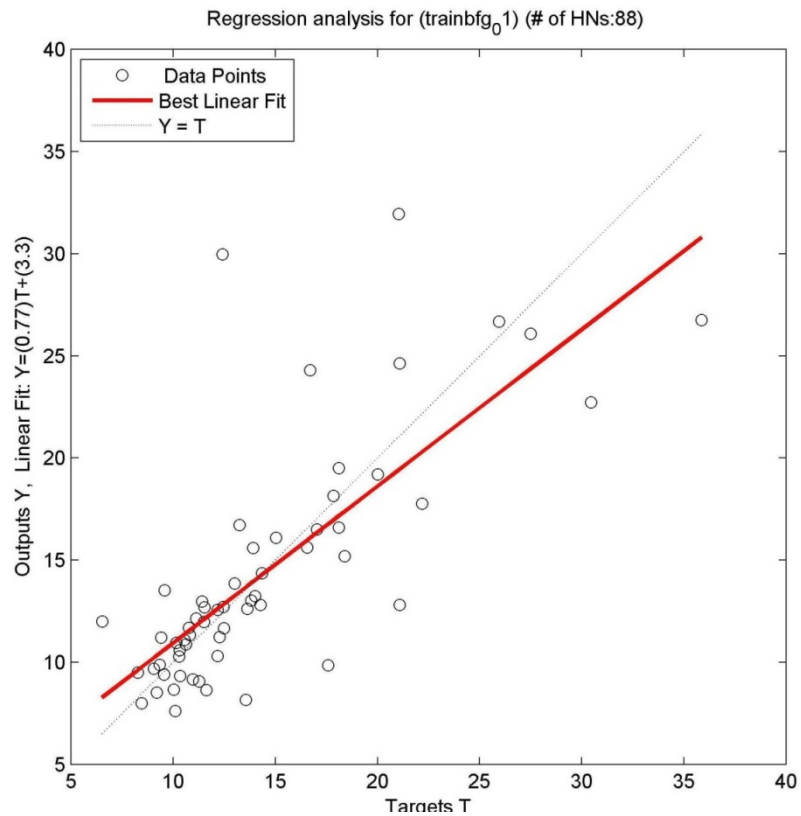


Figure A.22 “Linear Regression” Graph of **Surface B8** Data Combination (*Training Algorithm: trainbfg; Actv. Func.: logsig + logsig; No. of HD: 88*)

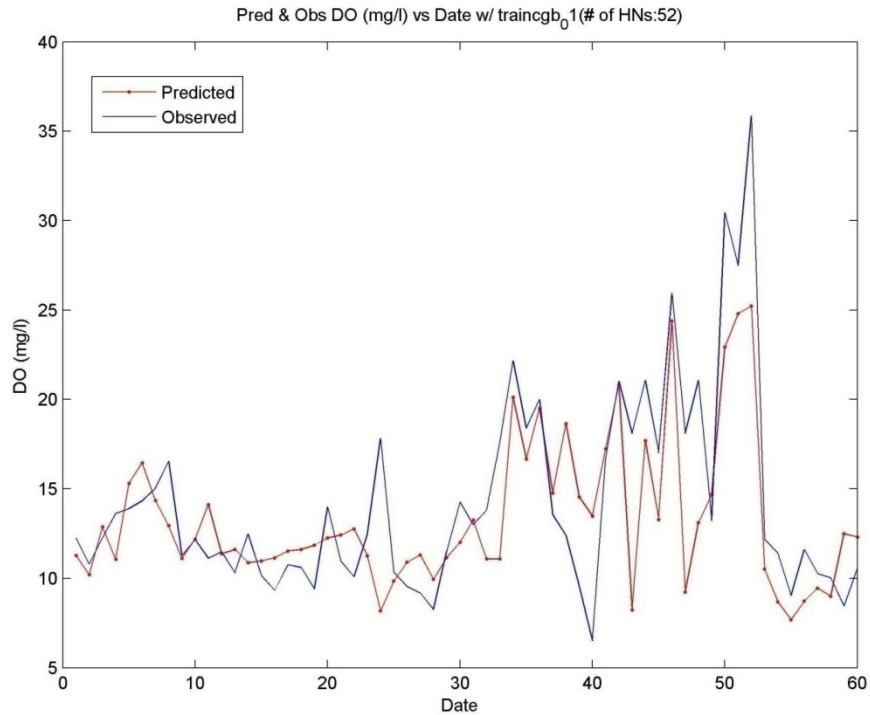


Figure A.23 “Predicted DO vs. Real DO” Graph of **Surface B9** Data Combination (*Training Algorithm: traincgb; Actv. Func.: logsig + logsig; No. of HD: 52*)

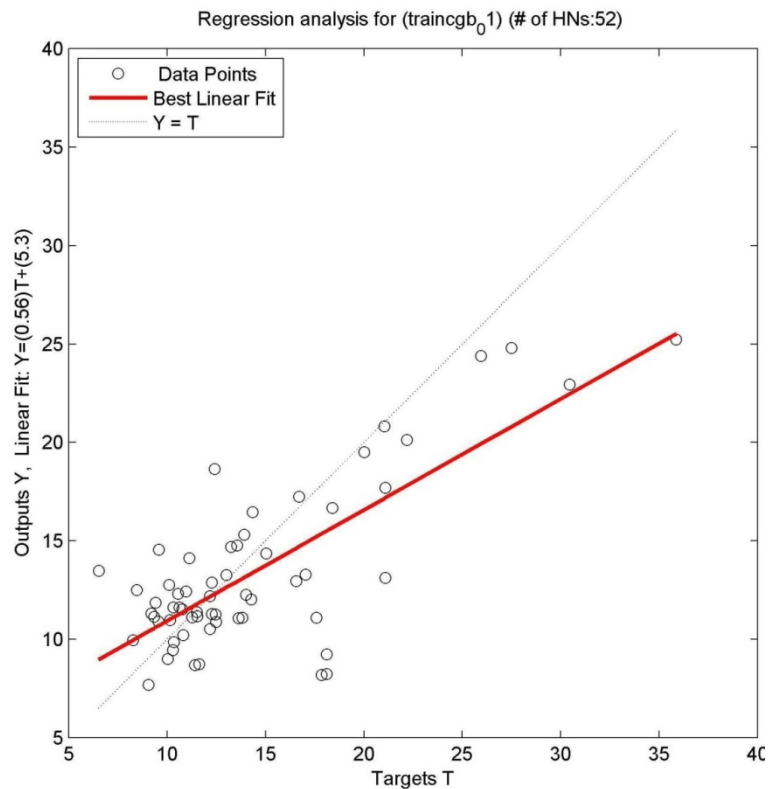


Figure A.24 “Linear Regression” Graph of **Surface B9** Data Combination (*Training Algorithm: traincgb; Actv. Func.: logsig + logsig; No. of HD: 52*)

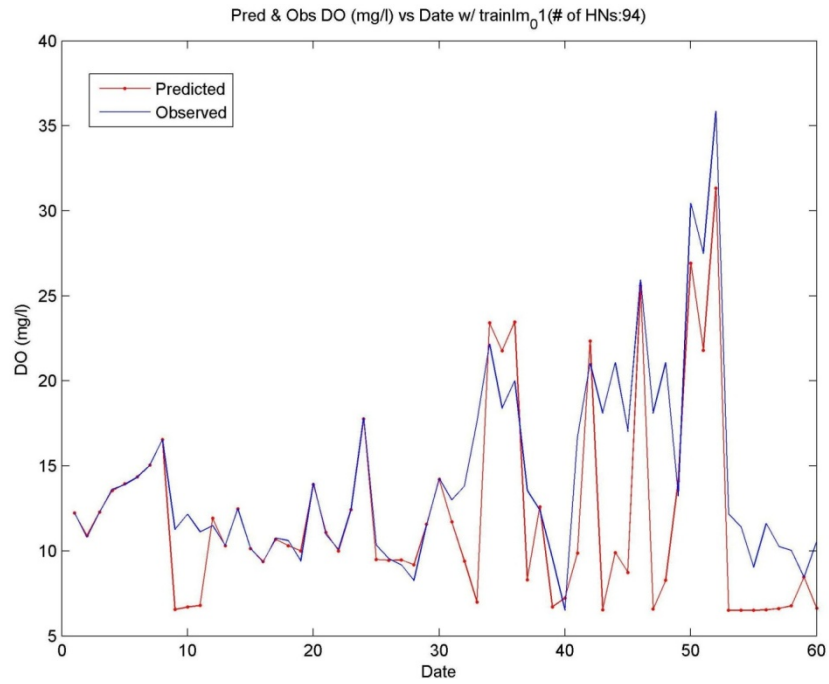


Figure A.25 “Predicted DO vs. Real DO” Graph of **Average A9** Data Combination (*Training Algorithm: trainIm; Actv. Func.: logsig + logsig; No. of HD: 94*)

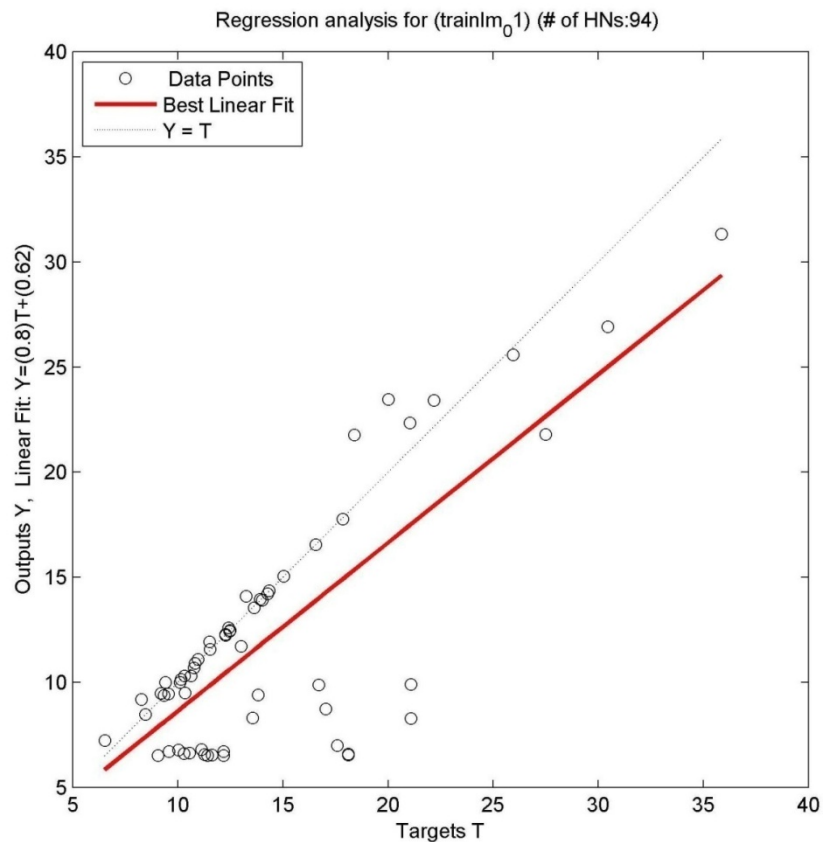


Figure A.26 “Linear Regression” Graph of **Average A9** Data Combination (*Training Algorithm: trainIm; Actv. Func.: logsig + logsig; No. of HD: 94*)

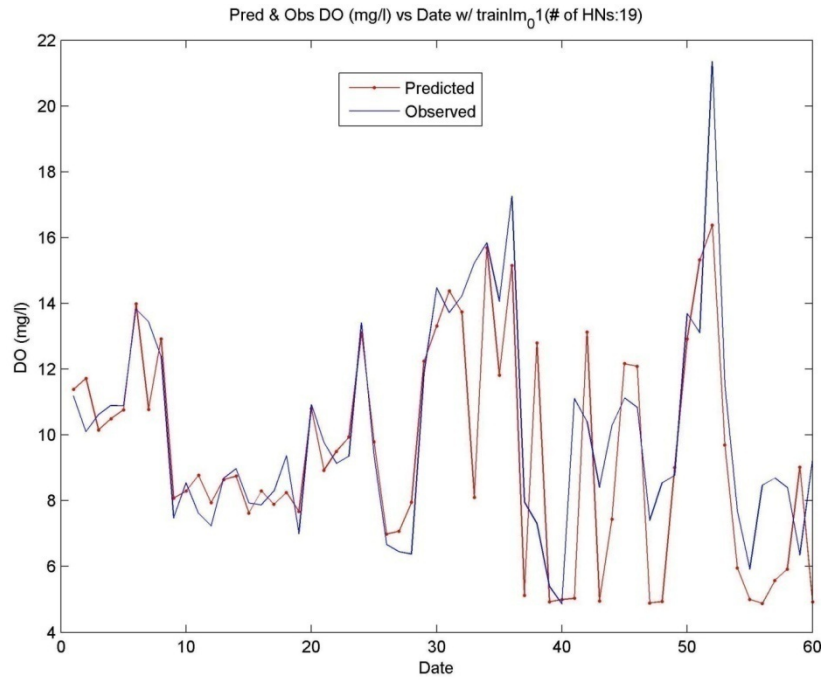


Figure A.27 “Predicted DO vs. Real DO” Graph of **Average B5** Data Combination (*Training Algorithm: trainlm; Actv. Func.: logsig + logsig; No. of HD: 19*)

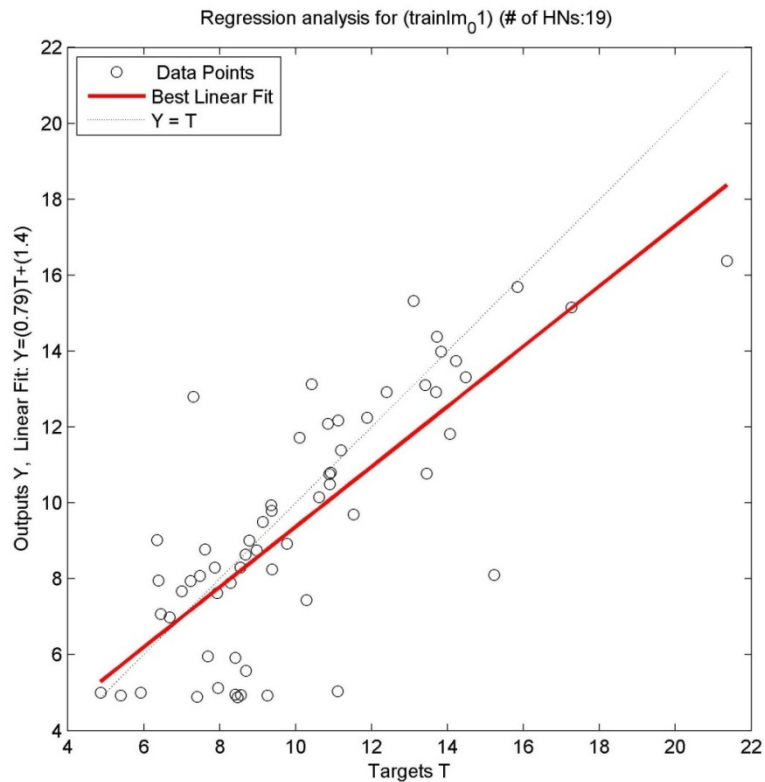


Figure A.28 “Linear Regression” Graph of **Average B5** Data Combination (*Training Algorithm: trainlm; Actv. Func.: logsig + logsig; No. of HD: 19*)

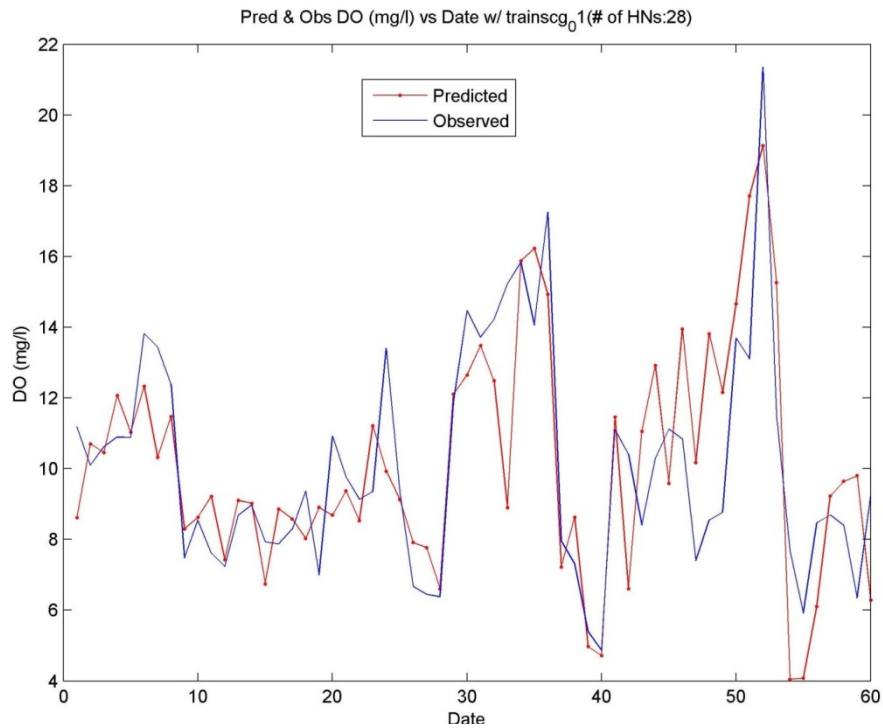


Figure A.29 “Predicted DO vs. Real DO” Graph of **Average B6** Data Combination (*Training Algorithm: trainlm; Actv. Func.: tansig + tansig; No. of HD: 28*)

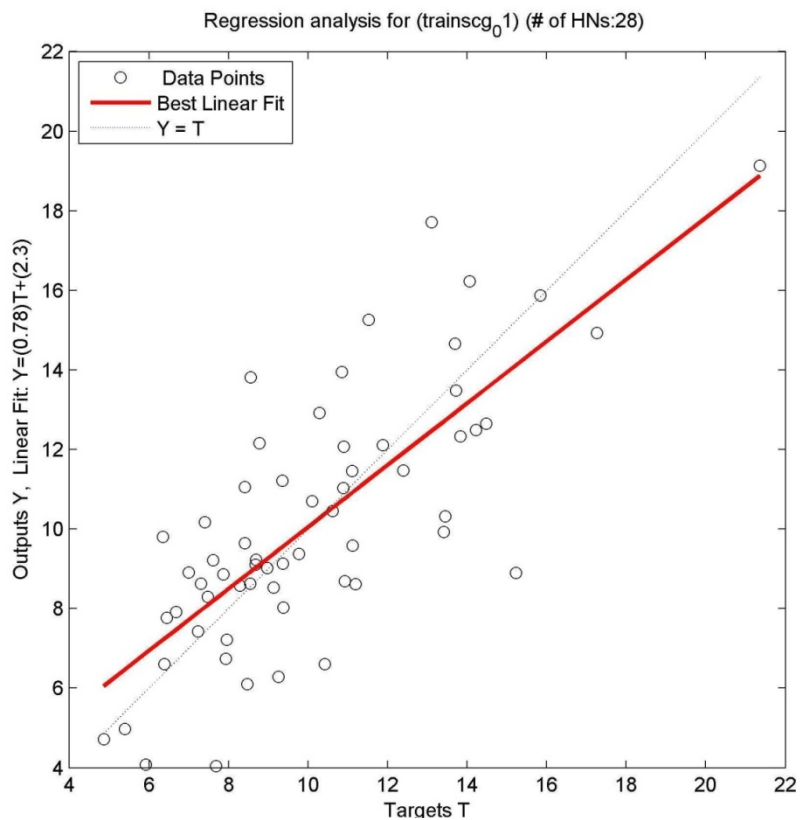


Figure A.30 “Linear Regression” Graph of **Average B6** Data Combination (*Training Algorithm: trainlm; Actv. Func.: tansig + tansig; No. of HD: 28*)

APPENDIX B

DATA USED DURING MODELING STUDIES

Table B.1 Surface Data

DATE	Point	Temp (c)	pH	Chl-a (µg/l)	Cond. (ms/cm)	DO (mg/l)	TKN (mg/l)	NO3 (mg/l)	Total P (mg/l)	PO4 (mg/l)	Alkalinity (mg/l as CaCO3)	Precipitation (Max)mm	Wind (max) knot	Ambient air temp (C)
26.03.2005	1	9.17	8.27	1.10	2.79	12.25	0.12	0.07	0.37	0.39	572.00	0.00	10.00	15.00
	2	9.30	8.23	1.00	2.76	10.80	0.25	0.75	0.39	0.37	580.00	0.00	10.00	15.00
	3	9.86	8.04	11.10	2.82	12.25	0.27	0.90	0.39	0.36	558.00	0.00	10.00	15.00
	4	9.96	7.73	8.00	2.80	13.63	2.50	0.50	0.33	0.32	580.00	0.00	10.00	15.00
13.04.2005	1	13.53	8.31	3.20	2.77	13.90	0.35	0.11	0.35	0.34	582.00	0.00	13.00	23.20
	2	13.35	8.31	64.00	2.78	14.33	0.48	0.91	0.35	0.35	578.00	0.00	13.00	23.20
	3	14.03	8.37	4.30	2.77	15.03	0.46	0.89	0.35	0.35	638.00	0.00	13.00	23.20
	4	14.40	8.27	12.90	2.76	16.55	2.75	0.92	0.35	0.33	610.00	0.00	13.00	23.20
22.04.2005	1	14.09	8.48	14.50	2.76	11.26	0.67	0.83	0.44	0.41	664.00	5.00	19.00	19.30
	2	14.65	8.36	1.70	2.78	12.16	0.67	0.90	0.43	0.41	648.00	5.00	19.00	19.30
	3	15.43	8.37	0.80	2.78	11.12	0.73	0.85	0.43	0.40	620.00	5.00	19.00	19.30
	4	16.75	8.30	0.20	2.77	11.49	3.11	1.00	0.42	0.39	658.00	5.00	19.00	19.30
06.05.2005	1	14.86	8.74	2.00	2.91	10.30	1.23	0.11	0.52	0.48	645.00	0.00	16.00	24.80
	2	15.57	8.57	2.10	2.89	12.48	1.12	0.11	0.51	0.48	660.00	0.00	16.00	24.80
	3	15.69	8.71	2.20	2.93	10.14	1.34	0.62	0.53	0.48	635.00	0.00	16.00	24.80
	4	15.93	8.71	2.90	2.94	9.32	0.78	0.02	0.55	0.50	653.00	0.00	16.00	24.80
26.05.2005	1	20.63	8.10	1.90	2.74	10.76	0.00	0.20	0.63	0.56	646.00	0.00	12.00	27.50
	2	20.65	8.10	1.90	2.74	10.61	0.00	0.29	0.67	0.57	630.00	0.00	12.00	27.50
	3	20.61	8.10	1.90	2.72	9.41	0.00	0.29	0.62	0.56	646.00	0.00	12.00	27.50
	4	20.83	8.10	2.70	2.72	14.00	0.00	0.20	0.61	0.51	668.00	0.00	12.00	27.50
08.06.2005	1	22.09	8.35	1.00	2.87	10.95	0.00	0.11	0.78	0.63	676.00	0.00	9.00	29.10
	2	22.82	8.42	6.10	2.86	10.09	0.00	0.29	0.86	0.64	590.00	0.00	9.00	29.10
	3	23.77	8.45	19.90	2.85	12.45	0.00	0.20	0.68	0.58	708.00	0.00	9.00	29.10
	4	25.50	9.92	8.30	2.86	17.83	0.00	0.11	0.72	0.51	700.00	0.00	9.00	29.10
01.07.2005	1	24.90	8.74	2.40	2.91	10.34	0.00	0.07	0.73	0.56	656.00	0.00	11.00	27.40
	2	25.43	8.77	30.60	2.92	9.55	0.00	0.10	0.74	0.52	654.00	0.00	11.00	27.40
	3	24.03	8.75	28.20	2.90	9.18	0.00	0.07	0.73	0.54	660.00	0.00	11.00	27.40
	4	25.20	8.77	0.20	2.91	8.26	0.00	0.07	0.72	0.55	644.00	0.00	11.00	27.40
13.07.2005	1	25.98	8.89	5.80	2.96	11.51	0.06	0.06	0.67	0.57	638.00	0.00	16.00	33.00
	2	26.52	8.92	0.90	2.97	14.27	0.22	0.05	0.70	0.59	628.00	0.00	16.00	33.00
	3	26.85	8.92	5.00	2.97	13.01	0.28	0.06	0.67	0.59	606.00	0.00	16.00	33.00
	4	27.28	8.91	19.30	2.98	13.80	0.45	0.07	0.67	0.57	650.00	0.00	16.00	33.00
25.07.2005	1	25.80	9.38	224.96	2.92	17.58	0.50	0.13	0.40	0.18	622.00	2.00	12.00	27.60
	2	26.75	9.46	63.60	2.92	22.19	0.81	0.14	0.41	0.17	594.00	2.00	12.00	27.60
	3	27.33	9.43	51.50	2.93	18.39	0.81	0.11	0.39	0.19	632.00	2.00	12.00	27.60
	4	27.71	9.47	62.10	2.92	20.01	0.81	0.15	0.38	0.16	594.00	2.00	12.00	27.60
08.08.2005	1	24.73	9.44	186.77	2.94	13.55	1.29	0.19	0.42	0.21	542.00	0.80	7.00	27.20
	2	26.71	9.62	0.60	2.94	12.40	1.06	0.32	0.51	0.19	594.00	0.80	7.00	27.20
	3	26.00	9.58	404.20	2.94	9.56	0.76	0.26	0.48	0.20	658.00	0.80	7.00	27.20

	4	26.32	9.58	313.30	2.95	6.51	1.57	0.24	0.43	0.18	620.00	0.80	7.00	27.20
22.08.2005	1	24.02	9.66	174.30	2.93	16.70	0.11	0.35	0.39	0.03	276.00	0.00	7.00	30.70
	2	24.16	9.70	118.60	2.94	21.03	0.11	0.39	0.45	0.02	228.00	0.00	7.00	30.70
	3	25.81	9.60	782.30	2.94	18.09	0.31	0.31	0.42	0.05	294.00	0.00	7.00	30.70
	4	26.99	9.71	180.90	2.93	21.09	0.17	0.52	0.47	0.03	238.00	0.00	7.00	30.70
05.09.2005	1	24.02	9.66	174.33	2.93	17.04	1.18	0.55	0.34	0.01	402.00	0.00	9.00	19.00
	2	24.24	9.72	125.70	2.93	25.95	2.24	0.19	0.32	0.04	408.00	0.00	9.00	19.00
	3	25.81	9.60	782.30	2.94	18.09	2.35	0.20	0.34	0.09	408.00	0.00	9.00	19.00
	4	26.99	9.71	180.90	2.93	21.09	2.80	0.15	0.46	0.22	420.00	0.00	9.00	19.00
19.09.2005	1	20.59	9.52	161.50	3.03	13.24	2.07	0.20	0.25	0.02	450.00	0.00	8.00	26.50
	2	21.97	9.78	107.60	3.02	30.44	1.62	0.24	0.38	0.02	440.00	0.00	8.00	26.50
	3	22.40	9.73	11.60	3.03	27.49	2.04	0.48	0.33	0.02	444.00	0.00	8.00	26.50
	4	22.36	9.86	49.60	2.99	35.87	2.24	0.42	0.66	0.02	456.00	0.00	8.00	26.50
03.10.2005	1	17.61	9.60	164.40	3.01	12.17	0.62	0.51	0.20	0.02	424.00	26.00	10.00	15.40
	2	18.49	9.57	158.10	3.02	11.41	1.62	0.63	0.19	0.02	568.00	26.00	10.00	15.40
	3	18.73	9.56	408.60	3.02	9.04	0.17	0.74	0.30	0.02	596.00	26.00	10.00	15.40
	4	18.83	9.61	563.20	3.02	11.61	0.45	0.87	0.38	0.02	590.00	26.00	10.00	15.40
19.10.2005	1	13.18	9.98	354.80	3.05	10.28	1.62	0.22	0.27	0.01	576.00	0.00	9.00	6.30
	2	13.61	9.93	251.10	3.05	10.02	1.74	0.20	0.25	0.01	584.00	0.00	9.00	6.30
	3	13.60	9.85	158.10	3.05	8.45	2.13	0.34	0.23	0.02	604.00	0.00	9.00	6.30
	4	13.14	9.92	469.80	3.05	10.55	1.46	0.53	0.25	0.01	574.00	0.00	9.00	6.30

Table B.2 Average Data

DATE	Point	Temp (C)	pH	Chl-a (µg/l)	Cond. (ms/cm)	DO (mg/l)	TKN (mg/l)	NO3 (mg/l)	Total P (mg/l)	PO4 (mg/l)	Alkalinity (mg/l as CaCO3)	Precipitation (Max)mm	Wind (max) knot	Ambient air temp (C)
26.03.2005	1	8.81	7.78	2.00	2.78	11.19	0.14	0.23	0.26	0.26	565.33	0.00	10.00	15.00
	2	8.66	8.22	1.40	2.77	10.10	0.55	0.75	0.42	0.40	576.67	0.00	10.00	15.00
	3	8.91	7.99	4.67	2.82	10.62	0.29	0.76	0.41	0.39	566.67	0.00	10.00	15.00
	4	9.73	7.73	7.57	2.78	10.90	1.07	0.86	0.33	0.33	572.00	0.00	10.00	15.00
13.04.2005	1	12.31	8.23	11.23	2.77	10.89	0.52	0.61	0.39	0.37	586.00	0.00	13.00	23.20
	2	11.97	8.27	2663	2.76	13.83	0.87	0.87	0.39	0.38	558.67	0.00	13.00	23.20
	3	12.68	8.31	3.93	2.77	13.44	0.55	0.87	0.48	0.47	604.00	0.00	13.00	23.20
	4	13.72	8.25	11.67	2.75	12.39	1.32	0.96	0.49	0.45	600.00	0.00	13.00	23.20
22.04.2005	1	13.72	8.41	5.57	2.69	7.48	0.84	0.90	0.52	0.50	640.67	5.00	19.00	19.30
	2	13.93	8.29	1.73	2.71	8.55	1.10	0.88	0.49	0.47	625.67	5.00	19.00	19.30
	3	14.68	8.32	0.70	2.72	7.61	0.73	0.94	0.43	0.41	620.00	5.00	19.00	19.30
	4	15.15	8.23	0.77	4.54	7.23	1.67	0.87	0.46	0.43	621.33	5.00	19.00	19.30
06.05.2005	1	14.45	8.74	27.50	2.91	8.68	1.27	0.30	0.59	0.55	628.33	0.00	16.00	24.80
	2	14.78	8.57	1.87	2.89	8.97	1.27	0.11	0.54	0.50	652.00	0.00	16.00	24.80
	3	14.73	8.71	1.97	2.90	7.93	1.01	0.70	0.52	0.47	636.67	0.00	16.00	24.80
	4	15.73	8.71	2.30	2.94	7.87	1.08	0.14	0.56	0.50	624.33	0.00	16.00	24.80
26.05.2005	1	20.50	8.12	7.77	2.71	8.29	0.00	0.20	0.64	0.57	659.33	0.00	12.00	27.50
	2	20.30	8.10	1.87	2.73	9.37	0.00	0.29	0.70	0.63	669.33	0.00	12.00	27.50
	3	19.75	8.10	3.03	2.68	7.00	0.00	0.29	0.66	0.58	654.00	0.00	12.00	27.50
	4	20.57	8.10	22.40	2.74	10.92	0.00	0.20	0.71	0.62	682.67	0.00	12.00	27.50
08.06.2005	1	21.25	8.35	12.00	8.74	9.77	0.00	0.10	0.86	0.69	636.67	0.00	9.00	29.10
	2	21.40	8.30	5.53	2.84	9.13	0.00	0.29	0.81	0.65	574.00	0.00	9.00	29.10
	3	21.57	8.25	15.17	2.83	9.36	0.00	0.20	0.75	0.63	697.33	0.00	9.00	29.10
	4	22.59	8.69	18.60	2.85	13.41	0.00	0.11	0.91	0.69	676.67	0.00	9.00	29.10
01.07.2005	1	24.56	8.70	17.79	2.82	9.37	0.00	0.08	0.73	0.54	652.67	0.00	11.00	27.40
	2	22.96	8.54	12.90	2.85	6.67	0.00	0.10	0.77	0.55	644.67	0.00	11.00	27.40
	3	22.27	8.50	11.03	2.83	6.45	0.00	0.21	1.20	0.86	642.67	0.00	11.00	27.40
	4	23.90	8.62	2.00	2.87	6.38	0.19	0.07	0.85	0.66	639.00	0.00	11.00	27.40
13.07.2005	1	25.73	8.89	6.40	2.96	11.88	0.15	0.09	0.67	0.57	632.00	0.00	16.00	33.00
	2	26.15	8.91	5.00	2.96	14.48	0.28	0.06	0.69	0.59	622.67	0.00	16.00	33.00
	3	26.24	8.90	7.43	2.96	13.72	0.32	0.06	0.68	0.59	600.00	0.00	16.00	33.00
	4	27.12	8.91	16.20	2.98	14.22	0.43	0.07	0.67	0.57	640.67	0.00	16.00	33.00
25.07.2005	1	25.69	9.21	103.09	2.94	15.22	0.34	0.13	0.49	0.32	628.00	2.00	12.00	27.60
	2	25.79	9.14	37.93	2.94	15.84	0.85	0.09	0.55	0.39	620.67	2.00	12.00	27.60
	3	26.10	9.11	33.93	2.95	14.06	0.73	0.11	0.54	0.40	611.33	2.00	12.00	27.60
	4	26.76	9.20	40.80	2.94	17.27	0.62	0.13	0.48	0.31	622.67	2.00	12.00	27.60
08.08.2005	1	23.66	9.29	161.02	2.87	7.96	1.17	0.21	0.41	0.23	589.33	0.80	7.00	27.20
	2	24.19	8.99	41.77	2.93	7.31	1.34	0.20	0.45	0.24	622.67	0.80	7.00	27.20
	3	24.05	9.07	209.90	2.94	5.39	1.05	0.17	0.47	0.30	662.00	0.80	7.00	27.20
	4	24.97	9.21	168.33	2.88	4.86	1.58	0.26	0.45	0.23	638.00	0.80	7.00	27.20
22.08.2005	1	23.57	7.33	133.53	2.89	11.11	0.14	0.40	0.72	0.15	226.17	0.00	7.00	30.70
	2	22.96	9.08	61.70	2.92	10.42	0.15	0.58	0.38	0.03	276.00	0.00	7.00	30.70
	3	23.66	9.26	298.67	2.96	8.41	0.29	0.41	0.68	0.26	268.67	0.00	7.00	30.70
	4	24.52	9.49	103.97	2.93	10.28	0.15	0.33	0.41	0.14	266.67	0.00	7.00	30.70
05.09.2005	1	22.22	9.36	102.48	2.85	11.12	1.25	0.45	0.51	0.11	400.00	0.00	9.00	19.00
	2	21.37	9.12	69.60	2.91	10.84	1.94	0.20	0.31	0.03	405.33	0.00	9.00	19.00
	3	22.00	9.06	289.13	2.93	7.40	2.17	0.18	0.35	0.12	410.00	0.00	9.00	19.00
	4	21.95	9.38	341.07	2.96	8.55	2.80	0.12	0.48	0.25	426.00	0.00	9.00	19.00

19.09.2005	1	19.23	9.38	140.47	2.95	8.77	2.11	0.11	0.26	0.05	451.33	0.00	8.00	26.50
	2	20.02	9.44	107.43	2.96	13.69	1.88	0.14	0.27	0.06	447.33	0.00	8.00	26.50
	3	20.26	9.24	39.37	2.98	13.10	2.24	0.24	0.24	0.06	455.33	0.00	8.00	26.50
	4	20.58	9.55	48.87	2.98	21.35	2.39	0.27	0.42	0.05	446.00	0.00	8.00	26.50
03.10.2005	1	17.31	9.60	105.93	3.01	11.52	0.75	0.78	0.21	0.01	449.33	26.00	10.00	15.40
	2	18.05	9.22	147.87	2.93	7.68	1.68	0.58	0.22	0.02	572.00	26.00	10.00	15.40
	3	18.30	9.43	227.07	2.97	5.92	1.16	0.64	0.27	0.02	615.33	26.00	10.00	15.40
	4	18.30	9.42	335.20	2.90	8.47	1.30	0.62	0.34	0.04	554.67	26.00	10.00	15.40
19.10.2005	1	12.82	9.89	239.40	2.97	8.69	1.96	0.23	0.37	0.04	593.33	0.00	9.00	6.30
	2	13.55	9.94	238.40	2.96	8.40	1.90	0.19	0.25	0.01	598.00	0.00	9.00	6.30
	3	13.53	9.76	165.73	3.02	6.35	1.94	0.29	0.21	0.02	593.33	0.00	9.00	6.30
	4	12.83	9.86	343.93	2.88	9.25	1.53	0.42	0.25	0.02	584.67	0.00	9.00	6.30

APPENDIX C

MATLAB SCRIPT THAT USED IN ANN TRAINING

Script that Has One Hidden Layer

```
echo off
fortr={'bfg';'br';'cgb';'cgf';'cgp';'gdm';'gdx';'lm';'oss';'rp';'scg'};
trspaces={" ";' ';'" ";'" ";'" ";'" ";'" ";'" ";'" ";'" ";'" };
[numf1 numf2]=size(fortr);
tn=(norm01(TarAll));
% spaces1={'      '};
% spaces2={'    '};
r=0;
for all=1:1
    pn = eval(char(strcat(['(norm01(Originp',[num2str(all,'%02d')],[')';'])]));
    [R,Q] = size(pn);
    iitr = 1:30;
    iival = 31:45;
    iitst = 46:1:60;
    validation.P = pn(:,iival);
    validation.T = tn(:,iival);
    testing.P = pn(:,iitst);
    testing.T = tn(:,iitst);
    ptr = pn(:,iitr);
    ttr = tn(:,iitr);

    for y= 1:numf1 % defines which training function will be used... after '_' shows which input
matrix used.
        DizinYarat(y)=strcat(['mkdir
.nettrain'],[fortr(y)],['_'],strcat([num2str(all,'%02d')],[trspaces(y)],['_']));
        DizinYaratti(y,:)=char(DizinYarat(y));
        eval(DizinYaratti(y,:));
    end
```

```

for j= 1:numf1 %changing the training function
    for i=1:99 % changing number of hidden neurons in the net...
        if all==1

NetOlustur(i)=strcat(['nettrain'],[fortr(j)],[num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[
'newff(minmax(ptr),[' , ...
    num2str(i,'%02d')],[ '
1],{'logsig','logsig'},'train',[fortr(j)],[ '];],[trspaces(j,:),[trspaces(j,:)]);
    NetOlusturdu(i,:)=char(NetOlustur(i));
    eval(NetOlusturdu(i,:));
elseif all==2 | all>=8

NetOlustur(i)=strcat(['nettrain'],[fortr(j)],[num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[
'newff(minmax(ptr),[' , ...
    num2str(i,'%02d')],[ '
1],{'logsig','logsig'},'train',[fortr(j)],[ '];],[trspaces(j,:),[trspaces(j:)],spaces2(1,:));
    NetOlusturdu(i,:)=char(NetOlustur(i));
    eval(NetOlusturdu(i,:));
else

NetOlustur(i)=strcat(['nettrain'],[fortr(j)],[num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[
'newff(minmax(ptr),[' , ...
    num2str(i,'%02d')],[ '
1],{'logsig','logsig','logsig'},'train',[fortr(j)],[ '];],[trspaces(j,:),[trspaces(j,:)]);
    NetOlusturdu(i,:)=char(NetOlustur(i));
    eval(NetOlusturdu(i,:));
end

NetAyarla1(i)=strcat(['nettrain'],[fortr(j)],
num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]),['.trainParam.epochs=10000;',...
    'nettrain'],[fortr(j)],
num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]),['.trainParam.goal=0.000;',...
    [trspaces(j,:),[trspaces(j,:)]);
NetAyarlandi1(i,:)=char(NetAyarla1(i));
eval(NetAyarlandi1(i,:));
% asagidaki Egit te ptr, ttr, validation, testing degisecek...

```



```

Egiti(i)=strcat(
['[nettrain]',[fortr(j)],[num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[' ,tres'],[fortr(j)],...
[num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[ '=train(nettrain)',[fortr(j)],
[num2str(i,'%02d')],[ '_'],...
strcat([num2str(all,'%02d')]),[' ,ptr,ttr,[],[],validation,testing);
clf;close(gcf);',[trspaces(j,:),[trspaces(j,:),[trspaces(j,:)]];
Egitti(i,:)=char(Egiti(i));
eval(Egitti(i,:));
% asagidaki Dene de pn, OrgOutAll degisecek...
Dene(i)=strcat(['[restrain]',[fortr(j)], num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),
[''],'=sim(nettrain' ],[fortr(j)], ...
num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[' ,pn'],'[Prestrain]',[fortr(j)],
num2str(i,'%02d')],[ '_'],...
strcat([num2str(all,'%02d')]),['= unnorm01(TarAll
'],'[restrain]',[fortr(j)],[num2str(i,'%02d')],[ '_'],...
strcat([num2str(all,'%02d')]),[';'],[trspaces(j,:),[trspaces(j,:),[trspaces(j,:),[trspaces(j,:)]];
Denedi(i,:)=char(Dene(i));
eval(Denedi(i,:));
%
HesaplaCizdir3(i)=strcat(['fig'],[fortr(j)],
[num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),['Reg=figure;'],...
['[m'],[fortr(j)],num2str(i,'%02d')', [' ,b'],[fortr(j)],
num2str(i,'%02d')],[ '_'],[fortr(j)],num2str(i,'%02d')'],...
[']=postreg(Prestrain' ],[fortr(j)], ...
num2str(i,'%02d')],[ '_'],strcat([num2str(all,'%02d')]),[' ','TarAll'],[ '_'],[ 'title("Regression
analysis for (train')],...
fortr(j),[ '_'],strcat([num2str(all,'%02d')]),[ '_'],[ '# of HNs:',[num2str(i,'%02d')],[ '_'],[ '
'],';'],trspaces(j)],...
trspaces(j),trspaces(j),trspaces(j),trspaces(j),trspaces(j));
Hesapladi3(i,:)=char(HesaplaCizdir3(i));
eval(Hesapladi3(i,:));

regresValue=eval(char(strcat(['r'],[fortr(j)],num2str(i,'%02d'))));

if regresValue>=0.75

```

```

Kaydet3(i)=strcat(['print("-r300","-
djpeg",".\nettrain'],[fortr(j)],['_'],strcat([num2str(all,'%02d')]),...
    ['\fig'],[fortr(j)],num2str(i,'%02d'),'Reg'' ');'],'close(gcf);'],trspaces(j),trspaces(j));
Kaydetti3(i,:)=char(Kaydet3(i));
eval(Kaydetti3(i,:));

HesaplaCizdir1(i)=strcat(['\fig'],[fortr(j)],
num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]),...
    ['=figure; plot(trres'],[fortr(j)], num2str(i,'%02d'), [' _'],strcat([num2str(all,'%02d')]),...
    ['.epoch,trres'], [fortr(j)], num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]), ...
    ['.perf,"r", trres'],[fortr(j)],num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]), ...
    ['.epoch,trres'], [fortr(j)], num2str(i,'%02d'), [' _'],strcat([num2str(all,'%02d')]),...
    ['.vperf,"g",trres'],[fortr(j)],num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]),...
    ['.epoch,trres'], [fortr(j)], num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]),...
    ['.tperf,"-b"];'],'xlabel("# of epoch");'],...

[trspaces(j,:),[trspaces(j,:),[trspaces(j,:),[trspaces(j,:),[trspaces(j,:),[trspaces(j,:),[trspaces(j,:)]];
Hesapladi1(i,:)=char(HesaplaCizdir1(i));
eval(Hesapladi1(i,:));

HesaplaCizdir11(i)=strcat(['title("MSE vs. # of Epochs (train'
,[fortr(j)],['_'],strcat([num2str(all,'%02d')]),...
    [''],[' (# of HNs:'],num2str(i,'%02d'),[')'];'],...
    [trspaces(j)],['legend("Training","Validation","Test",0); ylabel("Mean Squared Error");']);
Hesapladi11(i,:)=char(HesaplaCizdir11(i));
eval(Hesapladi11(i,:));

Kaydet1(i)=strcat(['print("-r300","-
djpeg",".\nettrain'],[fortr(j)],['_'],strcat([num2str(all,'%02d')]),['\fig'],[fortr(j)], ...

num2str(i,'%02d'),['_'],strcat([num2str(all,'%02d')]),[''];close(gcf);'],[trspaces(j,:),[trspaces(j,:)]];
Kaydetti1(i,:)=char(Kaydet1(i));
eval(Kaydetti1(i,:));

DATE=[1:1:60];

```

```

HesaplaCizdir2(i)=strcat(['fig'],[fortr(j)], num2str(i,'%02d'),'OwP'], ['=figure;
plot(DATE,Prestrain'],[fortr(j)],...
    num2str(i,'%02d'),'_',strcat([num2str(all,'%02d')]),['.-r"'],[DATE,TarAll,"-b");'],...
    ['legend("Predicted","Observed",0);'],[ylabel("DO (mg/l)");],[xlabel("Date");'],...
    ['title("Pred & Obs DO (mg/l) vs. Date w/ train
'],[fortr(j)],['_'],strcat([num2str(all,'%02d')]),...
    ['# of HNs:',num2str(i,'%02d'),'');],[trspaces(j)],[trspaces(j)],[trspaces(j)];
Hesapladi2(i,:)=char(HesaplaCizdir2(i));
eval(Hesapladi2(i,:));
% kontrol et... OwP ne olacak?
Kaydet2(i)=strcat(['print("-r300","-
djpeg",".\nettrain'],[fortr(j)],['_'],strcat([num2str(all,'%02d')]),['\fig'],[fortr(j)], ...
    num2str(i,'%02d'),'OwP"'); close(gcf);],[trspaces(j)],[trspaces(j)];
Kaydetti2(i,:)=char(Kaydet2(i));
eval(Kaydetti2(i,:));

end

% hesaplacizdir3 bolunecek buradan r alinacak, kendilerinin regression kısmi tum resim
cizimlerinin
% ustune yerlestirilecek ki bosuna anlamsiz cizimler icin zaman harcanmasin..
Kaydet4(i)=strcat(['save .\nettrain'],[fortr(j)],['_'],strcat([num2str(all,'%02d')]),
['\nettrain'],[fortr(j)],num2str(i,'%02d'),...
    ['_',strcat([num2str(all,'%02d')]),['net* Prestrain* restrain* trres*'],[' m'],fortr(j),['*'],['
b'],fortr(j)],...
    [' r'],fortr(j),['*'],[';close(gcf);'],['clear net* Prestrain* restrain* trres* fig* m'],fortr(j),['*
b'],fortr(j)],...
    ['*
r'],fortr(j),['*'],trspaces(j),trspaces(j),trspaces(j),trspaces(j),trspaces(j),trspaces(j),trspaces(j),trspaces(
j)];
Kaydetti4(i,:)=char(Kaydet4(i));
eval(Kaydetti4(i,:));

end

end

end

```

Normalization Fuction (norm01.m)

```
% Normalize the given column vector to the range 0-1 using
%  $x\_norm[i]=(x[i]-x\_min)/(x\_max-x\_min)$ 
function [D]=norm01(A)
[Q R]=size(A);
tr=zeros(Q,R);
for i=1:Q
    alt=min(A(i,:));
    ust=max(A(i,:));
    for j=1:R
        tr(i,j)=(A(i,j)-alt)/(ust-alt);
    end
end
D=tr';
```

Inverse Normalization Fuction (unnorm01.m)

```
% Rewind the Normalized matrix from the given source matrix unnormalize.
function [D]=unnorm01(A,B)
[Q R]=size(A);
tr=zeros(Q,R);
for i=1:Q
    alt=min(A(i,:));
    ust=max(A(i,:));
    for j=1:R
        tr(i,j)=(B(i,j)*(ust-alt)+alt);
    end
end
D=tr'
```