DECISION MAKING IN ENGINEERING PREDICTION SYSTEMS

by

HAKAN I. YASARER

B.S., Mustafa Kemal University, 2004
M.S., Kansas State University, 2010

AN ABSTRACT OF A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Civil Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2013

Abstract

Access to databases after the digital revolutions has become easier because large databases are progressively available. Knowledge discovery in these databases via intelligent data analysis technology is a relatively young and interdisciplinary field. In engineering applications, there is a demand for turning low-level data-based knowledge into a high-level type knowledge via the use of various data analysis methods. The main reason for this demand is that collecting and analyzing databases can be expensive and time consuming. In cases where experimental or empirical data are already available, prediction models can be used to characterize the desired engineering phenomena and/or eliminate unnecessary future experiments and their associated costs. Phenomena characterization, based on available databases, has been utilized via Artificial Neural Networks (ANNs) for more than two decades. However, there is a need to introduce new paradigms to improve the reliability of the available ANN models and optimize their predictions through a hybrid decision system. In this study, a new set of ANN modeling approaches/paradigms along with a new method to tackle partially missing data (Query method) are introduced for this purpose. The potential use of these methods via a hybrid decision making system is examined by utilizing seven available databases which are obtained from civil engineering applications. Overall, the new proposed approaches have shown notable prediction accuracy improvements on the seven databases in terms of quantified statistical accuracy measures. The proposed new methods are capable in effectively characterizing the general behavior of a specific engineering/scientific phenomenon and can be collectively used to optimize predictions with a reasonable degree of accuracy. The utilization of the proposed hybrid decision making system (HDMS) via an Excel-based environment can easily be utilized by the end user, to any available data-rich database, without the need for any excessive type of training.

DECISION MAKING IN ENGINEERING PREDICTION SYSTEMS

by

HAKAN YASARER

B.S., Mustafa Kemal University, 2004

M.S., Kansas State University, 2010

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Civil Engineering

College of Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2013

Approved by;

Major Professor
Yacoub M. Najjar

Abstract

Access to databases after the digital revolutions has become easier because large databases are progressively available. Knowledge discovery in these databases via intelligent data analysis technology is a relatively young and interdisciplinary field. In engineering applications, there is a demand for turning low-level data-based knowledge into a high-level type knowledge via the use of various data analysis methods. The main reason for this demand is that collecting and analyzing databases can be expensive and time consuming. In cases where experimental or empirical data are already available, prediction models can be used to characterize the desired engineering phenomena and/or eliminate unnecessary future experiments and their associated costs. Phenomena characterization, based on available databases, has been utilized via Artificial Neural Networks (ANNs) for more than two decades. However, there is a need to introduce new paradigms to improve the reliability of the available ANN models and optimize their predictions through a hybrid decision system. In this study, a new set of ANN modeling approaches/paradigms along with a new method to tackle partially missing data (Query method) are introduced for this purpose. The potential use of these methods via a hybrid decision making system is examined by utilizing seven available databases which are obtained from civil engineering applications. Overall, the new proposed approaches have shown notable prediction accuracy improvements on the seven databases in terms of quantified statistical accuracy measures. The proposed new methods are capable in effectively characterizing the general behavior of a specific engineering/scientific phenomenon and can be collectively used to optimize predictions with a reasonable degree of accuracy. The utilization of the proposed hybrid decision making system (HDMS) via an Excel-based environment can easily be utilized by the end user, to any available data-rich database, without the need for any excessive type of training.

Table of Contents

List of Figures

List of Tables

Acknowledgement

I would like to thank many individuals who have contributed, either directly or indirectly, to the completion of this dissertation at Kansas State University.

I would like to express my deepest and sincere appreciation to Professor Yacoub Najjar for his support and encouragement throughout the course of this work, which would not have been possible without his generous help and guidance. I am indebted to Dr. Najjar for his patience in reviewing this manuscript. I am also very grateful to my committee members, Professor Hayder A. Rasheed, Asad Esmaeily, and Medhat M. Morcos for their advice and valuable comments. Many thanks to Professor Gary Brase for accepting to chair my final examination and for his friendly attitude.

Financial support received from Kansas State University Transportation Center to help conduct this research study is gratefully acknowledged. Thanks are also due to the Civil Engineering Department at Kansas State University for graduate research and teaching assistantships provided. In addition, I owe great appreciation to the entire Department of Civil Engineering, which includes faculty, staff, and graduate students.

I have been thankful to know the Cochran family, and especially the deceased Mary Lee Cochran for the motherly friendship and caring. Also, I am appreciative to Wayne Goins, director of jazz studies, for the valuable life and music lessons he taught me.

My deepest gratitude goes to my family for their unflagging love and support throughout my life. Additional gratitude goes to my family in-law: James and Marlene Witthaus. Most importantly I am indebted to my wife, Lindsey M. W. Yasarer who has inspired me to do the best of everything. It would not have been possible without her unconditional love, consistent support, and patience during the research and writing of this dissertation.

CHAPTER 1

1-INTRODUCTION

1.1 Overview

The digital revolution has increased the capability to capture, process, store, distribute, and transmit information worldwide. Large databases are increasingly available over the internet and cover a wide range of topics. Their expanding usage in different areas will continue to grow with significant progress in computing technologies. Even though access to databases has become relatively simple, raw data can rarely be used directly. Its full value is driven from (a) the ability to extract information, which is useful for decision support or exploration and (b) understanding the phenomenon governing the data source. The overall process of knowledge discovery in databases consists of turning low-level data into high-level knowledge.

The development of computer hardware and software has inspired new approaches for data processing and analysis. Soft computing has been recognized as a low cost solution yielding analysis tools to solve complex problems in many areas of engineering. During the last century, data processing applications have increasingly been developed and used to analyze various databases in various research areas. In recent years, data-based modeling has become more popular in engineering applications. The Artificial Neural Networks (ANNs) approach, which is considered one component of soft computing, is one of the most reliable and commonly used knowledge discovery methods in databases due to its capability to directly learn complex nonlinear relationships.

ANNs is a mathematical or computational model that attempts to emulate the structure and/or functional aspects of biological neural networks. ANNs provide an analytical alternative to classical mathematics and traditional techniques which are often limited by assumptions of normality, linearity, variable individuality, etc. Unlike conventional computing techniques, soft computing models focus on partial exactness through an approximation with a tolerance of imprecision. Soft computing models exploit biological processes, predicate logic, the partial belongingness concept, parallel processing and techniques which mimic the human mind as well as nature (Lav et. al, 2009). ANNs-based material modeling approach has received increasing interest in the engineering area during the past 20 years. Essentially, ANNs approach is considered to be the best function approximation technique that is well suited for proper material behavior characterization. In a typical modeling process, ANNs-based model is trained to attain a specific knowledge through training or retraining via a mathematically-based process. As a result, the trained model stores the extracted knowledge, features embodied in the database, within its connection weights. ANNs possess the following unique advantages in information processing tasks:

1. ANNs are capable of directly learning complex nonlinear relationships from a large body of datasets without the need for any simplifying assumptions;
2. Model prediction accuracy can be improved by adding new training datasets which can internally adjust the model's connection weights in order to capture new features hidden within the new datasets;
3. ANNs have the ability to extract information from incomplete or partially incorrect datasets;
4. ANNs can be used to develop general purpose models to characterize various responses of material behavior;
5. ANNs can derive relationships and associations directly from the experimental data without the need for much theory support;
6. ANNs can be used to examine the effect of an individual input on the output parameter without the need to physically conduct additional experiments.

The most commonly used ANNs in engineering applications are multilayer backpropagation networks. A recent study by Yasarer (2010) has shown that the application of backpropagation ANNs has proven to be an effective modeling method for material characterization in engineering applications. The success of ANNs approach has been validated on many engineering applications reported in the literature [i.e. Nazarian (2004), Tutumluer and Seyhan (1998), and Meier and Rix (1994)]. Backpropagation ANNs approach has been successfully used in engineering applications by Ghaboussi et al 1990, 1991, 1994; Najjar and Basheer (1996), Najjar et al. (1999), Najjar and Ali (1999); and Yasarer and Najjar (2010).

It is essential that the ANNs approach is continually modified and improved so that the optimal network for individual phenomena can be optimized so that the ANNs approach can be applied to increasingly challenging problems and complex datasets.

1.2 Problem Statement

Many research investigations have been conducted to find alternative methods that can generate efficient, rational, and practical prediction models. Among these approaches, ANNs approach became popular due to its efficient and reliable results. However, the need for improving the statistical accuracy of this approach has become essential. For this reason, two of the questions that this research attempted to answer are:

1. "Can we improve or optimize the prediction accuracy of ANN models?"
2. "Can we develop more than one ANN model for a single database using different methodologies?"

A secondary issue related to ANN modeling is incomplete datasets. Datasets with missing variables are called incomplete datasets. In order for ANN models to be utilized, complete input parameters have to be provided. Otherwise the models are not valid and cannot be used in any circumstances. The problem of incomplete datasets is very common within engineering databases. Often, incomplete datasets cannot be used for ANN modeling and when incomplete data is removed, the resulting sample of complete cases may be too small to obtain statistically significant trends. There is a wide variety of methods for handling missing data, which vary a great deal in their mathematical complexity. The need for a simple solution is apparent.

Finally, there is a need to integrate multiple modeling frameworks into one decision-support system. Typically, engineering prediction models have one solution that is adapted directly by the user. Accordingly the engineer or the scientist does not have any options to choose from. In other words, the provided output is the only solution available to the user, even though most engineering judgments should be made with guidance from multiple options. Providing more than one option or output within a framework for evaluating the various options is essential to making reliable decisions.

1.3 Objectives

The overall objective of this study is to explore and expand the potential use of Artificial Neural Network modeling for seven civil engineering databases along with a new method for partially missing datasets in databases. According to the stated problems in Section 1.2, the overall scope for seven databases included the following tasks:

1. Develop a static ANN network. The potentials of static ANN are investigated. Effect of input parameters on the output based on the performance evaluation criteria (statistical accuracy measures and graphical evaluation) is utilized to determine the optimal architecture of the neural network models.

2. Utilize the initial estimates generated by the static ANN network in step 1, to develop the desired Feedback-ANN Network Model. The datasets used in the model development of the static ANN are also used for the model development of Feedback-ANN Network Model. The optimal network is determined based on similar statistical measures.

3. Use the initial estimates generated by the model in step 1, develop an Auto-associative network. ANN modeling criteria is similarly followed in this step. This network provides predictions of inputs and output together.

4. Convert the static model database into a dynamic model database. Utilizing the new database and the initial estimates from the static ANN network, develop an appropriate Dynamic-sequential network. Similarly, the optimal internal structure for the Dynamic-sequential network model is determined based on the same statistical accuracy measures.

5. Develop a Query method application to populate partially missing datasets as well as to generate predictions by using the entire database. First, the Query method application is developed with the datasets and tested on validation datasets. Second, the Query method application is developed and tested using all available datasets.

6. Develop a hybrid decision making system (HDMS) with a user-friendly interface that integrates the predictions from all ANN models as well as the Query method application developed in Steps 1 to 5. The developed HDMS interface will be designed to produce a single weighted prediction value along with the most likely prediction range.

As stated before, the main objectives of this study are to explore and improve the ability of backpropagation ANNs along with a method to replace missing variables in datasets through a hybrid decision making system. To achieve these objectives, the listed tasks are followed for seven databases sequentially. In the following chapters, the databases used, the new ANN approaches/paradigms and the Query method application along with their development phases and their corresponding prediction accuracy measures will be discussed in details.

1.4 Organization of the Dissertation

Chapter 1- Introduction: This chapter presents a brief discussion on ANNs-based modeling approach and advantages of using ANN modeling. Also, brief summaries of the contents of each chapter are presented.

Chapter 2- Background: This chapter contains a brief literature review related to the research conducted in this study. Several relevant publications on ANN modeling approaches that contributed significance to this research study are highlighted.

Chapter 3- Artificial Neural Network: This chapter discusses the aspects of ANN computational algorithms. Basic definition, elements, and Backpropagation learning algorithm used in ANN approach are discussed in details. Statistical prediction accuracy measures used to identify the best performing ANN models are also defined in this chapter.

Chapter 4- Database Description: All the databases to be used for the proposed ANN modeling approaches and the Query method applications are described in details. The description of each parameter is presented and explained.

Chapter 5- **Static ANN Network:** This chapter defines the fundamentals of the static ANN network as well as the model development stages for the seven databases. Static ANN networks for each database are discussed in details. Corresponding graphical results and their statistical accuracy measures for all seven databases are presented at the end of the chapter.

Chapter 6- Feedback ANN Network: This chapter discusses the facts about the procedure and the significance of Feedback ANN. The model development phases for the Feedback ANN network are argued in details. Prediction accuracy comparisons in terms of graphical and statistical accuracy measures for the developed ANN are presented in this chapter. Prediction improvement tables are presented at the end of Chapter 6.

Chapter 7- **Auto-associative Network:** The usage of the Auto-associative network in other engineering areas is explained, and the model development process for each database is presented in their relevant sections with details. Similarly, model accuracy plots and the statistical accuracy tables are given at the end of the chapter.

Chapter 8- **Dynamic-sequential Network:** This chapter outlines the essentials of the Dynamic-sequential network as well as the model development stages for the seven databases. Dynamic-sequential networks for each database are presented in details. Corresponding graphical results and their statistical accuracy measures for all seven databases are presented at the end of the chapter. The improvement performance of the Dynamic-sequential network is highlighted.

Chapter 9- **Query Method:** This chapter states the basics of the Query method, and then the application calculation procedures are presented. Corresponding graphical results and their statistical accuracy measures for all seven databases are similarly presented at the end of the chapter. Some screen-shots from the Excel-based application developed to produce the Query method are also shown.

Chapter 10- Hybrid Decision Making System: Integration of all ANN approaches and the Query method application for seven databases are described in this chapter. Sample screen-shots for the hybrid decision making system for 3 databases are presented. Recommended value statistics are also placed at the end of the chapter.

Chapter 11- Summary, Conclusion, and Recommendations: Summary of the research work performed in this study and major conclusions obtained are presented in this chapter. Recommendations for future research studies are also presented.

CHAPTER 2

2- BACKGROUND

2.1 ANN Modeling Approach

During the 1990s a new period of engineering material characterization emerged with the utilization of the Artificial Neural Networks (ANNs) approach to properly characterize the behavior of geo-materials, such as soil, concrete, Portland cement concrete (PCC) pavement, asphalt concrete (AC) (i.e. Ghaboussi et al. (1991), Najjar and Basheer (1996), and Najjar et al. (1999)). Material modeling is a fundamental phenomenon in engineering research and practice. A model is typically developed to describe the material constitutive/mechanical behavior under certain boundary conditions. Material models serve as the basis for numerical calculations and guidance for analyzing, designing, constructing and rehabilitating structures, including the material. In this chapter, significant studies that guided and which are relevant to the research presented in this dissertation are presented to provide beneficial background information.

Neural network approach were applied for automated inversion of dispersion curves from the spectral analysis of surface waves (SASW) test data on a four-layer AC pavement by Gucunski and Krstic (1996). SASW method is a seismic technique for in situ evaluation of elastic moduli and layer thicknesses for layered systems, such as pavements and soils. The objective of the SASW test is to obtain the experimental dispersion curve and, through an inversion procedure, obtain the profile of elastic moduli of the layered system. The inversion process in practice uses an average of dispersion curves for different receiver spacing. Results of theoretical studies indicate that differences in dispersion curves for various spacing are a result of interference of a number of body and surface waves. The development and application of neural networks to perform the inversion procedure for SASW testing of asphalt concrete (AC) pavements was proposed by Gucunski and Kristic (1996). The most important feature of the developed network is that training of the network was done by the dispersion curves for individual receiver spacings. The training set consists of dispersion curves for seven receiver spacings and 78 dimensionless frequencies, while output is presented by elastic moduli and layer thicknesses of a four-course AC pavement. The best developed model is a five-layer back-propagation model with jumps. The model perfectly predicts thicknesses and shear wave velocity for all layers, except the thickness of the sub-base layer. The obtained neural network model is compared to the previously developed model for back-calculation of moduli from the SASW test based on the averaged dispersion curve. Although both approaches can accurately define profiles, each has some advantages in evaluation of the thickness of the subbase.

The use of intelligent and soft computing techniques in the field of geomechanical and pavement engineering has emerged during 2000s. A probabilistic approach to the solution of inverse problems in nondestructive testing and engineering geophysics was applied by Hadidi et al. (2007). Interpretation of geophysical data often requires the solution of an inverse problem. There are two general approaches to the solution of inverse problems, deterministic and probabilistic approaches. Usually, in engineering geophysics inversion is carried out using a deterministic approach, where a single set of results is identified as the interpreted outcome. In complex inverse problems the deterministic solution process is often guided by an interpreter, who uses his knowledge, experience, or judgment to guide the process. However, it assumes the uncertainties in data and quantitative models are negligible. A technique for the evaluation of the probabilistic solution using Monte Carlo Markov Chains (MCMC) with Neighborhood Algorithm (NA) approximation is introduced and explained in Hadidi's paper (2007). The study demonstrates an application of MCMC with NA in the health monitoring of transportation infrastructure using non-destructive testing (NDT) Hadidi et al. (2007).

Hsu et al. (1995) presented a new procedure (entitled linear least squares simplex, or LLSSIM) for identifying the structure and parameters of three-layer feed forward ANN models and demonstrates the potential of such models for simulating the nonlinear hydrologic behavior of watersheds. The nonlinear ANN model approach is shown to provide a better representation of the rainfall-runoff relationship of the medium-size Leaf River basin near Collins, Mississippi, than the linear ARMAX (autoregressive moving average with exogenous inputs) time series approach or the conceptual SAC-SMA (Sacramento soil moisture accounting) model. Because the ANN approach presented here does not provide models that have physically realistic components and parameters, it is by no means a substitute for conceptual watershed modeling. However, the ANN approach provides a practical and effective alternative to the ARMAX time series approach for developing input-output simulation and forecasting models in situations that do not require modeling of the internal structure of the watershed.

Another example of a successful ANN application is a study about modeling hydration of cementitious materials, established by Riding et al. (2012). The study presented the development of a model for predicting the adiabatic temperature development of concrete mixtures based on material properties (for example, cement chemistry and fineness and supplementary cementitious materials (SCM) chemistry), mixture proportions, and chemical admixture types and dosages. The model was developed from 204 semi-adiabatic calorimetry results and validated from a separate set of 58 semi-adiabatic tests. The final model provided a useful tool to assess the temperature development of concrete mixtures and thereby enable the prevention of thermal cracking and delayed ettringite formation in concrete structures.

The ANN modeling approach has not only been limited to engineering databases, it has also been utilized by other fields, such as psychology and neuroscience where the methodology of ANN was derived. For example, Levine (2002) used the neural network modeling in several areas of psychology including sensory processes, short-term memory, pre-attentive vision, attention, and code development; control of individual movements and movement sequences; classical and operant conditioning, and reinforcement learning; involvement of several brain areas in cognitive-emotional interactions; categorization and classification; decision making; language understanding; reasoning and analogy; mental and cognitive disorders; and a few areas of social psychology. One simple example is given of the process of generating equations for a neural network model, with the terms of the equations being motivated by the psychological operations that those terms describe.

2.2 Auto-Associative Network Approach

Auto-associative network has also been widely utilized in other engineering areas. A detailed definition of the Auto-associative network is outlined by Daszykowski et al. (2003). Auto-associative neural networks (AANNs) provide an elegant method for data compression and visualization, which are subjects that have always generated a great deal of excitement in engineering and scientific fields. Since multidimensional datasets are difficult to interpret and visualize, much attention has been focused on how to compress them efficiently. Usually, the compression of dimensionality is considered as the first step of exploratory data analysis. Here, we focus our attention on auto-associative neural networks as a tool for data compression and visualization. AANNs can deal with linear and nonlinear correlation among variables, what makes them a very powerful tool in exploratory data analysis. In the literature, AANNs are often referred as nonlinear principal component analysis (PCA), and due to their specific structure they are also known as bottleneck neural networks. In Daszykowski et al. (2003), AANNs are discussed in detail and different training modes are described and illustrated on real examples. The usefulness of AANNs for nonlinear data compression and visualization purposes is proven with the aid of chemical data sets, being the subject of analysis. The comparison of AANNs with well-known PCA is also presented.

In another example, the neural auto-associative technique has been applied to image compression in a study by Basso and Kunt (1992). Particular attention was given to the preprocessing stage in image creation. The validity of some of the theoretical results is discussed and an experimental study of the mapping capabilities of the network based on a nonlinear parameterized activation function is presented. In order to test the image reconstruction capabilities of the neural technique, comparisons with more traditional image processing tools such as Karhunen-Loeve Transform (KLT) are shown. A parallel implementation of a linear version of the neural technique on the Associative String Processor (ASP) machine is presented. Despite the linear structure of the ASP and the use of fixed arithmetic for the implementation, promising results are shown in terms of learning speed and quality of the reconstructed images.

In another study by Marseguerra and Zoia (2005) Auto-Associative Neural Networks (RAANN) are applied to a series of signals produced by the Halden simulator of the 1200 MWe (megawatt of electricity) BWR (Boiling water reactor) Forsmark-3 plant in Sweden. The applications concern the:

- correction of drifts and gross errors in sensors, for diagnostic and control purposes,
- cluster analysis, to individuate a failed component and the intensity of the failure,
- forecasting system signals, for safety or economic purposes,
- and reconstruction of unmeasured signals (virtual sensors).

In the accomplishment of the above, the geometric interpretation of the mapping performed by the network has provided a reasoned choice of the most critical free parameter, i.e., the number of hidden nodes of the bottleneck layer, thus allowing a deep understanding of the network functioning and also avoiding the traditional and troubling procedure of selection by trial-and-error. The theoretical basis of this analysis is founded on the idea of dimension and in particular of fractal dimension, which has been used as a numerical estimator of the factors.

Desjardins et al. (2006) have proposed an Auto-associative neural network to model the classification processes and the selective recovery of information to perform the matching task. Neural network is an important paradigm that has received attention from the society of researchers in information retrieval, especially the auto-associative neural networks. These networks are capable of discovering patterns of terms among documents. The unique layer network is trained with the documents of the collection and then used to recall the most relevant documents to specific queries. The model has been tested on a TREC ("Text Retrieval Conference") sub-collection. The results are compared against the vector space model. The experiment shows higher levels of global precision and recall. The recall-precision curves show an important improvement on the precisions for the low levels of recall, which indicates a faster retrieval of the most highly relevant documents. With this study, the strength of the Auto-associative neural network has been shown in information retrieval for general collections Desjardins et al. (2006).

2.3 Missing Data Adjustments

Among highly complex modeling techniques and algorithms, the missing data problems have become wide-spread. Missing data adjustments for partially scaled variables were studied in 1987 by Little and Sue. Missing data is a pervasive problem in sample surveys. Two common strategies for dealing with the problem are direct analysis of the incomplete data and imputation. In the first approach, the missing values are left as gaps in the data set, identified by special missing data codes, and the treatment of missing data is deferred to the analysis stage. Given data in this form, most statistical analysis packages discard cases that contain incomplete information (complete-case analysis) or restrict attention to cases where the variable of interest is observed (available-case analysis). Little and Sue propose (1987) two methods for handling missing data on a set of partially-scaled variables, one based on maximum likelihood for a general model for mixed continuous and categorical variables, and one based on imputation from a matched complete record. Preliminary empirical work based on data from the Survey of Income and Program Participation (SIPP) shows that both of these methods have promise.

Gheyas and Smith (2009) proposed a non-parametric multiple imputation algorithm (GMI) for the reconstruction of missing data, based on Generalized Regression Neural Networks (GRNN). They compare GMI with popular missing data imputation algorithms: EM (Expectation Maximization) MI (Multiple Imputation), MCMC (Markov Chain Monte Carlo) MI, and hot deck MI. A separate GRNN classifier is trained and tested on the dataset imputed with each imputation algorithm. The imputation algorithms are evaluated based on the accuracy of the GRNN classifier after the imputation process. The effectiveness of the proposed algorithm was showed on twenty-six real datasets.

2.4 Decision Making in Engineering Systems

The importance of decision making in system engineering was studied by Roth (2007). Roth discusses that engineering design is inherently a social activity, as are all applied disciplines. Roth's discussion strives to illuminate the human biases present in modeling reality, the types of decision making often employed, the need for a holistic systems approach, the need to consider the human side of engineering, along with the complexities involved, and the need for collaboration to solve today's toughest problems. Suggestions for possible engineering problem solving methodologies involving human tendencies are introduced in Roth's study.

A review paper by Ascough et al. (2008) discusses the importance of decision making in ecological and environmental issues. Some of the important highlights associated with this research are: (1) the development of methods for quantifying the uncertainty associated with human input; (2) the development of appropriate risk-based performance criteria that are understood and accepted by a range of disciplines; (3) improvement of fuzzy environmental decision-making through the development of hybrid approaches (e.g., fuzzy-rule-based models combined with probabilistic data-driven techniques); (4) development of methods for explicitly conveying uncertainties in environmental decision-making through the use of Bayesian probability theory; (5) incorporating adaptive management practices into the environmental decision-making process, including model divergence correction; (6) the development of approaches and strategies for increasing the computational efficiency of integrated models, optimization methods, and methods for estimating risk-based performance measures; and (7) the development of integrated frameworks for comprehensively addressing uncertainty as part of the environmental decision-making process.

This brief literature review discussed topics related to the initial applications of ANN in various fields, the development of auto-associative ANN approaches and their applications, the search for solutions to missing data in engineering databases, and some discussion related to decision-making systems. Each of these subtopics is a broad research area utilized by many different disciplines. This review serves as a sample of the available research and a platform from which this research has sprung forth into newly developed modeling methods using the ANN structure.

CHAPTER 3

3- ARTIFICIAL NEURAL NETWORK

3.1 Definition and Elements

3.1.1 Definition

An artificial neural network (ANN) is a method based on the operation of biological neural networks. In other words, it is a simulation of biological neural system. ANN is a mathematical model or computational model that attempts to emulate the structure and/or functional aspects of biological neural networks. The interest in neural networks re-emerged only after some important theoretical results were attained in the early eighties, notably after the discovery of the error back-propagation scheme. Nowadays, artificial neural networks can be most adequately characterized as 'computational models' with particular properties such as the ability to adapt, learn, generalize, cluster or organize data in an operation based on parallel processing. However, many of the mentioned properties can be attributed to existing models for which the neural network approach can be suited better in certain applications. Parallel processing is often described with biological systems. However, there is still so little known about biological systems. Models developed by artificial neural network approach can be identified as oversimplification of the biological systems (Krose and Smagt, 1996). Artificial neural networks are highly interconnected structures consisting of many simple processors (neurons) that perform massively parallel computation for data processing and knowledge representation. ANNs approach is represented by mathematical algorithms designed to imitate methods of information processing and knowledge acquisition of the human brain (Pham 1994). ANNs systems typically consist of the same following basic components:

i. a neuron or node,
ii. an activation function associated to each node,
iii. a real-valued weight associated with each link between two nodes,
iv. a real-valued bias associated with each node,

v. a transfer function,

vi. a propagation rule, and

vii. a learning rule.

The ANNs have generalization capability which is highly dependent on the size of training samples, range of data domain, and density of solution space. Generalization process by an ANNs approach is achieved, very much similar to the human nervous system, by increasing the acquainted knowledge through the use of high number of experimentations.

3.1.2 Elements

The most important element in every ANN architecture is the neuron which is similar to the biological neurons. It is considered as a cell with a built-in activation function connected to other neurons by a set of connections. Main elements of an Artificial Neural Network are the input layer, hidden layer(s), output layer, and connection weights. An example of an ANN structure is depicted in Figure 3.1. Prediction accuracy of the network depends on its interconnected weights. A network usually performs the following three sequential tasks (Najjar et. al, 1996):

a. Input variables fed to the input layer,
b. Processing of information within the hidden layer,
c. Production of outputs at the output layer.

The input layer contains the input nodes and does not perform any mathematical operation. The number of the input nodes is based on input variables which are assumed to influence the output. The number of the input variables affects the performance of the network. Information is received, processed and forwarded to the hidden nodes by the input layer. The hidden layer may contain one or more layers consisting of a set of nodes which processes information within the network body. The hidden layer which is a transition layer between input layer and output layer is the most important element in the network. The hidden layer processes the information passed on from the input layer and feeds it forward towards the output layer. In other words, it facilitates the flow of information between the input nodes and the output node via the connecting links. The accuracy of the developed models is considerably affected by the number of the hidden layers as well as the number of neurons involved within each layer. Connection weights are the interconnecting links between the neurons in sequential layers. Each neuron is connected to every other neuron in the next layer via links which have individual and adjustable connection weights. There are no side connections used in this modeling approach.

3.2 Backpropagation Learning Algorithm

Backpropagation neural networks consist of a number of layers including a specified number of neurons. The input layer includes the input neurons corresponding to parameters which are assumed to affect the outcome of the phenomenon. The output layer consists of the output neuron(s) which represent(s) the solution of the problem. The hidden layer located between the input layer and the output layer is not designed to have any direct contact with the outside environment. It has been shown (Hornik et al., 1989; Funahashi, 1989; Cybenko, 1989; Hartman et al., 1990) that only one layer of hidden units can approximate any function with finitely many discontinuities to arbitrary precision, provided that the activation functions of the hidden units are non-linear (the universal approximation theorem). In most applications, a feed-forward network with a single layer of hidden units is used.

A sigmoidal function which is the most widely used function is where the input passes through to calculate the output of a neuron at the output layer. The calculated outputs are then compared to actual outputs to determine the error which is consequently used for error function determination. Then, the error function is used to adjust the error starting from the connection weights linked with the output, and backward to the input layers. In other words, the generated error by the network is used to adjust the connection weights. The connection weights are initially not known and typically assigned random or specified values. The output value obtained using the initial connection weights may not be close to the target value. The error correction is done based on the calculated error and the initial connection weights are adjusted by propagating the error backwards. With the new adjusted connection weights between input layer nodes and hidden layer nodes as well as hidden layer nodes and output layer node, the inputs are forwarded once again to determine the new output value accordingly, and then the new error is determined and is used to adjust the connection weights. The forward activation of signals and the backpropagation of error are continuously repeated on all training datasets until the error is reduced to a predetermined minimum or an allowed tolerance (Najjar *et al.*, 1997; Najjar and Zhang, 2000). The final connection weights which produce an error within the allowed tolerance range are then stored to represent the network. The final network can be used to predict the desired output(s) of a new dataset that have no actual output values. Note that, backpropagation ANN is a feedforward network and the backpropagation term does not mean the same with feedbackward propagation since the backpropagation is used for the error distribution in contrast to direction of signals' flow. In other words, the training algorithm starts with a feedforward of the input variables, followed by backpropagation of the associated error and connection weights' adjustment.

3.3 Learning Algorithm

The learning process of a standard Backpropagation Neural Network is demonstrated in this section.

Nodal Input Values

The nodes in a certain layer are connected to all other nodes in the following layer. Each node receives signals from all other neurons in previous layer and integrates those signals as a weighted average. For instance, input value for neuron "A" is the sum of the integrated signals multiplied by their corresponding connection weights. The input value for a neuron "A" can be expressed with the following equation:

$$(Input)_A = \sum(node\,value) \times connection\ \ weight$$ Equation 3-1

As depicted in Figure 3.2, the input of one node (i.e., Neuron A) is the all incoming signals and collective effect signal calculated as the weighted sum of all incoming signals is calculated according to the following equation:

$$Net_j^L = \sum_{i=1}^{m} w_{ji}^L \, Out_i^{(L-1)} \qquad \text{Equation 3-2}$$

Where Net_j^L refers to the excitation of neuron j in the L^{th} layer, w_{ji}^L represents the numerical value of the interconnection weight between neuron i in the $(L-1)^{th}$ layer and neuron j in the L^{th} layer. $Out_i^{(L-1)}$ is the output from the i^{th} neuron in the $(L-1)^{th}$ layer. Finally, Net_j^L is nonlinearly transferred via an appropriate activation function.

Activation Function: Sigmoidal Function

To calculate the output of a neuron, the input (i.e., excitation) must be processed through a transfer function because the input might either be very large or negative. In order to avoid large or negative values and to introduce nonlinearity in the model, the neuron's input experiences an additional nonlinear transformation to produce an output based on the following equation:

$$(Out)_A = f(input)_A \qquad \text{Equation 3-3}$$

Where "f" is a transfer function and "$(input)_A$" is the input value for node A previously calculated using Equation 3.1.

In this study, the Sigmoidal function, among the most common activation functions, was used as the activation function. The Sigmoidal function is the most widely used activation function in Backpropagation networks. The final output signal is positive, continuous and has a specified interval between 0 and 1. Sigmoidal function is expressed as

$$f(Input) = \frac{1}{1 + e^{-(Input)}} \qquad \text{Equation 3-4}$$

Since a neuron receives a total excitation (i.e., input) which is equivalent to " Net ", then the output from the neuron can be expressed as

$$a = f(Net) = \frac{1}{1 + e^{-(Net)}} \qquad \text{Equation 3-5}$$

As " Net " reaches a high (approx. 4.0) or low (approx. -4.0) values, activation stabilizes at values between 0 and 1, respectively.

Weight Adjustment

At the last stage of the backpropagation algorithm, the latest adjusted weights are updated by adding the weight adjustment values to the previous weight values. While the inputs are processed forward through every single layer of the network to produce outputs, the error between predicted and target values is used to adjust the connection weights. The incremental change for the current weight can be calculated as follows:

$$\Delta w_{ji}^{L} = w_{ji}^{L(new)} - w_{ji}^{L(previous)} \qquad \text{Equation 3-6}$$

where “new” and “previous” stand for the current and previous iterations. According to Backpropagation neural network algorithm (Zupan and Gasteiger, 1993), incremental change, Δw_{ji}^{L} can be computed using the Delta-rule:

$$\Delta w_{ji}^{L} = \eta \delta_{j}^{L} Out_{i}^{L-1} \qquad \text{Equation 3-7}$$

where η is the learning rate which controls the size of the updating process. The error factor, δ, reflects the weighted error on the connection ji. The Out_{i}^{L-1} term represents the output from the i^{th} neuron in the $(L-1)^{th}$ layer.

Learning Process

The learning process of a neural network is given as follows:

1) Input vectors are marked as X_1, X_2, ……, X_n, 1 where n refers to total number of input variables and last input stands for the threshold or the bias.
2) Propagate the input vectors, X_1, X_2, ……, X_n, via the connection weights to compute the output vectors, Out^{1} using the Equation 3.3 until consequently reaching Out^{last}.
3) Itemize initial weights, w_{ji}^{L} and update connection weights on output layer using the equation:

$$\Delta w_{ji}^{last} = \eta \delta_{j}^{last} Out_{i}^{last-1} + \mu \Delta w_{ji}^{last(previous)} \qquad \text{Equation 3-8}$$

Where δ is the correction factor (i.e., the weighted error) and is computed as

$$\delta_j^{last} = \left(y_j - Out_j^{last}\right) Out_j^{last}\left(1 - Out_j^{last}\right) \qquad \text{Equation 3-9}$$

in which y_j is target value of component, j in the output vector, Y. The function shown in Equation 3.8 is called generalized Delta-rule with a momentum rate (μ) where, $(0 < \mu < 1)$ (Rumelhart *et al.*, 1986). The current connection weight is updated by adding the adjustment to the previous connection weight. Biases are similarly updated on the last layer based on the following equation:

$$\Delta b_{ji}^{last} = \eta \delta_j^{last} + \mu \Delta b_{ji}^{last(previous)} \qquad \text{Equation 3-10}$$

4) All weights on any hidden layer are updated by using the following equation:

$$\Delta w_{ji}^{L} = \eta \delta_j^{L} Out_i^{L-1} + \mu \Delta w_{ji}^{L(previous)} \qquad \text{Equation 3-11}$$

Where δ is the correction factor and is computed as

$$\delta_j^{L} = \left(\sum_{k=1}^{r} \delta_k^{L+1}\, w_{kj}^{L+1}\right) Out_j^{L}\left(1 - Out_j^{L}\right) \qquad \text{Equation 3-12}$$

The biases are corrected within the hidden layer(s) using

$$\Delta b_{ji}^{L} = \eta \delta_j^{L} + \mu \Delta b_{ji}^{L(previous)} \qquad \text{Equation 3-13}$$

5) Steps (1) through (4) are repeated for each training dataset.

6) Steps (1) through (5) are repeated until the predicted output meets the corresponding target output within a predetermined tolerance or the training iterations reaches the maximum limit.

3.4 Initial Number of Hidden Nodes

The number of initial hidden nodes and the maximum allowed hidden nodes in ANN model development are specified by the user. ANN process starts with a user-specified initial hidden node and goes up to the maximum allowed number of hidden nodes. At the end of this process, ANN structures which have the least number of hidden nodes and the best prediction accuracy are chosen to be reevaluated in terms of statistical accuracy measures as well as graphical accuracy measures. The maximum number of hidden nodes, HN(max), can be calculated by the following equation:

$$HN(\max) \leq \frac{(number\ of\ training\ datasets) - (number\ of\ ouput\ variables)}{(number\ of\ input\ variables) + (number\ of\ output\ variables) + 1} \qquad \text{Equation 3-1}$$

Note that, choosing too many hidden nodes could lead to over-fitting situation. On the other hand, very few hidden nodes may not be enough to obtain a model for a complex phenomenon. Concerning the number of hidden layers, networks with one hidden layer are more adequate and efficient. In this research, only one hidden layer was used for the optimal ANN architecture.

3.5 Model Selection Criteria

In order to compare the performance of generated networks and to select the best performing network, statistical accuracy measures such as the Coefficient of Determination (also known as R^2), the Mean Absolute Relative Error (MARE), and the Mean Root Squared Error (MRSE) are evaluated. Training, testing, validation and overall performance parameters should be considered during the evaluation process. The level of agreement between the predicted and actual output values is interpreted based on statistical measures of the network producing the minimum values of MRSE and MARE; and the highest R^2. The MRSE value can be expressed by the following equation:

$$MRSE = \frac{\sqrt{\sum^{No}\sum^{P}(y' - y)^2}}{(\# of\ outputs)_{No}(\# of\ datasets)_P} \qquad \text{Equation 3-1}$$

Where y' is the predicted output, which is produced by the network and y is the actual value. The MARE value is computed by the following equation:

$$MARE(\%) = \frac{\sum^{No}\sum^{P}\left(\frac{y' - y}{y}\right)\times 100}{(\# of\ outputs)_{No}(\# of\ datasets)_P} \qquad \text{Equation 3-2}$$

R^2 can be expressed with the following Equation:

$$R^2 = \frac{S^2{}_{y'y}}{S_{y'y'} \times S_{yy}} \qquad \text{Equation 3-3}$$

Where $S_{y'y} = \sum y'y - \frac{\sum y' \sum y}{\# \ of\ datasets}$, Equation 3-4

$$S_{y'y'} = \sum y'y' - \frac{(\sum y')^2}{\# \ of\ datasets}\text{, and} \qquad \text{Equation 3-5}$$

$$S_{yy} = \sum y\, y - \frac{(\sum y)^2}{\#\ of\ datasets} \qquad \text{Equation 3-6}$$

Further information about ANN can be found from the following references: Rumelhart and McClelland (1986); Hopfield (1982); Haykin (1999); Rumelhart et al. (1986); Fausett (1994); Basheer (1998); Ali (2000); Herz et al. (1991); Wu and Ghaboussi (1995); Ghaboussi (1994); Ghaboussi et al. (1991). ANN approach is utilized to develop prediction models for various databases in the following chapters. Model development process for each database is described in details. The corresponding statistical accuracy measures and the graphical comparisons of each ANN model for the best performing networks are shown at the end of their corresponding chapters.

3.6 Figures

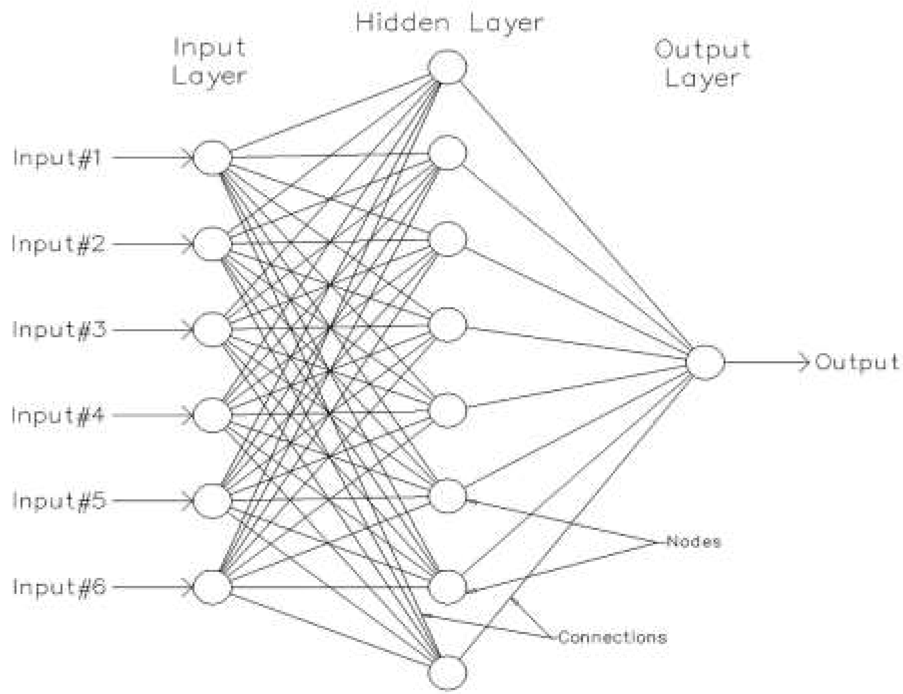

Figure 3-1 Structure of an Artificial Neural Network (ANN)

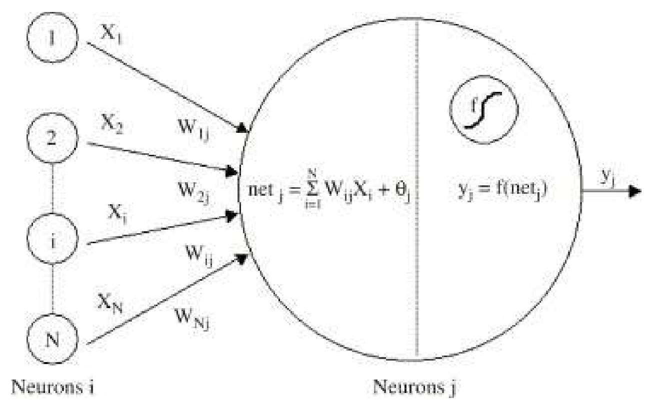

Figure 3-2 Activation Process of a Neuron

CHAPTER 4

4- DATABASE DESCRIPTION

To effectively demonstrate the potential use of Query method and new ANN modeling approaches, seven databases with different characteristics were chosen to be used in this study. Database characteristics in engineering, for extreme cases, may vary from synthetic databases, such as those obtained from computer software or through digital instrumentation, to human factor-involved databases, in which highly associated parameters may not be available. In this study, databases with a combination of various characteristics are included to evaluate the performance of the newly presented methods. Some databases used in this research are;

- Synthetic or digital databases: these types of databases are usually obtained though finite element analysis software. Also, instruments, such as digital sensors, are used to constantly collect data and build optimal databases with less error.
- Human factor-involved databases: these types of databases are typically considered as low-level databases because there are human related factors, for which there may or may not be information available. For example, for a speed limit study it may not be possible to collect information about driver's age or vehicle's comfort level.
- Databases with categorical variables: Some databases include categorical variables whose relationships with phenomena cannot be expressed mathematically, but are vital to the database modeling. For this reason, each category is considered as a binary variable and only one of them can be active at a time. For example, in concrete research studies cement type is one of the parameters proven to affect the final concrete parameters the most. For instance, considering that there are four type of cements used, then 4 additional variables are included in the database. However, only one of the cement types is active at a time. Other types will have no effect on the model.

- Databases with experimental observations: This type of database typically contains experimentally obtained variables. The inputs are controlled variables that are defined in the experimental set-up, while the output is an observation from an experiment. The exact number and type of variables will depend on the specific experiment.
- Databases with multiple outputs: This type of database has multiple outputs that are related to a set of inputs. However, in this study each output is treated separately in relation to its inputs. In other words, one model for each output was developed by using the same input parameters.
- Databases with small variance: In this type of database both the input and the output may have small variance, accordingly it is challenging to obtain a statistically significant relationship between inputs and output(s) in the model.

In order to evaluate the desired models, six real databases representing various engineering applications and one synthetic database were examined. More information about the databases is given herein:

4.1 Database 1

The first database used in this study was generated by using non-linear trigonometric functions. This database represents a simple case, where the data is generated synthetically and accordingly the model was expected to easily recognize the pattern in the database. Therefore, the prediction accuracy was anticipated to be high and the error low. The first database was generated by using the following non-linear trigonometric functions;

$$Y(X_1, X_2, X_3, X_4, X_5, X_6) = 100 + 2.5 * \left(X_1{}^{1.7}\right) + 4 * \left(X_2{}^{2.2}\right) + 5 * log_{10}(X_3) + 2 * (X_1 * X_4)^{0.5} + 4 * \left(X_2{}^{0.5}\right) * \left(X_5{}^{1.5}\right) * \left(X_6{}^{0.2}\right) - log_{10}\left(\frac{X_2 * X_1 * X_3 * X_5}{X_4}\right) - 5 * (X_1) * \left(X_2{}^{0.2}\right) * \left(X_3{}^{0.1}\right) * \left(X_4{}^{0.1}\right) * \left(X_6{}^{0.1}\right)$$

Equation 4-1

Where X_1, X_2, X_3, X_4, X_5, and X_6 are the inputs and Y is the output.

To develop the proposed models for database 1, seven inputs and one output were considered even though Equation 4.1 does not have any component shown as a seventh input. So the seventh variable in the database does not have any correlation with the output (i.e. Y). In order to evaluate the performance and predictability of the new approaches discussed in the following chapters, the additional seventh input variable was included as an un-correlated variable. In this case, the reliability of the new approaches in these types of circumstances was accordingly taken into consideration, which is very common in engineering applications. A total of 300 datasets were used to build the desired database, and then these datasets were divided into 157, 72, and 71 sub-datasets, respectively, for training, testing, and validation. For the Query method application, 229 datasets were used to develop the application and 71 datasets were used to validate the method.

4.2 Database 2

In order to determine the speed limit on highways, various speed studies have been done, which have determined that sensible and cautious drivers will most likely drive at the speed dictated by roadway and traffic conditions rather than relying on the posted speed limits. Actual field studies were carried out to determine the 85th percentile speed at which the drivers felt comfortable to drive at. However, carrying out such field studies for all highway sections is a costly and time-consuming process. For this reason, the database which includes real field measurements was used to develop database 2, which is provided by a government agency. Database 2 has been built by considering six inputs and one output, which respectively are:

Inputs:

1- Percent Pass (%)
2- Annual average daily traffic (AADT)
3- Present serviceability Index (PSI)
4- Surface Width
5- Shoulder Type A
6- Shoulder Type B

7- Shoulder Width

Output:

1- 85th Percentile Speed (mph)

For database 2, a total of 100 datasets were divided into sub-databases, respectively, 55 for training, 23 for testing, and 22 for validation. Seventy eight datasets were used to develop the Query Method application and 22 datasets were utilized to validate the application. Further information about speed studies can be found in the literature (i.e., Najjar *et al.,* 2000)

4.3 Database 3

Database 3 was collected from the literature and utilized to be used in model development. The influence of cement type, curing condition, and testing age on the chloride permeability of concrete mixes was evaluated by conducting Rapid Chloride Permeability test on 126 samples as was reported in the literature (Guneyisi et al., 2009). In this database, five different cement types and two water-cement ratios were deployed. After casting concrete samples, they were subjected to three different curing conditions and tested at the age of 28, 90, and 180 days to determine the chloride permeability of concrete samples through the rapid chloride permeability test. Database 3 has been built by considering 12 inputs and 1 output, which respectively are:

Inputs:

1. (CT1) Cement Type (CEM I=1, CEM II/A-M=0, CEM II/B-M =0, CEM V/A=0, and CEM III/A=0)
2. (CT2) Cement Type (CEM I=0, CEM II/A-M=1, CEM II/B-M =0, CEM V/A=0, and CEM III/A=0)
3. (CT3) Cement Type (CEM I=0, CEM II/A-M=0, CEM II/B-M =1, CEM V/A=0, and CEM III/A=0)
4. (CT4) Cement Type (CEM I=0, CEM II/A-M=0, CEM II/B-M =0, CEM V/A=1, and CEM III/A=0)

5. (CT5) Cement Type (CEM I=0, CEM II/A-M=0, CEM II/B-M =0, CEM V/A=0, and CEM III/A=1)
6. (W/C) Water-cement Ratio
7. (Ag/C) Aggregate-cement Ratio
8. (SP/C) Superplasticizer-cement Ratio
9. (CC1) Curing Condition (UC=1, CC=0, and WC=0)
10. (CC2) Curing Condition (UC=0, CC=1, and WC=0)
11. (CC3) Curing Condition (UC=0, CC=0, and WC=1)
12. (A) Testing Age

Output:

1. (Q) Total charge passed through the concrete sample (coulombs)

Instead of using six inputs, twelve inputs were used because the cement type was categorized in five groups and curing condition was categorized in three groups. As stated previously, the reason for the categorizations of cement type and curing condition is that there is no mathematical relation among the sub-categories that can be expressed numerically. Since only one of the sub-categories can be used at a time, categorical variables were used to model these inputs parameters to evaluate the correlation between cement type and the permeability response as well as curing condition and the permeability response. For this reason, five different cement types were considered as individual inputs which are, respectively, CEM I (CT1), CEM II/A-M (CT2), CEM II/B-M (CT3), CEM V/A (CT4) and CEM III/A (CT5) and curing condition as UC (CC1), CC(CC2) and WC (CC3). For instance, if cement type and curing condition are specified ,respectively, CEM I and Uncontrolled curing, then CT1 is coded as “1”, all other cement types, CT2, CT3, CT4, and CT5, are coded as “0” and CC1 is coded as “1” while other curing conditions, CC2 and CC3, are coded as “0”. Further information can be found in Yasarer, 2010. A total of 126 datasets were used to build the desired database; 63, 32 and 31 sub-databases were used, respectively, for training, testing and validation purposes. Ninety five datasets were used to develop the Query Method application and 31 datasets were utilized to validate the application.

4.4 Database 4

Database 4 is provided by Kansas Department of Transportation (KDOT) for the development of the Rapid Chloride permeability models. The samples included in the database are either prepared in the laboratory or collected in the field. Even though database 3 and database 4 explores same test method, the input variables considered for modeling and the data sources are quite different. Further information about the test method can be found in Yasarer, 2010. Based on the knowledge gained from experimental data analysis, database 4 has been built by considering six inputs and one output, which respectively are:

Inputs:

1- (A) Surface dry weight (grams)
2- (B) Saturated surface dry weight (grams)
3- (C) Weight in water (grams)
4- Curing time (days)
5- (G_s) Specific gravity
6- (W %) Percent of water absorbed

Output:

1- (Q) Total charge passed through the concrete sample (coulombs)

Although the number of inputs is given as six, the number of inputs could have been used as four by removing dependent variables such as specific gravity and water absorbed. However, the additional inputs will most likely lead the network towards the best correlation between the inputs and the output. A total of 265 datasets were used to build the desired database; 133, 66 and 66 sub-databases are used, respectively, for training, testing and validation purposes. In order to develop the Query Method application for database 4, 199 datasets were utilized and then 66 datasets were used for validation.

4.5 Database 5

Database 5 is similarly provided by KDOT to develop models to predict percent of voids in concrete as part of a test method established in ASTM C 642-97 standard. Similarly the samples included in the database were either prepared in the laboratory or collected in the field. In order to properly characterize percent of voids, a total of 325 datasets were used to build the desired database; 163, 81 and 81 datasets are used, respectively, for training, testing and validation purposes. Database 5 has three inputs and two outputs. For this reason, model development for database 5 was completed twice for each output by using the same three input variables. In this case, the models were optimized for one output at a time. The models developed for each output were referenced to according to their corresponding output number (i.e., Model 1 for output 1).

Based on the knowledge gained from experimental data analysis, Model 1 for database 5 has been built by considering three inputs and one output, which respectively are:

Inputs:

1- (A) Mass of oven-dried sample in air (grams)
2- (B) Mass of surface-dry sample in air after immersion (grams)
3- (CT) Curing Time (days)

Output:

1- (C) Mass of surface-dry sample in air after immersion and boiling (grams)

Model 2 for database 5 has been built by using the same three input parameters and one output, which respectively are:

Inputs:

1- (A) Mass of oven-dried sample in air (grams)

2- (B) Mass of surface-dry sample in air after immersion (grams)

3- (CT) Curing Time (days)

Output

1- (D) Apparent mass of sample in water after immersion and boiling (grams)

These two models were combined to calculate the void percent, which is a function of the two parameters predicted by model 1 and model 2. In order to develop the Query Method application for database 5, 244 datasets were utilized and then 81 datasets were tested for validation.

4.6 Database 6

The magnitude and timing of the temperature rise are very important factors on the hydration of cementitious systems, which should be controlled in order to prevent thermal cracking. Admixtures may play a significant role in the rate of temperature rise of a particular mixture. Accurate modeling of the progress of hydration requires an estimate of the effects of these chemical admixtures on the hydration of cementitious systems. Detailed information about this phenomenon can be found in the literature (i.e. Riding et al., 2012). Database 6 has been built by using the parameters considered in the experimental study, which are respectively:

Inputs:

1- Water/Cement ratio (w/c)
2- Low-range water reducing admixtures, Type A (LRWR)
3- Water-reducing and retarding admixtures (WRRET)
4- Mid-range water reducing admixture (MRWR)
5- Naphthalene-sulfonate-based high-range water-reducing admixture (NHRWR)
6- Polycarboxylate-based high-range water reducing admixture (PCHRWR)
7- Calcium-nitrate-based non-chloride accelerating admixture (ACCL)
8- Percent cement

9- Percent C_4AF

10- Percent C_3S

11- Percent C_3A

12- Percent Na_2O

13- Percent Na_2O Equivalent

14- Fly ash mass to total cementitious content ratio (FA)

15- Fly ash CaO mass to total fly ash content ratio (FA-Cao)

16- Percent Slag

Output 1:

1- Ultimate degree of hydration (α_u)

Output 2:

2- Hydration time parameter (hours), (τ)

Output 3:

3- Hydration shape parameter (β)

As listed above, Database 6 has 16 inputs and three outputs. Three models for each output were developed as similarly done for database 5. In this case, the same 16 input parameters were used each time to predict a different output. For example, 16 inputs and output 1 were used to develop one model, which is called Model 1 and similarly 16 inputs and output 2 to develop another model, which is called Model 2 as they were related to their corresponding output number. Therefore, three individual models to characterize the behavior were developed and used separately to be considered in the proposed model development processes. A total of 210 datasets were used to build the desired database; 105, 53 and 52 sub-databases are used, respectively, for training, testing and validation purposes. In order to develop the Query Method application for database 6, 158 datasets were utilized and then 52 datasets were considered for validation.

4.7 Database 7

Poor subgrade soil conditions can result in inadequate pavement support and reduce pavement life. Soils may be improved through the addition of chemical or cementitious additives. These chemical additives range from waste products to manufactured materials and include lime, Class C fly ash, Portland cement and proprietary chemical stabilizers (Najjar et al, 2009). Database 7 was built to predict the unconfined compression strength (UCS) of the stabilized soils. Database 7 has been built by considering 14 inputs and one output, which respectively are:

Inputs:

1- Passing No.200 sieve (%)
2- Plastic Limit
3- Plasticity Index
4- Maximum dry density (kg/m3)
5- Optimum Moisture Content
6- Cement content (by weight) (%)
7- Lime Content (by weight) (%)
8- Fly ash content (by weight) (%)
9- Cement kiln dust content (by weight) (%)
10- Stabilizer (EMC) (%)
11- Dry density (kg/m^3)
12- Moisture content (%)
13- Dry period (day)
14- Moisture Curing period (day)

Output:

1- Unconfined Compression Strength (psi)

A total of 792 datasets were used to build the desired database; 396, 198 and 198 sub-databases are used, respectively, for training, testing and validation purposes. In order to develop the Query Method application for database 7, 594 datasets were utilized and then 198 datasets were kept aside for validation.

CHAPTER 5

5- STATIC ANN NETWORK

In this chapter, Artificial Neural Network (ANN) approach explained in Chapter 3 was used to develop Static ANN models by using the databases described in Chapter 4. Model development procedure outlined in Chapter 3 was followed to model all seven databases. The static ANN method has been used by many researchers to characterize phenomena and considered to be the best approach to modeling in literature. For this reason, the output generated by the static ANN model was utilized by the new approaches to improve and/or optimize the models in this study.

The static ANN model was developed in four sequential stages. In the first stage, the ANN architecture was determined based on problem characteristics and ANN knowledge, and input and output categories were chosen accordingly. This step also includes classifying the datasets as training, testing or validation sets. In the second stage, the network was trained and tested on the experimental data to obtain the optimum number of hidden nodes and iterations for the ANN architecture determined in stage one. In the third stage, the best performing network obtained from the second stage was validated on the validation database. If accuracy measures from training, testing and validation database are highly comparable, then the model may not be trained on all data. In the fourth stage, the best performing network obtained in the second stage was retrained on all experimental data to increase the prediction accuracy and evaluate how well the ANN model characterized the desired behavior. Normally, retraining the network with all experimental data is expected to provide reliable predictions and better accuracy measures. However, it has been shown through several research studies by Najjar and Coworkers [Yasarer & Najjar (2010), Najjar & Mryyan (2009), and Najjar & McReynold (2003)] that stage four is recommended to arrive at a better performing network model. Architecture of a typical static ANN network is depicted in Figure 5-1. The optimal network structures for the static ANN models were selected based on statistical measures such as MRSE, MARE, and R^2, which are described in details in Chapter 3. The statistical accuracy measures of the static ANN models developed for databases 1 to 5 are shown together in Table 5-1, and the measures for databases 6 and 7 are shown together in Table 5-2. Information on the use of the four sequential stages on the seven databases and the criteria used to choose the optimal network structures is given in the following sections.

5.1 Static ANN Network Development of Database 1

Based on the knowledge gained from experimental data analysis, static ANN model architecture has been built by considering 7 inputs and 1 output. A total of 300 datasets are used to build the desired database; 157, 72 and 71 datasets are used, respectively, for training, testing and validation purposes. Based on statistical measures such as MRSE, MARE, and R^2, the optimal network structure of the static ANN model for Database 1 was found at 19 hidden nodes and 19,500 iterations. The corresponding accuracy measures for this network are $MRSE_{tr}$= 1.6151, $MARE_{tr}$= 2.0280%, R^2_{tr}= 0.9996 (for training database) and $MRSE_{ts}$=5.7671, $MARE_{ts}$= 2.7410%, R^2_{ts}=0.9978 (for testing database). The training and testing graphical comparison plots between predicted and actual values for the static ANN model developed for Database 1 are, respectively, shown in Figure 5-2 and Figure 5-3. Also, all the statistical accuracy measures for the training and testing are shown in Table 5-1. After the training and testing procedures using, respectively, 157 and 72 datasets, validation was conducted on the remaining 71 datasets. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 5-4. Once the validation stage is completed, all of the 300 datasets were used to retrain the network at the previously determined optimal structure to obtain the generalized response throughout the 300 datasets. The graphical comparison plot for the 300 datasets is shown in Figure 5-5. Statistical accuracy measures for validation and all data cases are also shown in Table 5-1. As can be seen from the table, static ANN network developed for database 1 has lower validation MRSE value than testing MRSE value. Typically statistical accuracy measures for validation datasets are not expected to be better than testing datasets. This indicates that the network performed well and generalized the phenomena. All data MRSE value is lower than testing and validation measures even though the value of MARE did not improve. R^2 values did not change significantly for testing, validation, or all data cases.

However, MRSE is considered as the main criterion to evaluate the performances of the networks. In Table 5-1, the *7-(8-19)-19500-1* notation denotes the determined architecture of the optimum network of Database 1 where each number, respectively, represents: number of inputs (7), initial number of hidden nodes (8), final number of hidden nodes (19), number of iterations (19500), and number of outputs (1). Final structure of the optimum network is depicted as *7-19-1,* which are, respectively: number of inputs, number of hidden nodes, and number of outputs.

5.2 Static ANN Network Development of Database 2

A database consisting of 100 datasets was used to develop a desired static ANN network for Database 2. As noted previously, the databases to be used for modeling are divided into three sub-categories such as training, testing, and validation. For database 2, during the first stage of modeling, 55 datasets are used for training, 23 for testing, and 22 for validation. The boundary of the training datasets is typically determined by the minimum and maximum of the input and output variables. Therefore, in order to obtain an optimum network with a wider input and output range, minimum and maximum of each input and output variable was considered in training stage. In this case, any input value within the minimum and maximum ranges of the database is applicable to the network. The input vector consisted of 7 parameters and the output vector consisted of 1 parameter were considered to be used in model development of database 2.

After examining the performance of several networks with different architectures, the best performing network was chosen based on the best statistical accuracy measures. Another criterion utilized to select the best performing network is the ideal network architecture. In other words, it is preferred to have less hidden nodes because a more complicated network structure with more hidden nodes can lead to memorization of the data. In this case, the network may generate unreliable predictions. Considering that the number of datasets in database 2 is considerably few, and the accuracy of the models is not expected to be very good because of the involvement of human-factors, fewer hidden nodes were chosen, even though there were network architectures with more hidden nodes and better statistical accuracy measures. The optimal structure for the static ANN network of Database 2 was chosen at 3 hidden nodes and 3100 iterations. A graphical comparison of training stage between the predicted and the actual values is depicted in Figure 5-6. Static ANN network for training stage yielded a mean root square error, $MRSE_{tr}$ of 0.4046, mean absolute relative error, $MARE_{tr}$ of 4.0297, and coefficient of determination, R_{tr}^2 of 0.6061. Similarly, graphical comparison of testing stage is shown in Figure 5-7 and statistical accuracy measures for this network are $MRSE_{ts}$ of 1.0121, $MARE_{ts}$ of 5.9550%, and R_{ts}^2 of 0.0020.

To further evaluate the optimal network, 22 datasets are used to validate the network. Figure 5-8 presents the graphical comparison of the predicted and the actual values. Corresponding statistical measures are given in Table 5-1. It is clear from the results that validation MRSE is lower than the testing MRSE. Once the validation stage is completed, The combined 100 datasets were used to retrain the network at the optimal structure. It can be inferred from the graphical plot in Figure 5-9 and the statistical accuracy measures in Table 5-1 that using the entire database to retrain the network has improved notably the statistical accuracymeasures.

5.3 Static ANN Network Development of Database 3

Another engineering phenomenon from an experimental study was considered to develop a static ANN network. As stated beforehand, this database is a combination of categorical and continuous variables; making this database a good candidate for non-linear modeling. To develop static ANN model for database 3, a total of 126 datasets were used. Sixty three and 32 of total datasets were, respectively, considered as training and testing datasets. The remaining 31 datasets were included in the validation stage after the optimal network was determined. An attempt to develop a static ANN network for database 3 was initiated with 12 inputs and 1 output. The best performing network structure was discovered at 6 hidden nodes and 200 iterations. The training and testing statistical measures for training and testing stages are shown in Table 5-1 and the graphical comparison plots are depicted in Figure 5-10 and Figure 5-11. As can be perceived from the table and the graphical plots, the training and testing stage yielded good accuracy even though MRSE values seem to have greater values than the previous databases. This is because of the output range, which changes from 0 to 14785. The value of the calculated errors such as MRSE and MARE cannot be used to interpret the accuracy of the models among the databases, but can be used within the different modeling stages of the same database (i.e., training, testing, validation, and all data cases). For this reason, the values of MRSE and MARE are recommended to be evaluated for the same database (i.e., only database 3). However, R^2 value can be used the quality of prediction between different databases because of its universal value ranging between 0 and 1 where "1" being the best possible prediction scenario while " 0" represents the worst prediction case.

After the training and testing procedures, validation was conducted on 31 datasets. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 5-12. The statistical accuracy measures for this network are $MRSE_{val}$= 211.5120, $MARE_{val}$= 15.439%, and R^2_{val}= 0.7221. Once the validation stage is completed, all of the 126 datasets were used to retrain the network at the optimal structure. The statistical accuracy measures for this network are $MRSE_{all}$= 63.7835, $MARE_{all}$= 12.719%, and R^2_{all}= 0.9364. The graphical comparison plot for the 126 datasets is shown in Figure 5-13. The resulting statistical accuracy measures for all static ANN network modeling stages are given in Table 5-1.

The statistical measures and the plots have indicated that the static ANN network for database 3 has performed well during the training stage, but the testing stage produced higher MRSE value, as was expected. However, change in MRSE value is more than expected. Similarly, MRSE error for the validation datasets is also high, which is about 7.16 times higher than that of the training MRSE, while MARE error is about 2.4 times higher. When all data combined and the network was re-trained, the statistical accuracy measures improved markedly.

This network yielded higher error values than database 1 an 2 networks because there were more connection weights that needed to be updated to reach the target output value, despite the fact that there were few datasets used to train the network. Statistical accuracy measures of all data, in terms of error, are promising considering that valuable data has been excluded from the training stage for the purposes of testing and validation. It is noteworthy to mention that the networks with more connections may need more datasets to extract more information in the training stage. If more data is not available, then the fourth stage of ANN model development is highly recommended to be produce a well performing final network.

5.4 Static ANN Network Development of Database 4

In order to develop static ANN network for Database 4, a total of 265 datasets; 133, 66, and 66 sub-datasets were, respectively, used for training, testing, and validation. The input vector consisted of 6 parameters and the output vector consisted of 1 parameter. The engineering phenomenon modeled in Section 5.3 for database 3 is the same as the phenomenon modeled in this section, but the parameters considered for input variables are different and obtained from a different source. The output variable has the same range as the output in Section 5.3. Accordingly, the MRSE error was expected to be similarly high because of the large range in output values.

To properly characterize the phenomenon, static ANN network approach with four sequential modeling stages were followed for database 4. In order for static ANN network to reach a least-error structure by training, the goal is to produce output values that are as close as possible to the actual values. The network structure is typically represented by the number of hidden nodes and the number of iterations to reach the optimized connection weights and threshold values for the network to generate outputs close to the actual values. For database 4, the optimal network structure for the static ANN model of database 4 was reached at 7 hidden nodes and 20000 iterations after the stage 2 was completed. Static ANN network for training stage yielded a mean root square error, $MRSE_{tr}$ of 68.4546, mean absolute relative error, $MARE_{tr}$ of 17.4401%, and coefficient of determination, R_{tr}^2 of 0.8554. Similarly, statistical accuracy measures for the testing stage are $MRSE_{ts}$ of 107.1671, $MARE_{ts}$ of 22.372%, and R_{ts}^2 of 0.8226. Graphical comparisons of testing and validation stages are, respectively, shown in Figure 5-14 and Figure 5-15. As can be seen from the graphical plots and the statistical accuracy measures depicted in Table 5-1, a good agreement between actual and predicted values is apparent. Once stage 2 was accomplished and the optimal network architecture was determined, stage 3 and stage 4 were sequentially initiated by using validation datasets and all datasets. The predictions by validation datasets and all datasets case were plotted against their corresponding actual values, respectively, in Figure 5-16 and Figure 5-17. Good agreement between the predictions and the actual values can also be observed in Table 5-1 in terms of statistical accuracy measures. In this case, MRSE value for validation datasets is the highest (as expected) while the all data MRSE value is the lowest. Also, all data case MARE value is higher than the training stage but is lower than the testing and validation stages. Accordingly, training on all data has produced a better performing network as was noted in the previous 4 cases.

5.5 Static ANN Network Development of Database 5

Database 5 has been built by considering 325 datasets; 163, 81, and 81 datasets that are for training, testing, and validation purposes. As previously mentioned in Chapter 4, database 5 has two outputs. For this reason, four sequential stages for static ANN model development process were conducted twice to arrive at two desired prediction models for two outputs. Only one output was considered at a time for optimized networks to be able to generate individual outputs. The reason for developing two individual models instead of one alone is that optimizing a network on two outputs that are not strongly and positively correlated causes a significant decrease in statistical accuracy measures (Yasarer, 2010). Model 1 and model 2 represent the networks with, respectively, output 1 and output 2. The optimal network structure for the model 1 was finalized at 4 hidden nodes and 19800 iterations. The corresponding accuracy measures for model 1 are $MRSE_{tr}$=0.1973, R^2_{tr}=0.9965, $MARE_{tr}$=0.178% (for training database) and $MRSE_{ts}$ =0.4684, R^2_{ts}=0.9846, $MARE_{ts}$= 0.227% (for testing database). The optimal network for Model 2 was reached at 4 hidden nodes and 19500 iterations. The corresponding accuracy measures of model 2 are $MRSE_{tr}$ =0.6420, R^2_{tr}=0.9285, $MARE_{tr}$=1.205% (for training database) and $MRSE_{ts}$ =0.8316, R^2_{ts}=0.9359, $MARE_{ts}$= 1.055% (for testing database). For the training and testing stages, model 2 has reduced the error more than model 1 even though values of the statistical measures for model 1 seem to be higher. In comparison with MRSE value from training stage, MRSE value for model 1 increased by about 137% in testing while MRSE value for model 2 increased by about 30% in testing. R^2 value for model 1 has decreased about 1.2% while R^2 value for model 2 has increased about 0.8%. It can be inferred that coefficient of determination (R^2) is not a good criterion for database 5 to evaluate the performance of the models because its value changes slightly while the change in MRSE values is noticeable.

The training and testing plots for model 1 are shown in Figure 5-18 and Figure 5-19. In the plots, the training and testing predictions are closely scattered around the 45 degree line, which means that predicted value is very close to actual value. Similarly the training and testing plots for model 2 are also given in Figure 5-20 and Figure 5-21. The corresponding statistical measures of model 1 and model 2 are presented in Table 5-1.

After the optimal network was determined, the validation for model 1 and model 2 was conducted on 81 datasets. The validation plots for model 1 and model 2 are, respectively, given in Figure 5-22 and Figure 5-23. After the validation stage is concluded, all of the 325 datasets were used to re-train the network at the optimal structure. The comparison plots of model 1 and model 2 for the 325 datasets are, respectively, shown in Figure 5-24 and Figure 5-25. The resulting statistical accuracy measures for the validation and the all data cases are depicted in Table 5-1. All data MRSE statistical measures for both model 1 and model 2 have the best results compared to their previous stages. Overall, model 1 has better statistical accuracy measures than model 2, even though model 2 has good accuracy measures.

5.6 Static ANN Network Development of Database 6

Another database with highly non-linear behavior and multiple outputs was used to develop static ANN network. 210 datasets were collected to build database 6 and divided into sub-databases: 105, 53, and 52 to be used, respectively, for training, testing, and validation. As stated in Chapter 4, database 6 has three outputs for which static ANN model development process was conducted three times to arrive at three desired prediction models for three outputs individually. In other words, static ANN model development process was repeated for each output and each model developed was called with its corresponding output number (i.e. Model 1 for output1). The number of sub-databases was the same for the three models.

After evaluating the performance of different network architectures, Static ANN network for Model 1 was determined at 3 hidden nodes and 5000 iterations. The optimal network was chosen among many other networks based on the obtained statistical accuracy measures. This network structure provided the optimal connection weights for the desired predictions. The training and testing accuracy measures for model 1 are presented in Table 5-2 along with the corresponding plots shown in Figure 5-26 and Figure 5-27 . According to the statistical measures, the optimal network performed well in the training stage as well as in the testing stage. However, MRSE value of the training, 0.0053 deteriorated to 0.0102 for the testing stage, which corresponds to a 92% increase in error. For the validation stage, the statistical measures changed slightly; however, for the all data stage, MRSE improves to a value of 0.0038, which translates into about 40% reduction in error (compared to training), while MARE value increased slightly. Even though R^2 value seems to decrease from 0.7130 for training stage to a value of 0.6612 for all data stage, the main criterion, which is MRSE, has a reasonable reduction in error. All the statistical measures for the validation and all data stages can be found in Table 5-2 and their corresponding plots are, in the given order, represented in Figure 5-28 and Figure 5-29.

The same database used to develop static ANN network for model 1 was utilized for Model 2 by considering 16 inputs and 1 output. Similarly, the optimal network for model 2 was reached at 3 hidden nodes and 13000 iterations. The accuracy of training and testing stages for the selected network architecture is given in Table 5-2 and the graphical evaluation plots are depicted in Figure 5-30 and Figure 5-31. Validation and all data stages were sequentially followed by the training and testing stages. Figure 5-32 and Figure 5-33, which are the plots for validation and all data predictions, indicate reasonably good correlation between the actual and predicted values. A good agreement between the actual and predicted values can easily be assessed from Table 5-2, even though the deviation of the error is very clear in the testing and validation stages. Nonetheless, the all data statistics represent the best accuracy measures in comparison with the accuracy measures of the testing and validation stages.

Model 3 to obtain a static ANN network was developed by considering the same input parameters, used for model 1 and model 2, and the output 3. Similarly, the model development process was sequentially followed in the order of training, testing, validation, and all data. The promising statistical measures were obtained with a structure of 4 hidden nodes and 7900 iterations after conducting the training and testing procedures. Table 5-2 presents all the statistical measures for model 3. Also, corresponding graphical comparisons for all stages are represented in Figure 5-34, Figure 5-35, Figure 5-36, and Figure 5-37. Even though some scatter is noted in these plots, most of the data is predicted reasonably well.

As can be noted from the tables and all the graphical plots, models for database 6 were successfully developed with minimal error for the three outputs. Overall comparison of these three outputs has showed that the least MRSE and MARE values were obtained by model 1 even though R^2 value for model 1 was the worst among the three outputs. This is because the output range of model 1 is considerably small. The same case can be told for model 3, as it is clear from the small MRSE value. The range of model 1 and model 3 are between 0 and 1.9 while model 2 has a range changing from 0 to 91. It is very clear from the differences in the ranges that the statistics of Model 2 can be considered as the best among the three models when considering the statistical accuracy measures and the applicable ranges.

5.7 Static ANN Network Development of Database 7

Last database was utilized to develop static ANN network is Database 7, which consists of 792 datasets divided into 396, 198, and 198 datasets for training, testing, and validation. By considering 15 inputs and 1 output, desired models were initiated with the training and testing stage. Every hundred iterations, the trained network was validated with the testing datasets. Training phase started with 1 hidden node to a maximum number of the allowed hidden nodes that was determined based on the number of training and testing datasets and number of outputs. All the statistical measures from the training and testing stages were considered when selecting the optimal network structure, which was obtained at 7 hidden nodes and 7900 iterations. The accuracy plots of the network with the optimal structure are illustrated in Figure 5-38 and Figure 5-39. The plots show the good correlation between actual and predicted results, even though there seem to be few outliers at the higher end even though there are not many data points available at that range. As can be observed from Table 5-2, the developed static ANN network has reasonably good statistics such as $MRSE_{tr}$= 1.2149, $MARE_{tr}$= 12.56%, and R^2_{tr}= 0.9834. Even though statistical accuracy measures for testing and validation stages deteriorated slightly, they are still considerably good. The accuracy of how well the validation datasets were predicted can be seen in Figure 5-40 and the corresponding statistics are shown in Table 5-2. Combining all datasets and re-training the network improved the model statistics noticeably. In this case, MRSE value of 1.2149 for training was reduced to a value of 0.8466, which can be translated into a 30% reduction. All data MARE and R^2 values changed slightly but the biggest improvement was obtained for the MRSE value. All data predictions are graphically depicted in Figure 5-41 and the statistical accuracy values are given in Table 5-2. As a result, static ANN network for database 7 was successfully developed and the statistical accuracy measures are reasonable. It is important to note again that the more datasets the database has, the better and more reliable the static ANN networks will be.

5.8 Concluding Remarks

In this chapter, a static artificial neural network with backpropagation learning algorithm was developed for seven databases. Effect of input parameters on the output(s) based on the performance evaluation criteria (statistical accuracy measures and graphical evaluation) was utilized to determine the optimal architecture of the neural network models. As seen from the graphical results depicted in Figure 5-2 to Figure 5-41 and the accuracy measures of the developed ANN models for each database listed in Table 5-1 and Table 5-2, the static ANN models successfully characterized their relevant phenomena. All of the models developed in this chapter have promising results. Even though some of the databases were previously considered in an ANN modelling process, they were fully re-developed due to the fact that the effort to develop each model was intended to be the same or similar for consistency.

As can be seen from Table 5-1 and Table 5-2, accuracy measures on training datasets are generally better than those attained on testing datasets. Similarly, accuracy measures on validation datasets are expected to be lower than those reported on training and testing datasets. By training on all data (once the optimized structure is identified), the new accuracy measures are expected to improve to a level that can compete with those reported on training datasets. In most cases, the measures on the all data network are better than those attained on the training datasets. For this reason, stage four training is highly recommended to arrive at a reasonably performing network model that utilizes the best generalization capability and the least memorization ability.

The results indicated that the methodology of using static artificial neural network with backpropagation learning algorithm is a useful, powerful tool not only for accurately predicting, but also to identifying correlations between output and inputs. However, it is necessary to mention that the accuracy of the neural network is highly dependent on the accuracy of the database. A significant amount of inaccurate data may lead to inappropriate and unreliable results. Note that, small databases may not be suitable to extract all important features from by the proposed network structure, which may generate inaccurate or unreliable predictions. This is the fact that this study proposes to investigate in order to arrive at optimized network architectures utilizing new modeling approaches. The role of static ANN model in hybrid decision making system is crucial since it is expected to provide the best initial estimate for most of the new modelling approaches that are explained in the following chapters.

5.9 Figures and Tables

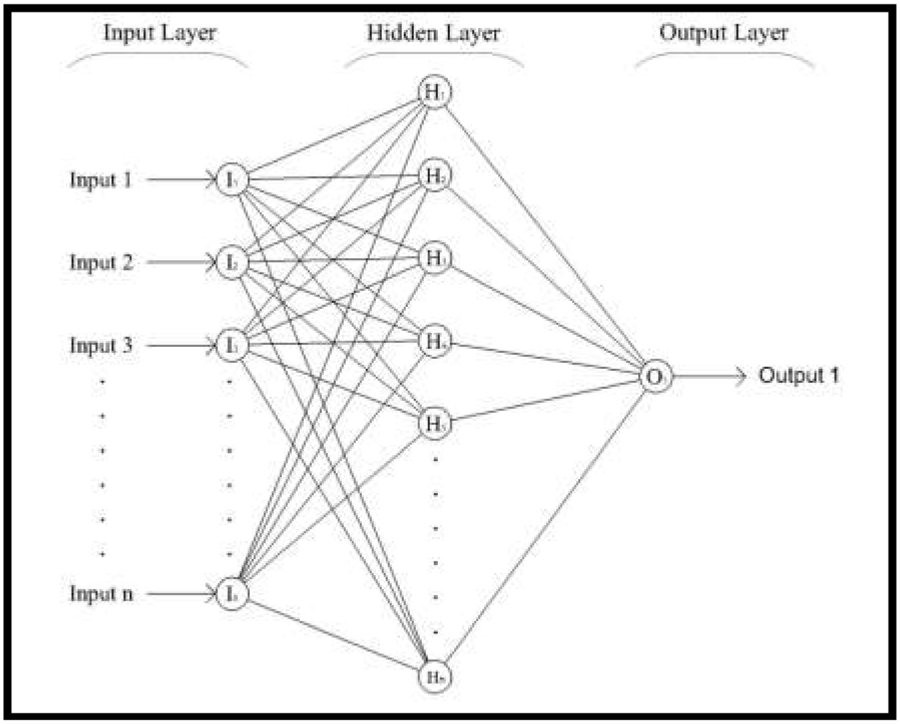

Figure 5-1 Architecture of a Static ANN Network

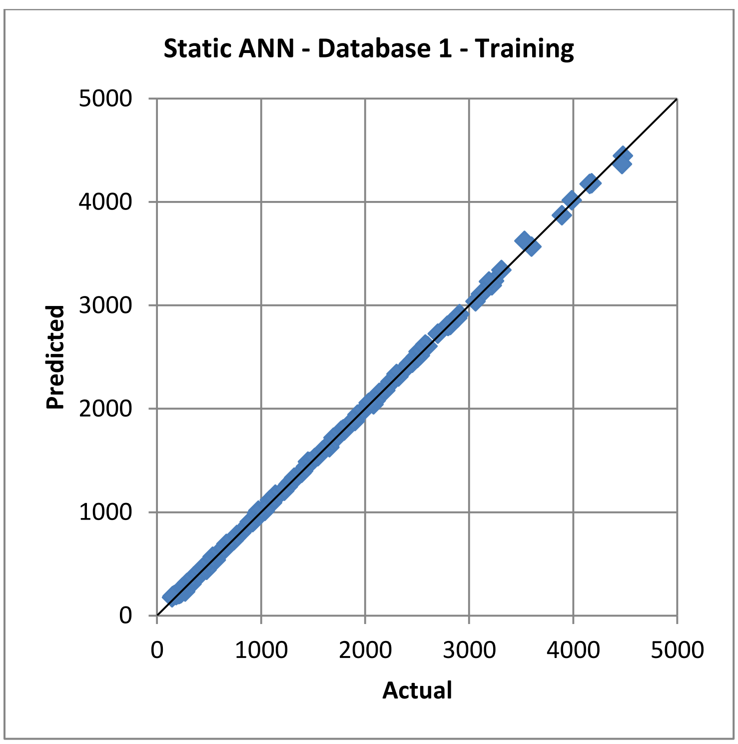

Figure 5-2 Static ANN Training Accuracy of Database 1

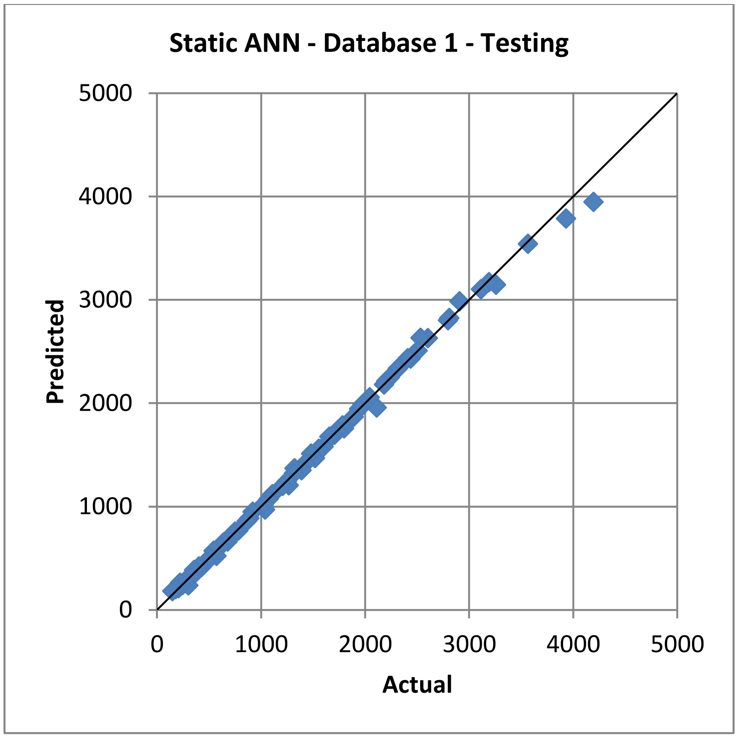

Figure 5-3 Static ANN Testing Accuracy of Database 1

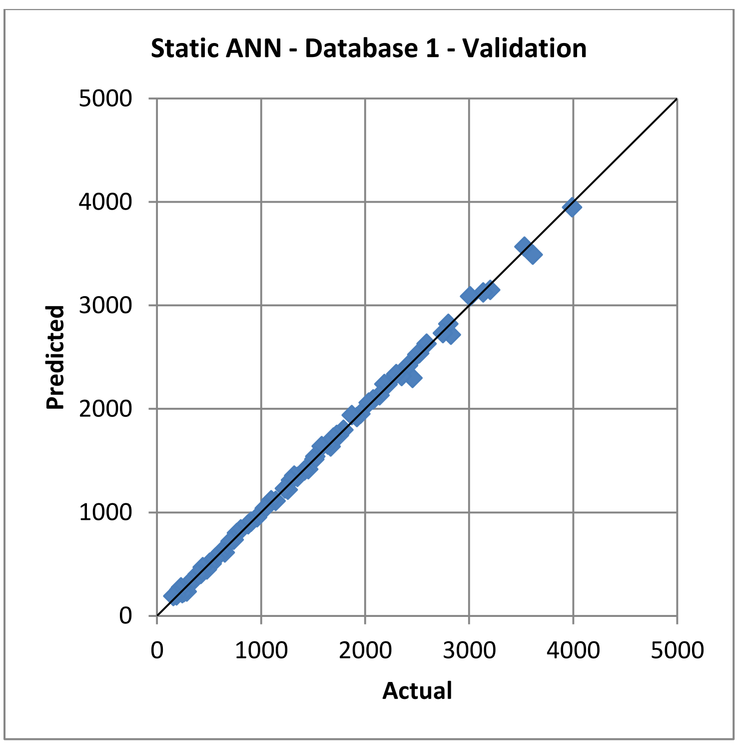

Figure 5-4 Static ANN Validation Accuracy of Database 1

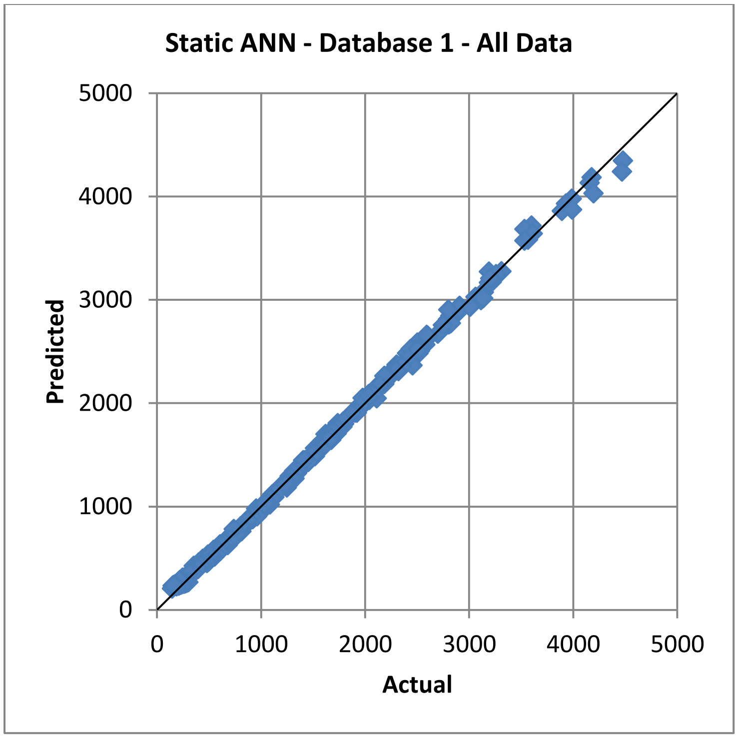

Figure 5-5 Static ANN All Data Accuracy of Database 1

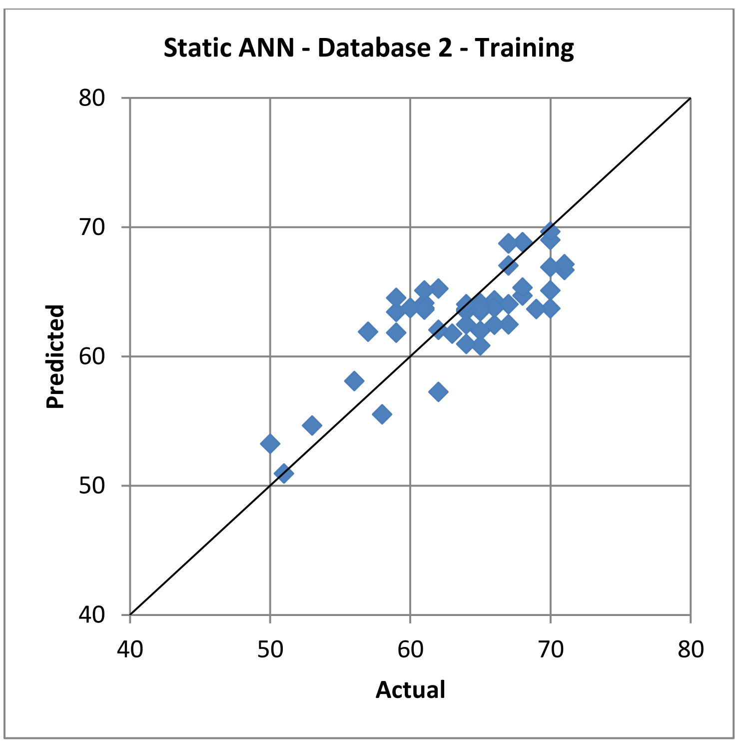

Figure 5-6 Static ANN Training Accuracy of Database 2

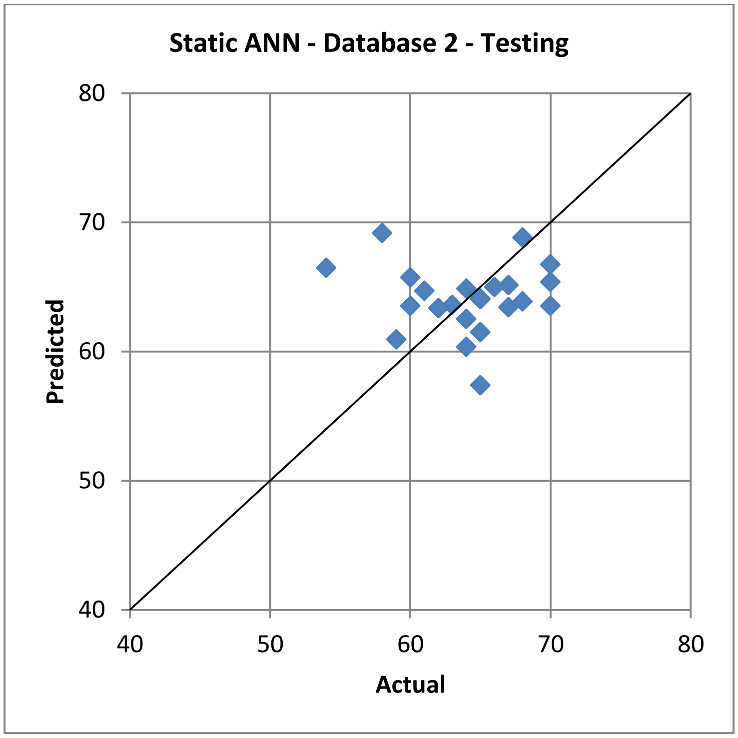

Figure 5-7 Static ANN Testing Accuracy of Database 2

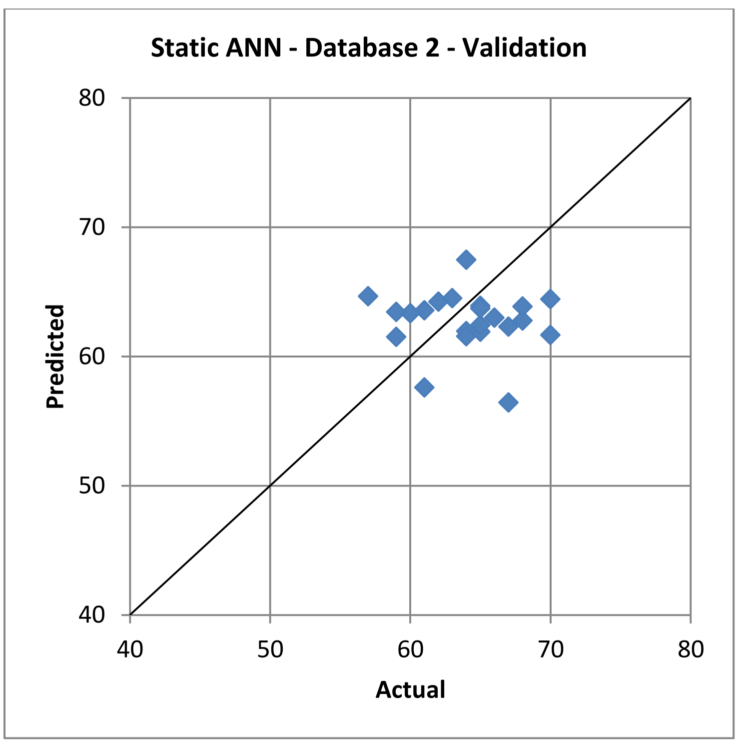

Figure 5-8 Static ANN Validation Accuracy of Database 2

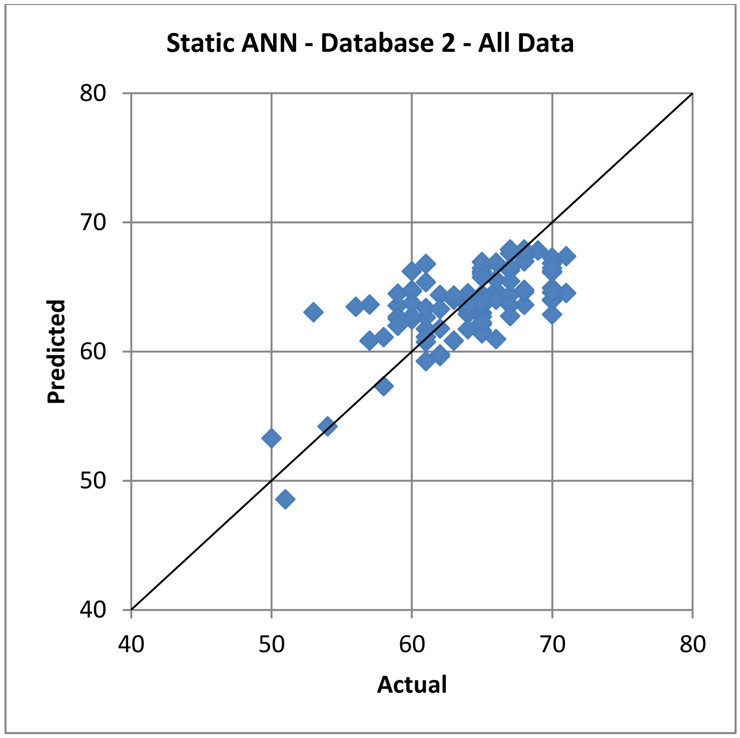

Figure 5-9 Static ANN All Data Accuracy of Database 2

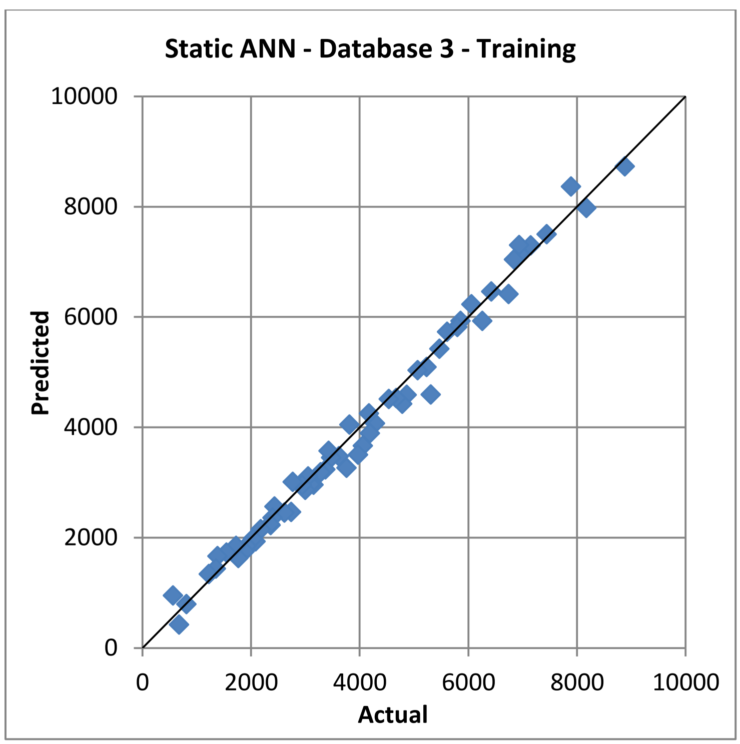

Figure 5-10 Static ANN Training Accuracy of Database 3

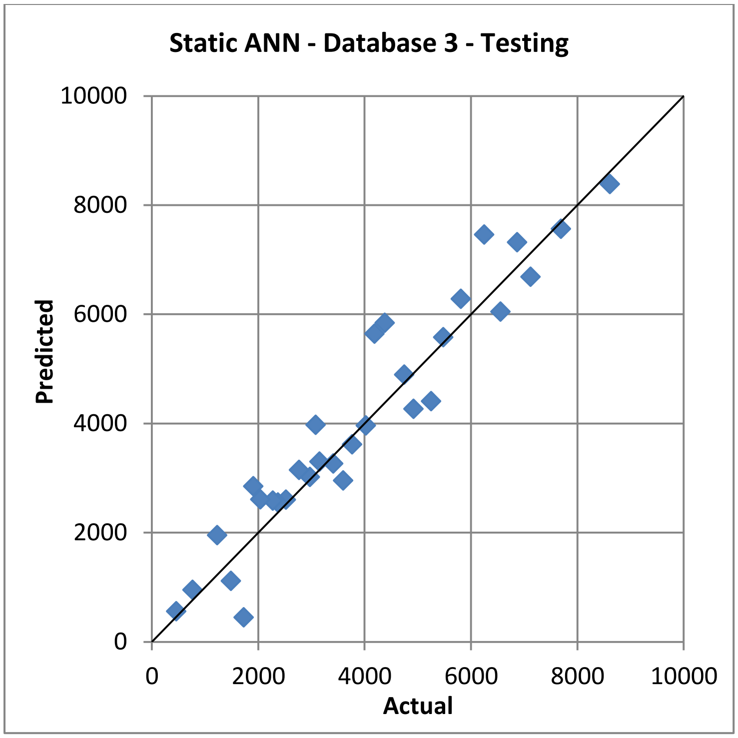

Figure 5-11 Static ANN Testing Accuracy of Database 3

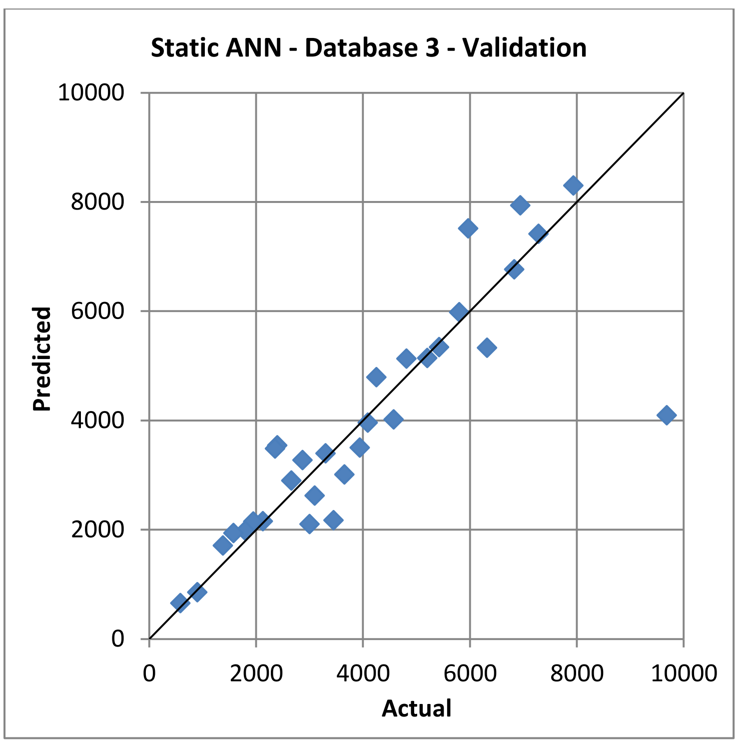

Figure 5-12 Static ANN Validation Accuracy of Database 3

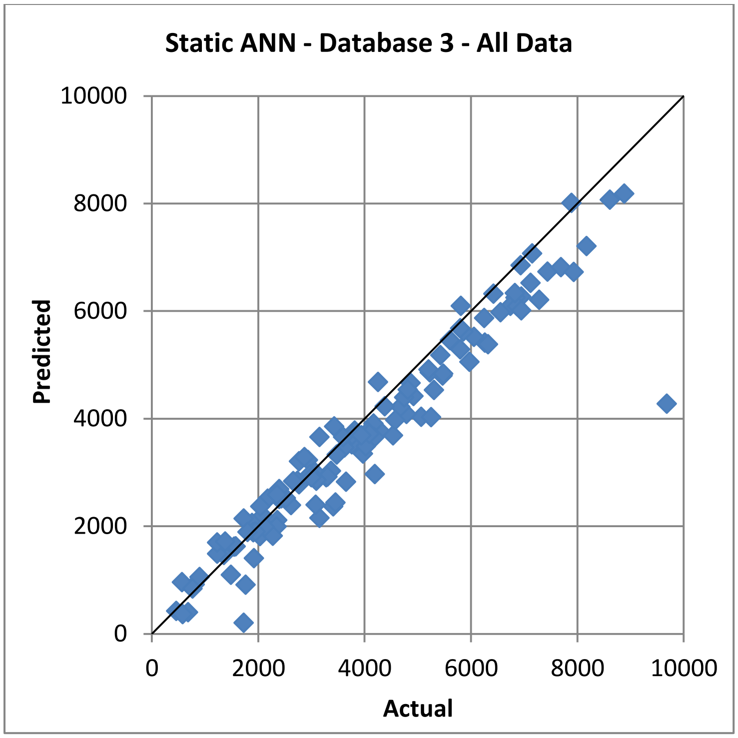

Figure 5-13 Static ANN All Data Accuracy of Database 3

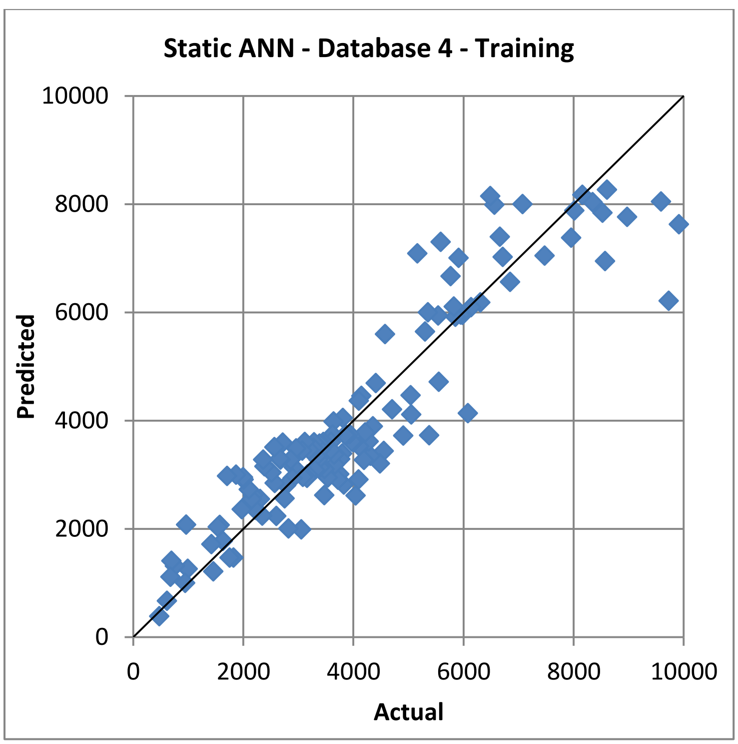

Figure 5-14 Static ANN Training Accuracy of Database 4

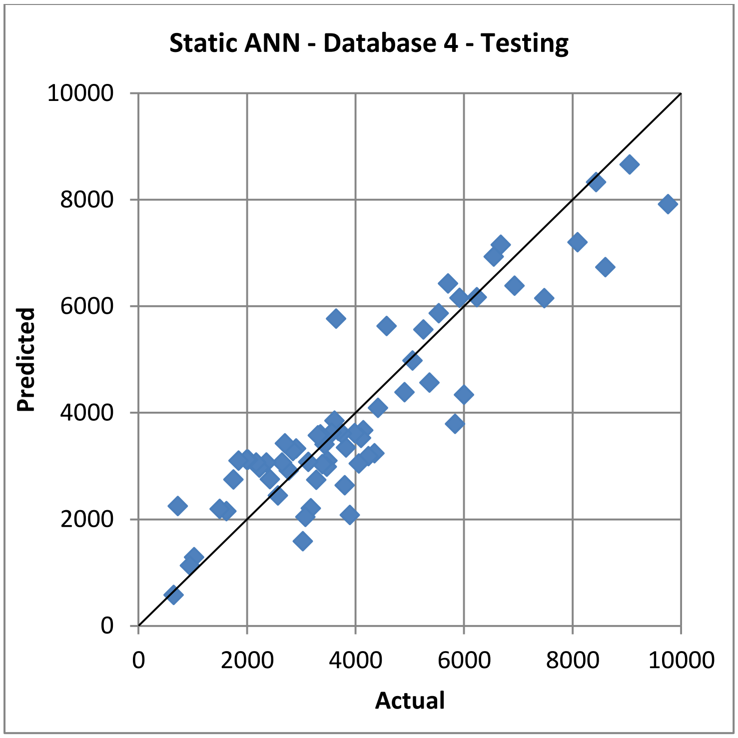

Figure 5-15 Static ANN Testing Accuracy of Database 4

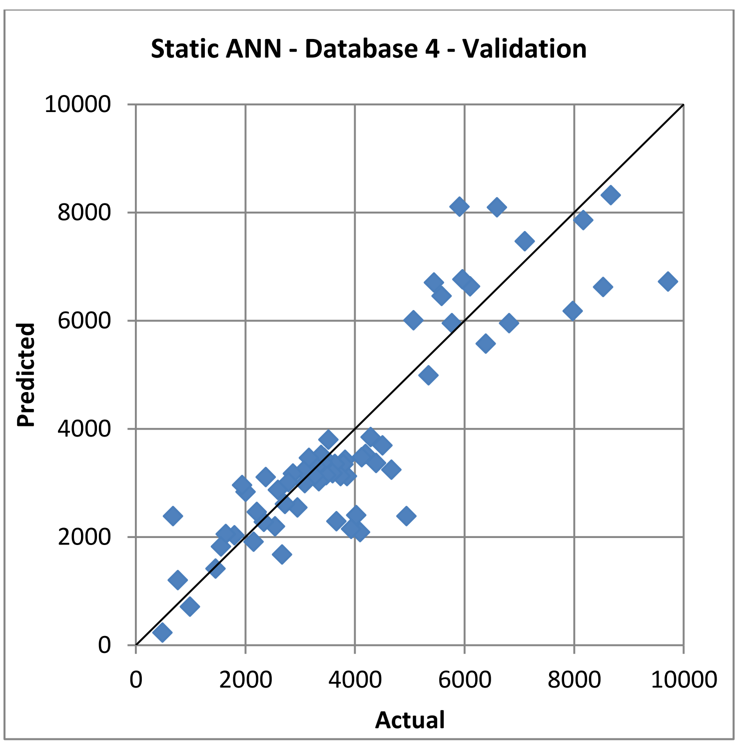

Figure 5-16 Static ANN Validation Accuracy of Database 4

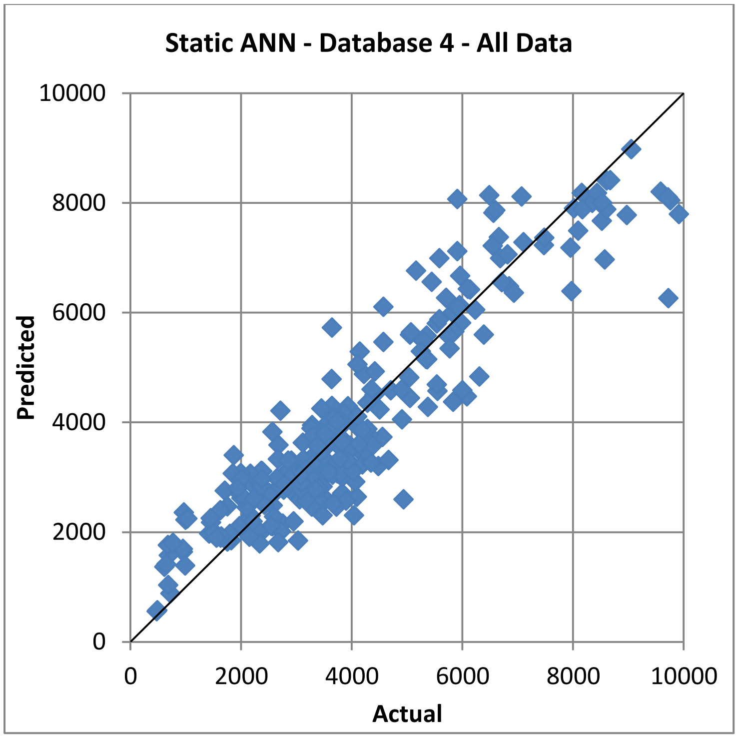

Figure 5-17 Static ANN All Data Accuracy of Database 4

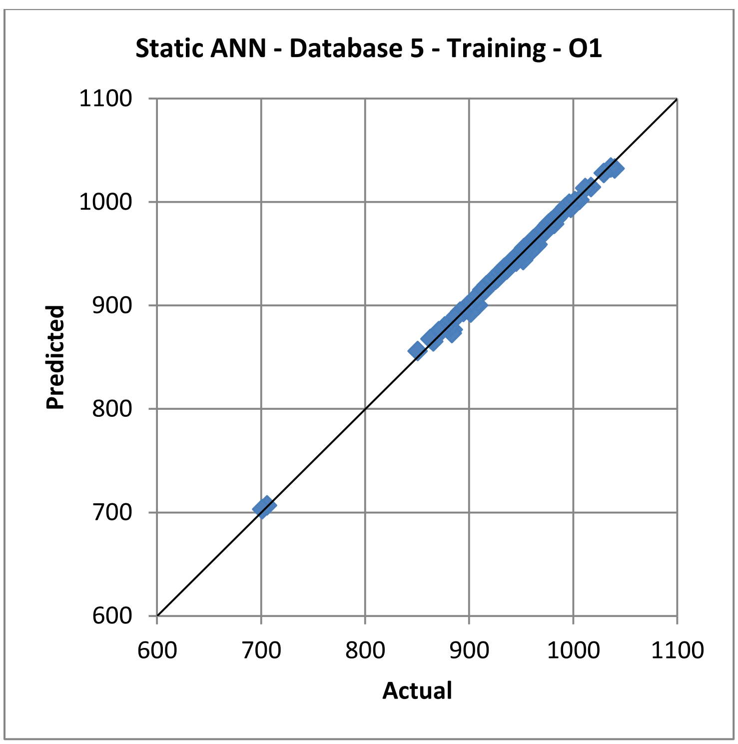

Figure 5-18 Static ANN Training Accuracy of Database 5, Output 1

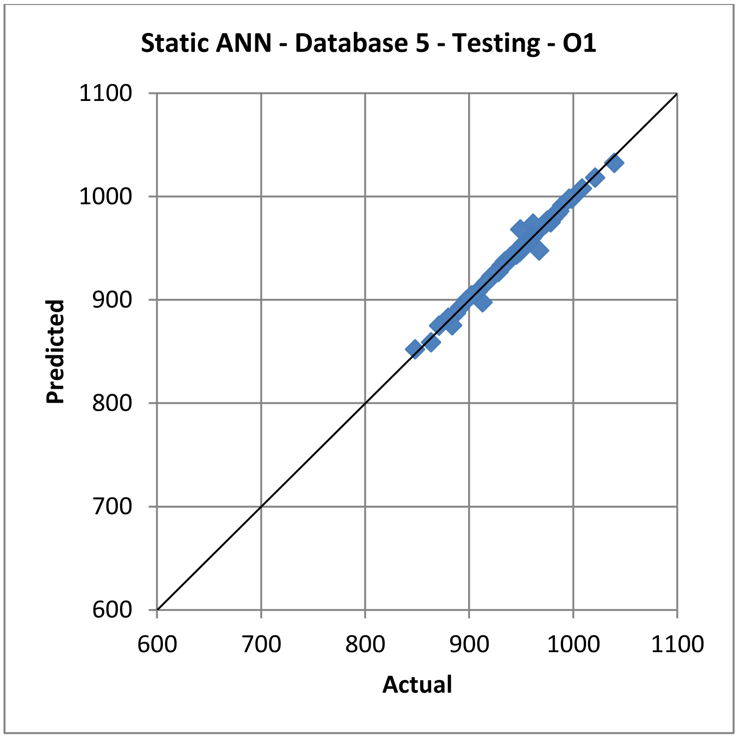

Figure 5-19 Static ANN Testing Accuracy of Database 5, Output 1

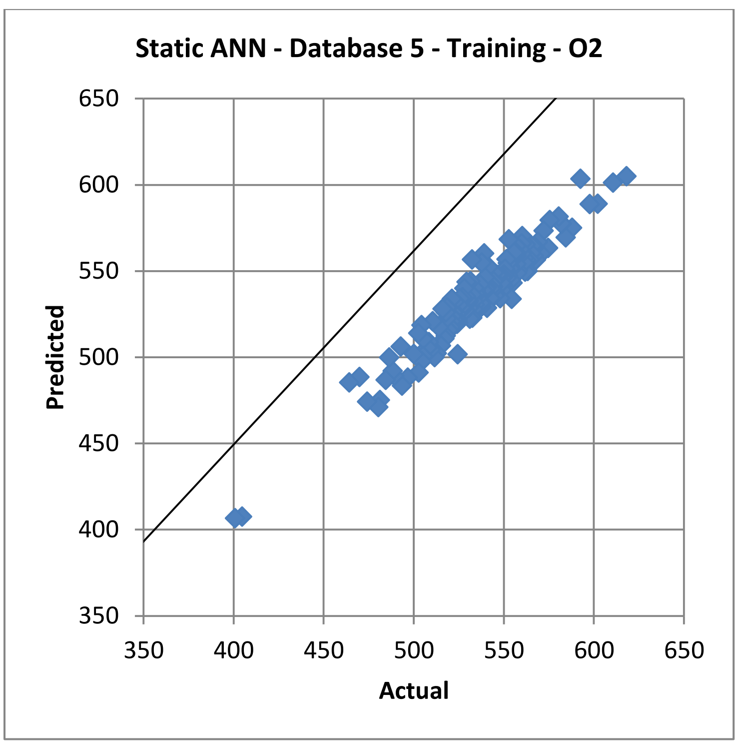

Figure 5-20 Static ANN Training Accuracy of Database 5, Output 2

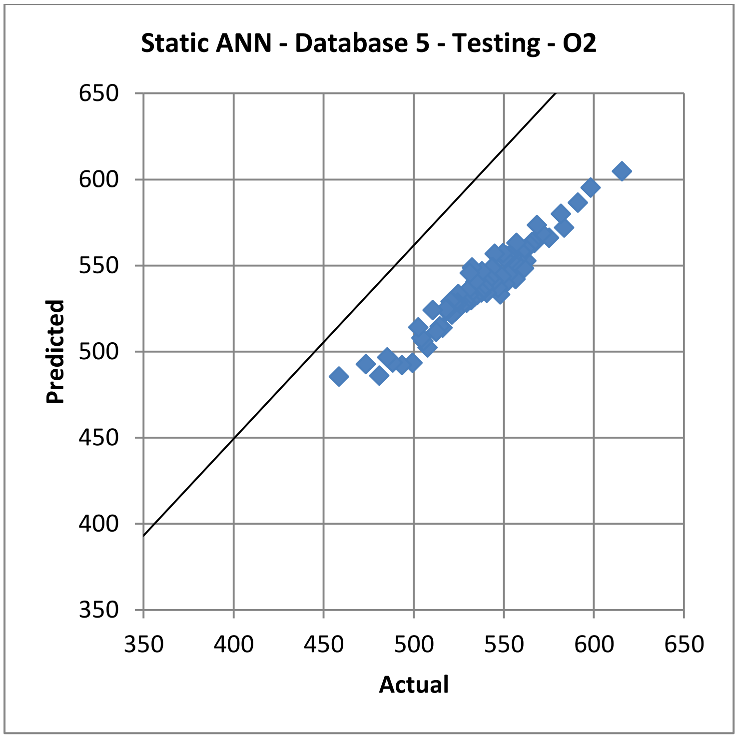

Figure 5-21 Static ANN Testing Accuracy of Database 5, Output 2

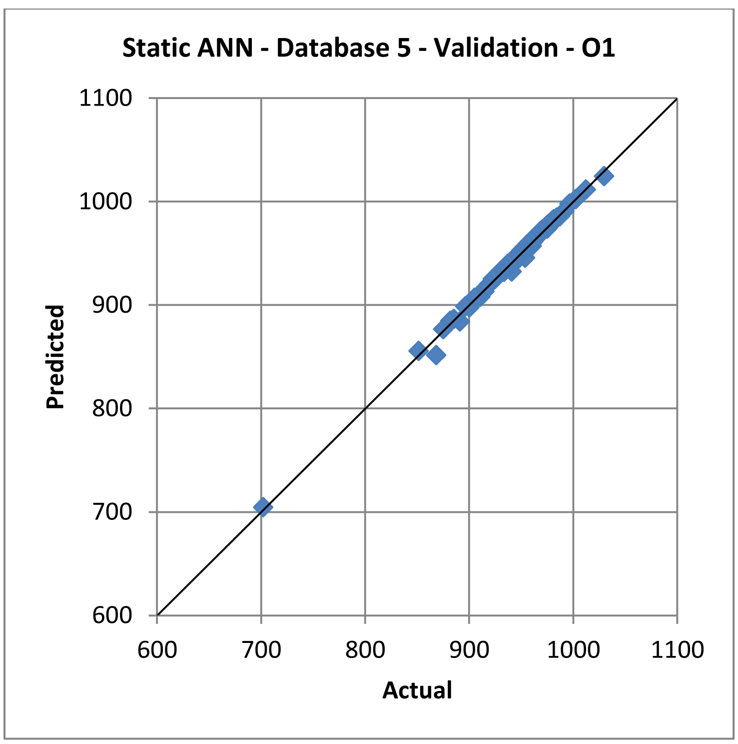

Figure 5-22 Static ANN Validation Accuracy of Database 5, Output 1

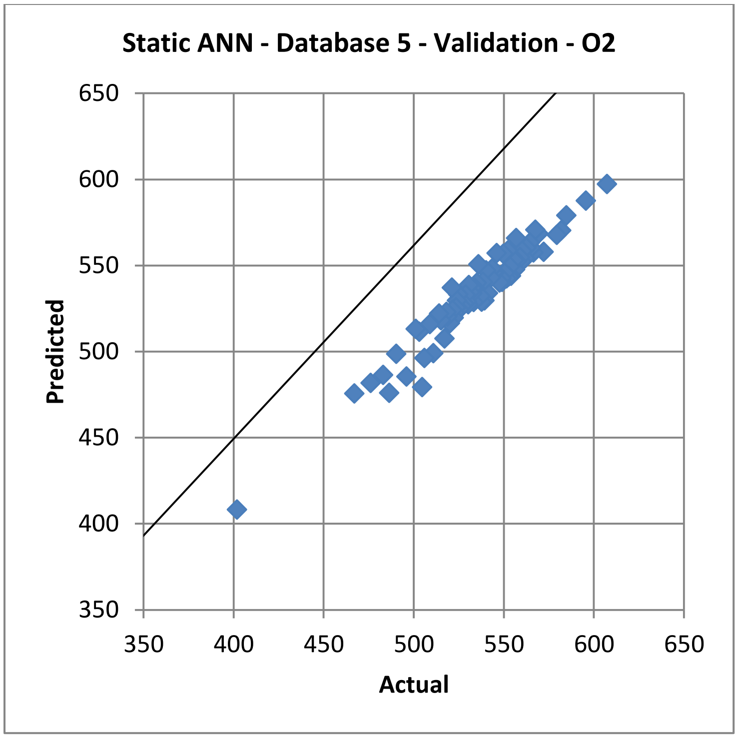

Figure 5-23 Static ANN Validation Accuracy of Database 5, Output 2

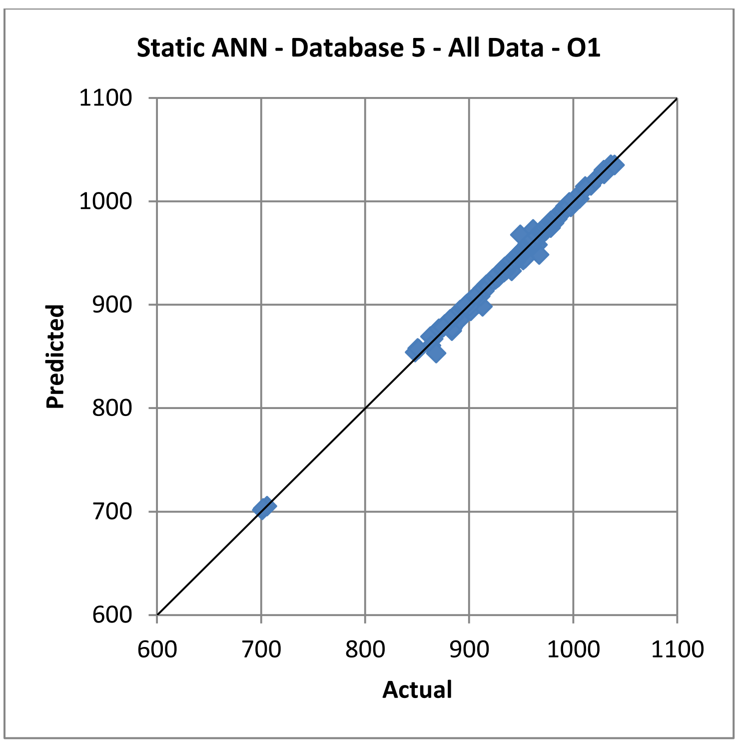

Figure 5-24 Static ANN All Data Accuracy of Database 5, Output 1

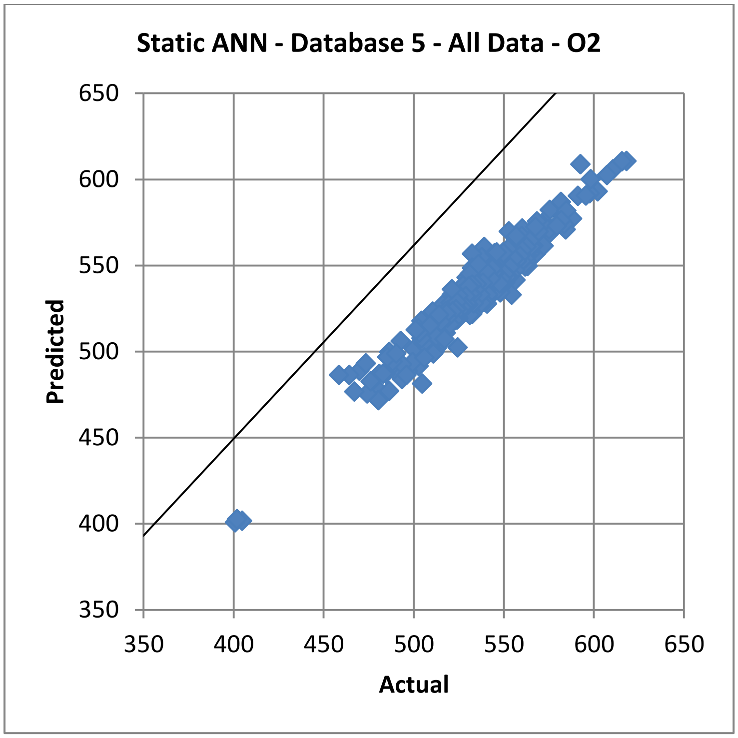

Figure 5-25 Static ANN All Data Accuracy of Database 5, Output 2

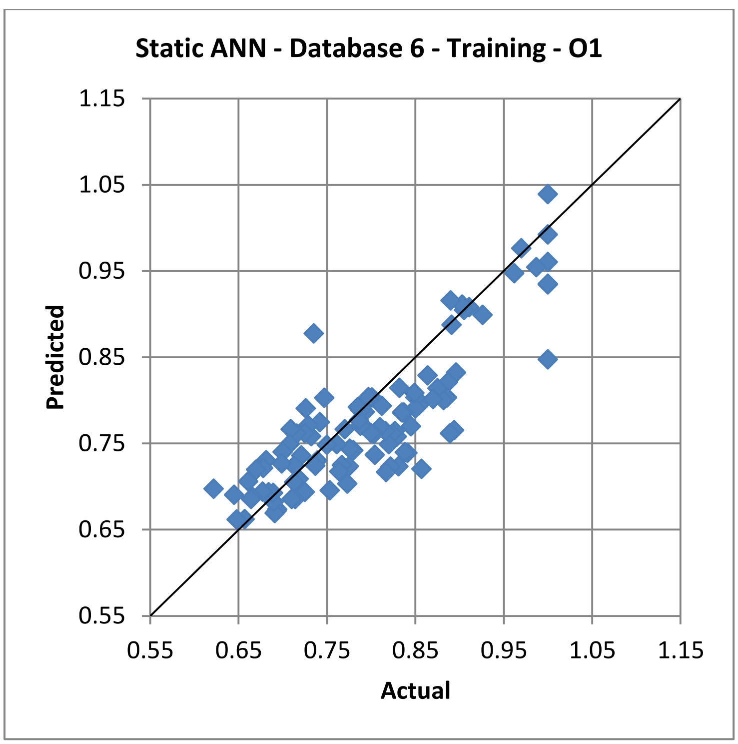

Figure 5-26 Static ANN Training Accuracy of Database 6, Output 1

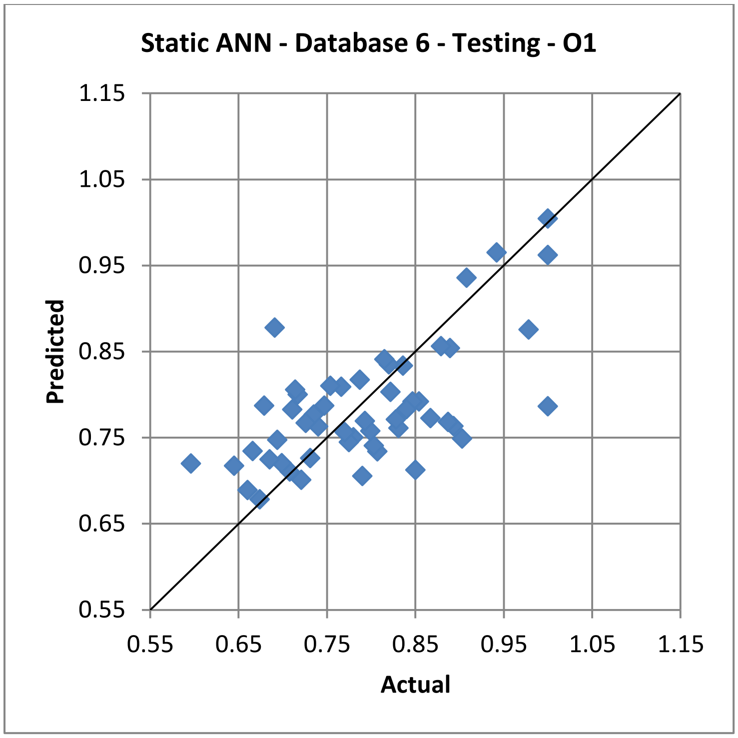

Figure 5-27 Static ANN Testing Accuracy of Database 6, Output 1

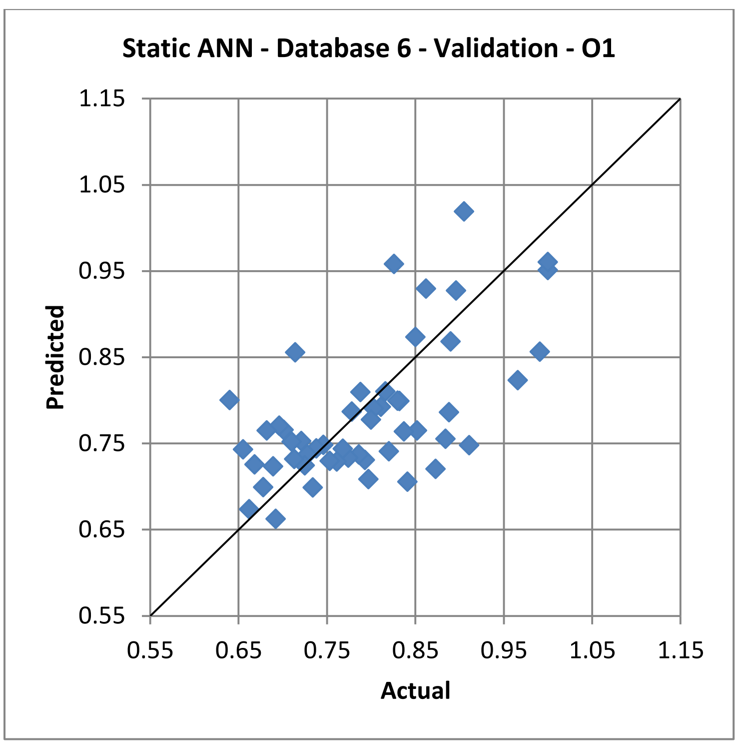

Figure 5-28 Static ANN Validation Accuracy of Database 6, Output 1

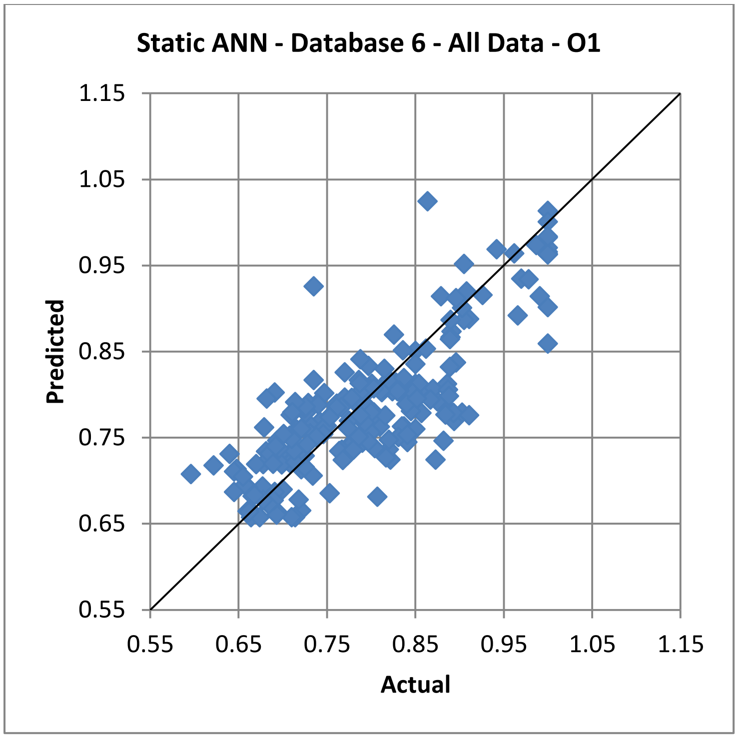

Figure 5-29 Static ANN All Data Accuracy of Database 6, Output 1

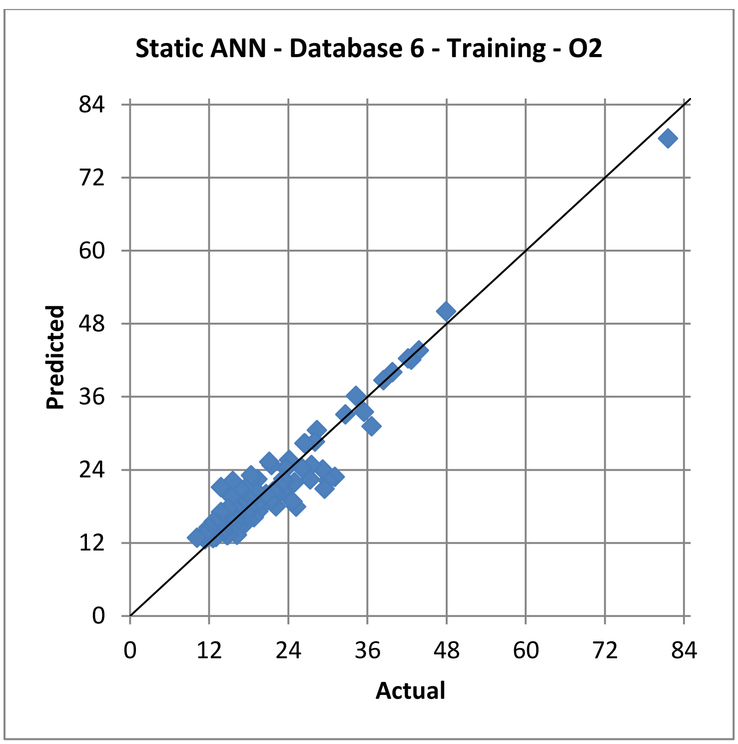

Figure 5-30 Static ANN Training Accuracy of Database 6, Output 2

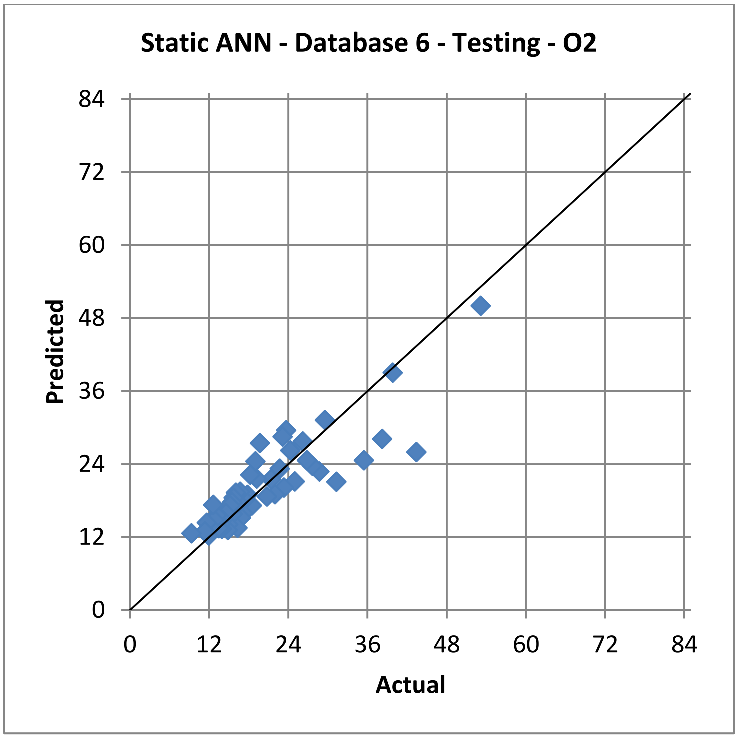

Figure 5-31 Static ANN Testing Accuracy of Database 6, Output 2

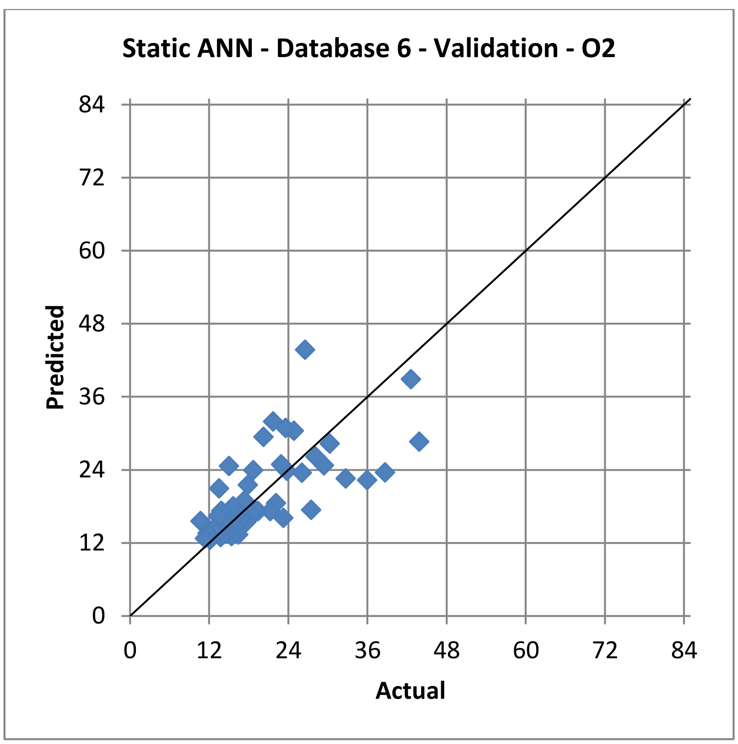

Figure 5-32 Static ANN Validation Accuracy of Database 6, Output 2

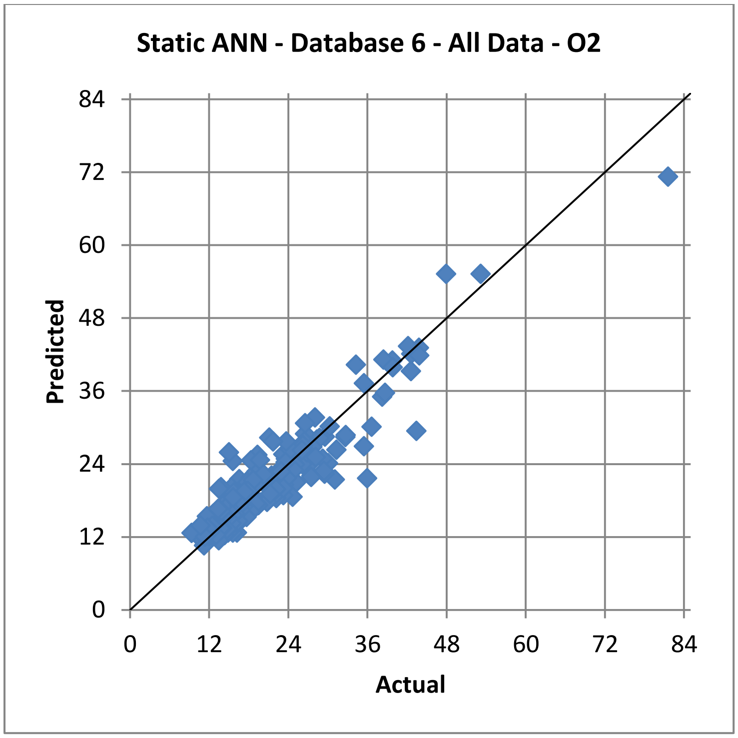

Figure 5-33 Static ANN All Data Accuracy of Database 6, Output 2

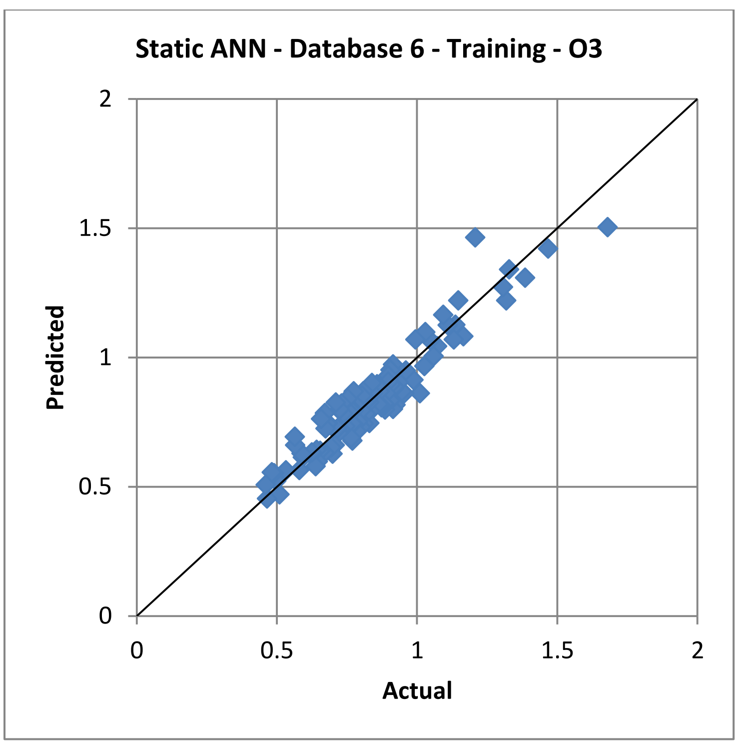

Figure 5-34 Static ANN Training Accuracy of Database 6, Output 3

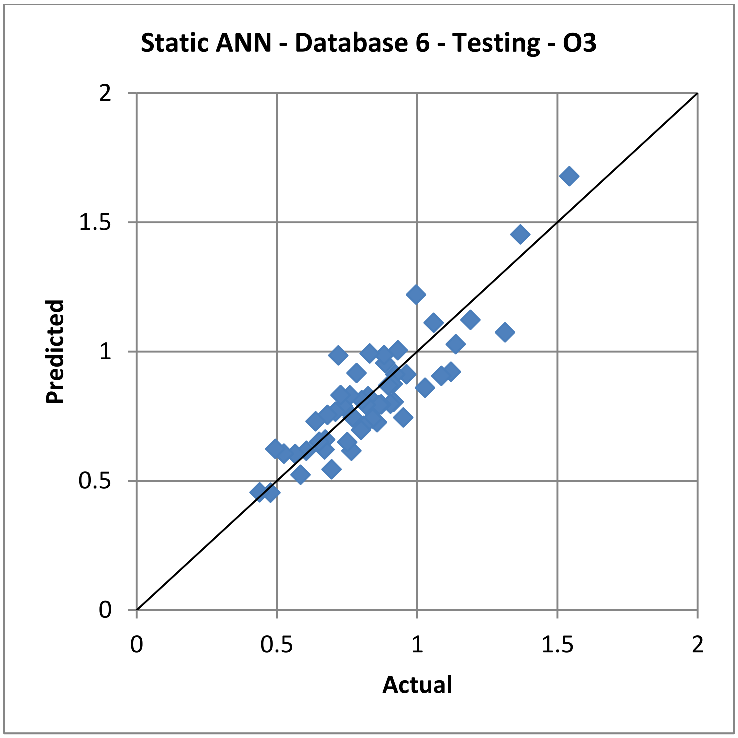

Figure 5-35 Static ANN Testing Accuracy of Database 6, Output 3

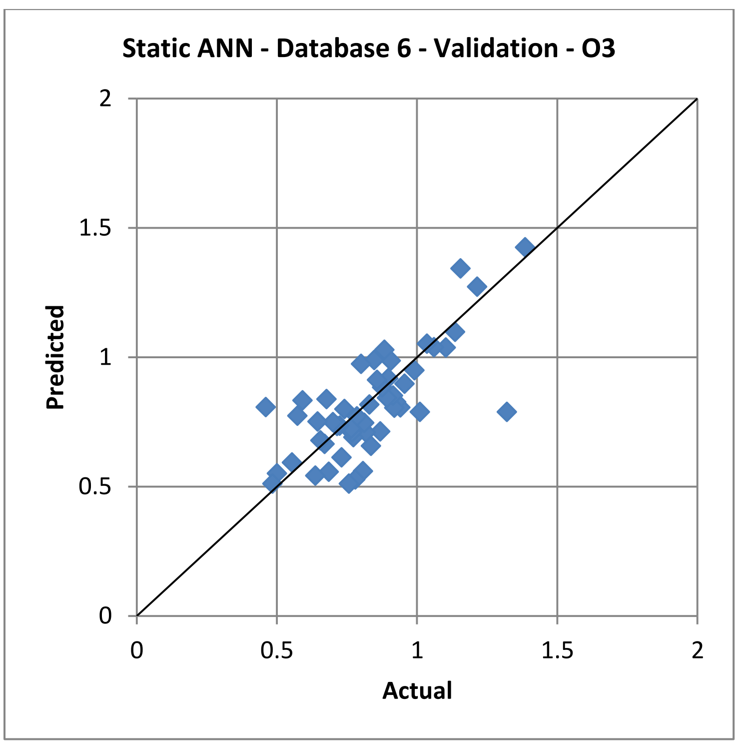

Figure 5-36 Static ANN Validation Accuracy of Database 6, Output 3

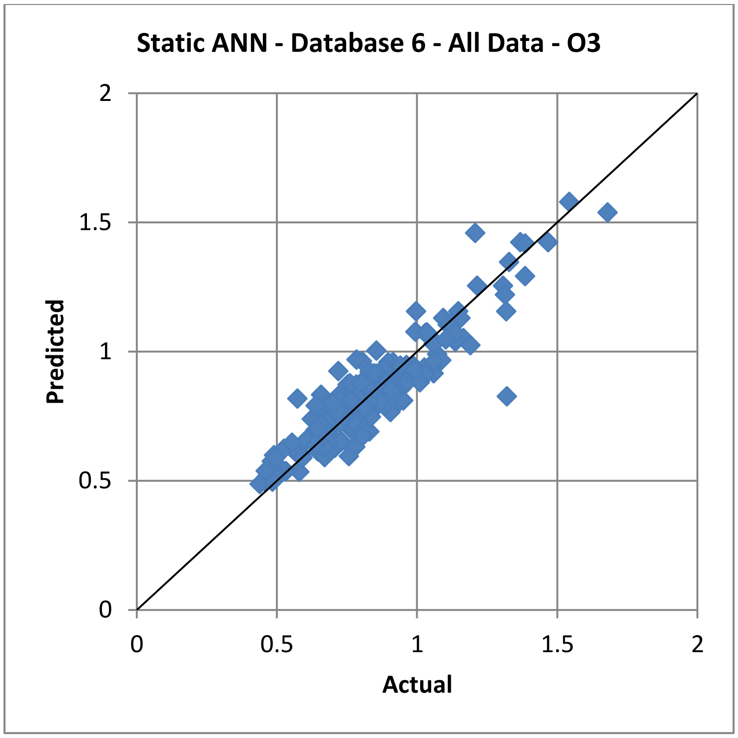

Figure 5-37 Static ANN All Data Accuracy of Database 6, Output 3

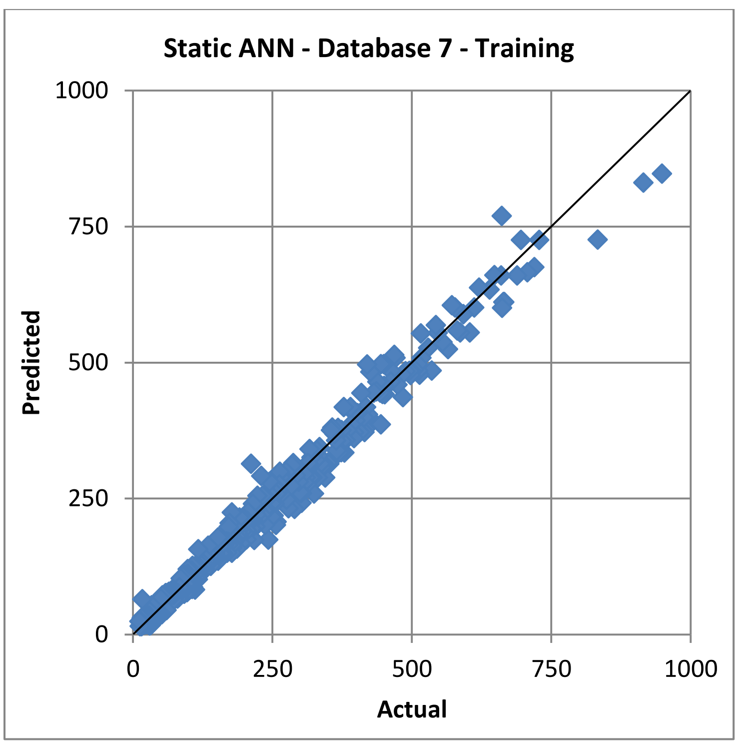

Figure 5-38 Static ANN Training Accuracy of Database 7

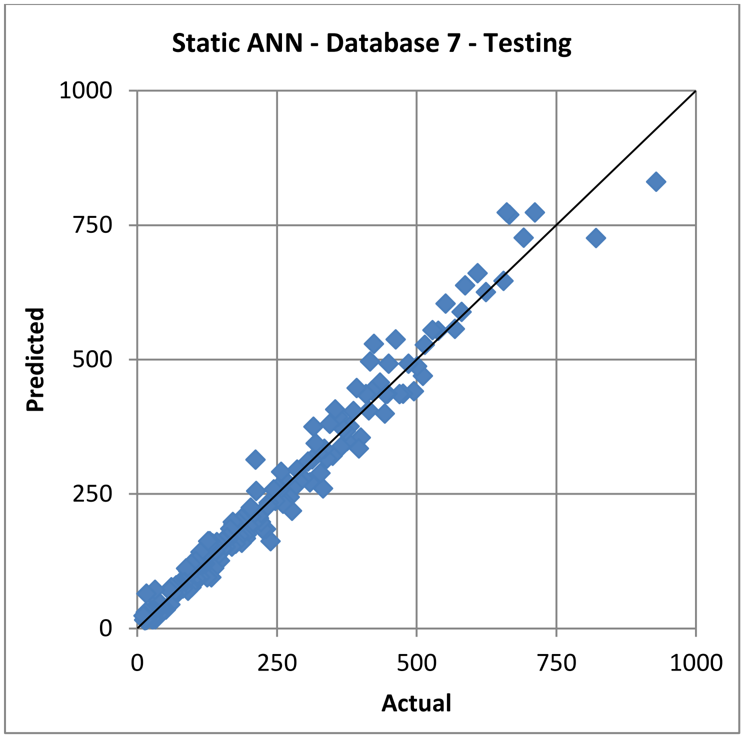

Figure 5-39 Static ANN Testing Accuracy of Database 7

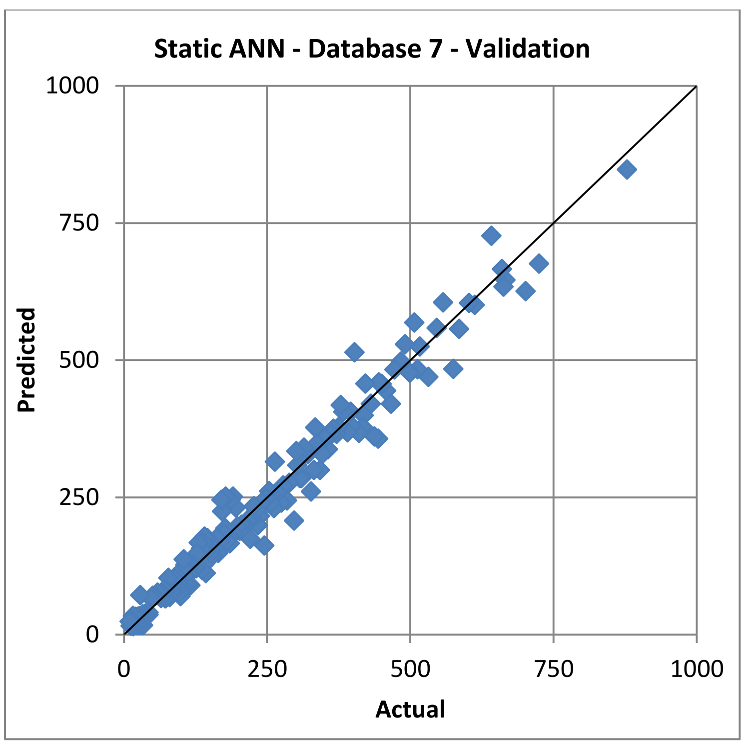

Figure 5-40 Static ANN Validation Accuracy of Database 7

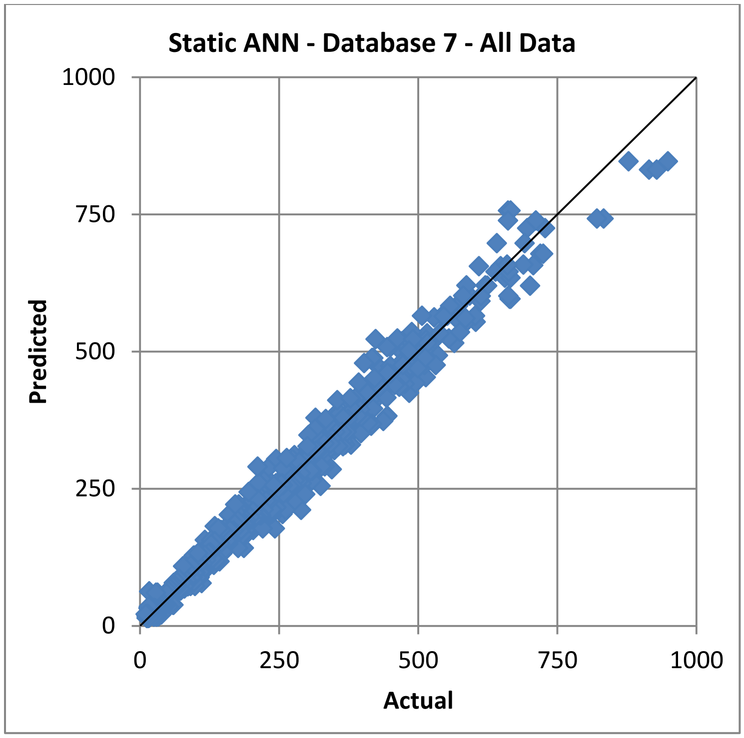

Figure 5-41 Static ANN All Data Accuracy of Database 7

Table 5-1 Statistical Accuracy Measures of Static ANN Models for Database 1 to Database 5

Accuracy Measures		STATIC ANN MODELS					
		Database 1	Database 2	Database 3	Database 4	Database 5	
		Output 1	Output 1	Output 1	Output 1	Output 1	Output 2
		7-(8-19)-19500-1	7-(2-3)-3100-1	12-(2-6)-200-1	6-(2-7)-20000-1	3-(2-4)-19800-1	3-(3-4)-19500-1
TR	MARE	2.028	4.0297	6.432	17.440	0.178	1.205
	R^2	0.9996	0.6061	0.9916	0.8554	0.9965	0.9285
	MRSE	1.6151	0.4046	29.5265	68.4546	0.1973	0.6420
TS	MARE	2.741	5.9550	16.854	22.372	0.227	1.055
	R^2	0.9978	0.0020	0.9406	0.8226	0.9846	0.9359
	MRSE	5.7671	1.0121	113.2199	107.1671	0.4684	0.8316
VAL	MARE	3.014	6.0170	15.439	21.604	0.207	1.180
	R^2	0.9984	0.0078	0.7221	0.7862	0.9949	0.9379
	MRSE	4.5703	0.9647	211.5120	118.7498	0.3321	0.8464
ALL DATA	MARE	4.069	3.9681	12.719	20.359	0.186	1.125
	R^2	0.9984	0.4554	0.9364	0.8549	0.9944	0.9333
	MRSE	2.3740	0.3203	63.7835	47.9782	0.1676	0.4255
FINAL STRUCTURE		**7 - 19 - 1**	**7 - 3 - 1**	**12 - 6 - 1**	**6 - 7 - 1**	**3 - 4 - 1**	**3 - 4 - 1**

Table 5-2 Statistical Accuracy Measures of Static ANN Models for Database 6 and Database 7

		STATIC ANN MODELS			
Accuracy Measures		Database 6			Database 7
		Output 1	Output 2	Output 3	Output 1
		16-(3-3)-5000-1	16-(1-3)-13000-1	16-(1-3)-19400-1	15-(4-7)-7900-1
TR	MARE	5.259	11.038	6.511	12.560
	R^2	0.7130	0.9202	0.9066	0.9834
	MRSE	0.0053	0.2761	0.0065	1.2149
TS	MARE	7.372	13.942	10.663	14.876
	R^2	0.4081	0.7554	0.7678	0.9735
	MRSE	0.0102	0.6057	0.0151	2.1657
VAL	MARE	7.337	19.056	13.515	15.064
	R^2	0.3851	0.4636	0.5444	0.9750
	MRSE	0.0105	0.8268	0.0201	2.0286
ALL DATA	MARE	5.416	11.529	8.009	12.380
	R^2	0.6612	0.8721	0.8377	0.9831
	MRSE	0.0038	0.2276	0.0059	0.8466
FINAL STRUCTURE		**16 - 3 - 1**	**16 - 3 - 1**	**16 - 3 - 1**	**15 -7 - 1**

CHAPTER 6

6- FEEDBACK ANN NETWORK

The human nervous system has more complicated networks than what can artificially be implemented. Artificial neural networks are an over-simplification of the human nervous system. The main distinction between these two systems is the complexity of the human nervous system. One of the reasons for this complexity is that networks in human nervous system may have one or more impulse coming from another network. Basically, knowledge learned through some other network can be fed into other networks.

For instance, babies first learn to roll around, next to sit, and then to stand, and to cruise and then to walk. They start walking slowly first, then once they have confidence about their balance, they walk with confidence. Later on, they even start running. A child learning to walk is a prominent example of the human learning process. Without learning how to sit up, they cannot stand or without learning how to walk, they cannot run. Now, to use this example in the context of artificial neutral networks and to explain it simply; one network trained for walking is an input for another network which may be for running and/or riding a bike. Using this ideology, a new ANN approach is proposed in this study. By training a network and then using the output from this network to feed into another network as an initial estimate is the main methodology of this approach. It is hypothesized that any leading information towards the output will improve the network's generalization capability. In other words, this new method is to improve the accuracy measures of the static ANN models developed in Chapter 5 by including the initial estimates from the static ANN model as another input. In this case, the number of input variables increased by one via including the initial estimate from the model developed in the previous chapter. By feeding the network with an initial estimate, the network may be able to understand the phenomena better. Moreover, generated output can be fed into the network and be iterated until the outcome gets stabilized. Architecture of proposed Feedback ANN network is depicted in Figure 6-1.

To develop Feedback ANN models for the seven databases described in Chapter 4, the same methodology used for static ANN model development was also followed for this new method.

As stated before, the number of datasets used for training, testing, and validation purposes were similar to those used to develop the Static ANN models. However, the Feedback ANN model development process has been completely re-done according to the four training stages described in Chapter 5. The initial estimates used to develop Feedback ANN network models were taken from the trained all networks. In other words, the best performing networks for static ANN models were chosen and the networks were retrained with all experimental data because it is expected to improve the statistical accuracy measures, and their generalization capability according to the results shown in Chapter 5. Hence, all data predictions from static ANN model development stage were included as an input in the model development of Feedback ANN Network models. Accordingly, the number of the inputs for all seven databases was increased by one. Similarly as stated in Chapter 5, the optimal network structures for the Feedback ANN models were selected based on statistical measures: MRSE, MARE, and R^2. The statistical accuracy measures of the static ANN models developed for databases 1 to 5 are shown together in Table 6-1, and the measures for databases 6 and 7 are shown together in Table 6-2.

Details of the use of the four sequential training stages on all seven databases and the desired criteria used to choose the optimal network structures for of Feedback ANN network models are presented in the following sections.

6.1 Feedback ANN Model Development of Database 1

Feedback ANN model architecture has been designed by considering 8 inputs and 1 output. One of the counted inputs is the initial estimate from the developed static ANN network described in the previous chapter. A total of 300 datasets are used to build the desired database; 157, 72 and 71 datasets are used, respectively, for training, testing and validation purposes. Based on statistical measures such as MRSE, MARE, and R^2, the optimal network structure of the Feedback ANN model for Database 1 was found at 19 hidden nodes and 19,500 iterations. The corresponding accuracy measures for this network are $MRSE_{tr}$= 3.1927, $MARE_{tr}$= 3.409%, R^2_{tr}= 0.9985 (for training database) and $MRSE_{ts}$=4.7437, $MARE_{ts}$= 3.991%, R^2_{ts}=0.9986 (for testing database). The training and testing graphical comparison plots between predicted and actual values for the Feedback ANN model developed for Database 1 are, respectively, shown in Figure 6-2 and Figure 6-3. Also, all the statistical accuracy measures for the training and testing are shown in Table 6-1. After the training and testing procedures using, respectively, 157 and 72 datasets, validation was conducted on the remaining 71 datasets. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 6-4. Once the validation stage is completed, all of the 300 datasets were used to retrain the network at the previously determined optimal structure to obtain the generalized response throughout the 300 datasets. The graphical comparison plot for the 300 datasets is shown in Figure 6-5. Statistical accuracy measures for validation and all data cases are also shown in Table 6-1. As can be seen from the table, Feedback ANN network developed for database 1 has higher validation MRSE value than testing MRSE value as expected. All data MRSE value is lower than testing and validation measures while the value of MARE improved slightly. R^2 values did not change significantly for testing, validation, or all data case. However, MRSE is considered as the main criterion to evaluate the performances of the networks. MRSE value for all data was decreased to a value of 2.1754 while the ones for testing and validation were increased to values of 4.7437 and 5.3192. It can be concluded that the datasets included in validation and testing stages carry important knowledge pertaining to the phenomenon and improved the statistical accuracy measures once included in the all data case. In Table 6-1, as indicated in Chapter 5, the *8-(2-4)-3200-1* notation specifies the determined architecture of the optimum network of Database 1 where each number ,respectively, represents: number of inputs (8), initial number of hidden nodes (2), final number of hidden nodes (4), number of iterations (3200), and number of outputs (1). Final structure of the optimum network is depicted as *8-4-1,* which are, respectively: number of inputs, number of hidden nodes, and number of outputs.

6.2 Feedback ANN Model Development of Database 2

A database consisting of 100 datasets was used to develop the desired Feedback ANN network for Database 2. As noted previously, the databases to be used for modeling are divided into three sub-categories such as training, testing, and validation. For database 2, during the first stage of modeling, 55 datasets are used for training, 23 for testing, and 22 for validation. The boundary of the training datasets was determined by the minimum and maximum of the input and output variables. Therefore, in order to obtain an optimum network with a wider input and output range, minimum and maximum of each input and output variable was considered in training stage. In this case, any input value within the minimum and maximum ranges of the database is applicable to the network. The input vector consisted of 8 parameters and the output vector consisted of 1 parameter were considered to be used in model development of database 2.

Similar network development procedure, to the one used in Chapter 5 for this database, was also followed here. Accordingly, the optimal structure for the Feedback ANN network of Database 2 was chosen at 4 hidden nodes and 1100 iterations. A graphical comparison of training stage between the predicted and the actual values is depicted in Figure 6-6. Feedback ANN network for training stage yielded a mean root square error, $MRSE_{tr}$ of 0.4059, mean absolute relative error, $MARE_{tr}$ of 4.0208%, and coefficient of determination, R^2_{tr} of 0.6596. Similarly, graphical comparison of testing stage is shown in Figure 6-7 and statistical accuracy measures for this network are $MRSE_{ts}$ of 0.9944, $MARE_{ts}$ of 5.368%, and R_{ts}^2 of 0.2694.

To further evaluate the optimal network, 22 datasets are used to validate the network. Figure 6-8 presents the graphical comparison of the predicted and the actual values. Corresponding statistical measures are given in Table 6-1. It can be seen that the validation MRSE is higher than the testing MRSE as oppose to those noted for the static ANN network. Once the validation stage is completed, all 100 datasets were used to retrain the network at the optimal structure. It can be concluded from the graphical plot in Figure 6-9 and the statistical accuracy measures in Table 6-1 that using entire database to retrain the network significantly improved the statistical accuracy measures. Overall, performance of the Feedback ANN network is better than that noted for the static ANN network.

6.3 Feedback ANN Model Development of Database 3

To develop Feedback ANN model for database 3, a total of 126 datasets were used. Sixty three and 32 of total datasets were, respectively, considered as training and testing datasets. The remaining 31 datasets were included in the validation stage after the optimal network was determined. An effort to develop a Feedback ANN network for database 3 was initiated with 13 inputs and 1 output. The best performing network structure was obtained at 5 hidden nodes and 100 iterations. The training and testing statistical measures for training and testing stages are shown in Table 6-1 and the graphical comparison plots are depicted in Figure 6-10 and Figure 6-11. As can be observed from the table and the graphical plots, the training and testing stage produced good accuracy. Validation was conducted on 31 datasets, after the training and testing stages. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 6-12. The statistical accuracy measures for this network are $MRSE_{val}$= 188.8319, $MARE_{val}$= 12.942%, and R^2_{val}= 0.7766. Once the validation stage is finalized, all of the 126 datasets were used to retrain the network at the optimal structure. The statistical accuracy measures for this network are $MRSE_{all}$= 52.9530, $MARE_{all}$= 9.985%, and R^2_{all}= 0.9466. The graphical comparison plot for the 126 datasets is shown in Figure 6-13. The resulting statistical accuracy measures for all Feedback ANN network modeling stages are given in Table 6-1.

The statistical measures and the plots indicate that the Feedback ANN network for database 3 has performed well during the training stage, but the testing stage produced higher MRSE value, as was expected. Similarly, MRSE value is even higher for the validation case, which is about 4.7 times higher than the training MRSE. Additionally, MARE error is about 1.45 times higher than the training MARE. When all data combined and the network was retrained, the statistical accuracy measures noticeably improved. It should be noted that the error increase from training MRSE to validation MRSE by Feedback ANN network is less than that noted for static ANN (i.e. 4.7 versus 7.16 times). Similarly, the error increase for MARE by Feedback ANN is less than the one noted for static ANN (i.e. 1.45 versus 2.4 times).

6.4 Feedback ANN Model Development of Database 4

To develop Feedback ANN network for Database 4, a total of 265 datasets; 133, 66, and 66 sub-datasets were, respectively, used for training, testing, and validation. The input vector consisted of 7 parameters, including the one from static ANN network. To properly characterize the phenomenon, the Feedback ANN network approach with four sequential modeling stages were followed for database 4. In this case, the optimal network structure was reached at 3 hidden nodes and 19900 iterations where the network performed is best. Feedback ANN network for training stage yielded a mean root square error, $MRSE_{tr}$ of 70.0604, mean absolute relative error, $MARE_{tr}$ of 20.825%, and coefficient of determination, R_{tr}^2 of 0.8485. Similarly, statistical accuracy measures for the testing stage are $MRSE_{ts}$ of 102.3868, $MARE_{ts}$ of 22.496%, and R^2_{ts} of 0.8369. Graphical comparisons of testing and validation stages are, respectively, shown in Figure 6-14 and Figure 6-15. As can be seen from the graphical plots and the statistical accuracy measures depicted in Table 6-1, a good agreement between actual and predicted values is apparent. The predictions for validation datasets and all datasets case were plotted against their corresponding actual values, respectively, in Figure 6-16 and Figure 6-17. Good agreement between the predictions and the actual values can also be evaluated numerically in Table 6-1 in terms of statistical accuracy measures. Even though error by validation datasets are typically expected to be higher than those by testing, for database 4 statistical accuracy measures are improved in validation stage. MARE values by validation datasets are even lower than that of by all data case. However, all data MRSE value is the lowest compared to the previous stages (i.e. training, testing, and validation).

6.5 Feedback ANN Model Development of Database 5

Database 5 has been built by considering 325 datasets; 163, 81, and 81 datasets that are for training, testing, and validation purposes. As previously mentioned in Chapter 4, database 5 has two outputs. For this reason, four sequential stages for static ANN model development process were conducted twice to arrive at two desired prediction models for two outputs. Only one output was considered at a time for optimized networks to be able to generate individual outputs. The optimal network structure for the model 1 was finalized at 4 hidden nodes and

19300 iterations. The corresponding accuracy measures of model 1 are $MRSE_{tr}$=0.2014, R^2_{tr}=0.9963, $MARE_{tr}$=0.183% (for training database) and $MRSE_{ts}$ =0.4768, R^2_{ts}=0.9841, $MARE_{ts}$= 0.226% (for testing database). The optimal network for Model 2 was reached at 3 hidden nodes and 14100 iterations. The corresponding accuracy measures of model 2 are $MRSE_{tr}$ =0.6391, R^2_{tr}=0.9293, $MARE_{tr}$=1.201% (for training database) and $MRSE_{ts}$ =0.8311, R^2_{ts}=0.9345, $MARE_{ts}$= 1.036% (for testing database). Training MRSE value for model 1 increased about 137% in testing while training MRSE value for model 2 increased about 30% in testing. R^2 value for model 1 has decreased about 1.2% while R^2 value for model 2 has increased about 0.6%. These numbers are very similar to those noted for the associated static networks presented in chapter 5.

The training and testing plots for model 1 are shown in Figure 6-18 and Figure 6-19. In the plots, the training and testing predictions are very close to the 45 degree line, which means that predicted values are very close to actual values. Similarly the training and testing plots for model 2 are also given in Figure 6-20 and Figure 6-21. The corresponding statistical measures of model 1 and model 2 are presented in Table 6-1.

After the optimal network was determined, the validation for model 1 and model 2 was conducted on 81 datasets. The validation plots for model 1 and model 2 are, respectively, given in Figure 6-22 and Figure 6-23. After the validation stage is concluded, all of the 325 datasets were used to re-train the network at the optimal structure. The comparison plots of model 1 and model 2 for the 325 datasets are, respectively, shown in Figure 6-24 and Figure 6-25. The resulting statistical accuracy measures for the validation and the all data cases are depicted in Table 6-1. All data MRSE statistical measures for both model 1 and model 2 have the best results compared to their previous stages. Similar to the case noted for static ANN, overall, model 1 has better statistical accuracy measures than model 2, even though model 2 has reasonably good accuracy measures.

6.6 Feedback ANN Model Development of Database 6

In this database, 210 datasets were divided into sub-databases: 105, 53, and 52 to be used, respectively, for training, testing, and validation purposes Similar to the case in Chapter 5, Feedback ANN model development process was repeated for each output and each model developed was associated with its corresponding output number (i.e. Model 1 for output1). The number of sub-databases was kept the same for the three models.

In this case, Feedback ANN network for Model 1 was determined at 2 hidden nodes and 10100 iterations. The optimal network was chosen among many other networks based on the noted statistical accuracy measures. This network structure provided the optimal connection weights for the anticipated predictions. The training and testing accuracy measures for model 1 are presented in Table 6-2 along with the corresponding plots shown in Figure 6-26 and Figure 6-27. According to the statistical measures, the optimal network performed well in the training stage as well as in the testing stage. However, MRSE value of the training, 0.0029 deteriorated to 0.0062 for the testing stage, which corresponds to a 113.8% increase in error. For the validation stage, the statistical measures changed slightly; however, for the all data stage, MRSE improves to a value of 0.0025, which translates into about 13.8% reduction in error, while MARE value increased about 15.8%. Even though R^2 value seems to decrease from 0.9050 for training stage to a value of 0.8561 for all data stage, the main criterion, which is MRSE, has a reasonable reduction in error. All the statistical measures for the validation and all data stages can be found in Table 6-2 and their associated plots are, in the given order, presented in Figure 6-28 and Figure 6-29.

The same database used to develop Feedback ANN network for model 1 was utilized for Model 2 by considering 17 inputs, including the one from static ANN network, and 1 output. The optimal network for model 2 was reached at 3 hidden nodes 15300 iterations. The accuracy of training and testing stages for the selected network architecture is given in Table 6-2 and the graphical evaluation plots are depicted in Figure 6-30 and Figure 6-31. Validation and all data stages were sequentially followed by the training and testing stages. Figure 6-32 and Figure 6-33, which are the plots for validation and all data predictions, indicate reasonably good correlation between the actual and predicted values. A good agreement between the actual and predicted values can easily be evaluated from Table 6-2. Again, the all data stage attains the best accuracy measures compared to testing and validation stages.

Similar procedure was followed to develop Feedback ANN network for model 3. The corresponding statistical accuracy measures were obtained with a structure of 3 hidden nodes and at 3100 iterations. Table 6-2 presents all the statistical measures for this model. Also, corresponding graphical comparisons are represented in Figure 6-34, Figure 6-35, Figure 6-36, and Figure 6-37. Even though some scatter is noted in these plots, most of the data is predicted reasonably well.

6.7 Feedback ANN Model Development of Database 7

Last database was utilized in this chapter to develop a 16 input 1 output Feedback ANN network is Database 7, which consists of 792 datasets divided into 396, 198, and 198 datasets for training, testing, and validation purposes. All the statistical accuracy measures from the training and testing stages were considered to choose the optimal network structure, which was obtained at 5 hidden nodes and 5200 iterations. The accuracy plots of the network with the optimal structure are illustrated in Figure 6-38 and Figure 6-39. The plots validate the good correlation between actual and predicted results, even though there seem to be few outliers at the higher end as was noted for the Static case. As can be observed from Table 6-2, the developed Feedback ANN network has reasonably good statistics where $MRSE_{tr}$= 1.1518, $MARE_{tr}$= 11.734%, and R^2_{tr}= 0.9850. Even though statistical accuracy measures for testing and validation stages deteriorated slightly, they are still considerably good. The accuracy of how well the validation datasets were predicted can be seen in Figure 6-40 and the corresponding statistics are shown in Table 6-2. Combining all datasets and retraining the network has improved the model statistics where the MRSE value of 1.1518 for training was reduced to a value of 0.8011, which can be translated into a 30% reduction. All data case for MARE and R^2 values were changed slightly but the biggest improvement was obtained for the MRSE value. All data predictions are graphically depicted in Figure 6-41 and the statistical accuracy quantities are given in Table 6-2. As a result, Feedback ANN network for database 7 was effectively developed and the statistical accuracy measures are adequate.

6.8 Concluding Remarks

In this chapter, a new ANNs approach is introduced and used on seven databases. This new ANN method utilizes the output from static ANN model along with the input parameters to generate new improved results. In other words, architecture of Feedback ANN network was developed by considering the output from static ANN model and the input parameters, which were used to develop static ANN models as well. Basically, this new method was proposed to improve the accuracy measures of the static ANN models developed in Chapter 5 by including the initial estimates from the static ANN model as another input.

As seen from the graphical results depicted in Figure 6-2 to Figure 6-41 and the accuracy measures of the developed Feedback ANN models for each database listed in Table 6-1 and Table 6-2, the Feedback ANN models have reliable results. Moreover, the statistical accuracy measures, such as MARE, R^2, and MRSE, from static ANN modeling network and Feedback ANN network modeling have been evaluated to determine the improvements/reductions in the statistical accuracy measures of the proposed Feedback ANN modeling process. The reduction of MARE for the seven databases can be seen in Table 6-3. The reduction of MARE for six databases (i.e., Databases 1, 2, 3, 4, 6, and 7) is ranging from 3% to 36%. Feedback ANN approach has shown a negative reductions on output 1 of database 5 and zero reduction on output 2 of database 5, which means that Feedback ANN did not outperform the static network on the this database. The possible reason that Feedback ANN was not effective for this database is that the correlation between inputs and outputs are highly linear and the static ANN model may have ultimately discovered the relationship. This is why the Feedback ANN approach could not improve the prediction any further.

The improvement of R^2 for the seven databases is ranging from 0% to 29%. R^2 values for most of the databases have changed slightly. The improvement results are depicted in Table 6-4. Databases 1, 5, and 7 did not indicate any improvement or reduction. Database 2 and output 1 of database 6 have shown the most improvement in R^2, which are, respectively, 17% and 29%. The improvements for the rest of the databases are nearly 1%. Similarly, the reduction of MRSE for the seven databases has been evaluated and the results are shown in Table 6-5. The reduction of MRSE for six databases (i.e. Databases 1, 2, 3, 4, 6, and 7) is ranging from 2% to

35%. Feedback ANN approach has shown a negative reduction on output 1 of database 5 and zero reduction on output 2 of database 5 as was noted for the performance of MARE.

As can be seen from the results presented in Table 6-3, Table 6-4, and Table 6-5, Feedback ANN approach has, overall, successfully improved the prediction accuracy of six databases, whose correlations between inputs and output(s) considered as non-linear. Database 5 is the only database, which has fallen out of this category and did not show either positive reduction in error measurements or positive improvements in R^2 values. In this case, static ANN network for database 5 has reached to its saturation point, where the network cannot perform any better and no further accuracy improvement is expected.

It can be observed from the architecture of the developed networks for the seven databases in Table 6-6, Feedback ANN networks have improved the optimal network structure compared to static ANN networks. In Table 6-6, for example, the notation for database 1 is shown as *7-(8-19)-19500-1,* which represents the determined architecture of the optimum network where each number stands for, in the written order: number of inputs (7), initial number of hidden nodes (8), final number of hidden nodes (19), number of iterations (19500), and number of outputs (1). The optimal network structure for most of the databases was found at a lesser hidden nodes or lesser iterations if the number of hidden nodes remained same. The only database that the number of hidden nodes was increased of is Database 2. Database 1 has the most noticeable change from 19 hidden nodes to 4 hidden nodes with better overall statistical accuracy measures. The number of hidden nodes for Database 3, Database 4, Database 5 – Output 2, Database 6 – Output 1, and Database 7 has decreased. Database 5 – Output 1, database 6 – output 2 and output 3 did not have any change in terms of hidden nodes. However, the number of iterations to arrive at their optimal network has decreased. Consequently, it can be inferred that Feedback ANN network approach has improved the statistical measures as well as the optimal network architectures by either decreasing the number of hidden nodes and/or the number of iterations.

6.9 Figures and Tables

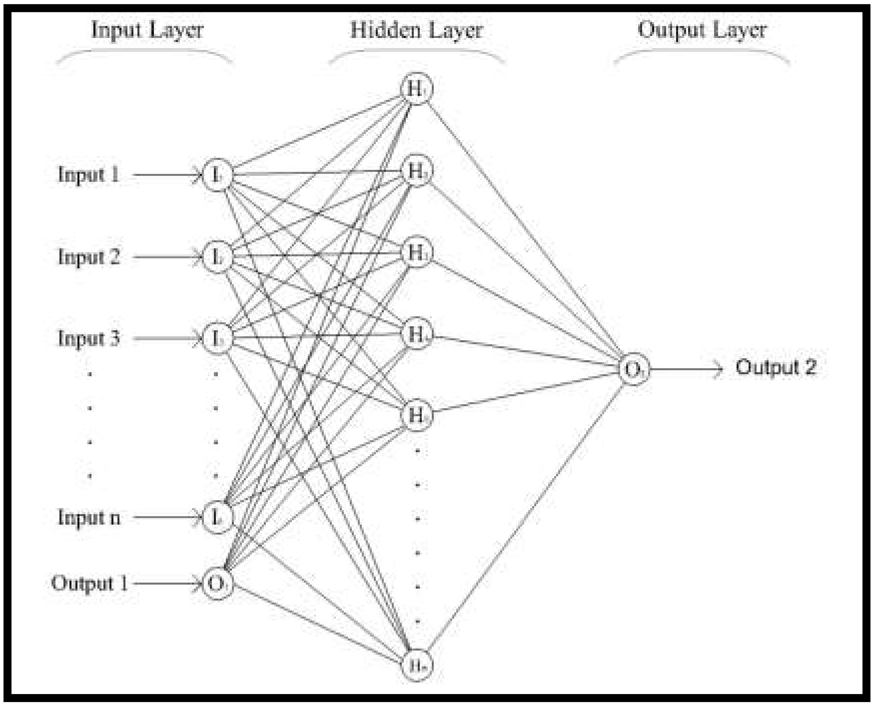

Figure 6-1 Architecture of a Feedback ANN Network

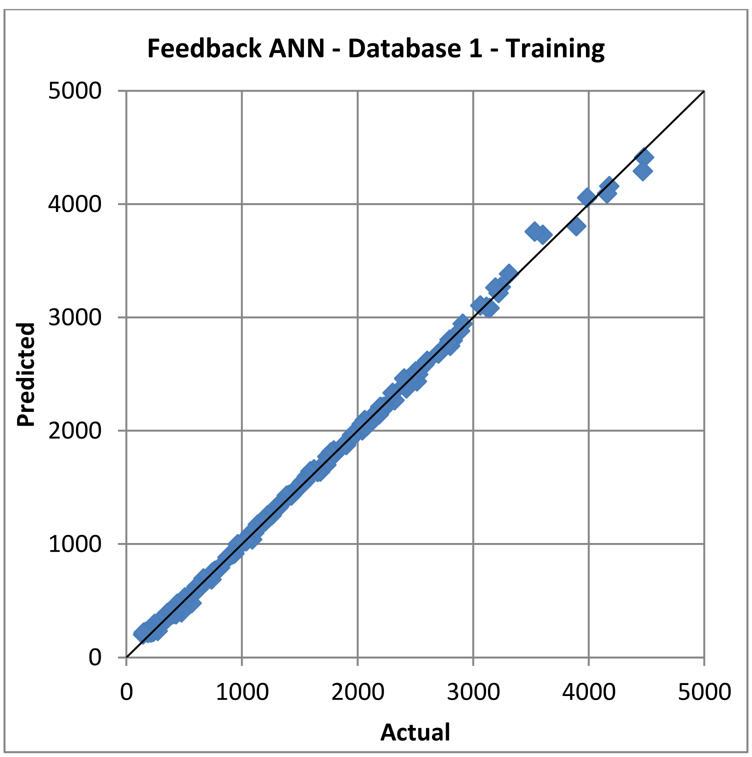

Figure 6-2 Feedback ANN Network Training Accuracy of Database 1

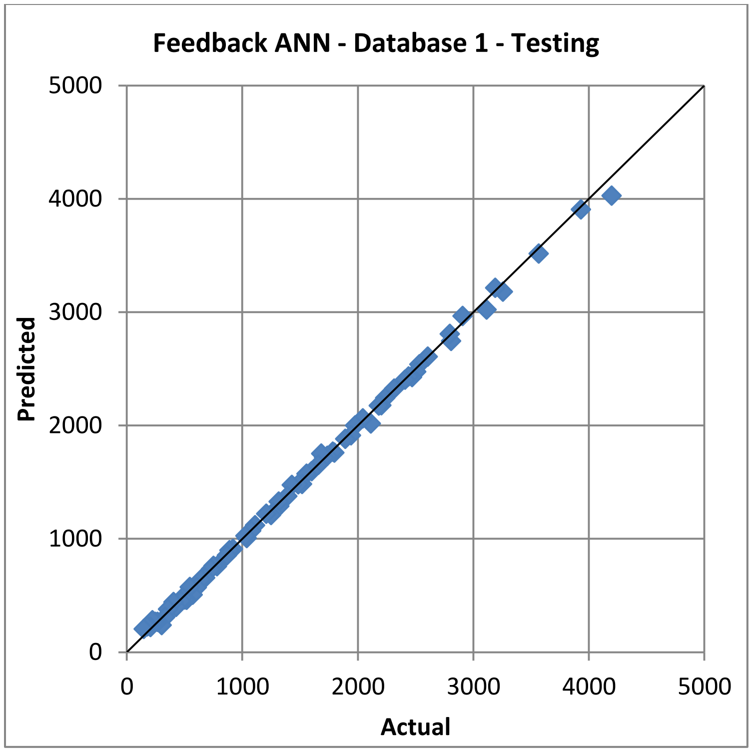

Figure 6-3 Feedback ANN Network Testing Accuracy of Database 1

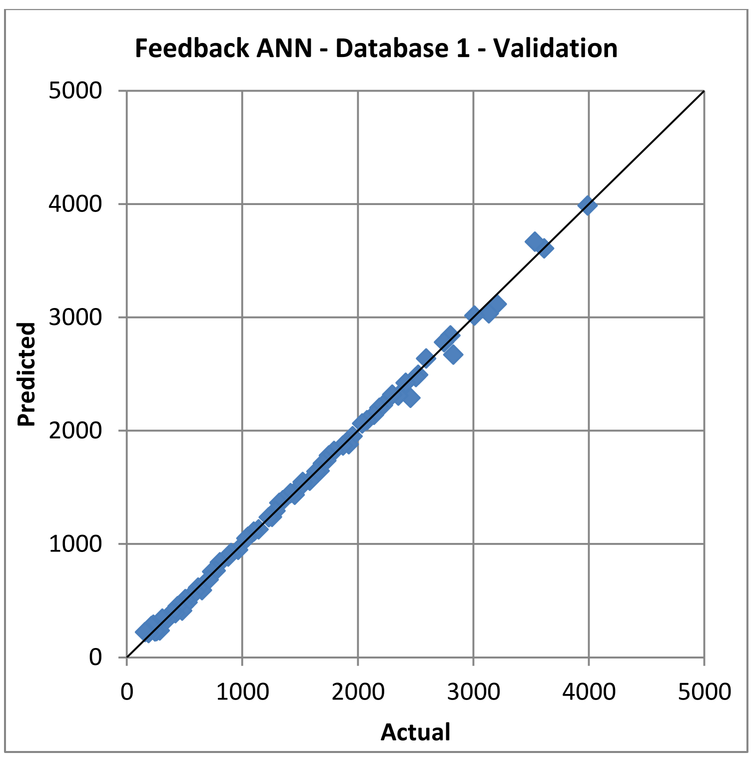

Figure 6-4 Feedback ANN Network Validation Accuracy of Database 1

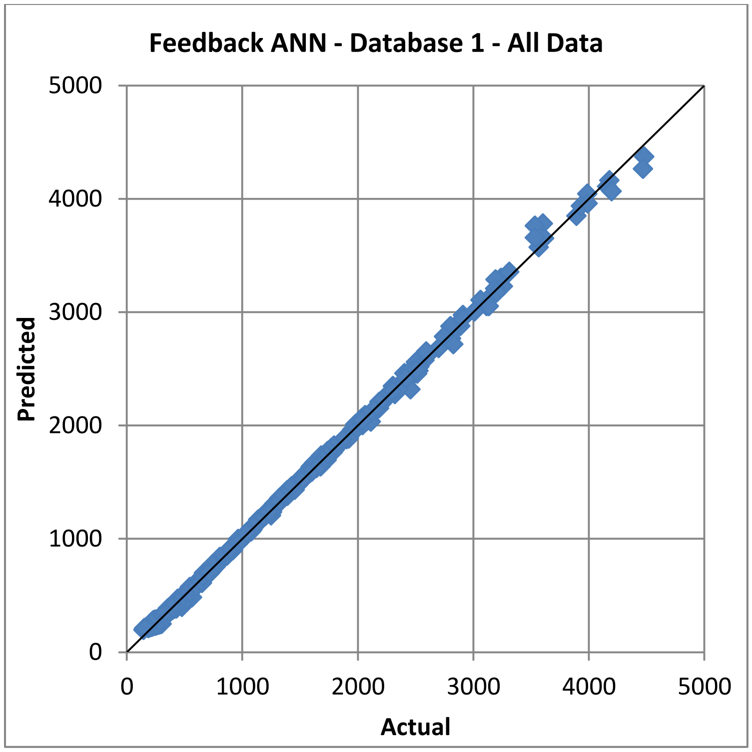

Figure 6-5 Feedback ANN Network All Data Accuracy of Database 1

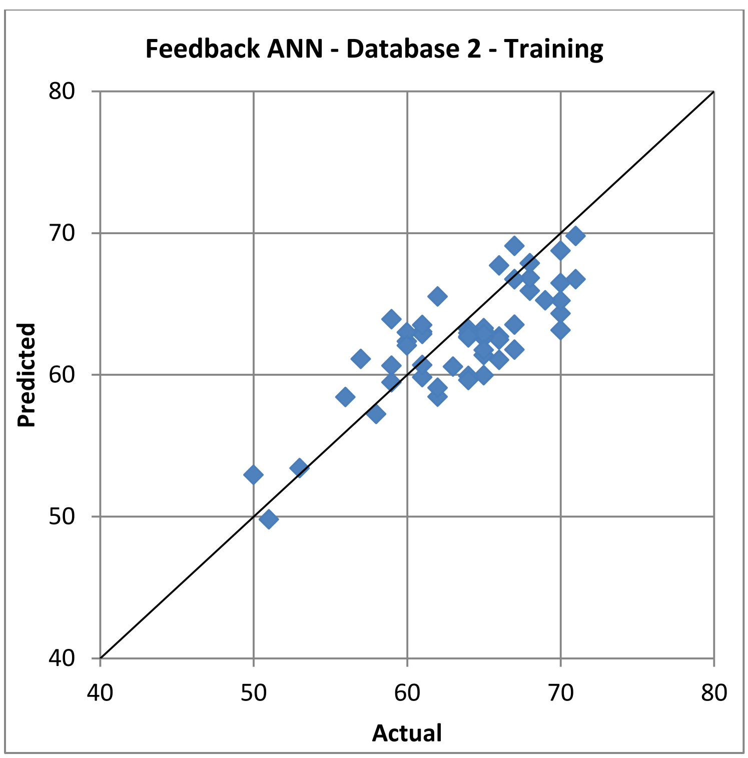

Figure 6-6 Feedback ANN Network Training Accuracy of Database 2

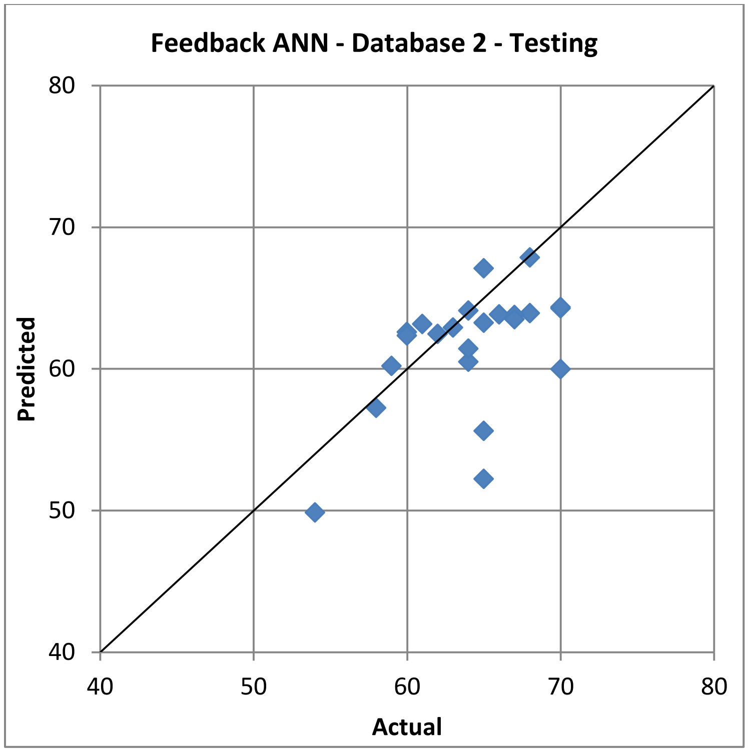

Figure 6-7 Feedback ANN Network Testing Accuracy of Database 2

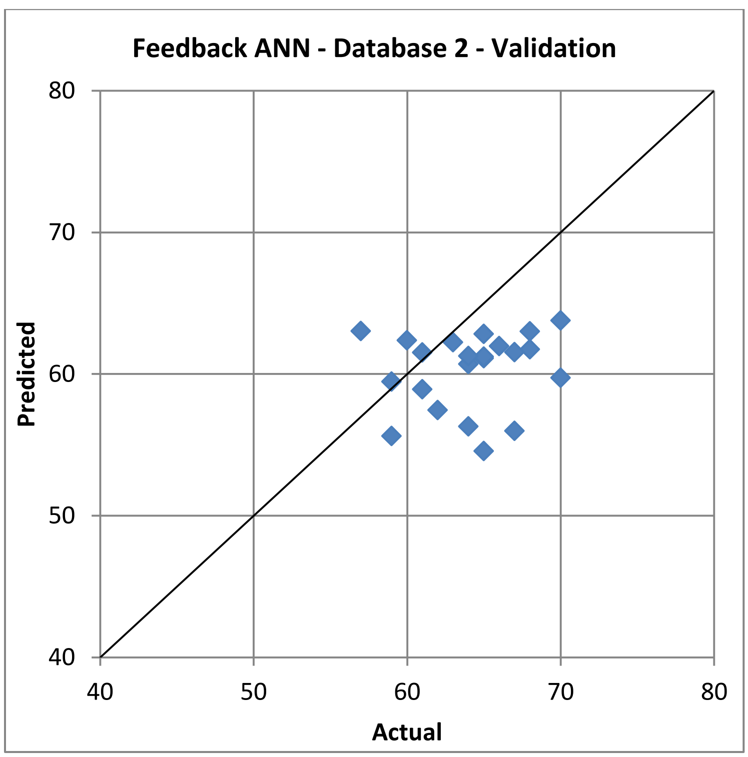

Figure 6-8 Feedback ANN Network Validation Accuracy of Database 2

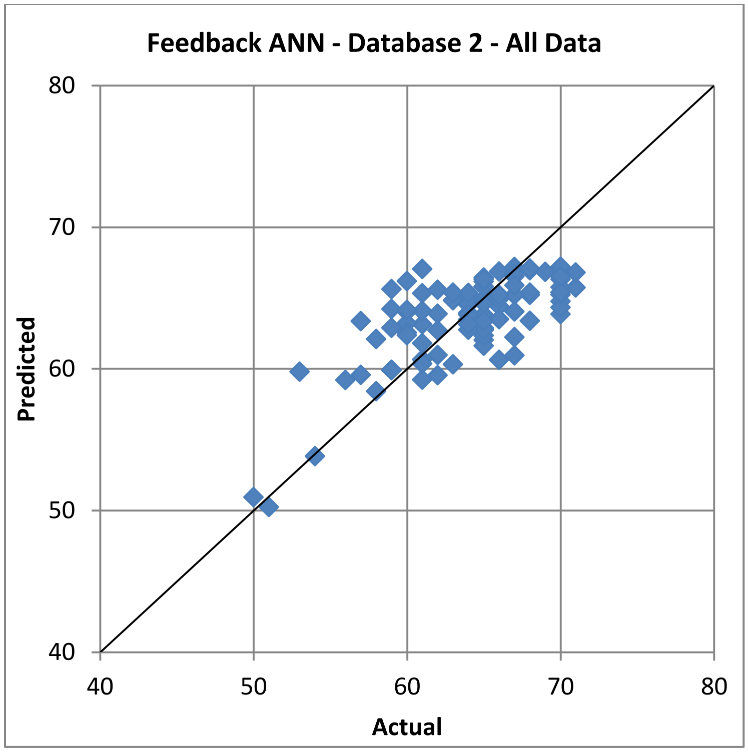

Figure 6-9 Feedback ANN Network All Data Accuracy of Database 2

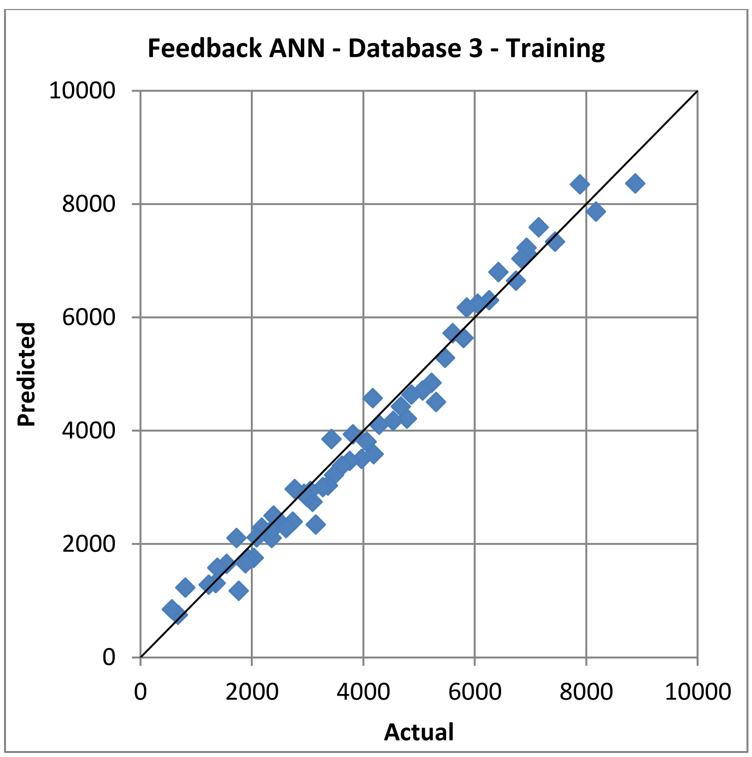

Figure 6-10 Feedback ANN Network Training Accuracy of Database 3

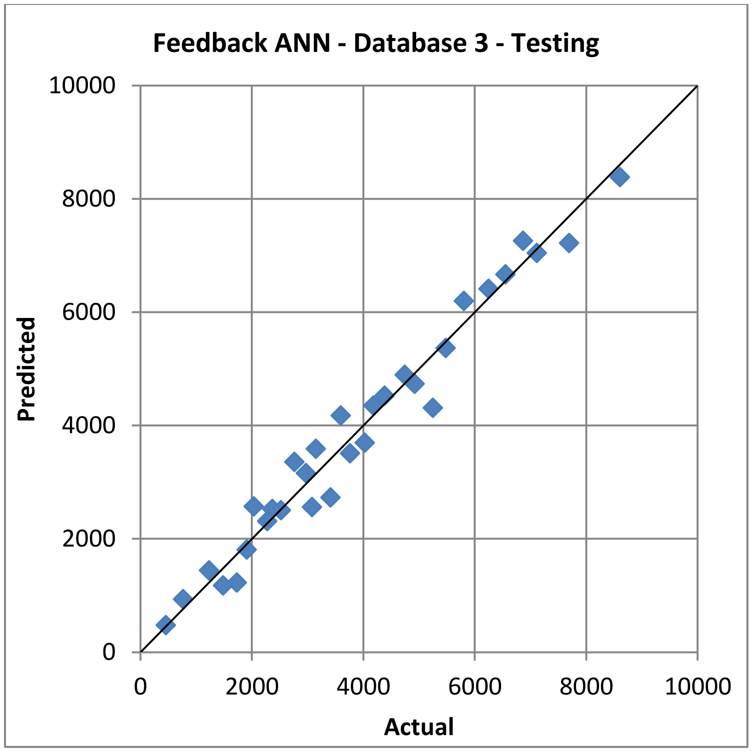

Figure 6-11 Feedback ANN Network Testing Accuracy of Database 3

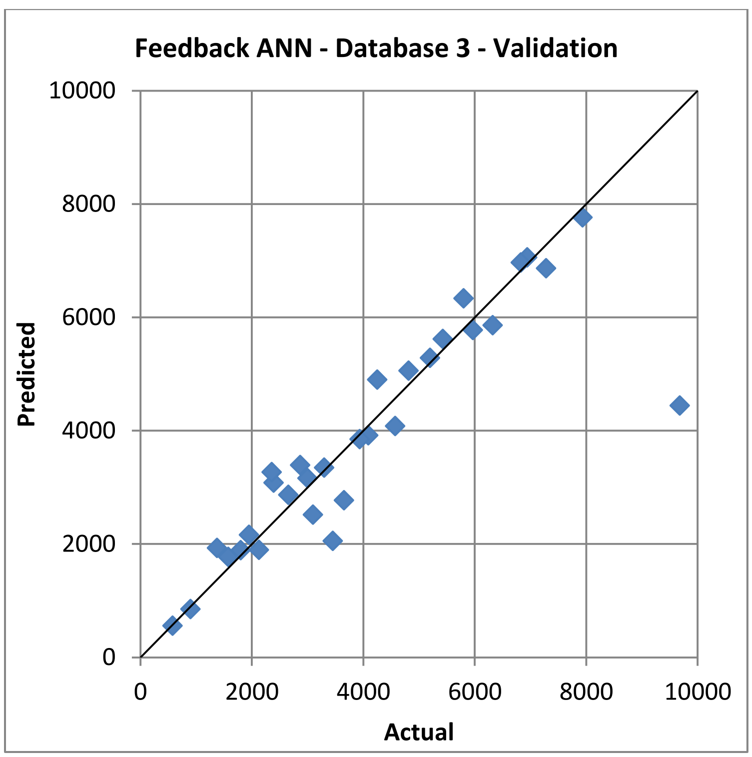

Figure 6-12 Feedback ANN Network Validation Accuracy of Database 3

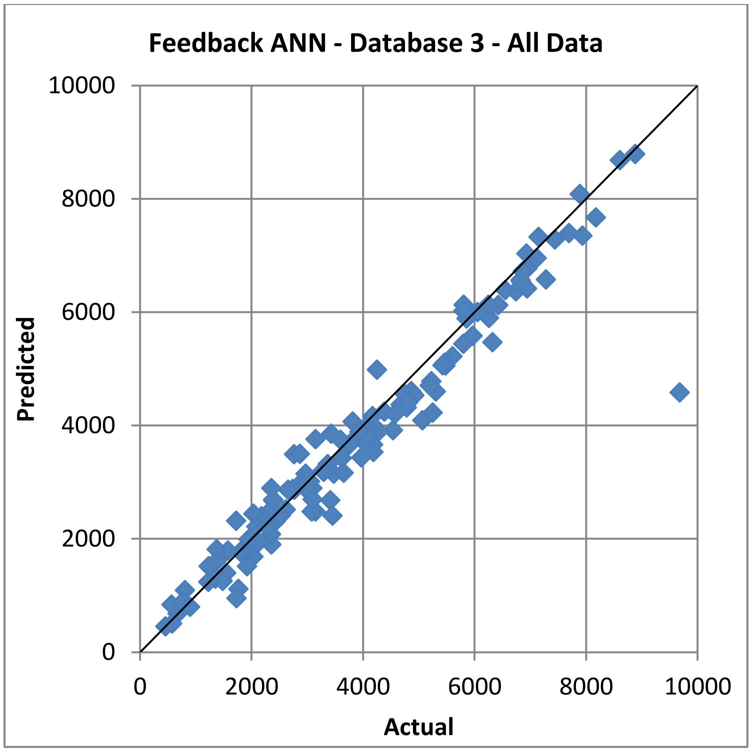

Figure 6-13 Feedback ANN Network All Data Accuracy of Database 3

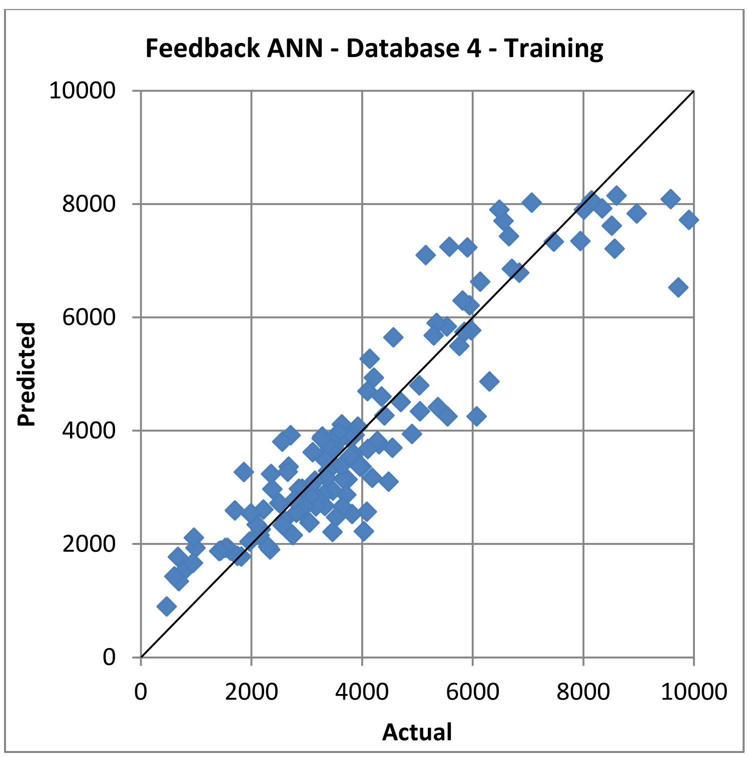

Figure 6-14 Feedback ANN Network Training Accuracy of Database 4

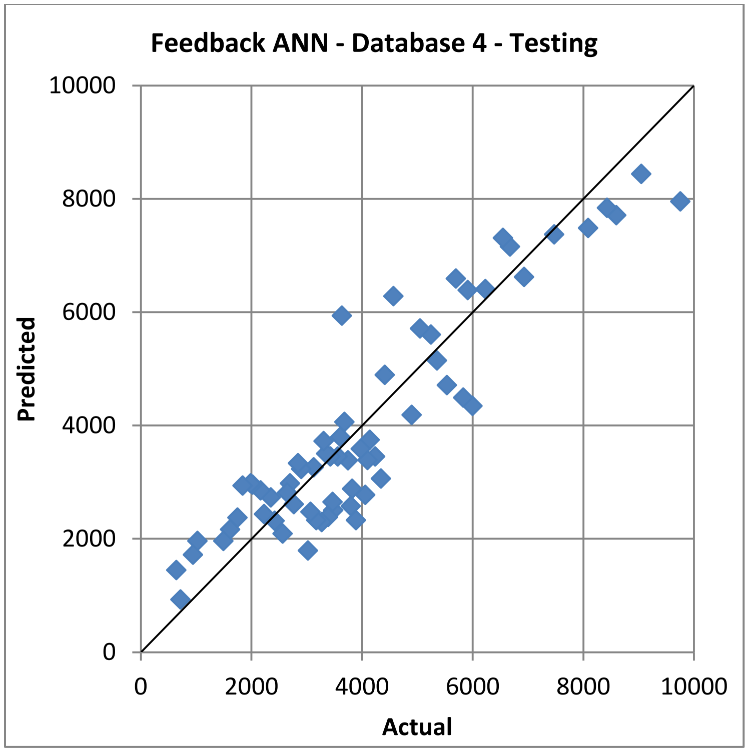

Figure 6-15 Feedback ANN Network Testing Accuracy of Database 4

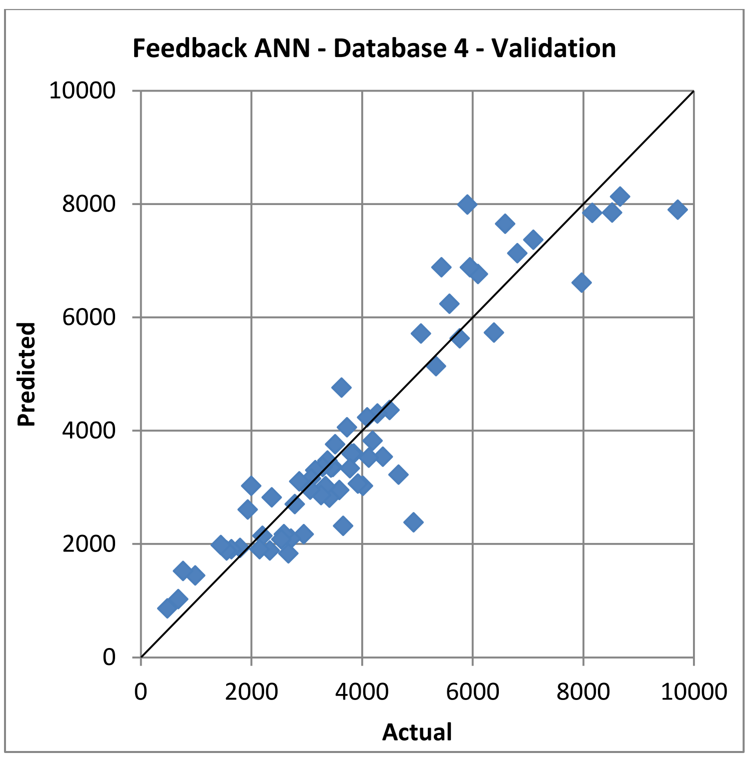

Figure 6-16 Feedback ANN Network Validation Accuracy of Database 4

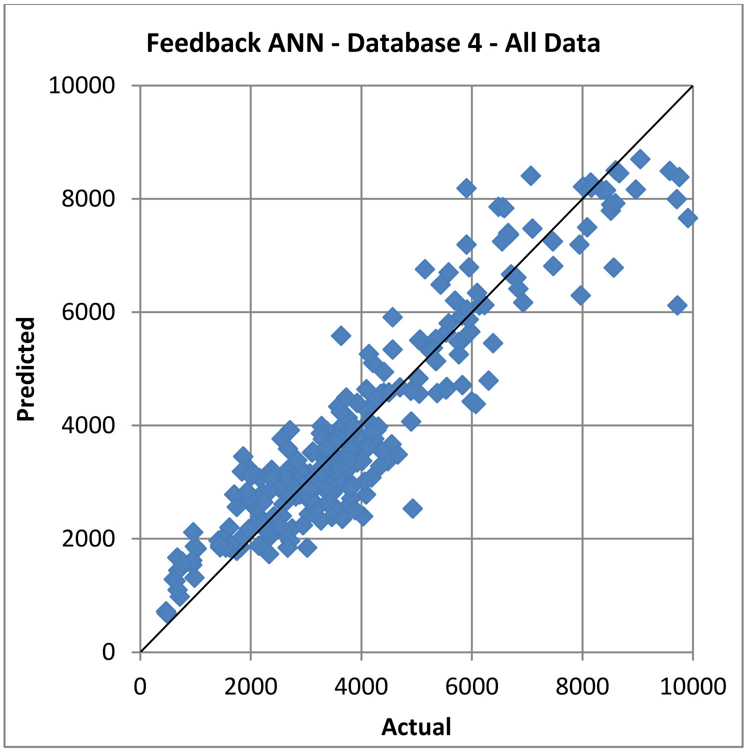

Figure 6-17 Feedback ANN Network All Data Accuracy of Database 4

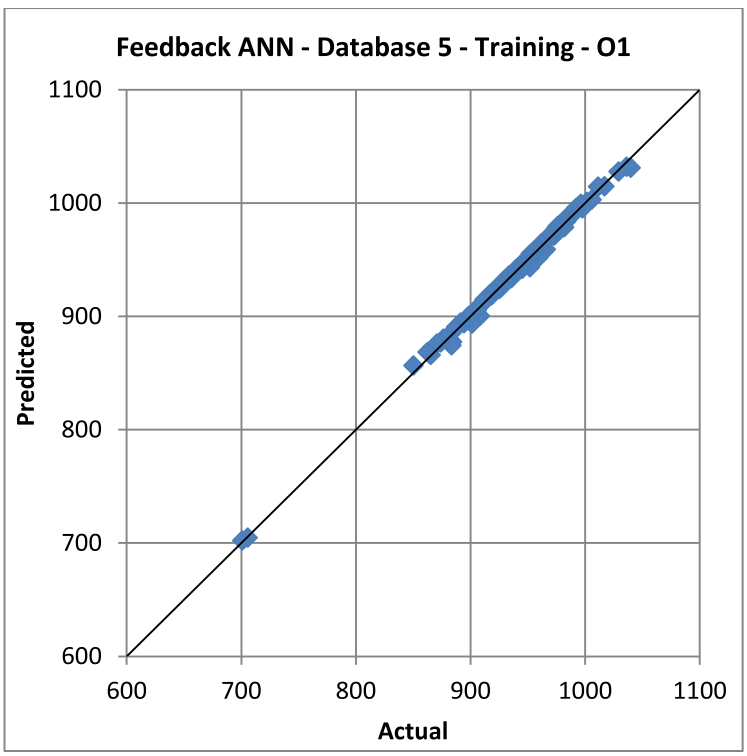

Figure 6-18 Feedback ANN Network Training Accuracy of Database 5, Output 1

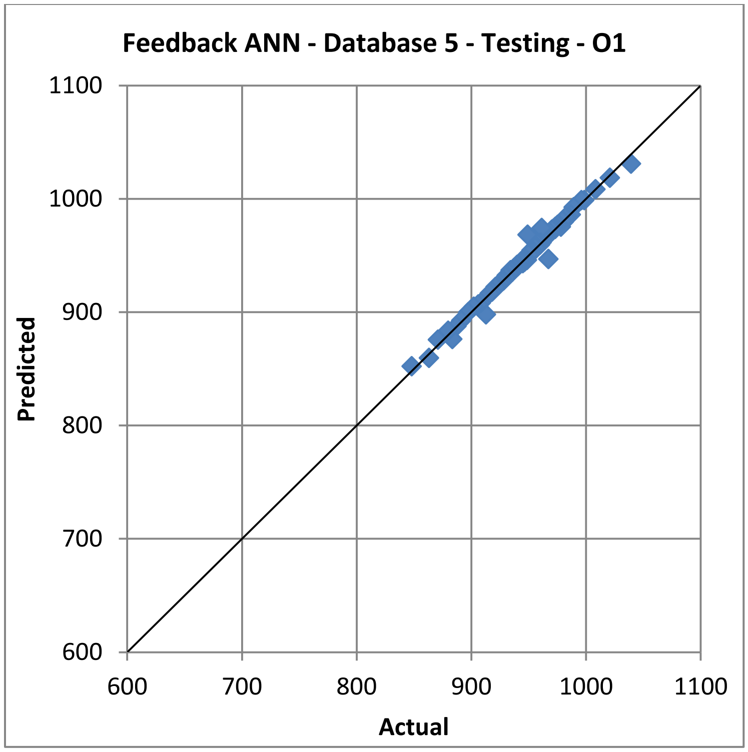

Figure 6-19 Feedback ANN Network Testing Accuracy of Database 5, Output 1

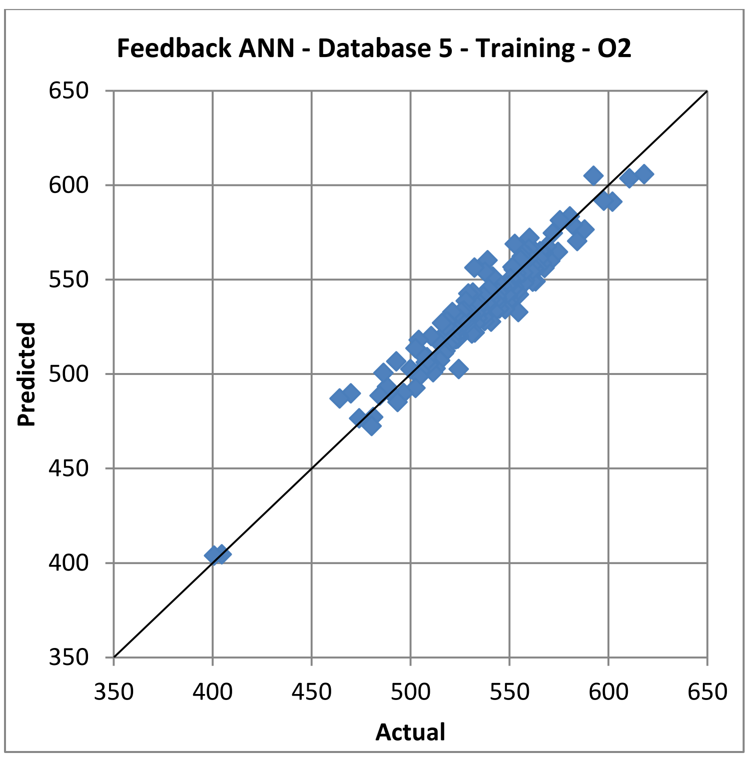

Figure 6-20 Feedback ANN Network Training Accuracy of Database 5, Output 2

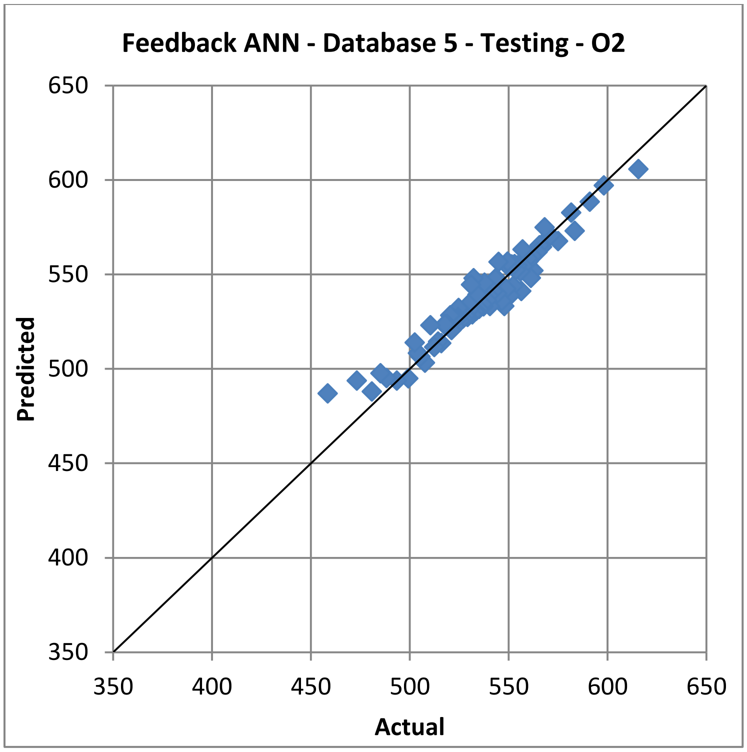

Figure 6-21 Feedback ANN Network Testing Accuracy of Database 5, Output 2

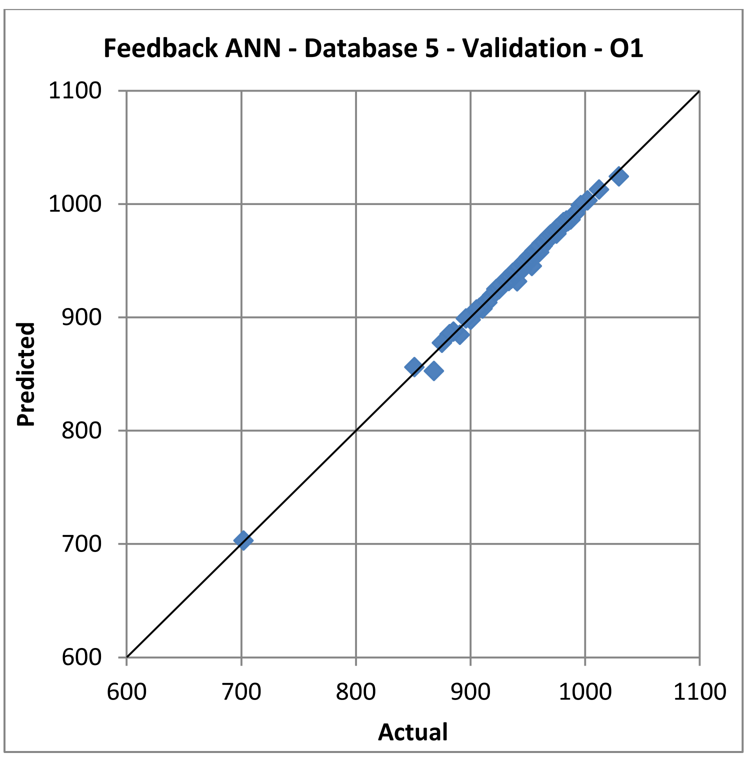

Figure 6-22 Feedback ANN Network Validation Accuracy of Database 5, Output 1

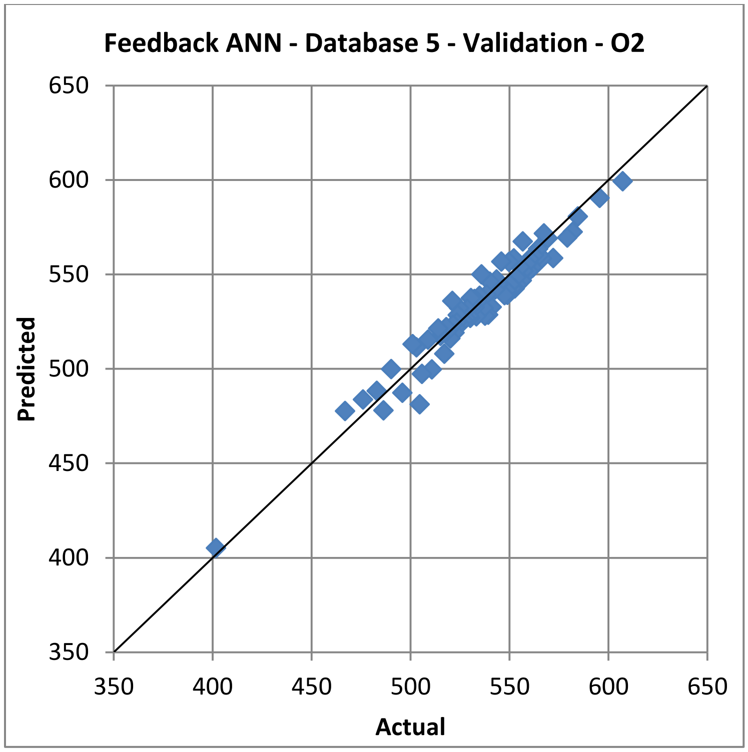

Figure 6-23 Feedback ANN Network Validation Accuracy of Database 5, Output 2

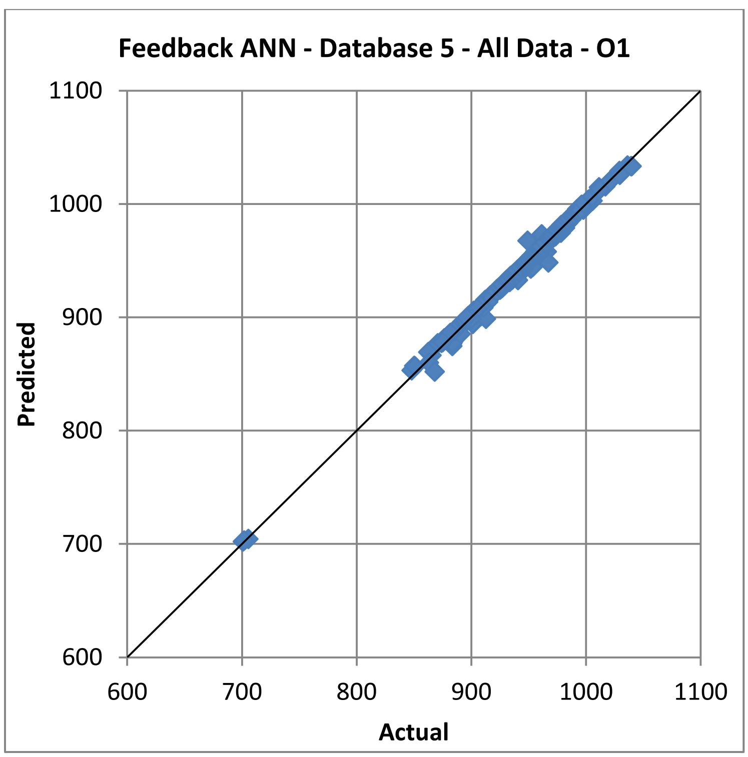

Figure 6-24 Feedback ANN Network All Data Accuracy of Database 5, Output 1

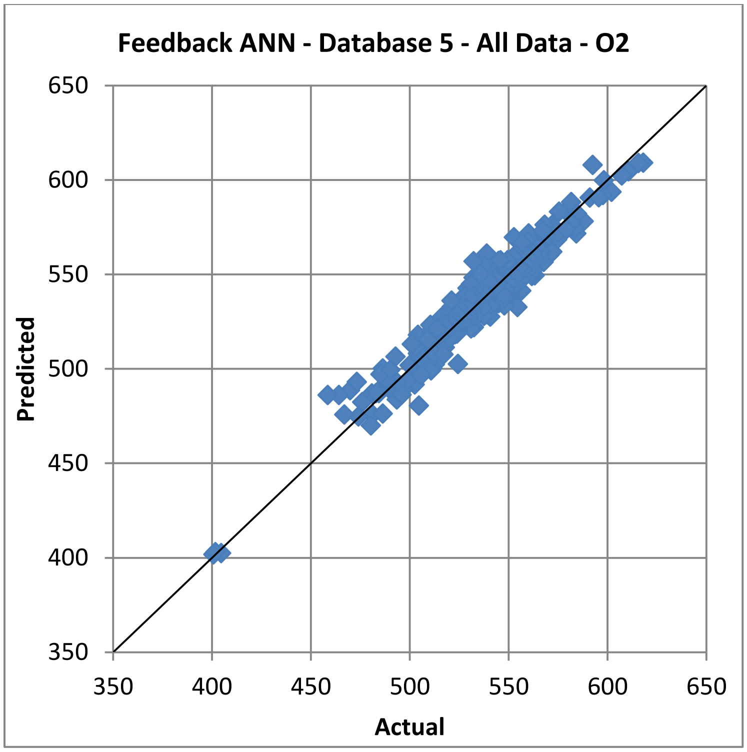

Figure 6-25 Feedback ANN Network All Data Accuracy of Database 5, Output 2

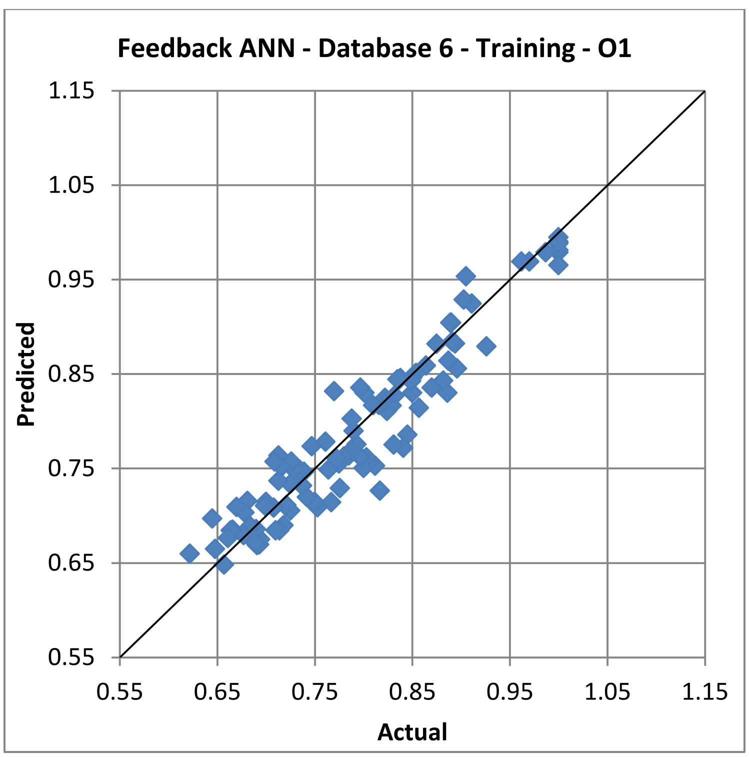

Figure 6-26 Feedback ANN Network Training Accuracy of Database 6, Output 1

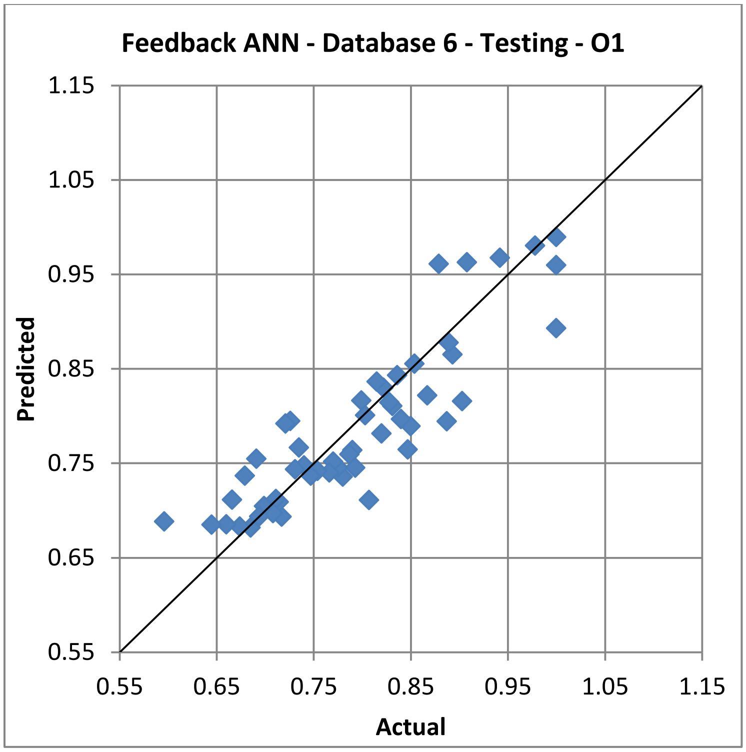

Figure 6-27 Feedback ANN Network Testing Accuracy of Database 6, Output 1

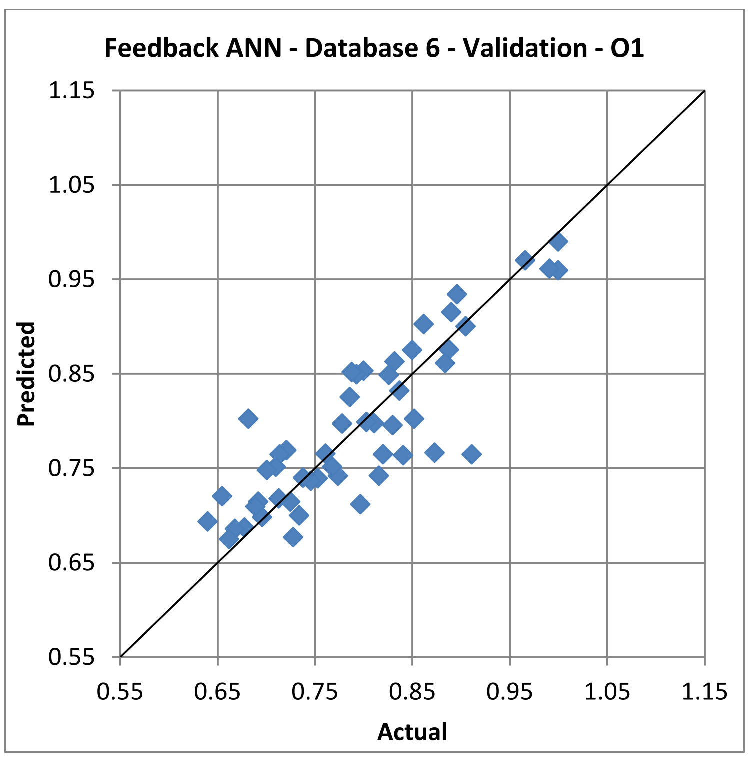

Figure 6-28 Feedback ANN Network Validation Accuracy of Database 6, Output 1

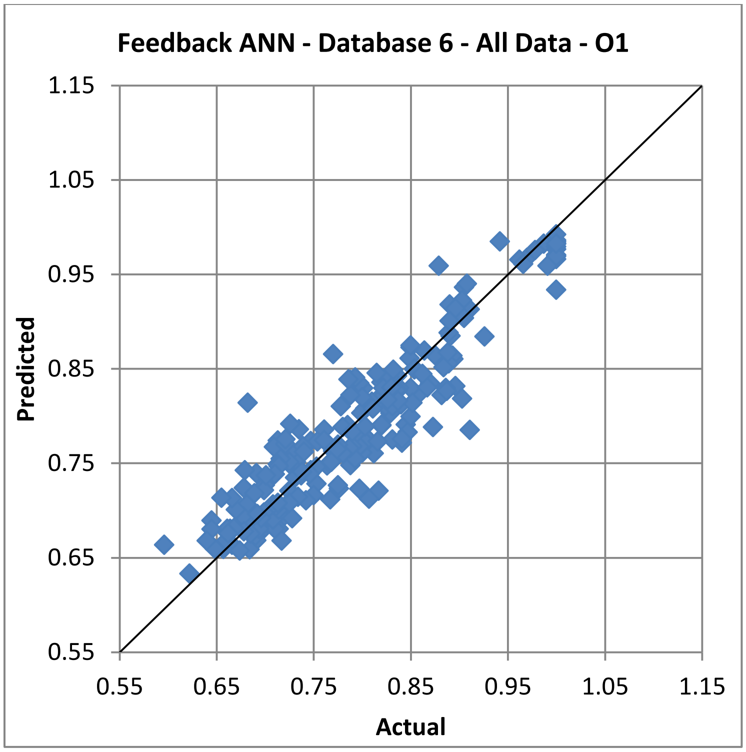

Figure 6-29 Feedback ANN Network All Data Accuracy of Database 6, Output 1

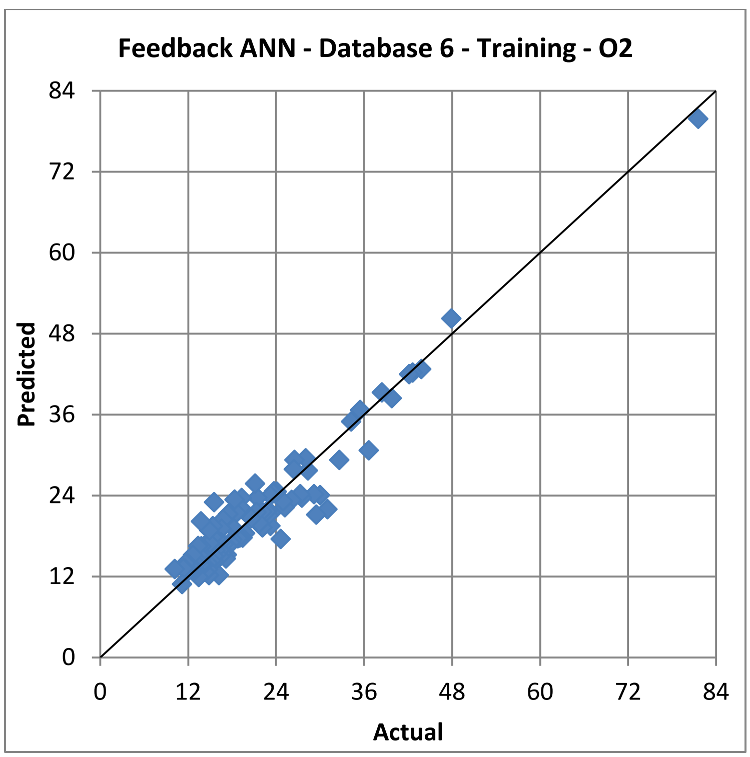

Figure 6-30 Feedback ANN Network Training Accuracy of Database 6, Output 2

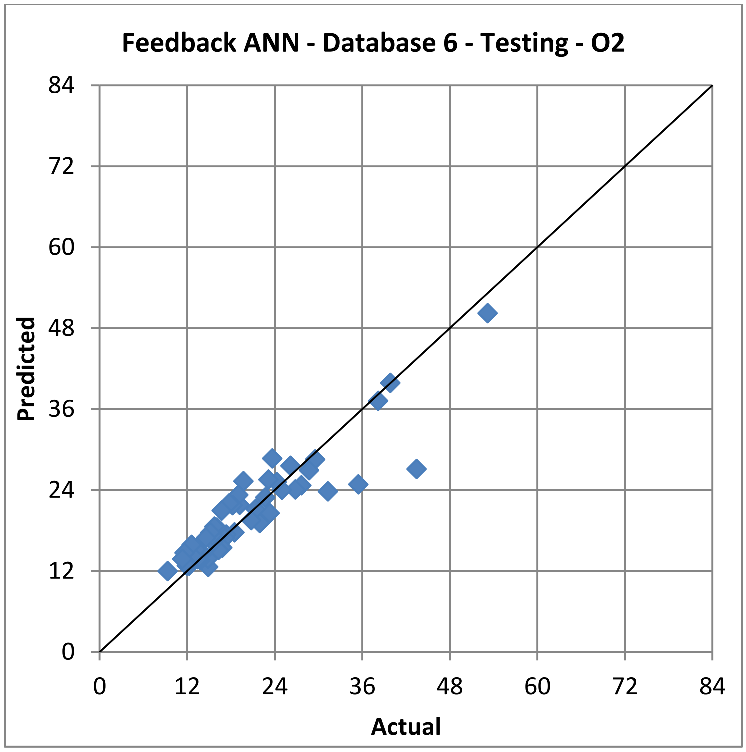

Figure 6-31 Feedback ANN Network Testing Accuracy of Database 6, Output 2

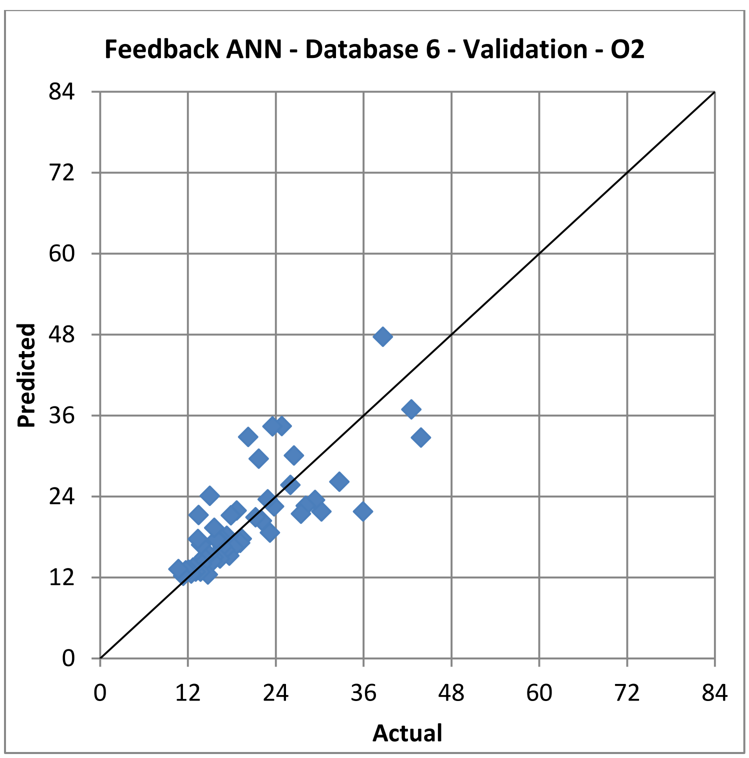

Figure 6-32 Feedback ANN Network Validation Accuracy of Database 6, Output 2

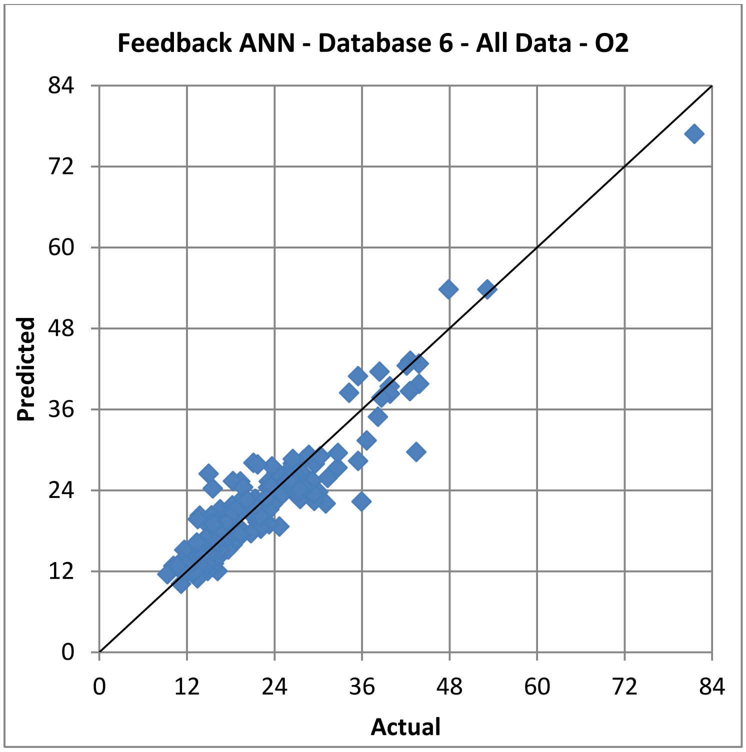

Figure 6-33 Feedback ANN Network All Data Accuracy of Database 6, Output 2

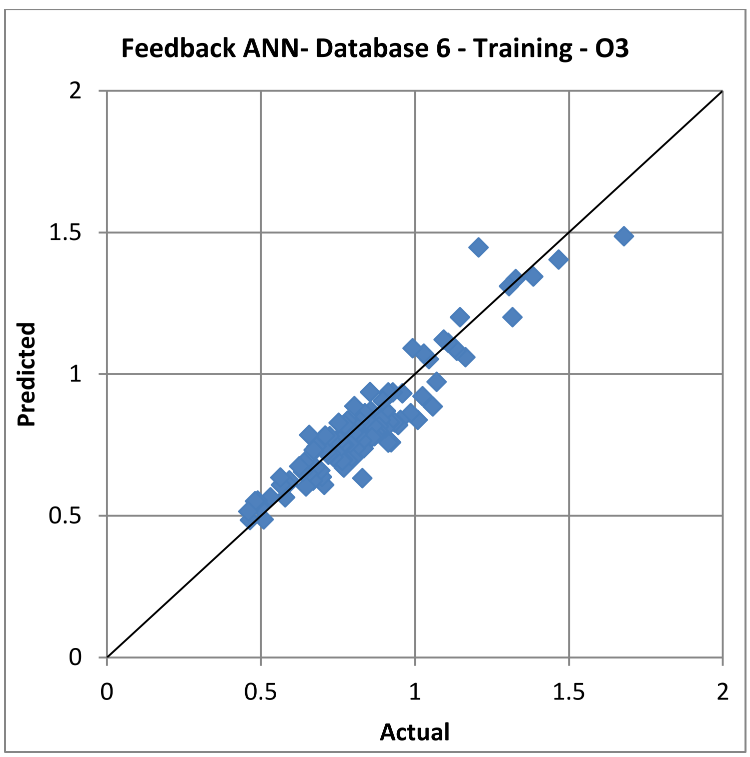

Figure 6-34 Feedback ANN Network Training Accuracy of Database 6, Output 3

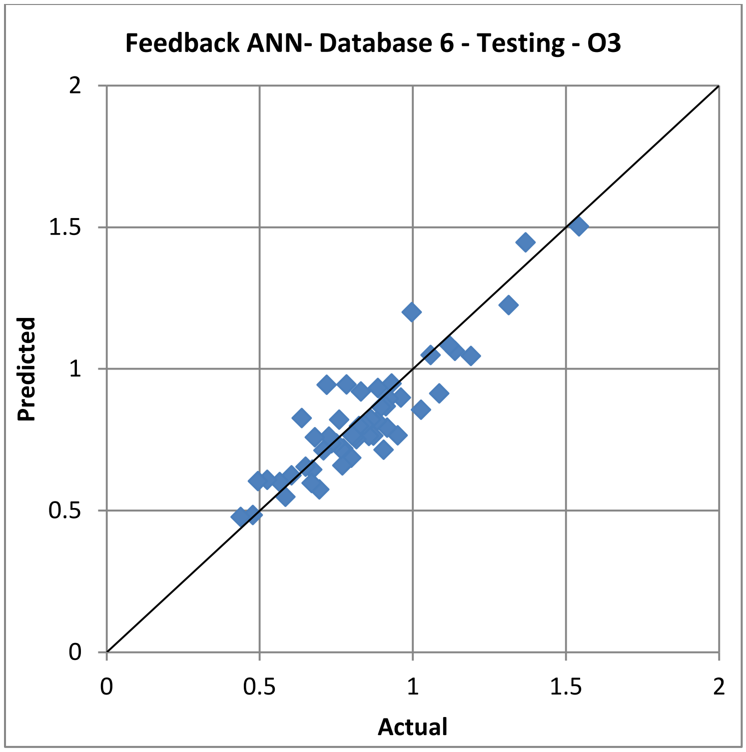

Figure 6-35 Feedback ANN Network Testing Accuracy of Database 6, Output 3

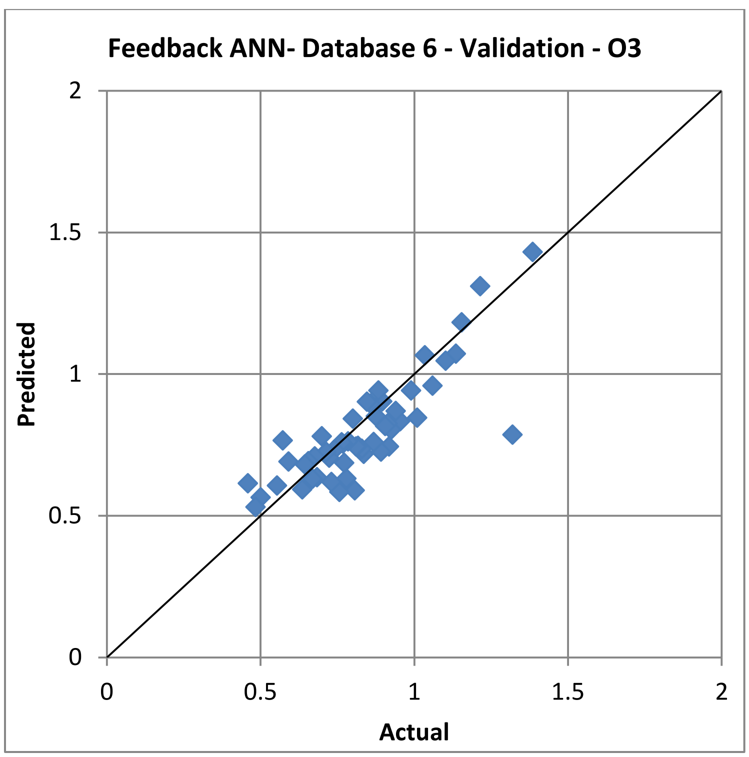

Figure 6-36 Feedback ANN Network Validation Accuracy of Database 6, Output 3

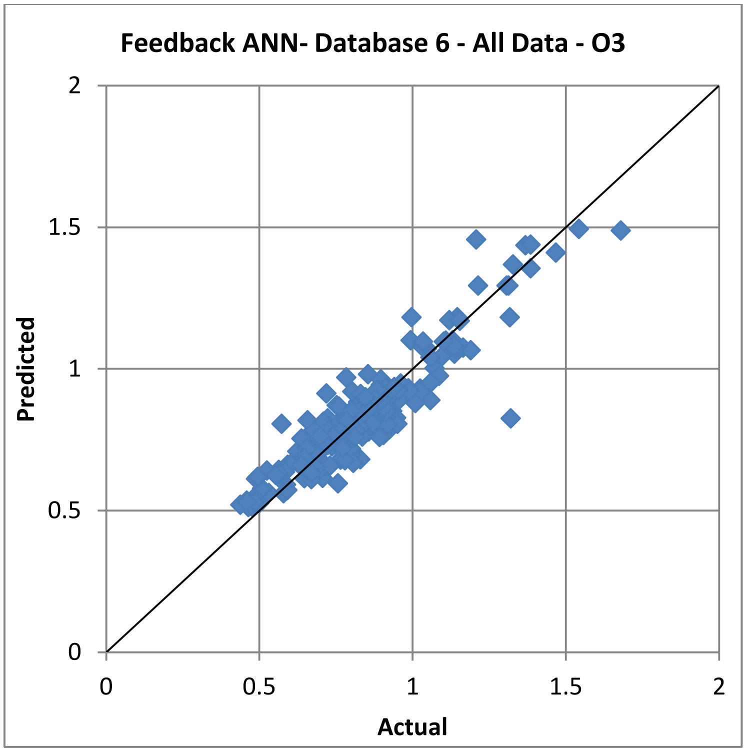

Figure 6-37 Feedback ANN Network All Data Accuracy of Database 6, Output 3

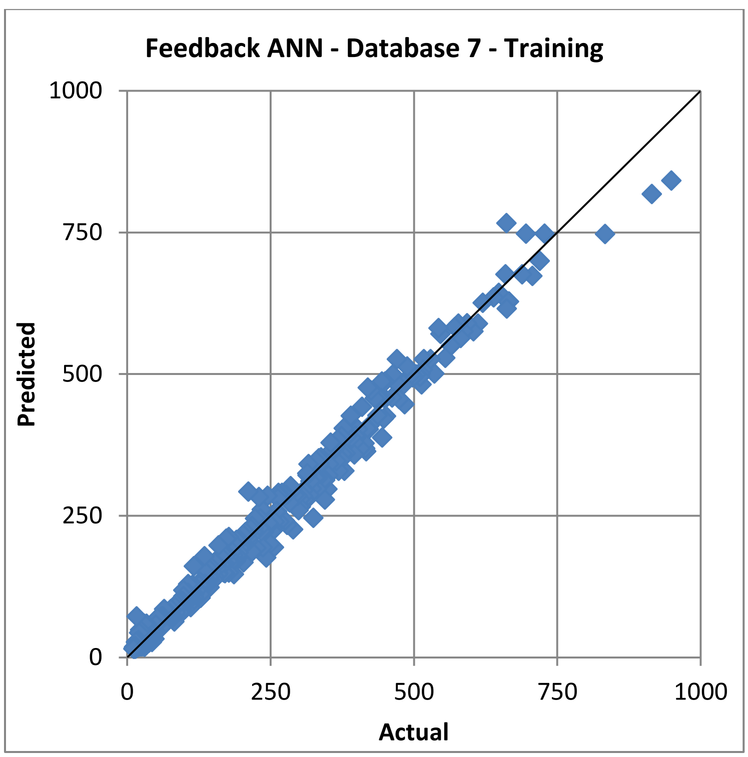

Figure 6-38 Feedback ANN Network Training Accuracy of Database 7

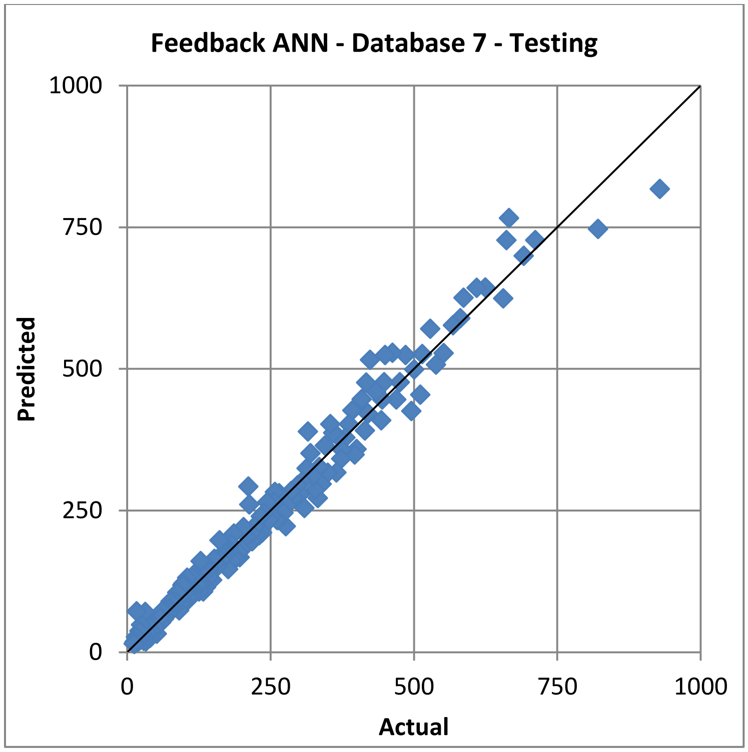

Figure 6-39 Feedback ANN Network Testing Accuracy of Database 7

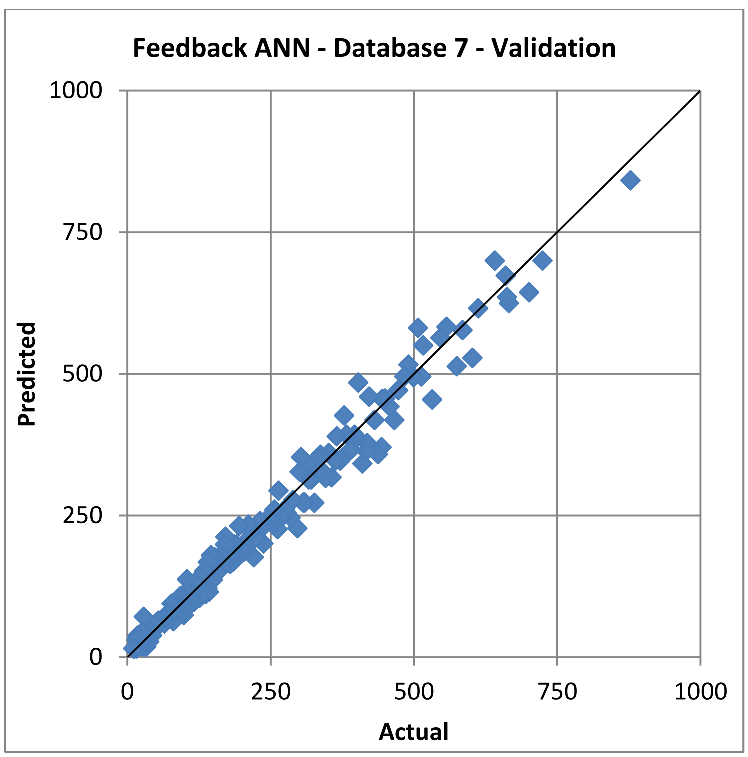

Figure 6-40 Feedback ANN Network Validation Accuracy of Database 7

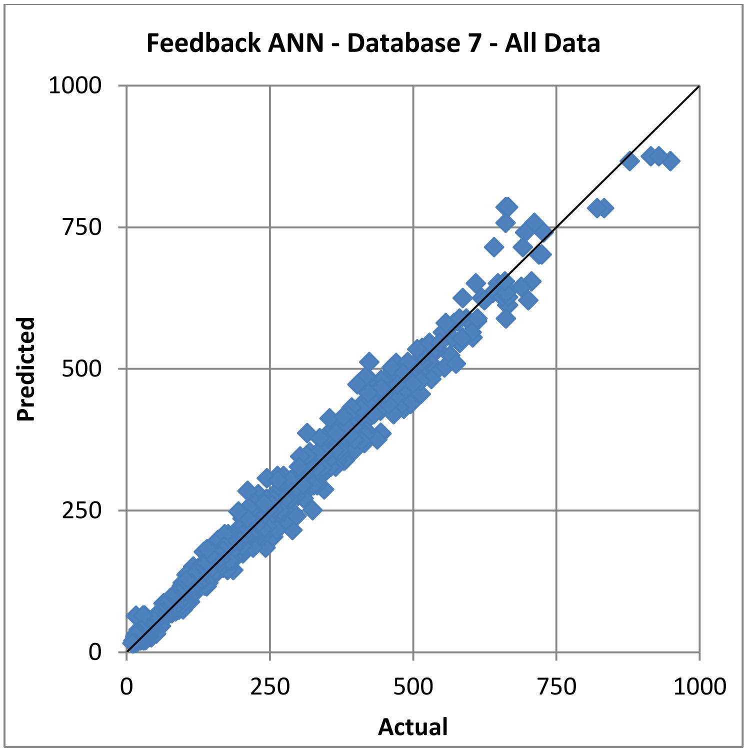

Figure 6-41 Feedback ANN Network All Data Accuracy of Database 7

Table 6-1 Statistical Accuracy of Feedback ANN Network Models for Database 1 to Database 5

Accuracy Measures		FEEDBACK ANN MODELS					
		Database 1	Database 2	Database 3	Database 4	Database 5	
		Output 1	Output 1	Output 1	Output 1	Output 1	Output 2
		8-(2-4)-3200-1	8-(3-4)-1100-1	13-(1-5)-100-1	7 -(2-3)-19900-1	4-(2-4)-19300-1	4-(3-3)-14100-1
TR	MARE	3.409	4.021	8.869	20.825	0.183	1.201
	R^2	0.9985	0.6596	0.9849	0.8485	0.9963	0.9293
	MRSE	3.1927	0.4059	40.5326	70.0604	0.2014	0.6391
TS	MARE	3.991	5.368	9.549	22.496	0.226	1.036
	R^2	0.9986	0.2694	0.9797	0.8369	0.9841	0.9345
	MRSE	4.7437	0.9944	64.9743	102.3868	0.4768	0.8311
VAL	MARE	4.116	7.161	12.942	17.999	0.205	1.138
	R^2	0.9979	0.0237	0.7766	0.8626	0.9951	0.9425
	MRSE	5.3192	1.1833	188.8319	93.7436	0.3303	0.8179
ALL DATA	MARE	3.281	3.699	9.985	19.470	0.190	1.129
	R^2	0.9986	0.5314	0.9466	0.8613	0.9942	0.9329
	MRSE	2.1754	0.2979	52.9530	46.9162	0.1703	0.4269
FINAL STRUCTURE		**8 - 4 - 1**	**8 - 4 - 1**	**13 - 5 - 1**	**7 - 3 - 1**	**4 - 4 - 1**	**4 - 3 - 1**

Table 6-2 Statistical Accuracy of Feedback ANN Network Models for Databases 6 and 7

Accuracy Measures		FEEDBACK ANN MODELS			
		Database 6			Database 7
		Output 1	Output 2	Output 3	Output 1
		17-(1-2)-10100-1	17 -(2-3)-15300-1	17 -(1-3)-3100-1	16-(4-5)-5200-1
TR	MARE	2.993	10.731	7.136	11.734
	R^2	0.9050	0.9239	0.8854	0.9850
	MRSE	0.0029	0.2690	0.0075	1.1518
TS	MARE	4.327	11.682	9.253	13.206
	R^2	0.7823	0.8372	0.8178	0.9780
	MRSE	0.0062	0.5004	0.0131	1.9500
VAL	MARE	4.594	16.611	10.400	11.459
	R^2	0.7331	0.6032	0.6862	0.9816
	MRSE	0.0066	0.7122	0.0164	1.7721
ALL DATA	MARE	3.467	11.099	7.749	11.504
	R^2	0.8561	0.8844	0.8467	0.9848
	MRSE	0.0025	0.2162	0.0057	0.8011
FINAL STRUCTURE		**17 - 2 - 1**	**17 - 3 - 1**	**17 - 3 - 1**	**16 - 5 - 1**

Table 6-3 Reduction of Mean Absolute Relative Error (MARE) for seven databases

Database #	OUTPUT	MARE		
		Static ANN	Feedback ANN	Reduction
Database 1	Output 1	4.069	3.281	19%
Database 2	Output 1	3.9681	3.6991	7%
Database 3	Output 1	12.719	9.985	21%
Database 4	Output 1	20.359	19.470	4%
Database 5	Output 1	0.186	0.190	-2%
	Output 2	1.125	1.129	0%
Database 6	Output 1	5.416	3.467	36%
	Output 2	11.529	11.099	4%
	Output 3	8.009	7.749	3%
Database 7	Output 1	12.380	11.504	7%

Table 6-4 Improvement of Coefficient of Determination (R^2) for seven databases

Database #	Output	R^2		
		Static ANN	Feedback ANN	Improvement
Database 1	Output 1	0.9984	0.9986	0%
Database 2	Output 1	0.4554	0.5314	17%
Database 3	Output 1	0.9364	0.9466	1%
Database 4	Output 1	0.8549	0.8613	1%
Database 5	Output 1	0.9944	0.9942	0%
	Output 2	0.9333	0.9329	0%
Database 6	Output 1	0.6612	0.8561	29%
	Output 2	0.8721	0.8844	1%
	Output 3	0.8377	0.8467	1%
Database 7	Output 1	0.9831	0.9848	0%

Table 6-5 Reduction of Mean Root Square Error (MRSE) for seven databases

Database #	Output	MRSE		
		Static ANN	Feedback ANN	Reduction
Database 1	Output 1	2.3740	2.1754	8%
Database 2	Output 1	0.3203	0.2979	7%
Database 3	Output 1	63.7835	52.9530	17%
Database 4	Output 1	47.9782	46.9162	2%
Database 5	Output 1	0.1676	0.1703	-2%
	Output 2	0.4255	0.4269	0%
Database 6	Output 1	0.0038	0.0025	35%
	Output 2	0.2276	0.2162	5%
	Output 3	0.0059	0.0057	3%
Database 7	Output 1	0.8466	0.8011	5%

Table 6-6 Comparison of the network arhictecture between Static ANN and Feedback ANN

Database #	Output	Optimal Structure	
		Static ANN	Feedback ANN
Database 1	Output 1	7-(8-19)-19500-1	8-(2-4)-3200-1
Database 2	Output 1	7-(2-3)-3100-1	8-(3-4)-1100-1
Database 3	Output 1	12-(2-6)-200-1	13-(1-5)-100-1
Database 4	Output 1	6-(2-7)-20000-1	7 - (2-3)-19900-1
Database 5	Output 1	3-(2-4)-19800-1	4-(2-4)-19300-1
	Output 2	3-(3-4)-19500-1	4-(3-3)-14100-1
Database 6	Output 1	16-(3-3)-5000-1	17-(1-2)-10100-1
	Output 2	16-(1-3)-13000-1	17-(2-3)-15300-1
	Output 3	16-(1-3)-19400-1	17-(1-3)-3100-1
Database 7	Output 1	15-(4-7)-7900-1	16-(4-5)-5200-1

CHAPTER 7

7. AUTO-ASSOCIATIVE NETWORK

A Feed-forward neural network involves acquisition of input-output models from examples using backpropagation training algorithm as stated in Chapter 3. The network learns a mapping from given inputs to desired output values by adjusting internal weights to minimize the error. In auto-associative networks, the knowledge to be extracted from a database is the identity function of the database, which is simply: {network inputs} = {network outputs}. Auto-associative networks are one of the classic ANN architectures used commonly in robotics, machine learning, and signal processing. They have been used for a wide variety of pattern processing problems such as cleaning up noisy pictures and recognizing known pictures when partially occluded (Hand, 2001). Some of the known applications that Auto-associative networks are typically used in are:

- Noise reduction
- Replacement of missing sensor values
- Gross error detection and correction
- Signal processing

The purpose of training a highly-parameterized, nonlinear network in these areas is that feed-forward networks trained on the identity function can perform several useful data screening tasks with appropriate internal architectures (Kramer, 1992). In other words, this particular type of network is trained to reproduce its inputs and its output(s). The network is forced to represent the input patterns in fewer dimensions, creating a compressed representation. These compressed representations may reveal interesting generalization about the data. Typical architecture of auto-associative network contain 3 hidden layers, which are, respectively, called mapping layer, bottle neck layer, and de-mapping layer (Kramer,1991). This approach has been used by some researchers (i.e. Bishop et al. (1992), Desjardins et al. (2006), and Sohn et al. (2005)) to reduce the dimensionality of the hidden layer in ANNs for commonly used applications listed above.

The auto-associative network approach has been used in some engineering areas for about two decades but not in civil engineering, where artificial intelligence is mostly referred to as a function approximation method.

In this chapter, the auto-associative network approach was explored by using civil engineering databases, which do not only consist of binary numbers and patterns as in other engineering areas. For this reason, model development of the auto-associative network with the databases mentioned in Chapter 4 was considered with only one hidden layer. More than one hidden layer combined with an insufficient number of databases may cause the network to memorize the data in the training phase. Consequently, to avoid this situation at first place, models were developed with one hidden layer only to maintain the generalization capability of the network. Even though future studies will look into expending this research by including more hidden layers, this study is limited to only one hidden layer networks. Since the strategy of this approach is based on mapping n input variables into n output variables, it still would not be wrong to call these networks as Auto-associative networks.

Due to the fact that the auto-associative network is optimized on not only output, but inputs as well, it can be used to validate partially missing input variables. Query method, which is another scope of this study, is used to replace missing input parameter(s) or/and output. The replaced value by Query method can be easily validated or iterated by using Auto-associative network. For example, if there is a missing data among the input dataset, query method replaces the value and the auto-associative network generates a reflection of the dataset, which helps to validate the value replaced by Query method. Moreover, Auto-associative network approach can be utilized to validate partially missing datasets as well as generate outputs.

Auto-associative network is based on mapping n input variables into n output variables. In order to obtain predictions from this network, initial estimate of the controlled variable (i.e. output) has to be included as an input. The methodology used in Chapter 6, which of using static ANN prediction as an input in the model development was also applied to Auto-associative network approach. The architecture of the proposed Auto-associative network is depicted in Figure 7-1. The four sequential training stages for all seven databases and their desired criteria to choose the optimal network structures of Auto-associative network models are explained in the following sections. Even though the developed models are optimized on both inputs and output, in this study only output variable was evaluated in terms of statistical accuracy measures. Therefore, presented results in the following chapters are limited to output variables.

7.1 Auto-associative Network Development of Database 1

In this database, auto-associative model architecture has been built by considering 8 inputs and 8 outputs. One of the counted inputs is the initial estimate from the developed static ANN network in Chapter 5. The seven inputs, excluding the initial estimate from static ANN network, are used as outputs. The eighth output, which is the uncontrolled parameter, is the actual value of the output variable. A total of 300 datasets are used to build the desired database; 157, 72 and 71 datasets are used, respectively, for training, testing and validation purposes. Based on statistical measures such as MRSE, MARE, and R^2, the optimal network structure of the Auto-associative model for Database 1 was found at 6 hidden nodes and 20,000 iterations. The corresponding accuracy measures for this network are $MRSE_{tr}$= 9.6230, $MARE_{tr}$= 6.262%, $R^2{}_{tr}$= 0.9868 (for training database) and $MRSE_{ts}$=16.5741, $MARE_{ts}$= 6.937%, $R^2{}_{ts}$=0.9802 (for testing database). The training and testing graphical comparison plots between predicted and actual values for the Auto-associative model developed for Database 1 are, respectively, shown in Figure 7-2 and Figure 7-3. Also, all the statistical accuracy measures for the training and testing are shown in Table 7-1. After the training and testing procedures using, respectively, 157 and 72 datasets, validation was conducted on the remaining 71 datasets. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 7-4. Once the validation stage is completed, all of the 300 datasets were used to retrain the network at the previously determined optimal structure to obtain the generalized response throughout the 300 datasets. The graphical comparison plot for the 300 datasets is shown in Figure 7-5. Statistical accuracy measures for validation and all data cases are also shown in Table 7-1. As can be seen from the table, Auto-associative network developed for database 1 has lower validation MRSE value than testing MRSE value. All data MRSE value is lower than training, testing and validation measures while the value of MARE deteriorated slightly. R^2 values did not change significantly for testing, validation, or all data. However, MRSE is considered as the main criterion to evaluate the performances of the networks. MRSE value for all data was decreased to a value of 7.4928 while the ones for testing and validation were increased to values of 16.5741 and 12.9810. In Table 7-1, as indicated in previous chapters, the *8-(3-6)-20000-8* notation specifies the determined architecture of the optimum network of Database 1 where each number ,respectively, represents: number of inputs (8), initial number of hidden nodes (3), final number of hidden nodes (6), number of iterations (20000), and number of outputs (8). Final structure of the optimum network is depicted as *8-6-8,* which are, respectively: number of inputs, number of hidden nodes, and number of outputs. Seven input parameters with an initial estimate from static ANN network are feed-forwarded to hidden layer with eight hidden nodes, then come out as eight outputs, one of which is the actual output and the rest are all input predictions.

7.2 Auto-associative Network Development of Database 2

The available 100 datasets were used to develop the desired Auto-associative network for Database 2. During the first stage of modeling, 55 datasets are used for training, 23 for testing, and 22 for validation The input vector consisted of 8 parameters and the output vector made up of the same 8 parameters were used in the model development process of database 2. The optimal structure for the Auto-associative network of Database 2 was chosen at 6 hidden nodes and 3100 iterations. A graphical comparison of training stage between the predicted and the actual is depicted in Figure 7-6. Auto-associative network for training stage yielded a mean root square error, $MRSE_{tr}$ of 0.4911, mean absolute relative error, $MARE_{tr}$ of 4.8125%, and coefficient of determination, R^2_{tr} of 0.4795. Similarly, graphical comparison of testing stage is shown in Figure 7-7 and statistical accuracy measures for this network are $MRSE_{ts}$ of 0.8469, $MARE_{ts}$ of 5.1122%, and R_{ts}^2 of 0.1798.

To further evaluate the optimal network, 22 datasets are used to validate the network. Figure 7-8 presents the graphical comparison of the predicted and the actual values. Corresponding statistical measures are given in Table 7-1. It is clear from the results that validation MRSE is higher than the testing MRSE as oppose to static ANN network. Once the validation stage is completed, all 100 datasets were used to retrain the network at the optimal structure. It can be concluded from the graphical plot in Figure 7-9 and the statistical accuracy measures in Table 7-1 that using the entire database to retrain the network enhances the statistical measures. Overall performance of the Auto-associative network is very similar to static ANN network in terms of statistical accuracy measures. For example, all data MRSE by Auto-associative network is 0.3425 while all data MRSE by static ANN network and by Feedback ANN network, respectively, resulted 0.3203 and 0.2979. The MRSE by auto-associative network is about 6.5% higher than that of the static ANN network and about 13% higher than that of the Feedback ANN network. The predictions via the Auto-associative network are not as accurate as those obtained by previous networks but still can be considered adequate especially when considering that the Auto-associative network is optimized on both inputs and output(s).

7.3 Auto-associative Network Development of Database 3

To develop Auto-associative model for database 3, the 126 datasets were used. Sixty three and 32 of total datasets were, respectively, considered as training and testing datasets. The remaining 31 datasets were included in the validation stage after the optimal network was determined. An effort to develop an Auto-associative network for database 3 was initiated with 13 inputs and 13 outputs. The best performing network structure was obtained at 8 hidden nodes and 20000 iterations. The training and testing statistical measures for training and testing stages are shown in Table 7-1 and the graphical comparison plots are depicted in Figure 7-10 and Figure 7-11. As can be observed from the table and the graphical plots, the training and testing stages produced good prediction accuracy. Validation was conducted on the remaining 31 datasets, after the training and testing stages. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 7-12. The statistical accuracy measures for this network are $MRSE_{val}$= 210.3098, $MARE_{val}$= 22.139%, and R^2_{val}= 0.7209. Once the validation stage is finalized, all of the 126 datasets were used to retrain the network at the optimal structure. The statistical accuracy measures for this network are

$MRSE_{all}$= 55.7573, $MARE_{all}$= 14.770%, and R^2_{all}= 0.9342. The graphical comparison plot for the 126 datasets is shown in Figure 7-13. The resulting statistical accuracy measures for all Auto-associative network modeling stages are given in Table 7-1.

The statistical measures and the plots have indicated that the Auto-associative network for database 3 has performed well during the training stage, but the testing stage resulted in a higher MRSE value, as was expected. Similarly, MRSE value for the validation datasets produced the highest MRSE value, which is about 3.6 times higher than the training MRSE value. When all data combined and the network was retrained, the statistical accuracy measures improved. It should be noted that the error increase from training MRSE to validation MRSE by Auto-associative network is less than the one noted for static ANN (i.e., 3.6 versus 7.16 times). Similarly, the error increase for MARE by Auto-associative is less than that by static ANN (i.e. 1.76 versus 2.4 times).

7.4 Auto-associative Network Development of Database 4

In order to develop Auto-associative network for Database 4, a total of 265 datasets; 133, 66, and 66 sub-datasets were, respectively, used for training, testing, and validation. The input vector consisted of 7 parameters, including the one from static ANN network, and the output vector consisted of 7 parameters as well (6 inputs and one output). To properly characterize the phenomenon, Auto-associative network approach with four sequential modeling stages were followed herein. Accordingly, the optimal network structure for the Auto-associative model of database 4 was determined at 7 hidden nodes and 20000 iterations where the network performed its best. Auto-associative network for training stage yielded a mean root square error, $MRSE_{tr}$ of 68.8665, mean absolute relative error, $MARE_{tr}$ of 20.557%, and coefficient of determination, R_{tr}^2 of 0.8539. Similarly, statistical accuracy measures for the testing stage are $MRSE_{ts}$ of 102.4501, $MARE_{ts}$ of 21.863%, and R^2_{ts} of 0.8363. Graphical comparisons of testing and validation stages are, respectively, shown in Figure 7-14 and Figure 7-15. As can be seen from the graphical plots and the statistical accuracy measures depicted in Table 7-1, a good agreement between actual and predicted values is apparent. Stage 3 and stage 4 were sequentially initiated by using validation datasets and all datasets. The predictions by validation datasets and all datasets were plotted against their corresponding actual values, respectively, in Figure 7-16 and Figure 7-17. Good agreement between the predictions and the actual values can also be observed in Table 7-1 in terms of statistical accuracy measures.

7.5 Auto-associative Network Development of Database 5

Database 5 has been built by considering 325 datasets; 163, 81, and 81 datasets that are for training, testing, and validation purposes. The optimal network structure for the model 1 was concluded at 4 hidden nodes and 20,000 iterations. The corresponding accuracy measures of model 1 are $MRSE_{tr}$=0.2676, R^2_{tr}=0.9936, $MARE_{tr}$=0.240% (for training database) and $MRSE_{ts}$ =0.5100, R^2_{ts}=0.9817, $MARE_{ts}$= 0.286% (for testing database). The optimal network for Model 2 was reached at 5 hidden nodes and 20,000 iterations. The corresponding accuracy measures of model 2 are $MRSE_{tr}$ =0.6344, R^2_{tr}=0.9299, $MARE_{tr}$=1.199% (for training database) and $MRSE_{ts}$ =0.8233, R^2_{ts}=0.9354, $MARE_{ts}$= 1.037% (for testing database). Training MRSE value for model 1 increased by about 90% in testing while training MRSE value for model 2 increased by about 30% in testing. The training and testing plots for model 1 are shown in Figure 7-18 and Figure 7-19. Similarly the training and testing plots for model 2 are also given in Figure 7-20 and Figure 7-21. The corresponding statistical measures of model 1 and model 2 are presented in Table 7-1.

After the optimal network was determined, the validations for model 1 and model 2 were conducted on the 81 datasets. The validation plots for model 1 and model 2 are, respectively, given in Figure 7-22 and Figure 7-23. After the validation stage is concluded, all of the 325 datasets were used to retrain the network at the optimal structure. The comparison plots of model 1 and model 2 for the 325 datasets are, respectively, shown in Figure 7-24 and Figure 7-25. The resulting statistical accuracy measures for the validation and the all data cases are depicted in Table 7-1. All data MRSE statistical measures for both model 1 and model 2 attain the best values when compared with the other stages.

7.6 Auto-associative Network Development of Database 6

The 210 datasets were divided into sub-databases: 105, 53, and 52 to be used, respectively, in training, testing, and validation tasks. Auto-associative network for Model 1 was determined at 7 hidden nodes and 20000 iterations. This network structure provided the optimal connection weights for the desired predictions. The training and testing accuracy measures for model 1 are resented in Table 7-2 along with the corresponding plots shown in Figure 7-26 and Figure 7-27. According to the statistical measures, the optimal network performed well in the training stage as well as in the testing stage. However, MRSE value of the training, 0.0033 deteriorated to 0.0061 for the testing stage, which corresponds to 85% increase in error. For the validation stage, the statistical measures changed slightly; however, for the all data stage, MRSE improves to a value of 0.0027, which translates into about 18% reduction in error, while MARE value increased by about 10%. All the statistical measures for the validation and all data stages can be found in Table 7-2 and their associated plots are, in the given order, represented in Figure 7-28 and Figure 7-29.

The same database used to develop Auto-associative network for model 1 was utilized for Model 2 by considering 17 inputs. The optimal network for model 2 was reached at 7 hidden nodes and 20,000 iterations. The accuracy of training and testing stages for the selected network architecture is given in Table 7-2 and the graphical evaluation plots are depicted in Figure 7-30 and Figure 7-31. Validation and all data stages were sequentially followed by the training and testing stages. Figure 7-32 and Figure 7-33, which are the plots for validation and all data predictions, indicate a reasonably good correlation between the actual and predicted values. A good agreement between the actual and predicted values can be noted in Table 7-2.

Auto-associative network model 3 was developed by considering the same input parameters, used for model 1 and model 2, and output 3. Similarly, the model development process was followed in the order of training, testing, validation, and all data cases. The statistical measures were obtained with a structure of 2 hidden nodes and 18,100 iterations. Table 7-2 presents all statistical measures for model 3. Also, corresponding graphical comparisons for the stages are represented in Figure 7-34, Figure 7-35, Figure 7-36, and Figure 7-37. Even though some scatter is noted in these plots, most of the data is predicted reasonably well.

7.7 Auto-associative Network Development of Database 7

Auto-associative network for Database 7 was developed using 792 datasets divided into 396, 198, and 198 datasets for training, testing, and validation. The network utilized 16 inputs and 16. Statistical measures from the training and testing stages were utilized to select the optimal network structure, which was obtained at 8 hidden nodes and 20,000 iterations. The accuracy plots are illustrated in Figure 7-38 and Figure 7-39. The plots indicate good correlation between actual and predicted results, even though there seem to be few outliers at the higher end. As can be observed from Table 7-2, the developed Auto-associative network has reasonably good statistics such as $MRSE_{tr}$= 1.8678, $MARE_{tr}$= 30.249%, and R^2_{tr}= 0.9660. The accuracy of the validation datasets can be seen in Figure 7-40 and the corresponding statistics are shown in Table 7-2. Combining all datasets and retraining the network improved the model accuracy statistics where the MRSE value of 1.8678 for training was reduced to a value of 1.4805, which can be translated into a 20% reduction. All data MARE and R^2 values changed slightly but the major improvement was obtained for the MRSE value. All data predictions are graphically depicted in Figure 7-41 and the statistical accuracy measures are given in Table 7-2.

7.8 Concluding Remarks

In this chapter, Auto-associative network approach with backpropagation learning algorithm was explored by using civil engineering databases. This method is based on mapping n input variables into n output variables. Effect of input parameters on the output based on the statistical evaluation criteria was utilized to determine the optimal architecture of the neural network models, while mapping input parameters on the output layer as well. The idea of using this method is to train a network to obtain the identity mapping, in other words, to develop an identity function. As stated before, this approach has been utilized for other engineering applications but it was introduced to civil engineering databases in this study. The Auto-associative network method utilizes the output from static ANN model along with the input parameters to generate new improved results as well as to provide reflection for predicted and missing values of input parameters.

As seen from the graphical results depicted in Figure 7-2 to Figure 7-41 and the accuracy measures of the developed Auto-associative network models for each database listed in Table 7-1 and Table 7-2, the Auto-associative network models have reliable results. Moreover, the statistical accuracy measures, such as MARE, R^2, and MRSE, from static ANN network and Auto-associative network modeling have been evaluated to determine the increase/reductions in the statistical accuracy measures of the proposed Auto-associative network models. Due to the fact that Auto-associative network is optimized on inputs and output(s), the statistical accuracy measures of the outputs were not expected to be as reliable as static ANN networks. However, the results indicated that for few cases Auto-associative network can perform better.

As can be seen in Table 7-3, only database 4 and database 6- Output 1 had a MARE reduction. The rest of the databases or outputs had an increase in error values. Similarly, R^2 value has increased for the same two databases as shown in Table 7-4. The corresponding statistical measure, MRSE has similar results in Table 7-5. However, there is more improvement for MRSE values than the other statistical measures. Database 3 and Database 6 – output1 had a few significant changes, one of which is 13% and the other one is 29% reduction in error. Database 3 had also 4% reduction in error. The Auto-associative network did not perform well on most of the databases in terms of error reduction but discovered the relationship between inputs and output. Even though the results from Auto-associative network are not comparable with those obtained via other previous approaches, they are still considerably promising. It is noteworthy to mention that Auto-associative network can not only be utilized to generate outputs, but can also be used for verification of the missing values in input parameters.

7.9 Figures and Tables

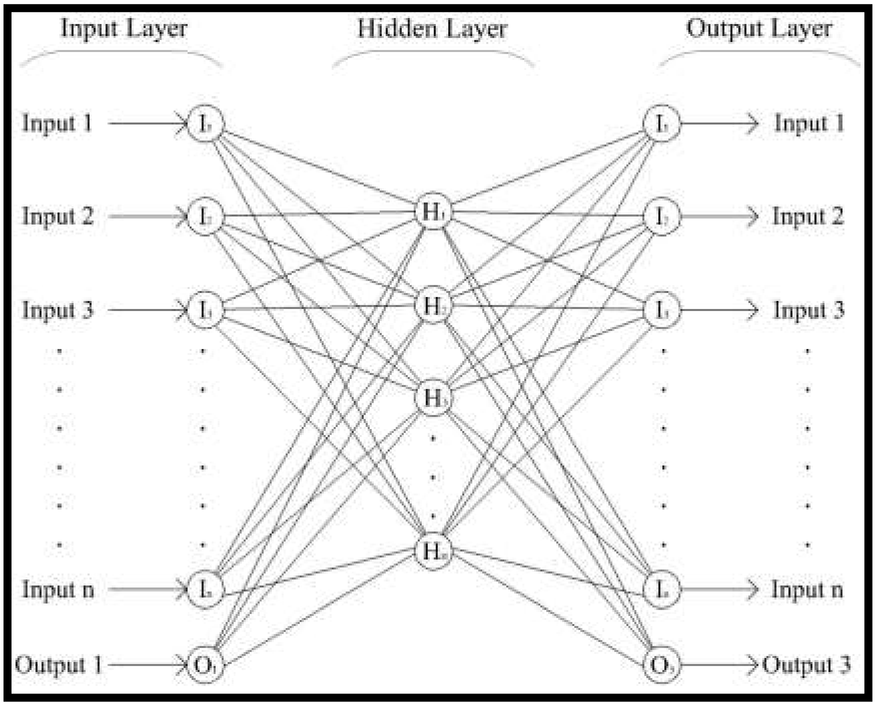

Figure 7-1 Architecture of an Auto-associative Network

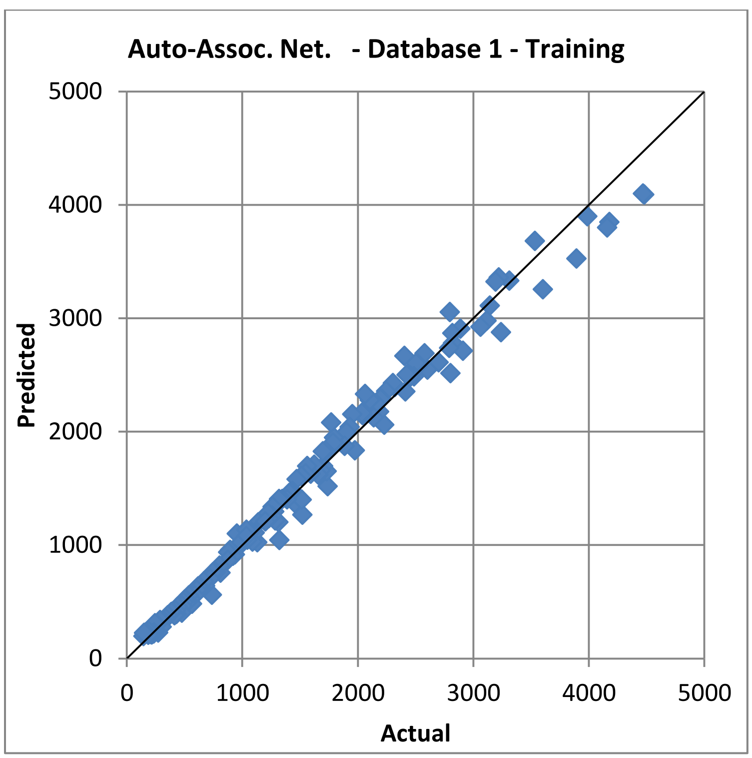

Figure 7-2 Auto-associative Network Training Accuracy of Database 1

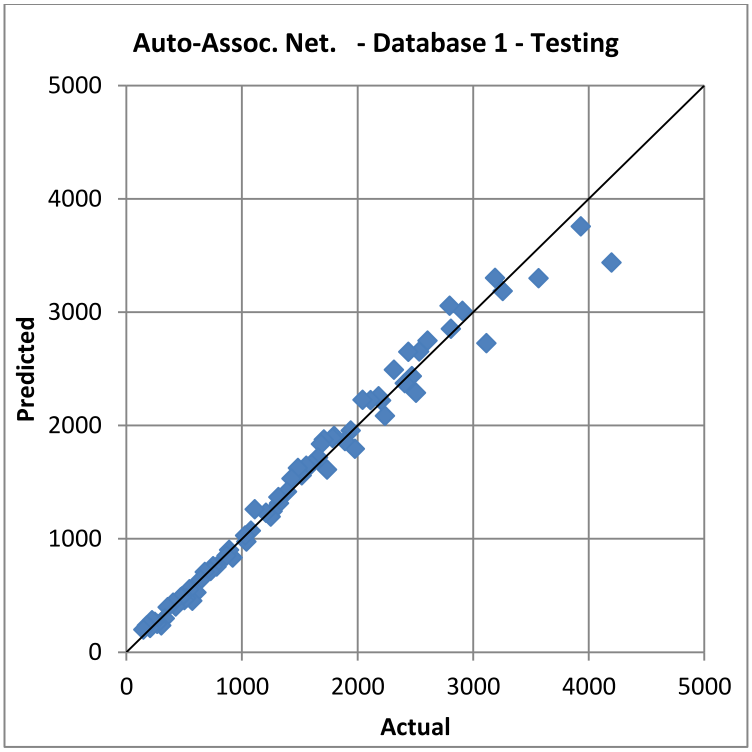

Figure 7-3 Auto-associative Network Testing Accuracy of Database 1

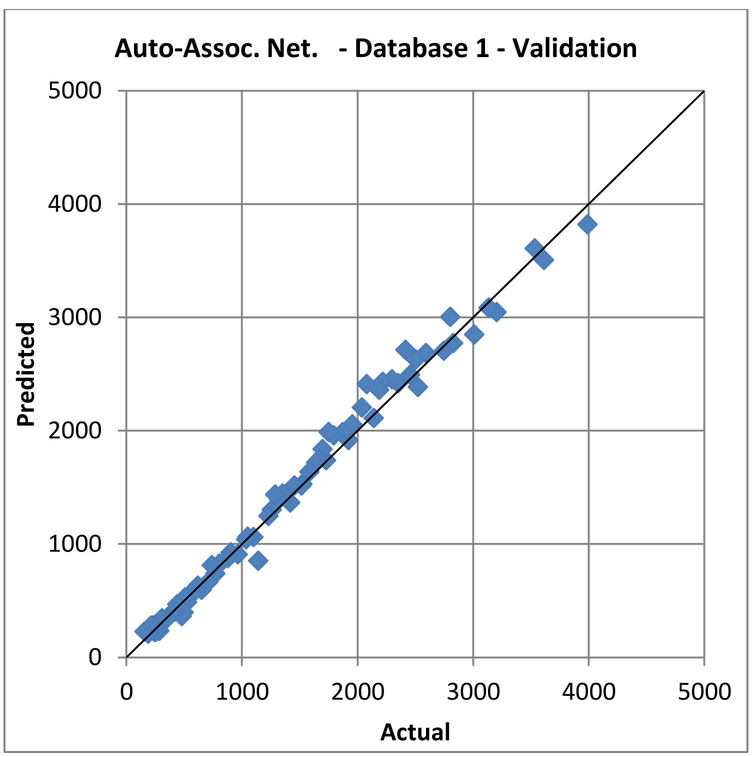

Figure 7-4 Auto-associative Network Validation Accuracy of Database 1

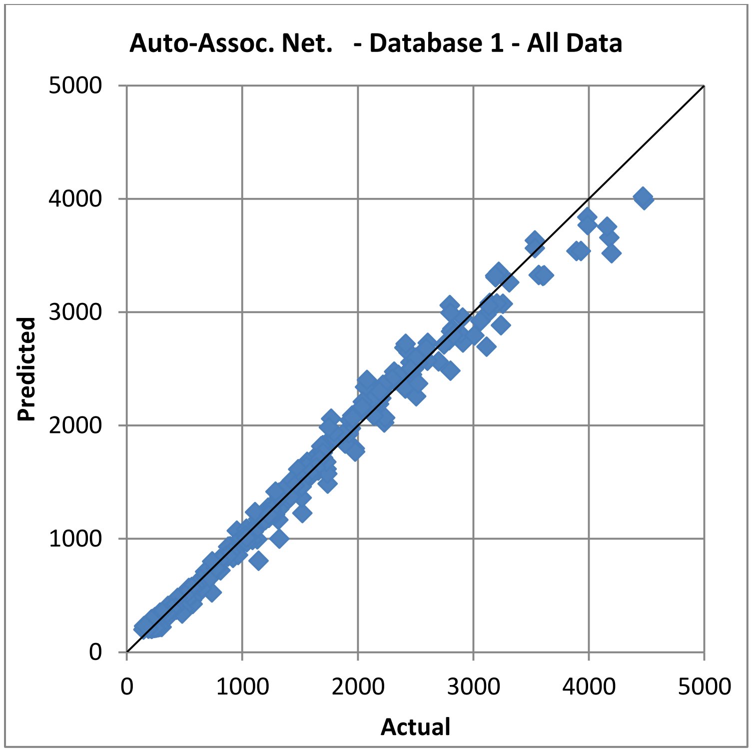

Figure 7-5 Auto-associative Network All Data Accuracy of Database 1

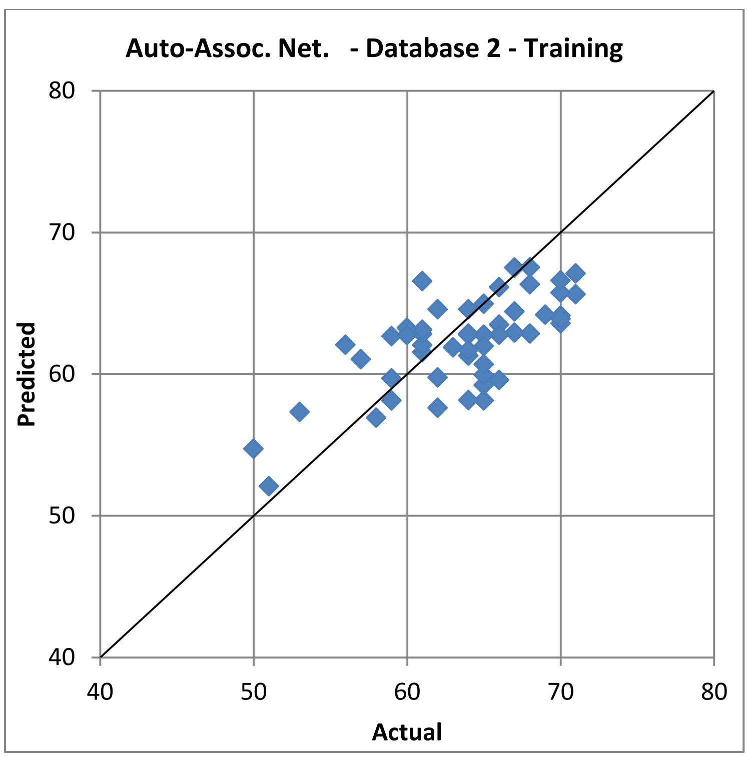

Figure 7-6 Auto-associative Network Training Accuracy of Database 2

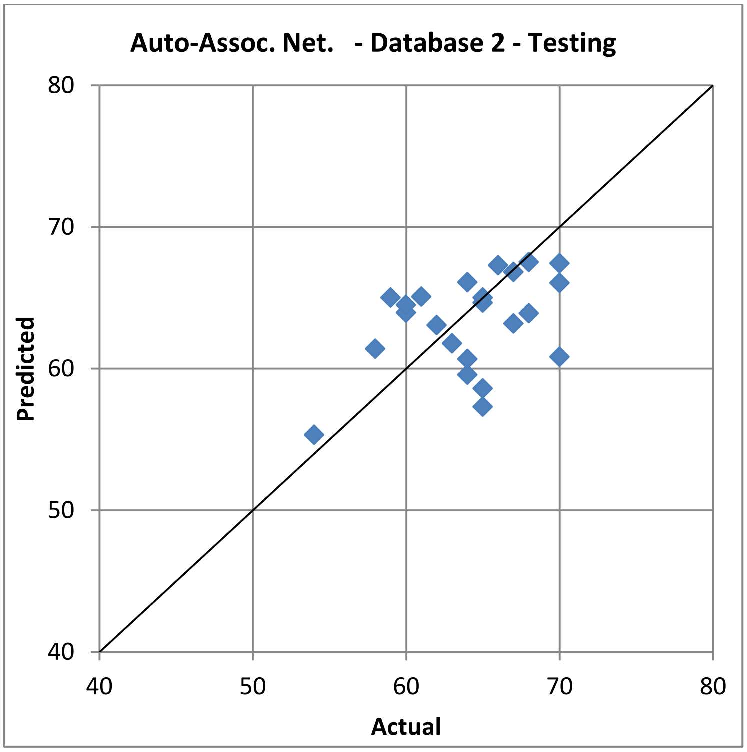

Figure 7-7 Auto-associative Network Testing Accuracy of Database 2

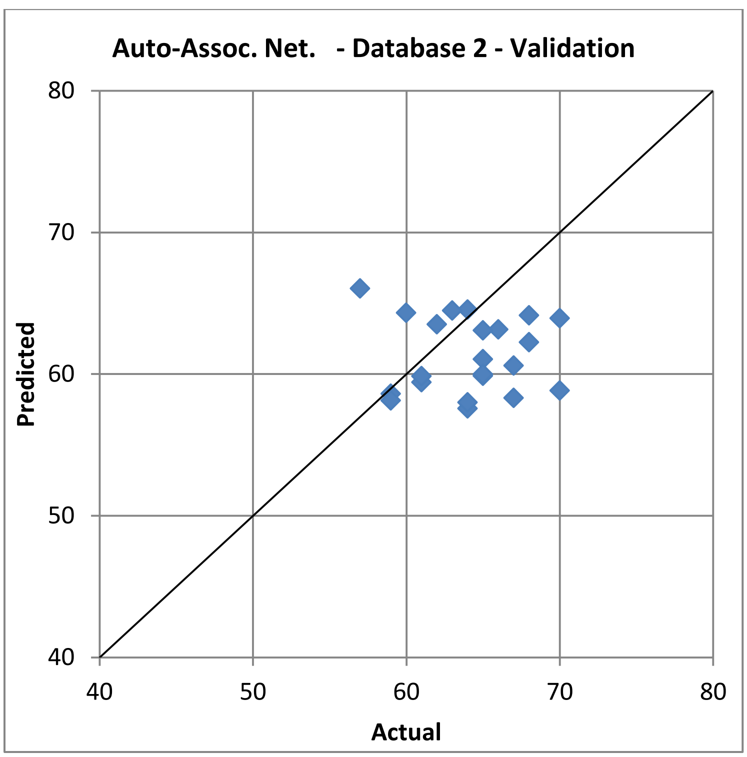

Figure 7-8 Auto-associative Network Validation Accuracy of Database 2

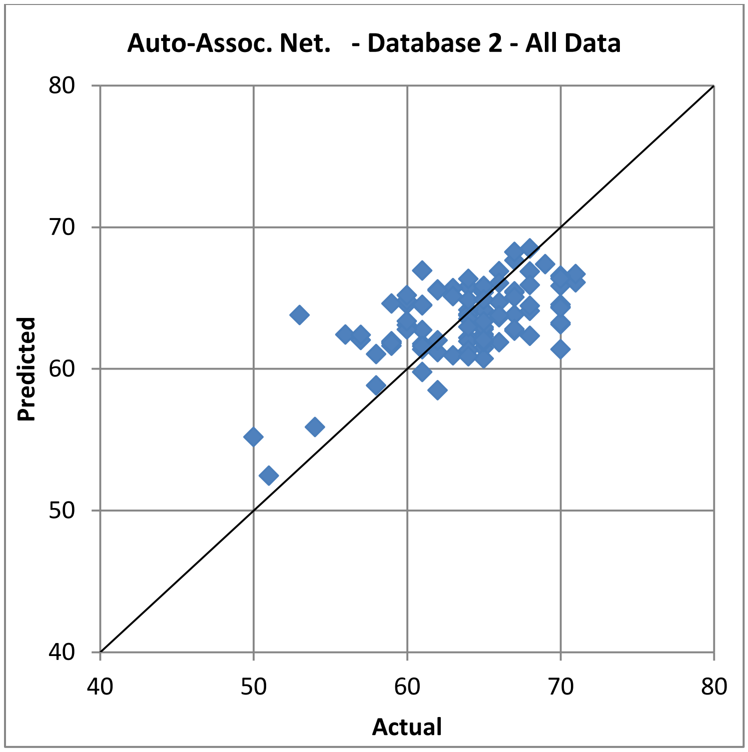

Figure 7-9 Auto-associative Network All Data Accuracy of Database 2

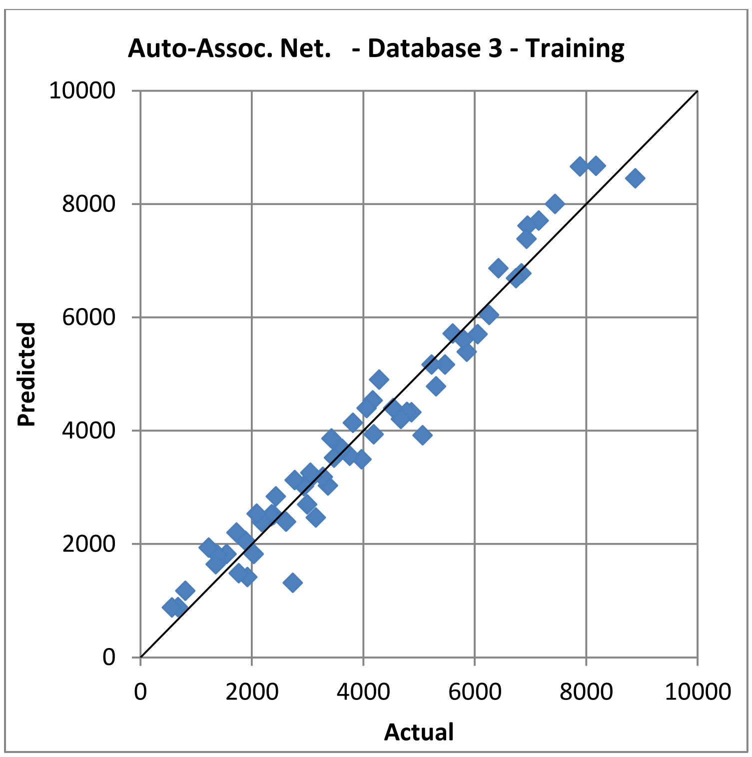

Figure 7-10 Auto-associative Network Training Accuracy of Database 3

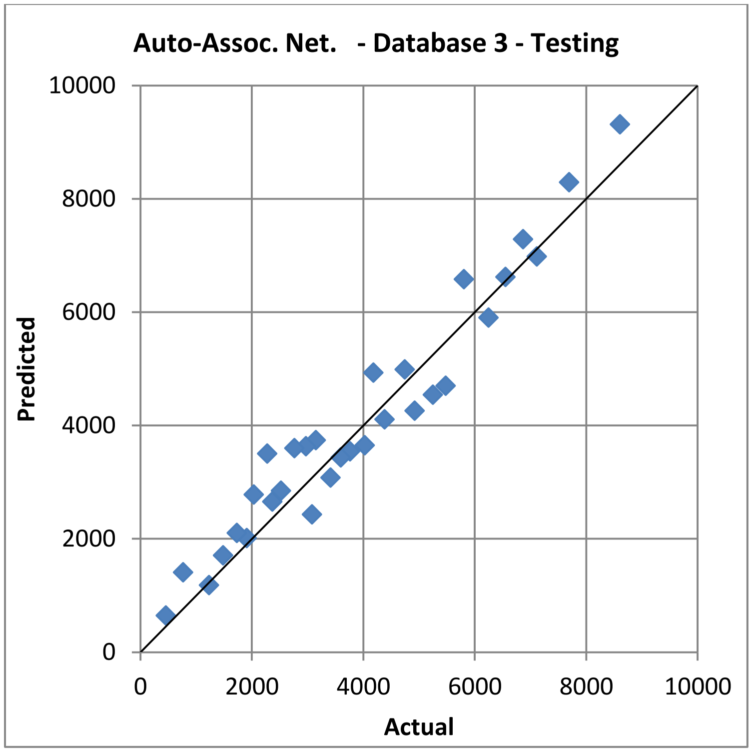

Figure 7-11 Auto-associative Network Testing Accuracy of Database 3

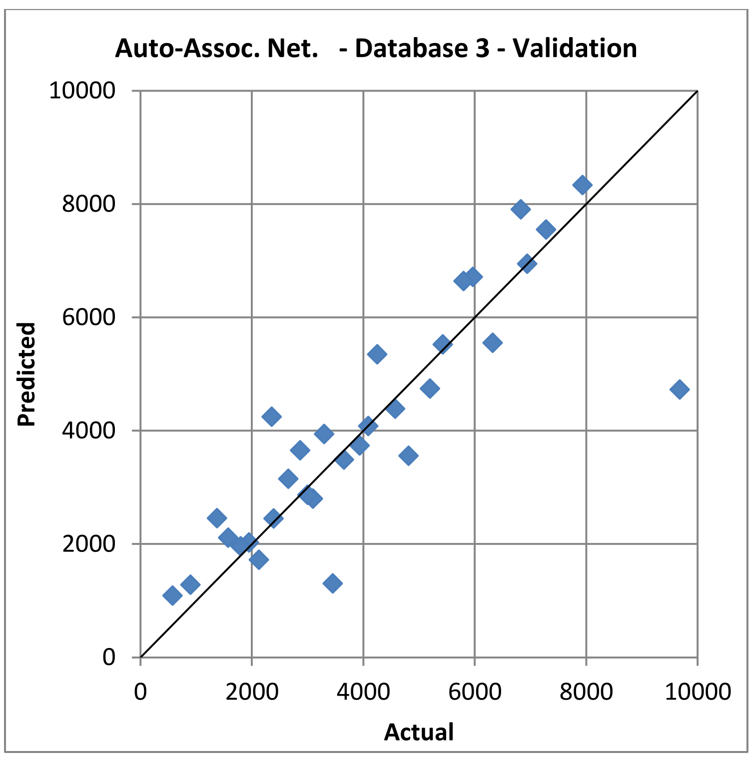

Figure 7-12 Auto-associative Network Validation Accuracy of Database 3

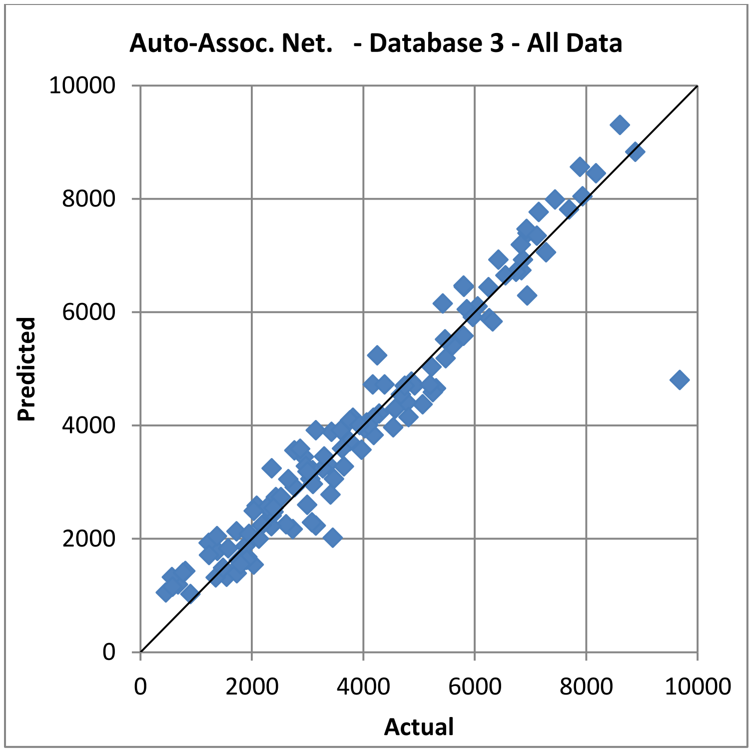

Figure 7-13 Auto-associative Network All Data Accuracy of Database 3

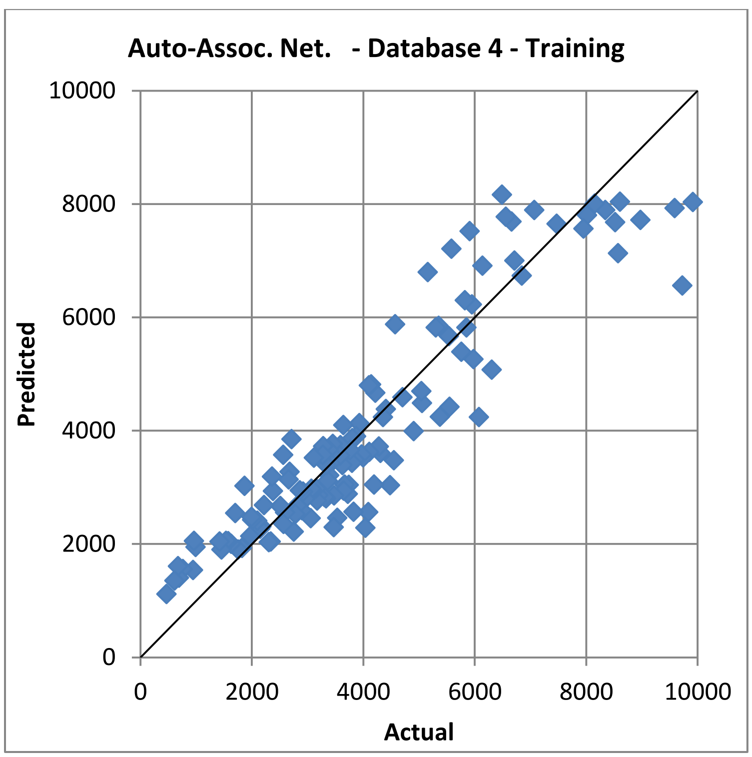

Figure 7-14 Auto-associative Network Training Accuracy of Database 4

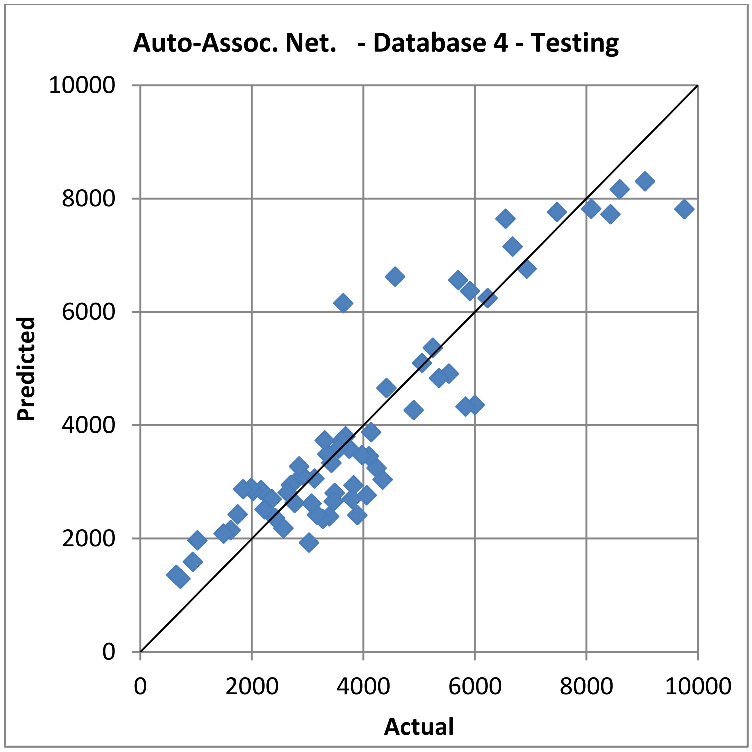

Figure 7-15 Auto-associative Network Testing Accuracy of Database 4

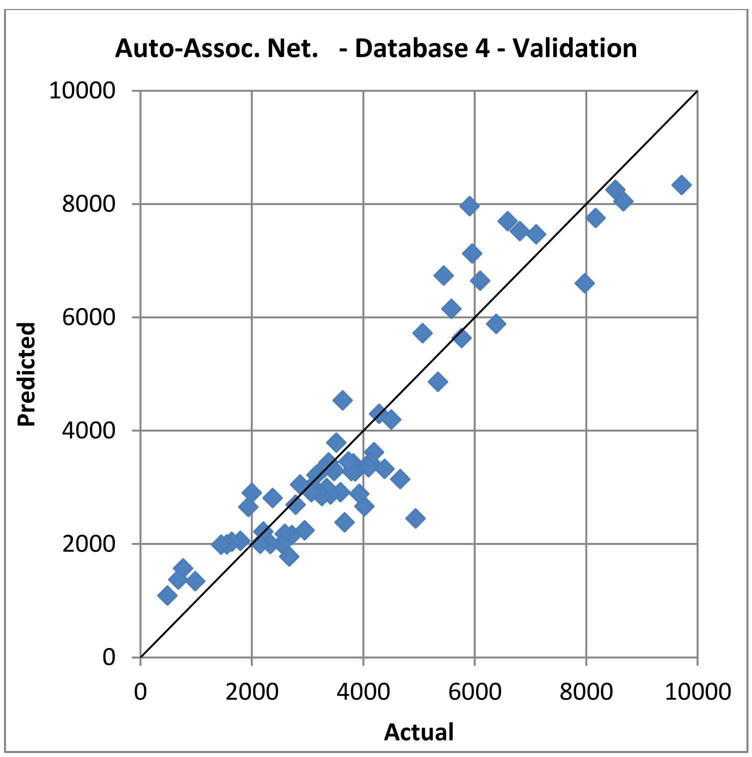

Figure 7-16 Auto-associative Network Validation Accuracy of Database 4

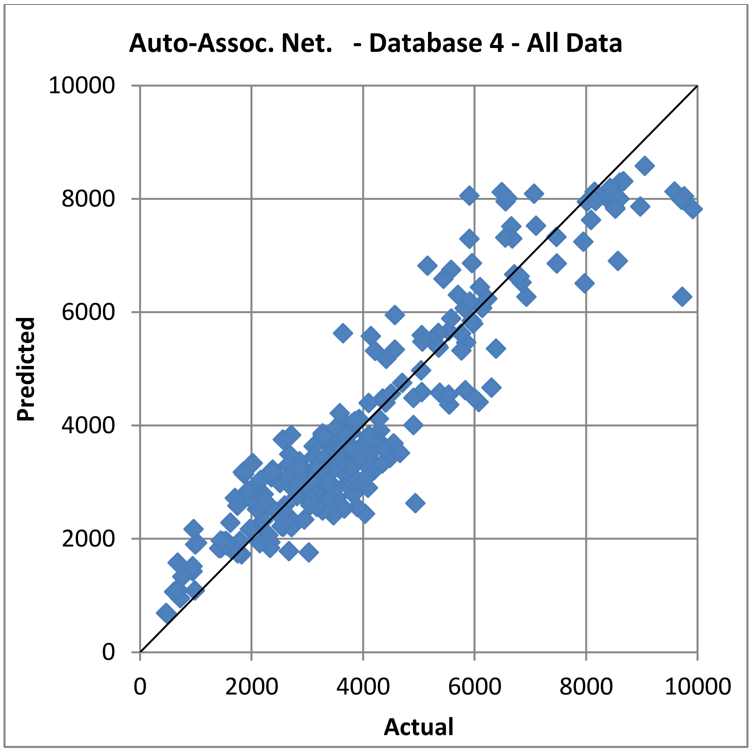

Figure 7-17 Auto-associative Network All Data Accuracy of Database 4

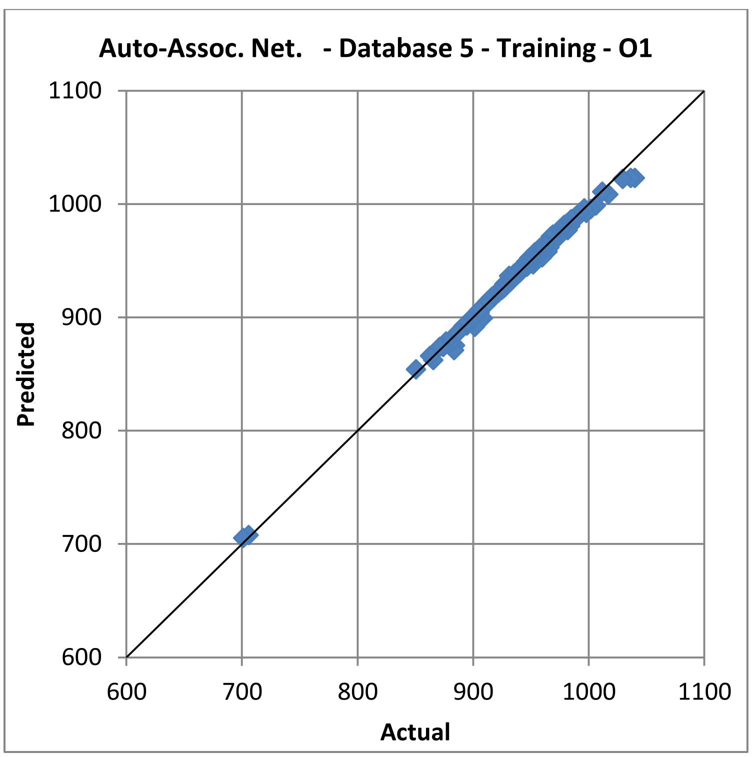

Figure 7-18 Auto-associative Network Training Accuracy of Database 5, Output 1

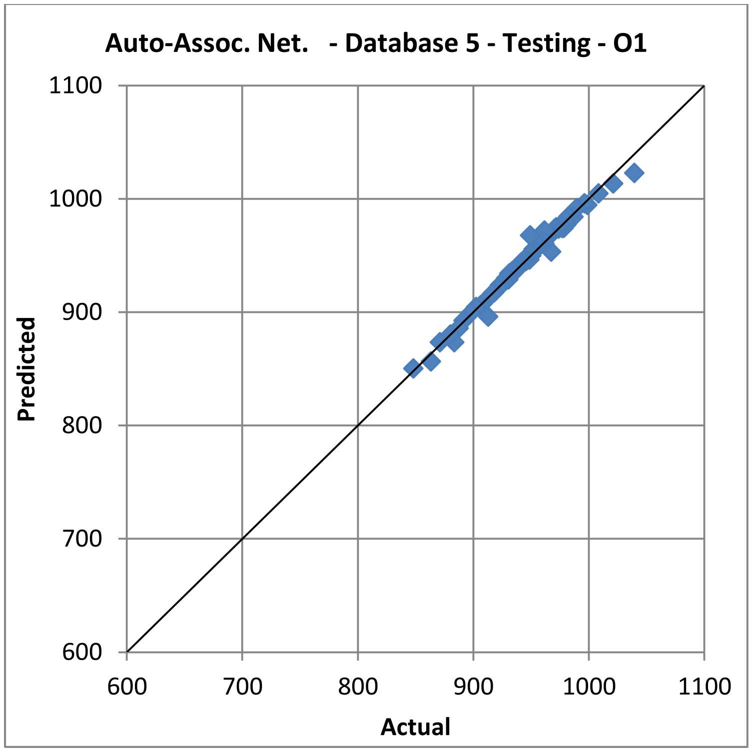

Figure 7-19 Auto-associative Network Testing Accuracy of Database 5, Output 1

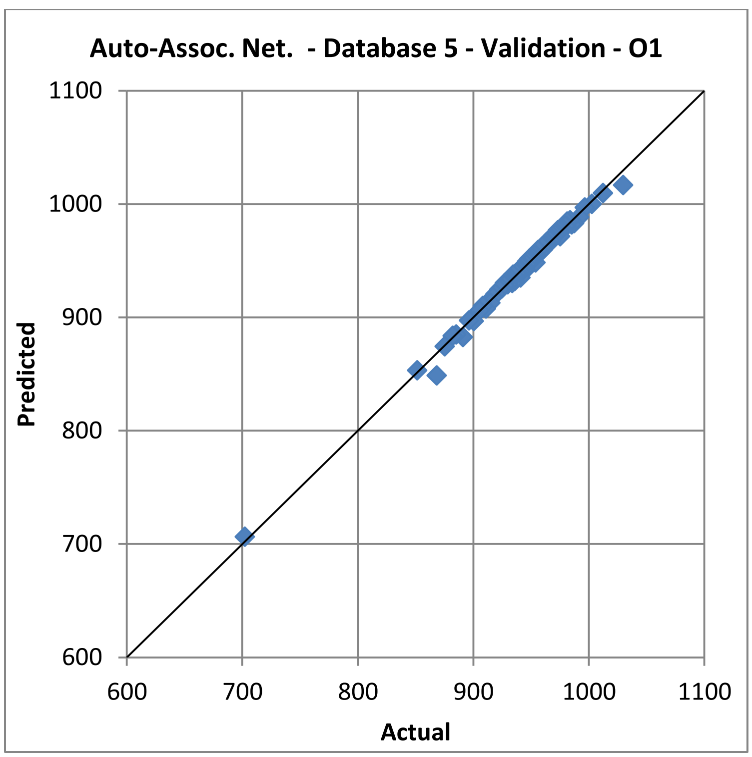

Figure 7-20 Auto-associative Network Validation Accuracy of Database 5, Output 1

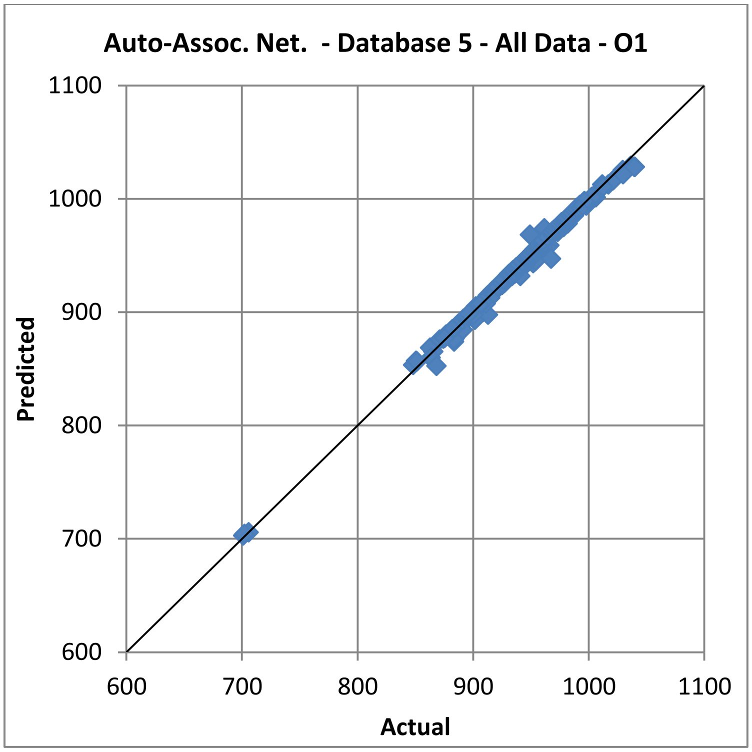

Figure 7-21 Auto-associative Network All Data Accuracy of Database 5, Output 1

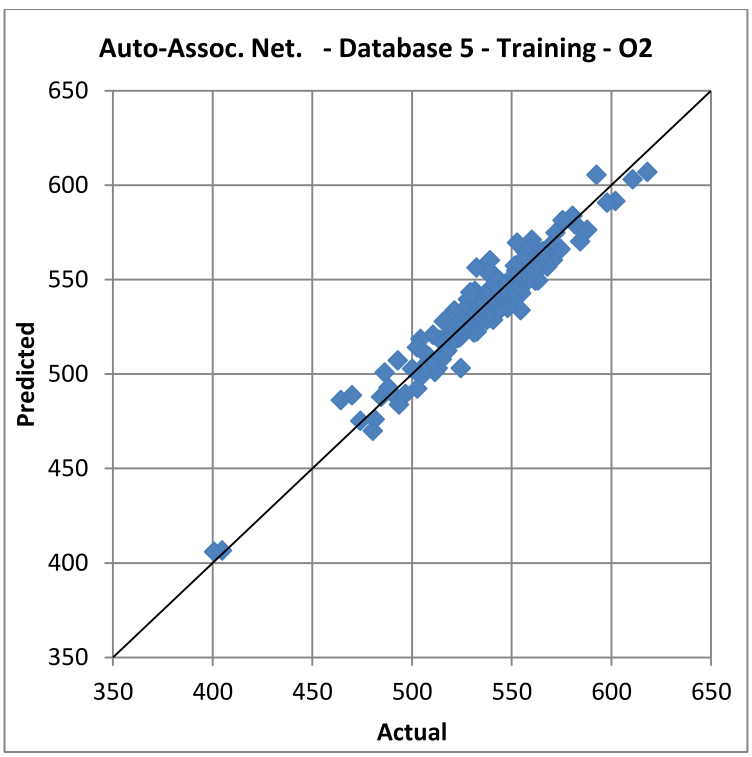

Figure 7-22 Auto-associative Network Training Accuracy of Database 5, Output 2

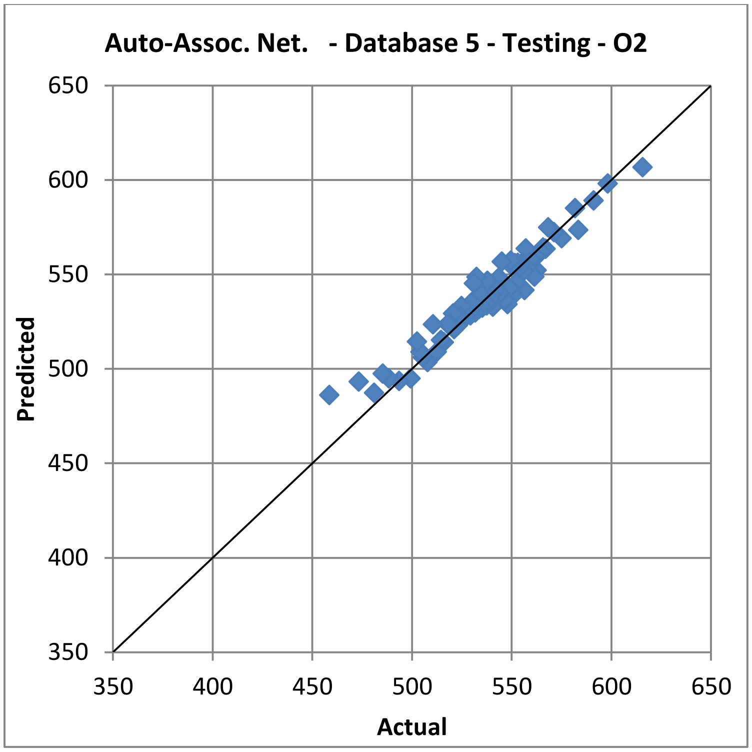

Figure 7-23 Auto-associative Network Testing Accuracy of Database 5, Output 2

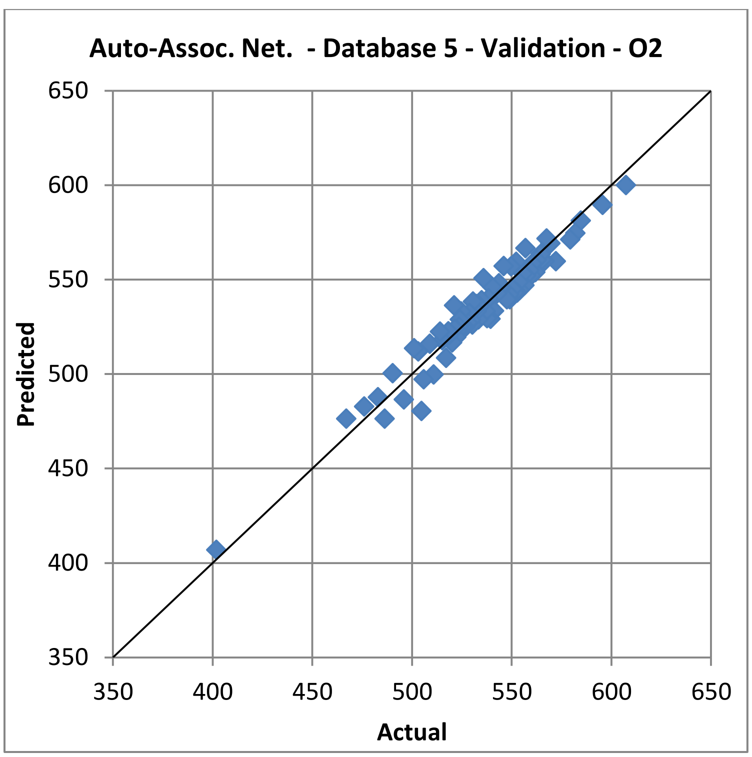

Figure 7-24 Auto-associative Network Validation Accuracy of Database 5, Output 2

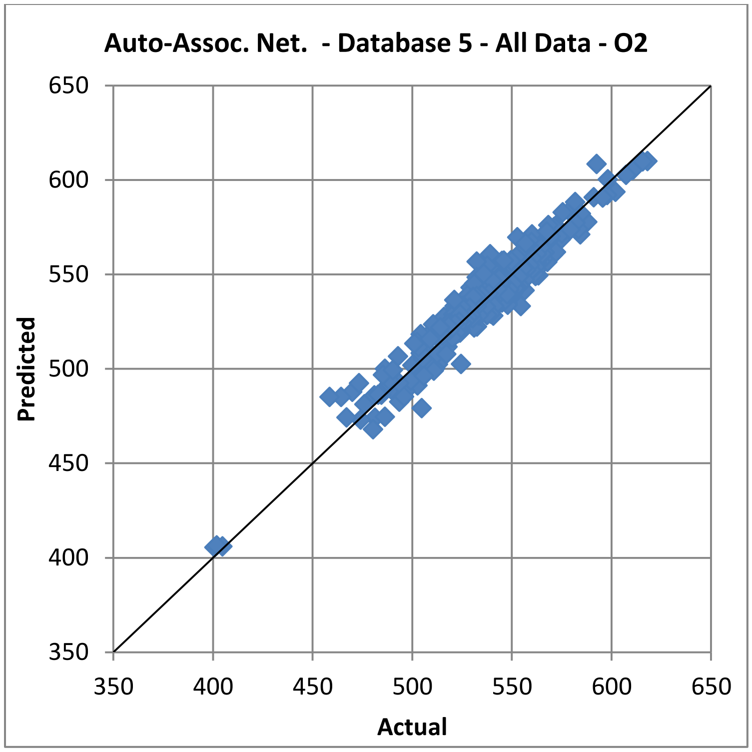

Figure 7-25 Auto-associative Network All Data Accuracy of Database 5, Output 2

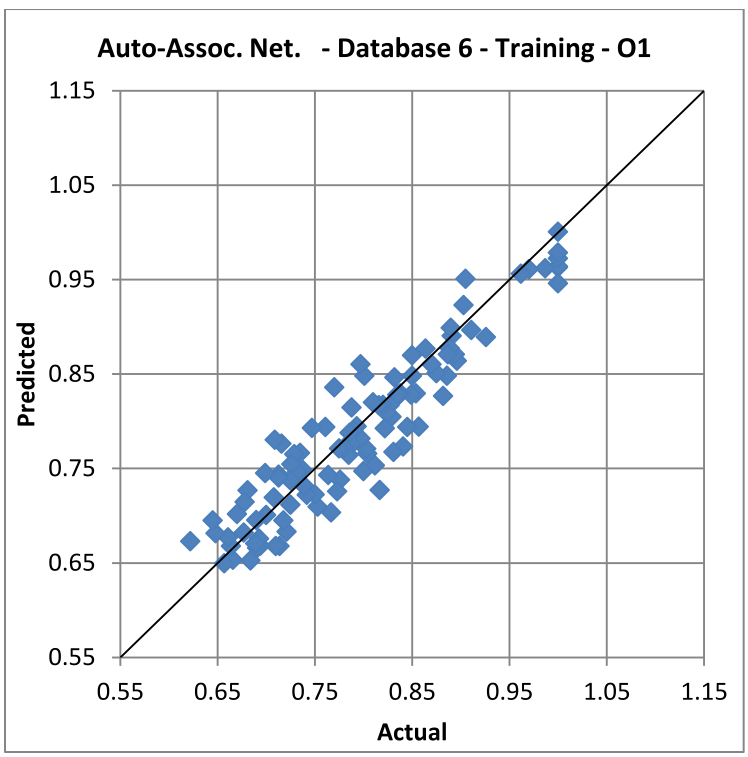

Figure 7-26 Auto-associative Network Training Accuracy of Database 6, Output 1

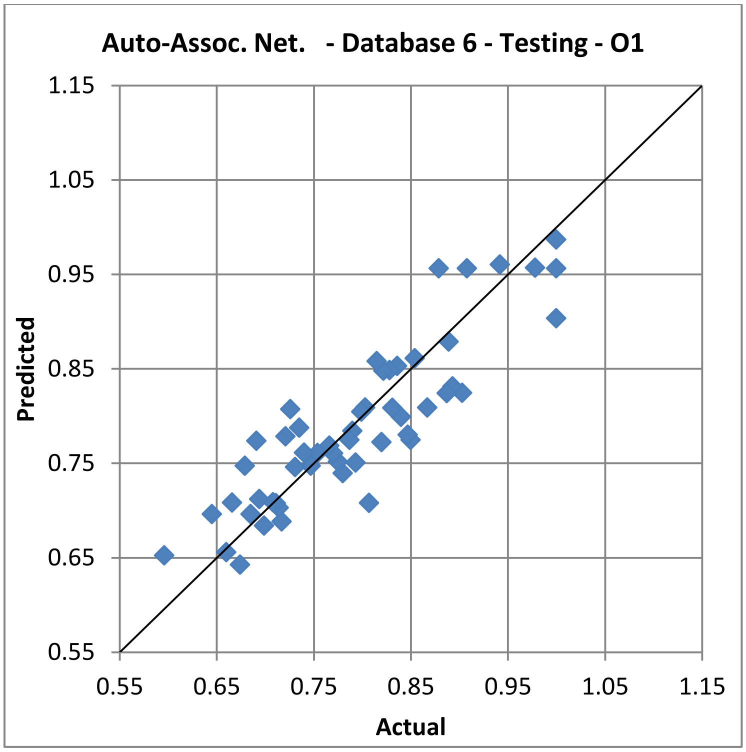

Figure 7-27 Auto-associative Network Testing Accuracy of Database 6, Output 1

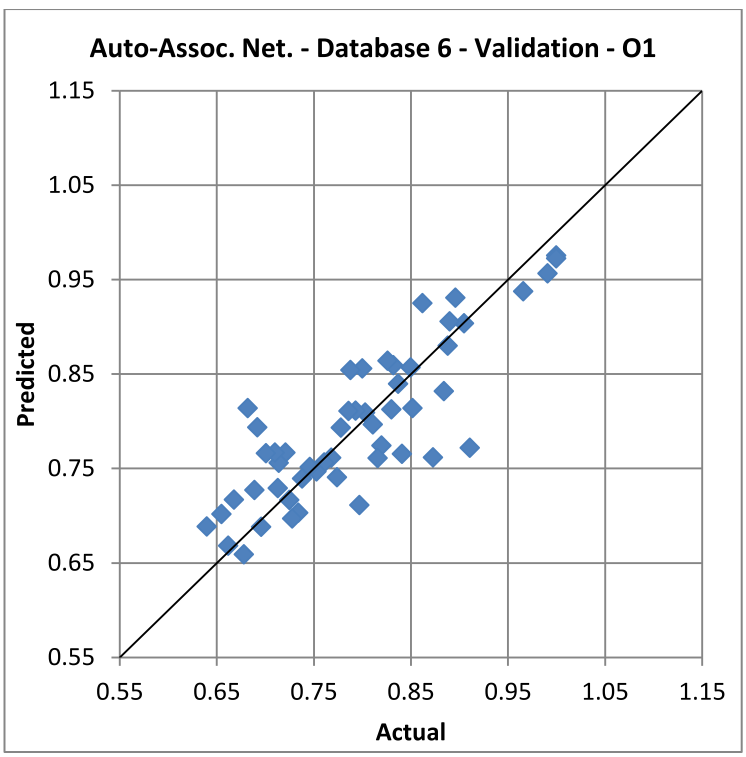

Figure 7-28 Auto-associative Network Validation Accuracy of Database 6, Output 1

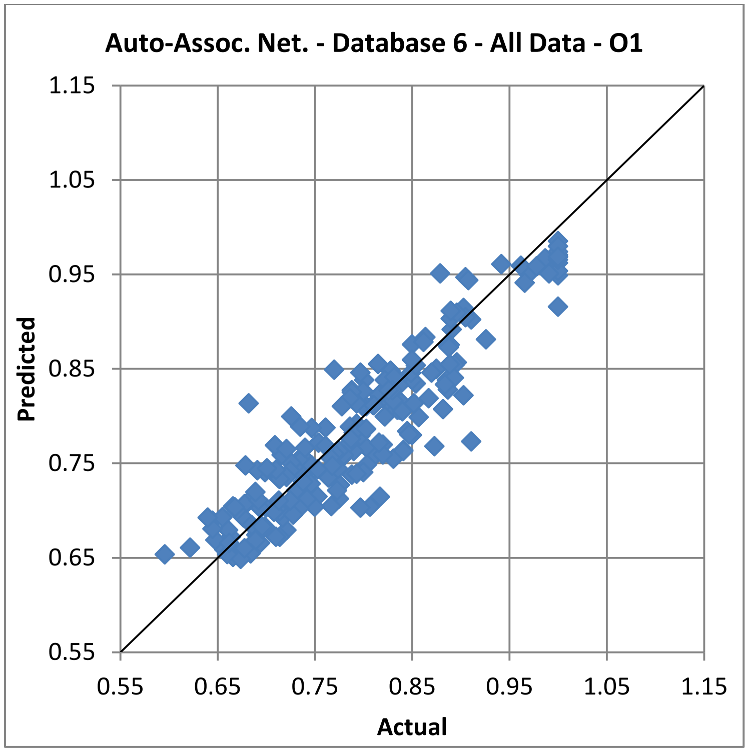

Figure 7-29 Auto-associative Network All Data Accuracy of Database 6, Output 1

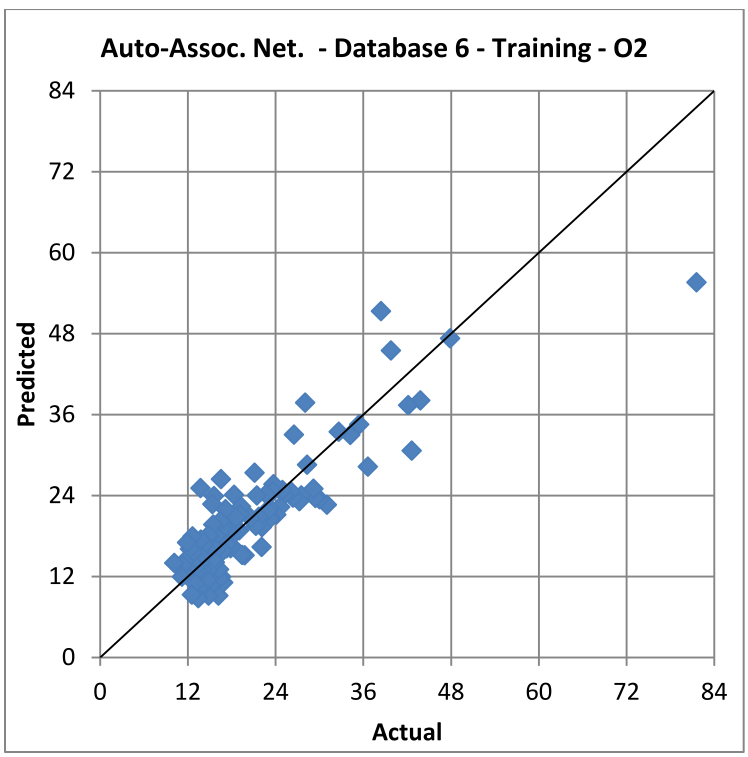

Figure 7-30 Auto-associative Network Training Accuracy of Database 6, Output 2

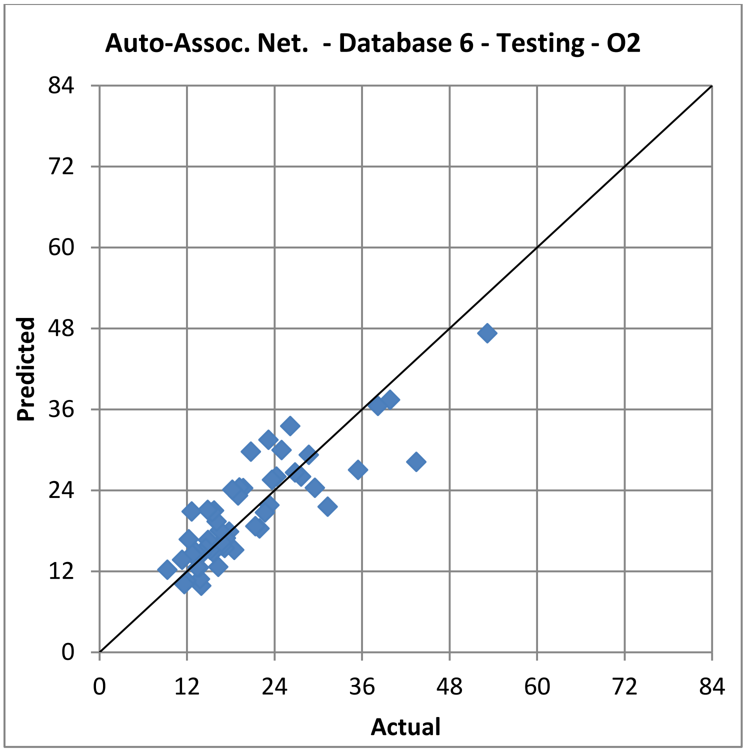

Figure 7-31 Auto-associative Network Testing Accuracy of Database 6, Output 2

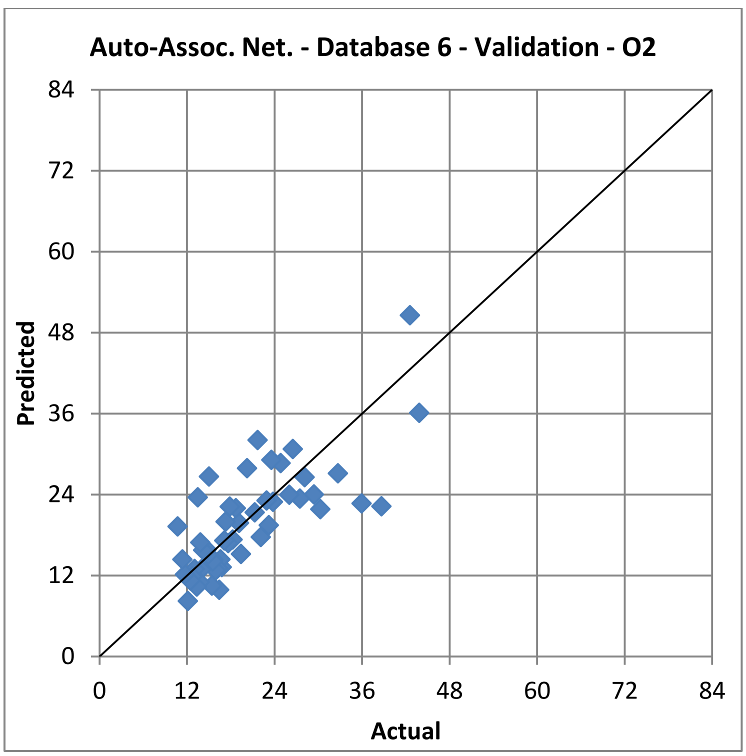

Figure 7-32 Auto-associative Network Validation Accuracy of Database 6, Output 2

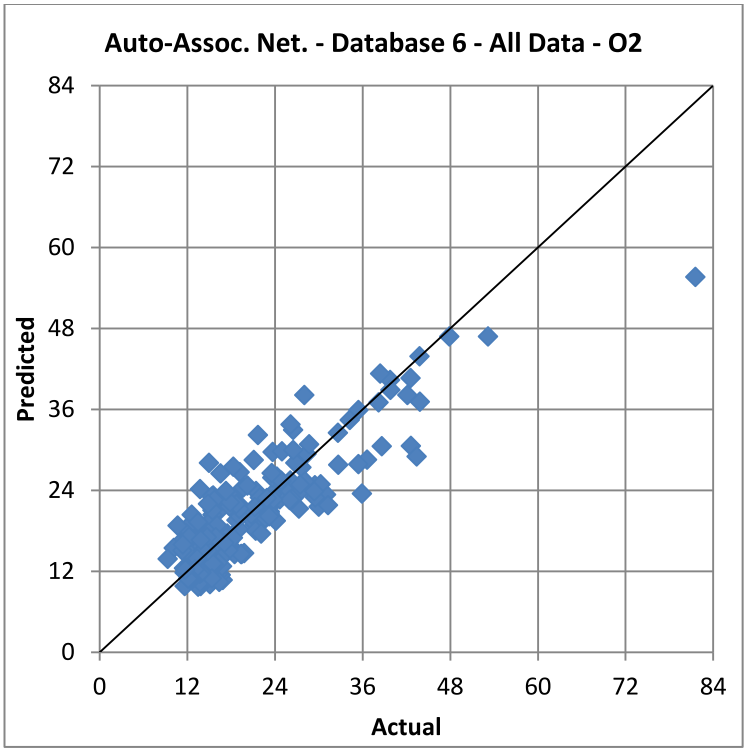

Figure 7-33 Auto-associative Network All Data Accuracy of Database 6, Output 2

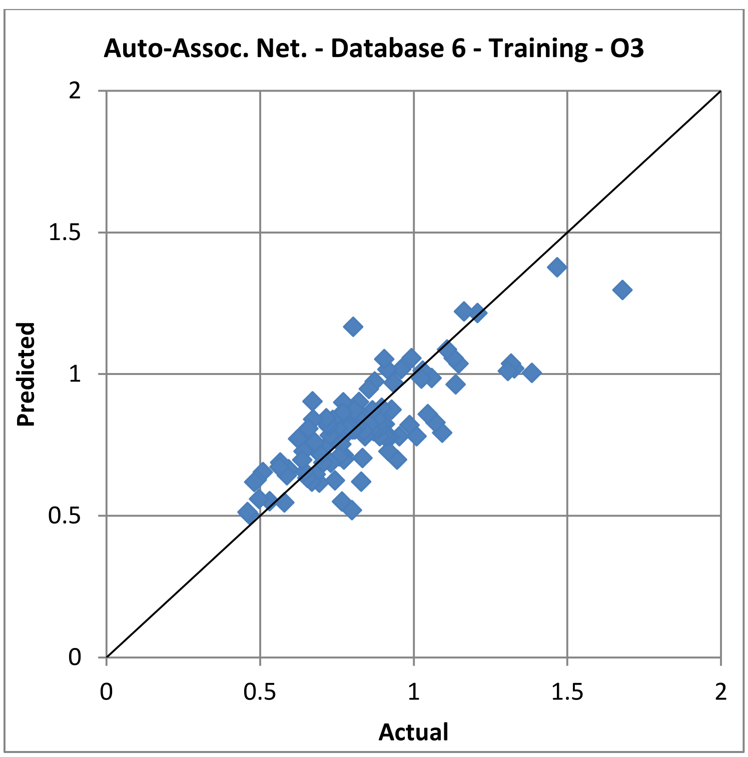

Figure 7-34 Auto-associative Network Training Accuracy of Database 6, Output 3

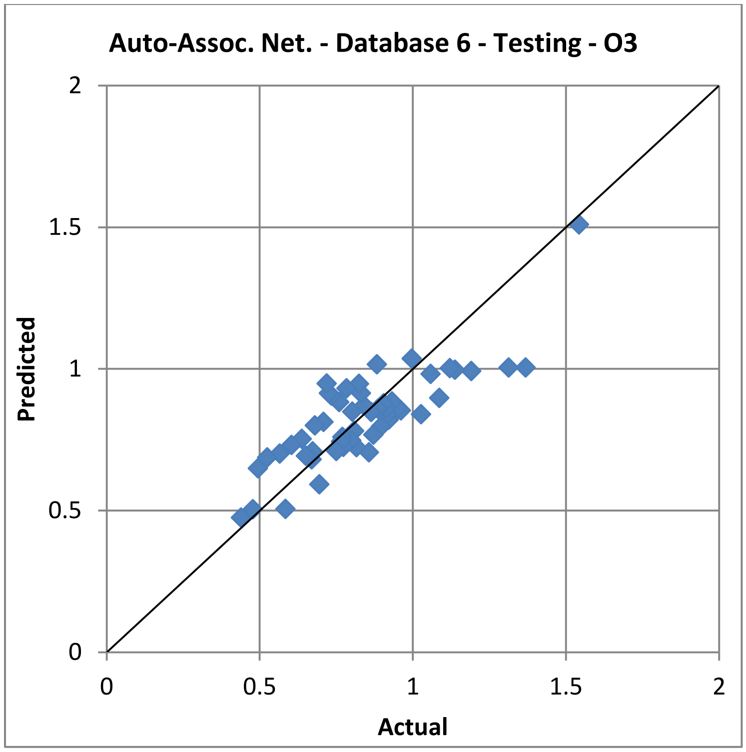

Figure 7-35 Auto-associative Network Testing Accuracy of Database 6, Output 3

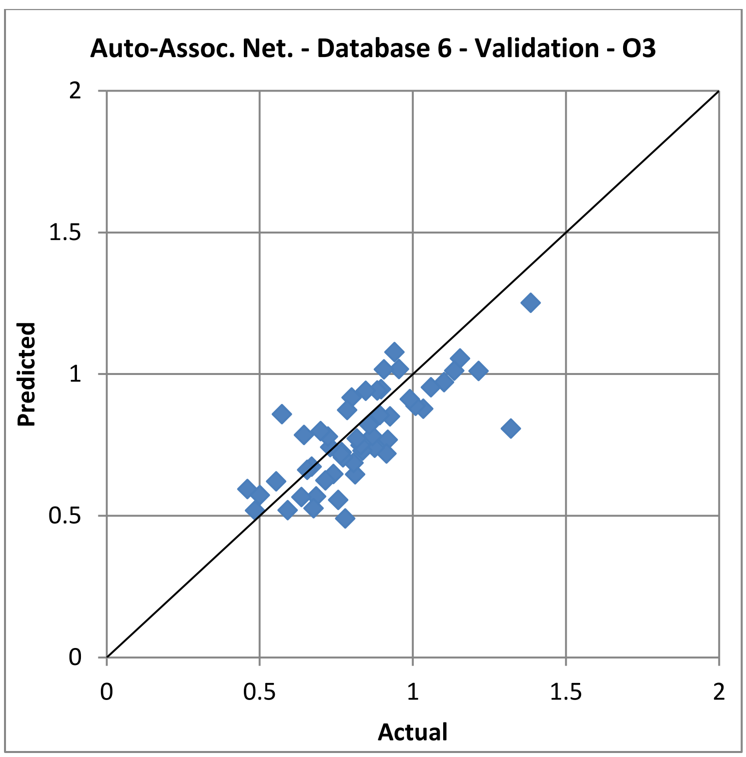

Figure 7-36 Auto-associative Network Validation Accuracy of Database 6, Output 3

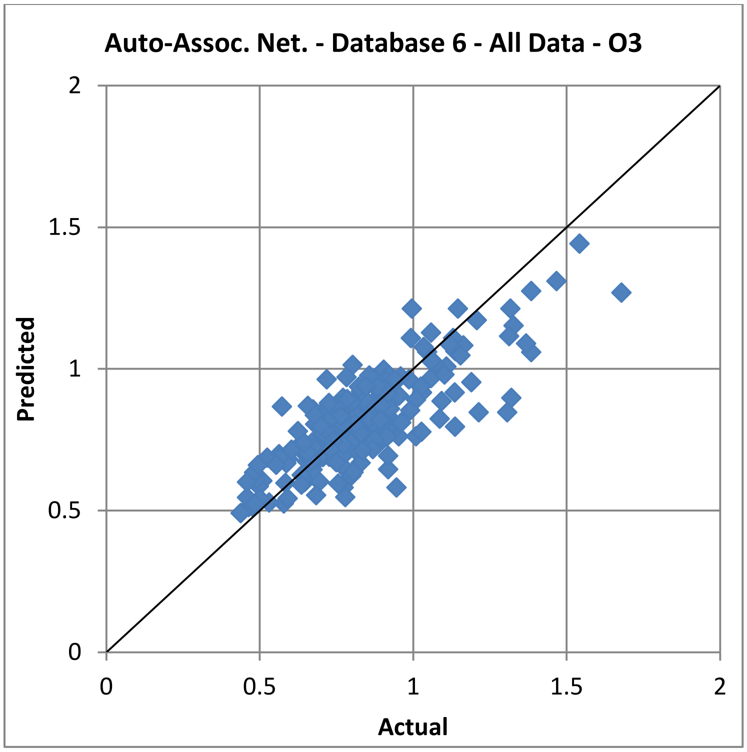

Figure 7-37 Auto-associative Network All Data Accuracy of Database 6, Output 3

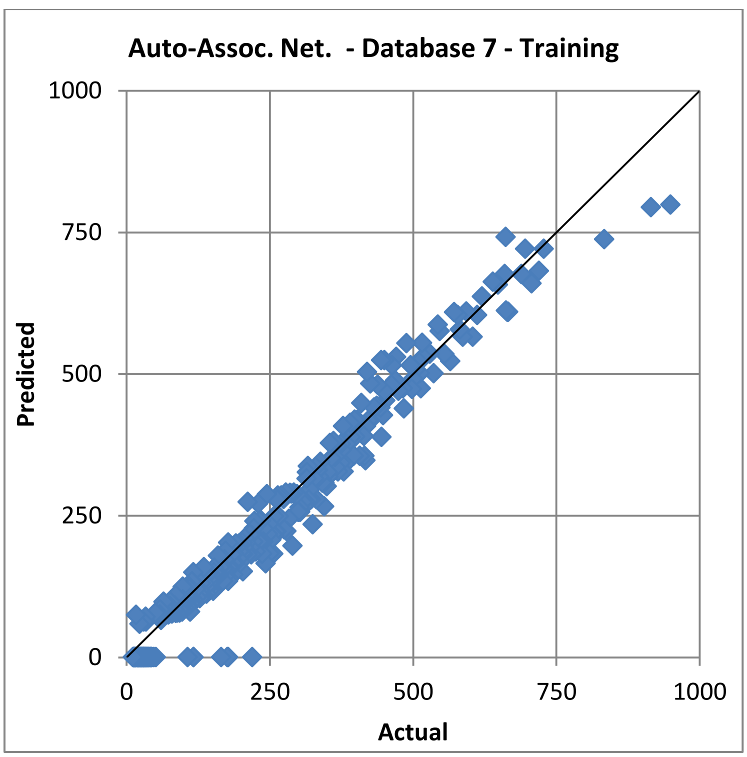

Figure 7-38 Auto-associative Network Training Accuracy of Database 7

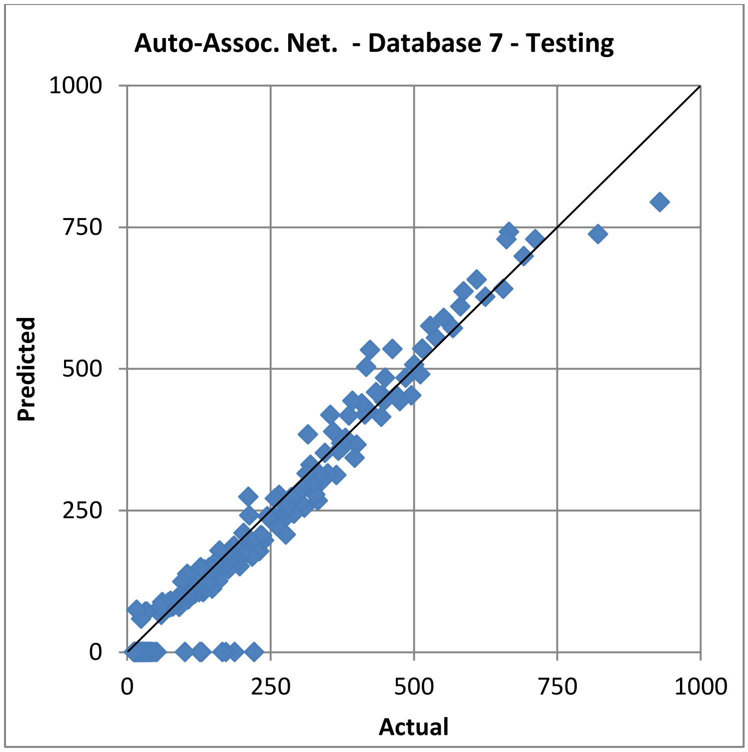

Figure 7-39 Auto-associative Network Testing Accuracy of Database 7

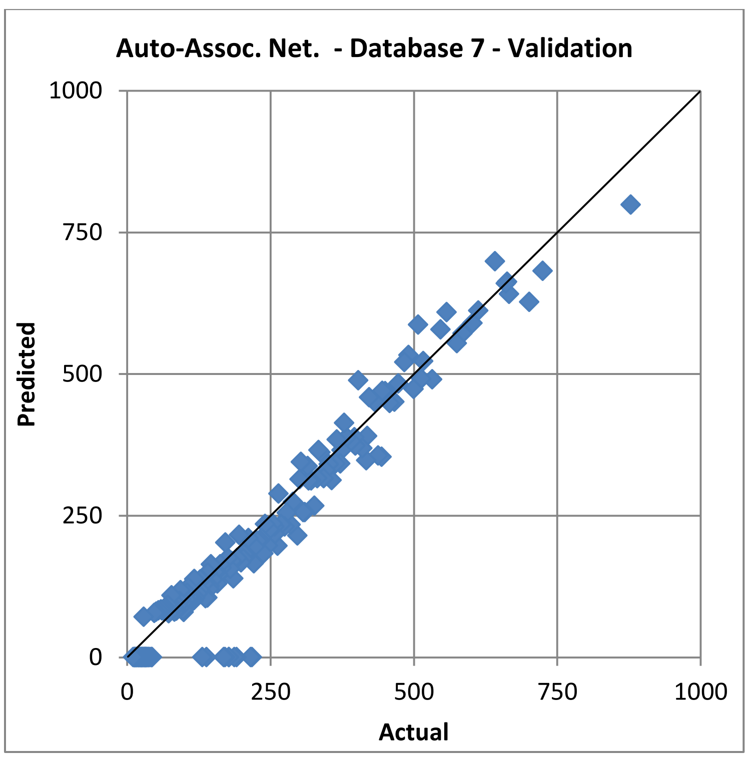

Figure 7-40 Auto-associative Network Validation Accuracy of Database 7

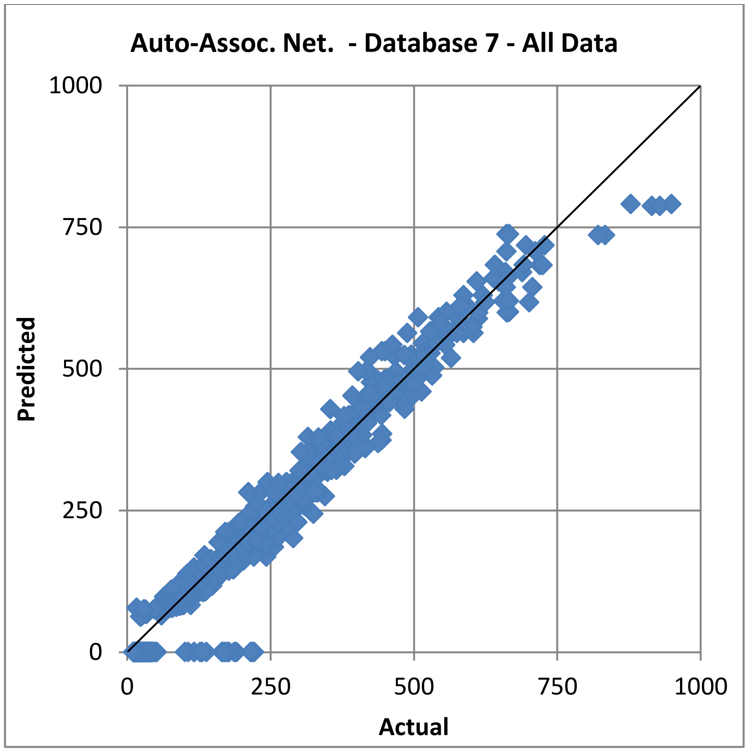

Figure 7-41 Auto-associative Network All Data Accuracy of Database 7

Table 7-1 Statistical Accuracy of Auto-associative Network Models for Database 1 to Database 5

Accuracy Measures		AUTO-ASSOCIATIVE NETWORK MODELS					
		Database 1	**Database 2**	**Database 3**	**Database 4**	**Database 5**	
		Output 1	Output 1	Output 1	Output 1	Output 1	Output 2
		8-(3-6)-20000-8	8-(6-6)-3100-8	13-(7-8)-20000-13	7 - (5-7)-20000-7	4-(4-4)-20000-4	4-(4-5)-20000-4
TR	MARE	6.262	4.8125	12.544	20.557	0.240	1.199
	R^2	0.9868	0.4795	0.9650	0.8539	0.9936	0.9299
	MRSE	9.6230	0.4911	58.4512	68.8665	0.2676	0.6344
TS	MARE	6.937	5.1122	16.594	21.863	0.286	1.037
	R^2	0.9802	0.1798	0.9519	0.8363	0.9817	0.9354
	MRSE	16.5741	0.8469	103.2161	102.4501	0.5100	0.8233
VAL	MARE	7.163	6.5996	22.139	20.260	0.257	1.140
	R^2	0.9882	0.0009	0.7209	0.8604	0.9925	0.9420
	MRSE	12.9810	1.1071	210.3098	95.4665	0.4049	0.8142
ALL DATA	MARE	7.269	4.3912	14.770	18.321	0.206	1.132
	R^2	0.9839	0.3788	0.9342	0.8653	0.9933	0.9329
	MRSE	7.4928	0.3425	55.7573	46.2396	0.1825	0.4266
FINAL STRUCTURE		**8 - 6 - 8**	**8 - 6 - 8**	**13 - 8 - 13**	**7 - 7 - 7**	**4 - 4 - 4**	**4 - 5 - 4**

Table 7-2 Statistical Accuracy of Auto-associative Network Models for Databases 6 and 7

Accuracy Measures		AUTO-ASSOCIATIVE NETWORK MODELS			
		Database 6			Database 7
		Output 1	Output 2	Output 3	Output 1
		17-(1-7)-20000-17	17 - (6-7)-20000-17	17 -(1-2)-18100-17	16-(7-8)-20000-16
TR	MARE	3.486	17.319	7.963	30.249
	R^2	0.8766	0.7564	0.8561	0.9660
	MRSE	0.0033	0.4821	0.0086	1.8678
TS	MARE	4.434	17.465	10.055	32.960
	R^2	0.7841	0.7311	0.7535	0.9533
	MRSE	0.0061	0.6290	0.0155	3.1479
VAL	MARE	4.764	20.188	13.207	33.130
	R^2	0.7112	0.5817	0.5665	0.9397
	MRSE	0.0068	0.7514	0.0203	3.5617
ALL DATA	MARE	3.855	18.220	8.779	31.243
	R^2	0.8347	0.7397	0.8038	0.9553
	MRSE	0.0027	0.3251	0.0066	1.4805
FINAL STRUCTURE		**17 - 7 - 17**	**17 - 7 - 17**	**17 - 2 - 17**	**16 - 8 - 16**

Table 7-3 Increase of Mean Absolute Relative Error (MARE) for seven databases

Database #	OUTPUT	MARE		
		Static ANN	Auto-associative	Increase
Database 1	Output 1	4.069	7.269	79%
Database 2	Output 1	3.9681	4.3912	11%
Database 3	Output 1	12.719	14.770	16%
Database 4	Output 1	20.359	18.321	-10%
Database 5	Output 1	0.186	0.206	11%
	Output 2	1.125	1.132	1%
Database 6	Output 1	5.416	3.855	-29%
	Output 2	11.529	18.220	58%
	Output 3	8.009	8.779	10%
Database 7	Output 1	12.380	31.243	152%

Table 7-4 Reduction of Coefficient of Determination (R^2) for seven databases

Database #	OUTPUT	R^2		
		Static ANN	Auto-associative	Reduction
Database 1	Output 1	0.9984	0.9839	1%
Database 2	Output 1	0.4554	0.3788	17%
Database 3	Output 1	0.9364	0.9342	0%
Database 4	Output 1	0.8549	0.8653	-1%
Database 5	Output 1	0.9944	0.9933	0%
	Output 2	0.9333	0.9329	0%
Database 6	Output 1	0.6612	0.8347	-26%
	Output 2	0.8721	0.7397	15%
	Output 3	0.8377	0.8038	4%
Database 7	Output 1	0.9831	0.9553	3%

Table 7-5 Increase of Mean Root Square Error (MRSE) for seven databases

Database #	OUTPUT	MRSE		
		Static ANN	Auto-associative	Increase
Database 1	Output 1	2.3740	7.4928	216%
Database 2	Output 1	0.3203	0.3425	7%
Database 3	Output 1	63.7835	55.7573	-13%
Database 4	Output 1	47.9782	46.2396	-4%
Database 5	Output 1	0.1676	0.1825	9%
	Output 2	0.4255	0.4266	0%
Database 6	Output 1	0.0038	0.0027	-29%
	Output 2	0.2276	0.3251	43%
	Output 3	0.0059	0.0066	12%
Database 7	Output 1	0.8466	1.4805	75%

CHAPTER 8

8. DYNAMIC-SEQUENTIAL NETWORK

Models developed purely from data significantly depend on database size. Artificial Neural Networks (ANNs) can be developed from data of any size. However, the generalization of developed models is affected by the size considerably since ANNs are required to generalize for unseen cases. Preferably, data to be used for training should be sufficiently large to cover the possible known variation in the application domain. In some engineering applications, experimental data is expensive and time-consuming to collect. Therefore, some databases may contain limited amounts of data. Developing prediction models with these databases can be challenging in terms of their reliability. Training a network with few datasets typically results in a network that memorizes the data rather than generalizing the desired phenomenon. ANNs approach is a powerful computational technique capable of mapping and capturing many features embedded within large datasets. As training of ANN models requires an adequate amount of datasets to be able to extract knowledge, there can often be insufficient data in the database to both train and test the ANN model. To provide a solution to this issue a new approach is proposed, dynamic-sequential network method, which solves the issue of insufficient data by converting a static problem into a dynamic problem. In other words, dynamic-sequential network uses a feed-forward neural network to obtain more reliable networks with consistent generalization aptitude.

Development of ANN requires partitioning of the database into three sub-databases as stated in previous chapters. The training database is used to update the weights of the network using a learning algorithm. Typically, for a static ANN network each dataset is used only once in training to update the connection weights and threshold values in every epoch. In other words, network training for static ANN network is completed sequentially by utilizing every dataset once during every iteration (i.e., epoch). Accordingly, it takes more iterations for a network to fully extract the information from all datasets. However, the network may end up memorizing the data, in other words over-fitting may occur. With this new approach, every dataset was fed with an initial estimate from a Static ANN network and used five times during every training epoch. Only the initial estimate is fed into the first iteration. After the first iteration, the ANN training program generates an output and feeds them back into the second iteration for the same dataset. This methodology is similar to modeling a material response. For instance, modeling a concrete behavior under a load has a similar modeling logic because every response from previous stage needs to be fed into the current stage. Using the same logic, a static database can be converted into a dynamic database by replicating the target dataset 5 times and including the initial estimate from the static ANN model. In this case, an initial estimate is fed into the network and the network generates an output, then the generated output is fed back into the next dataset to replace the initial estimate while keeping all other input values unchanged. This procedure is sequentially repeated 4 more times. Essentially every dataset is multiplied by five and the network is trained 5 times on the same dataset. Although the number of datasets is multiplied by five, statistical accuracy measures and graphical comparison plots presented in this chapter are only based on the last training step(the fifth sequence) because each dataset will be used in training five times and the fifth prediction value is what this research is intending to explore.

The architecture of a Dynamic-sequential network can be seen in Figure 8-1. To develop a Dynamic-sequential network, all datasets are duplicated five times and another input for the initial response is added. An initial estimate from static ANN network was considered as first guess for the first corresponding dataset. Then the prediction generated for the dataset after every training stage is used in the next training stage for the same data as an initial guess until the training is terminated.

In this chapter, Dynamic-sequential network method was explored to evaluate the consistency and applicability to civil engineering systems by utilizing seven databases. In order to verify the stability of Dynamic-sequential network approach, another statistical assurance concept was examined. In this concept, the previously developed networks were validated by using two different initial estimate configurations, one of which is the mean of the output, and the other one is “0”. The statistical accuracy measures of the validation datasets with different initial estimate configurations were calculated and the results were compared for each database.

Detailed information on the training stages for all seven databases and their determined optimal dynamic-sequential network structures are explained in the following sections.

8.1 Dynamic-Sequential Network Model Development of Database 1

In this case, dynamic-sequential model architecture was designed by considering 8 inputs and 1 output. One of the counted inputs is the initial estimate from the developed static ANN network explained in Chapter 5. A total of 300 datasets were converted into 1500 datasets by duplicating each datasets five times and then the new database was divided into sub-datasets; 785, 360, and 355 datasets were used, respectively, for training, testing, and validation purposes. Based on statistical measures MRSE, MARE, and R^2, the optimal network structure of the Auto-associative model for this database was found at 16 hidden nodes and 19,600 iterations. The corresponding accuracy measures, respectively, on the original 157 and 72 datasets for this network are $MRSE_{tr}$= 1.2130, $MARE_{tr}$= 1.632%, R^2_{tr}= 9998 (for training database) and $MRSE_{ts}$=6.6146, $MARE_{ts}$= 2.666%, R^2_{ts}=0.9970 (for testing database). The training and testing graphical comparison plots between the fifth sequenced prediction and the actual values are, respectively, shown in Figure 8-2 and Figure 8-3. Also, all the corresponding statistical accuracy measures for the training and testing stages are shown in Table 8-1. After the training and testing procedures using, respectively, 785 and 360 datasets, validation was conducted on the remaining 355 datasets. The graphical comparison plot, for the validation stage, between prediction and actual response is shown in Figure 8-4 by considering only the fifth sequenced of each dataset. Once the validation stage is completed, all of the 1500 datasets were used to retrain the network at the previously determined optimal structure to obtain the generalized response throughout the 300 datasets. The graphical comparison plot for the 300 datasets is shown in Figure 8-5. Statistical accuracy measures for validation and all data cases are also shown in Table 8-1. As noted in previous chapters, the *8-(1-16)-19600-1* notation specifies the determined architecture of the optimum network where each number, respectively, represents: number of inputs (8), initial number of hidden nodes (1), final number of hidden nodes (16), number of iterations (19600), and number of outputs (1). Final network structure is represented as *8-16-1,* which are, respectively: number of inputs, number of hidden nodes, and number of outputs. Dynamic-sequential network generates predictions as soon as the input parameters are entered. Static ANN network and Dynamic-sequential network work simultaneously to generate the desired output values.

8.2 Dynamic-Sequential Network Model Development of Database 2

A database consisting of 100 datasets was used to develop the desired Dynamic-sequential network for Database 2. As noted previously, the databases to be used for modeling were converted to a dynamic database first and then divided into three sub-categories such as training, testing, and validation. In this case, 500 datasets were used for modeling; 275 datasets are used for training, 115 datasets for testing, and 110 datasets for validation. The input vector consisted of 8 parameters and the output vector consisted of 1 parameter. The optimal structure for the Dynamic-sequential network was found at 3 hidden nodes and 20,000 iterations. A graphical comparison of training stage predictions against actual values is depicted in Figure 8-6. Dynamic-sequential network for training stage on 55 datasets yielded a mean root square error, $MRSE_{tr}$ of 0.3839, mean absolute relative error, $MARE_{tr}$ of 3.6220%, and coefficient of determination, R^2_{tr} of 0.6681. Similarly, graphical comparison of testing stage on 23 datasets is shown in Figure 8-7 and statistical accuracy measures for this network are $MRSE_{ts}$ of 1.0838, $MARE_{ts}$ of 5.9117%, and R_{ts}^2 of 0.3861.

To further validate the optimal network predictions, 110 datasets are used. Figure 8-8 presents the graphical comparison between the predicted and the actual values on the original 22 datasets. Corresponding statistical measures are given in Table 8-1. Once the validation stage is completed, the 500 datasets were used to retrain the network at the optimal structure. It can be concluded from the graphical prediction plot of 100 datasets in Figure 8-9 and the corresponding statistical accuracy measures in Table 8-1 that using the entire database to retrain the network greatly improves the statistical measures. Overall, performance of the Dynamic-sequential network has attained better statistical accuracy measures than those noted previously for the equivalent static ANN network.

8.3 Dynamic-Sequential Network Model Development of Database 3

To develop Dynamic-sequential model for database 3, a total of 126 datasets were converted to 630 datasets by duplicating every dataset five times. Three hundred fifteen and 160 of total datasets were, respectively, considered as training and testing datasets. The remaining 155 datasets were included in the validation stage after the optimal network was determined. The Dynamic-sequential network was initiated with 13 inputs and 1 output. The best performing network structure was obtained at 8 hidden nodes and 100 iterations. The training and testing statistical measures for training and testing stages on 315 and 160 datasets are shown in Table 8-1 and the corresponding graphical comparison plots are depicted in Figure 8-10 and Figure 8-11. As can be observed from the table and the graphical plots, the training and testing stage produced good accuracy.

Validation was conducted on the remaining 155 datasets, after the training and testing stages. The graphical comparison plot, for the validation stage results (using fifth sequenced predictions) and actual response is shown in Figure 8-12. The statistical accuracy measures on the original 31 datasets are $MRSE_{val}$= 250.2872, $MARE_{val}$= 27.787%, and R^2_{val}= 0.6193. Once the validation stage is finalized, all of the 630 datasets were used to retrain the network at the optimal structure. The statistical accuracy measures on the original 126 datasets are $MRSE_{all}$= 95.9043, $MARE_{all}$= 19.180%, and R^2_{all}= 0.9317. The graphical comparison plot of the 126 datasets is shown in Figure 8-13. The resulting statistical accuracy measures for all Dynamic-sequential network modeling stages are given in Table 8-1.

The statistical measures and the plots indicate that the Dynamic-sequential network for this database is performing fairly well. When all data combined and the network was retrained, the statistical accuracy measures showed notable improvement. It should be noted that the error increase from training MRSE to validation MRSE by Dynamic-sequential network is less than those by static ANN (i.e. 3.4 versus 7.6 times). Similarly, the error increase for MARE by Dynamic-sequential network is less than those by static ANN (i.e. 1.94 versus 2.4 times).

8.4 Dynamic-Sequential Network Model Development of Database 4

To develop Dynamic-sequential network for Database 4; 665, 330, and 330 data sets were used for training, testing, and validation tasks. The input vector consisted of 7 parameters, including the one from static ANN network, and the output vector consisted of 1 parameter. To properly characterize the phenomenon, the Dynamic-sequential network approach with four sequential modeling stages was followed. For this case, the optimal network structure for the Dynamic-sequential model was achieved at 7 hidden nodes and 20,000 iterations. Dynamic-sequential network on 133 datasets for training stage yielded a mean root square error, $MRSE_{tr}$ of 81.6443, mean absolute relative error, $MARE_{tr}$ of 20.581%, and coefficient of determination, R_{tr}^2 of 0.8710. Similarly, statistical accuracy measures for the testing stage are $MRSE_{ts}$ of 133.0335, $MARE_{ts}$ of 21.747%, and R^2_{ts} of 0.7959. Corresponding graphical comparisons of testing and validation stages are, respectively, shown in Figure 8-14 and Figure 8-15. As can be seen from the graphical plots and the statistical accuracy measures listed in Table 8-1, good agreement between actual and predicted values is evident. The predictions by validation datasets and all datasets case were plotted against their corresponding actual values, respectively, in Figure 8-16 and Figure 8-17. Good agreement between the predictions and the actual values can be seen in Table 8-1. The validation MRSE is higher than those of training and testing as expected. Similarly, MARE values for testing and validation increased compared to the one from training. However, the all data MRSE value is the lowest compared to those obtained in previous stages (i.e. training, testing, and validation). In other words, training MRSE value had a reduction of about 39% in error.

8.5 Dynamic-Sequential Network Model Development of Database 5

Database 5 utilizes 325 datasets; 163, 81, and 81 datasets that are for training, testing, and validation purposes. However, these datasets were converted to dynamic databases by reproducing each dataset five times. As previously mentioned in Chapter 4, database 5 has two outputs. For this reason, four sequential stages for static ANN model development process were conducted twice to arrive at two desired prediction models for the two outputs. The optimal network structure for the model 1 was finalized at 4 hidden nodes and 16,900 iterations. The corresponding accuracy measures of model 1 on 163 datasets are listed as $MRSE_{tr}$=0.1901, R^2_{tr}=0.9967, $MARE_{tr}$=0.165% (for training database) and $MRSE_{ts}$ =0.4613, R^2_{ts}=0.9851, $MARE_{ts}$= 0.213% (for testing database). The optimal network for Model 2 on 163 datasets was reached at 4 hidden nodes and 20000 iterations. The corresponding accuracy measures of model 2 for this network are $MRSE_{tr}$ =0.6649, R^2_{tr}=0.9283, $MARE_{tr}$=1.278% (for training database) and $MRSE_{ts}$ =0.8500, R^2_{ts}=0.9283, $MARE_{ts}$= 1.101% (for testing database). Training MRSE value for model 1 increased by about 143% in testing while training MRSE value for model 2 increased by about 27.8% in testing. The training and testing plots on 163 and 81 datasets for model 1 are shown in Figure 8-18 and Figure 8-19. In the plots, the training and testing predictions are closely scattered around the 45 degree line, which means that the predicted values are very close to the actual values. Similarly the training and testing plots for model 2 are also given in Figure 8-20 and Figure 8-21. The corresponding statistical accuracy measures for models 1 and 2 are presented in Table 8-1.

The validation for model 1 and model 2 was conducted on 405 datasets. The validation plots on 81 datasets for model 1 and model 2 are, respectively, given in Figure 8-22 and Figure 8-23. After the validation stage is concluded, all of the 1625 datasets were used to retrain the network at the optimal structure. The comparison plots of model 1 and model 2 for the 325 datasets are, respectively, shown in Figure 8-24 and Figure 8-25. The resulting statistical accuracy measures for the validation and the all data cases are depicted in Table 8-1. All data MRSE statistical measures for both model 1 and model 2 have the best results compared to their previous stages.

8.6 Dynamic-Sequential Network Model Development of Database 6

This database containing highly non-linear behavior and multiple outputs was used to develop appropriate Dynamic-sequential based networks. The original two hundred and ten datasets were converted to dynamic database by reproducing each dataset five times, and then divided the new database into sub-databases: 525, 265, and 260 to be used, respectively, for training, testing, and validation tasks. As stated in Chapter 4, database 6 has three outputs for which Dynamic-sequential model development process was conducted three times to arrive at three desired prediction models for three outputs individually. In other words, Dynamic-sequential model development process was repeated for each output. The number of sub-databases was kept the same for the three models.

Dynamic-sequential network for Model 1 was determined at 3 hidden nodes and 20,000 iterations. This network structure provided the optimal connection weights for the desired predictions. The training and testing accuracy measures on 105 and 53 datasets for model 1 is presented in Table 8-2 along the corresponding plots shown in Figure 8-26 and Figure 8-27. According to the statistical measures, the optimal network performed well in the training stage as well as in the testing stage. However, the MRSE value of the training, 0.0048 deteriorated to 0.0110 for the testing stage, which corresponds to 129.2% increase in error. For the validation stage, the statistical measures changed slightly; however, for the all data stage, MRSE improves to a value of 0.0035, which translates into about 27% reduction in error from training stage. All the statistical measures for the validation and all data stages, respectively on the original 52 and 105 datasets, can be found in Table 8-2 and their corresponding comparison plots are, in the same order, represented in Figure 8-28 and Figure 8-29.

Database used to develop Dynamic-sequential network for model 1 was also utilized for Model 2 by considering 17 inputs, including the one from static ANN network, and 1 output. The optimal network for model 2 was reached at 2 hidden nodes and 20,000 iterations. The accuracy on 102 and 53 datasets for the training and testing stages of the selected network architecture is given in Table 8-2 and the graphical evaluation plots are depicted in Figure 8-30 and Figure 8-31. Validation and all data stages were sequentially followed by the training and testing stages. Figure 8-32 and Figure 8-33, which are the plots for validation and all data predictions, indicate reasonably good agreement between the actual and predicted values. A good agreement between the actual and predicted values can easily be evaluated from Table 8-2.

Following the same modeling procedure, a Dynamic-sequential network for model 3 was developed by considering the same input parameters, used for model 1 and model 2, and output 3. The resulting statistical accuracy measures were obtained at a structure of 6 hidden nodes and 7,100 iterations. Table 8-2 presents all the statistical measures for model 3. Also, corresponding graphical comparisons for the stages are represented in Figure 8-34, Figure 8-35, Figure 8-36, and Figure 8-37. Even though scatter around the 45 degree line is noted in these plots, , most of the output values were predicted well.

8.7 Dynamic-Sequential Network Model Development of Database 7

Database 7 is the last database utilized in this chapter to develop an associated Dynamic-sequential network. The database consists of 792 datasets divided into 396, 198, and 198 datasets used for training, testing, and validation purposes. However, in order to obtain Dynamic-sequential network model, the database was reproduced five times and the number of datasets considered for modeling increased to 3,960 datasets, which was then divided into 1980, 990, and 990 sub-datasets to satisfy the model's training, testing and validation requirements. The optimal network structure was obtained as 6 hidden nodes and 20,000 iterations. The accuracy plots are illustrated in Figure 8-38 and Figure 8-39. The plots show good correlation between actual values and predicted results. As can be observed from Table 8-2, the developed Dynamic-sequential network has reasonably good statistics such as $MRSE_{tr}$= 1.1866, $MARE_{tr}$= 14.910%, and R^2_{tr}= 0.9855. Even though statistical accuracy measures for testing and validation stages deteriorated slightly, they are still considerably good. The accuracy of how good the validation datasets were predicted can be observed in Figure 8-40 and the corresponding statistics are shown in Table 8-2. Combining all datasets and retraining the network improved model statistics noticeably. All data predictions are graphically depicted in Figure 8-41 and the statistical accuracy measures are given in Table 8-2. As a result, Dynamic-sequential network was successfully developed and the statistical accuracy measures are adequate.

8.8 Concluding Remarks

In this chapter, a new approach of feed-forward neural networks was explored by utilizing civil engineering databases. When databases are limited due to the cost and difficulties to collect information, model developing tools are vital to those who struggle to make a decision. In these cases, reliability of the tools is really important and current methods may not be adequate to generate a model with generalization capability. Artificial neural networks can be developed from databases of any size, but the reliability of some databases may be questionable. For this reason, the new method introduced in this chapter tried to answer the question: can we improve the generalization capability of the current models or/and reduce the error? In order to implement this method, the database is converted to a dynamic database by duplicating each dataset five times and including an initial estimate from static ANN network for the first iteration only.

In order to develop Dynamic-sequential networks, seven civil engineering databases were used. Each database was converted to dynamic database first, and then the usual training, testing, and validation stages were performed. Even though the number of datasets for modeling was increased, the statistical measures and graphical plots were completed by considering the original size to validate the predictions at the end of fifth sequence. As can be perceived from the graphical results shown in Figure 8-2 to Figure 8-41 and the statistical accuracy measures listed in Table 8-1 and Table 8-2, the developed Dynamic-sequential network models attain good prediction accuracy. A good trend between predicted and actual values is apparent for all databases considered herein.

To assess the performance of the new models, the predictions of the developed models were compared (in terms of the same statistical accuracy measures) to the prediction obtained from their counterpart static ANN networks. The reduction of MARE for the seven databases can be seen in Table 8-3. The reduction for databases 1, 2, 4, and 6 – Output 1 are ranging from 8% to 62%. Database 1 had 62% MARE reduction, which is the highest among the databases. The rest of databases did not perform well and some had an increase in MARE. In Table 8-3, the databases, which did not perform better than static ANN network, are indicated as negative reductions. Table 8-4 presents the change in R^2 values by the Dynamic-sequential networks and static ANN networks. Only database 2 and database 6 – output 1 has shown improvement. The remaining databases either had negative improvements or did not have any changes. However, MRSE values by Dynamic-sequential networks depicted in Table 8-5 has more reduction than the MARE case. The positive reduction by Dynamic-sequential networks is ranging from 1% to 67%. Five databases; Databases 1, 2, 5 – output 1, 6 –output 1, and 7 had reduction in MRSE values even though some of them is small. As can be seen from the results presented in Table 8-3, Table 8-4, and Table 8-5, Dynamic-sequential network improved several of the statistical accuracy measures.

In order to verify the stability of Dynamic-sequential network approach, statistical assurance concept was examined. The previously developed networks were validated by using two different initial estimate configurations, one of which is the mean of the output, and the other one is “0”. The statistical accuracy measures of the validation datasets with different initial estimate configurations were calculated and the results were compared for each database. This concept intended to imitate a situation where there is no initial estimate available. In addition, the networks were optimized on initial values from static ANN and now their performance without an initial estimate is intended to be investigated. The statistical accuracy measures for all seven databases are given in Table 8-6 to Table 8-15. The three initial estimates presented in the tables are static ANN, Value of “0”, and the Average value of the variable. Value “0” represents the case where the initial estimates for validation datasets were considered as “0”. All databases, except database 4 and database 7, have shown good trends. The statistical accuracy measures of those databases have matched successfully with the others. Even though some databases have shown slight changes, they are considerably small. For example, the MRSE of Value “0” is 350.2783, while the MRSE of static ANN is 347.9069. The difference between these two measures is negligible. For database 4, using the initial estimate values as “0” caused an increase in error. MRSE of static ANN and Average value was, respectively, calculated as 154.4276 and 156.8112, while the MRSE of Value “0” had a value of

382.1132. The error by Value “0” is about 2.46 times higher than those by the static ANN. Although database 7 is another database with this exception, increase in error for database 7 is not as high as those noted for database 4. The best statistical measures for database 7 were noted when using initial estimate from the static ANN model. Using average value for initial estimate yielded MRSE value of 2.0871 which is about 7.8% higher than that by static ANN. For database 4 and database 7, using the initial estimate from static ANN seems to provide more accurate predictions. The decision to choose the best option in terms of initial estimate can be easily made by conducting the statistical assurance analysis once the optimal structure of the Dynamic-sequential network models is determined.

Generally, once the Dynamic-sequential network is trained with the initial estimates from static ANN network, then any value fed into the network (as first estimate) will be stabilized by the network within the 5 sequential iterations. It is recommended that the stability of any developed Dynamic-sequential network be examined to assure convergence regardless of the initial estimate.

8.9 Figures and Tables

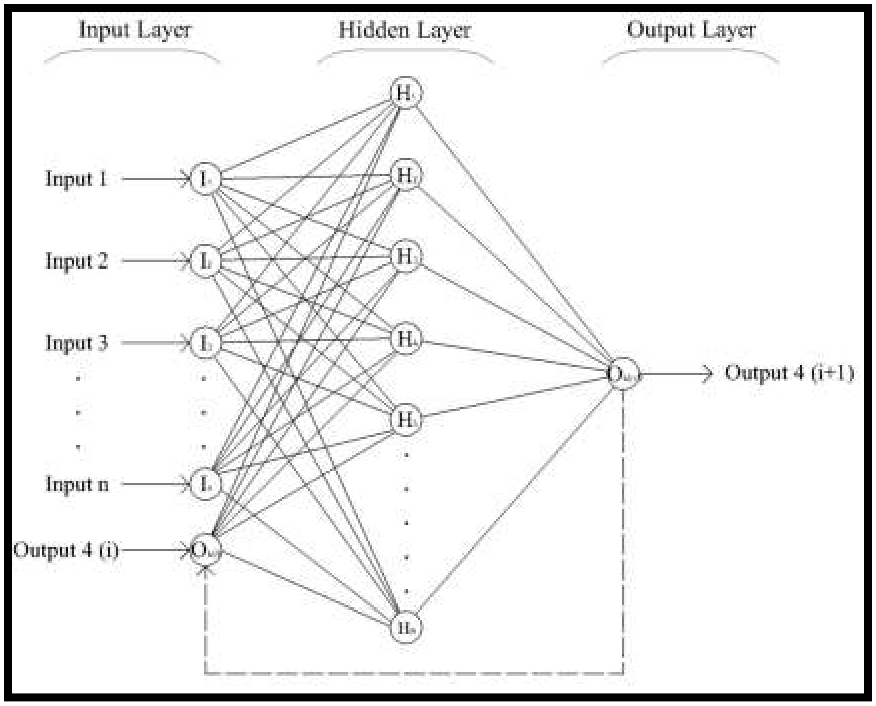

Figure 8-1 Architecture of a Dynamic-Sequential Network

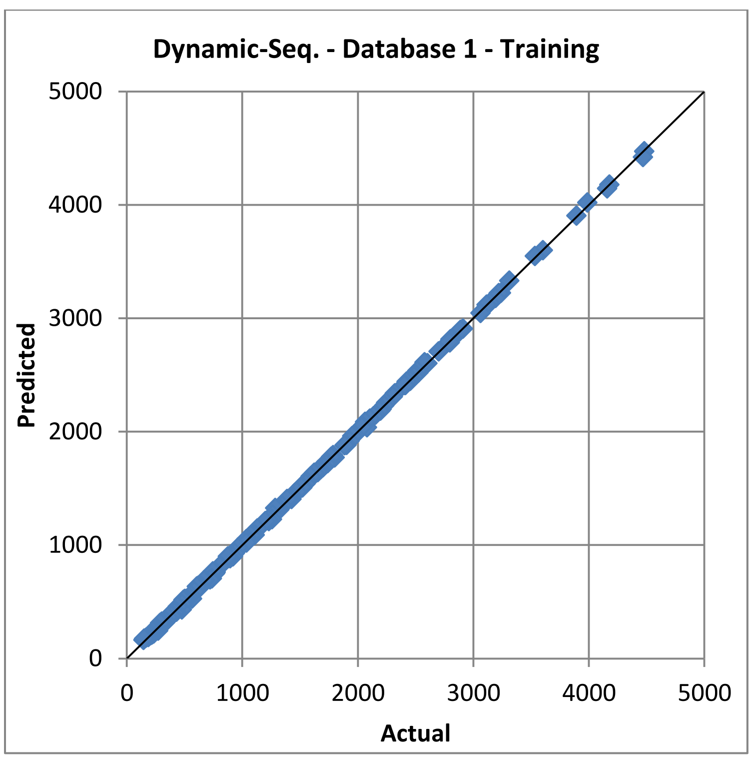

Figure 8-2 Dynamic-Sequential Network Training Accuracy of Database 1

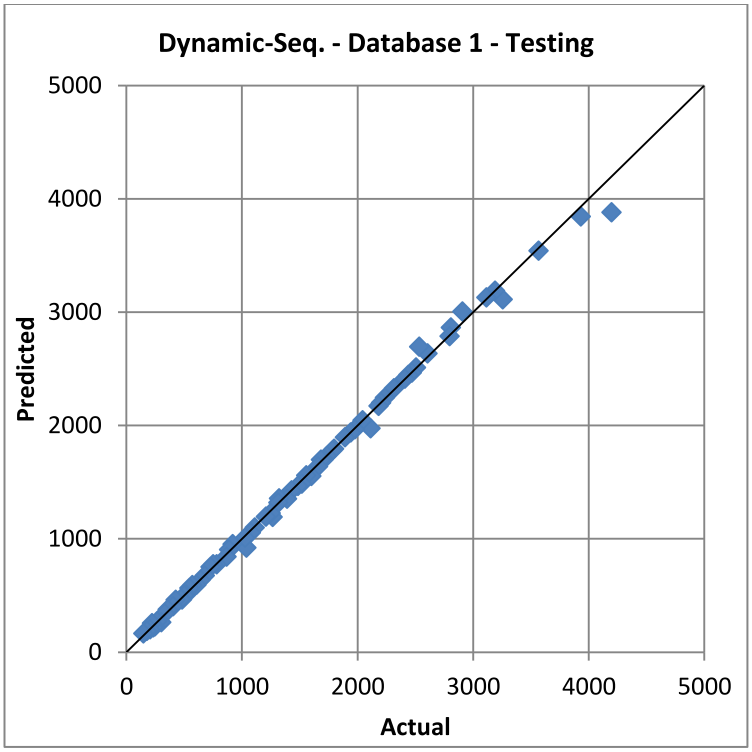

Figure 8-3 Dynamic-Sequential Network Testing Accuracy of Database 1

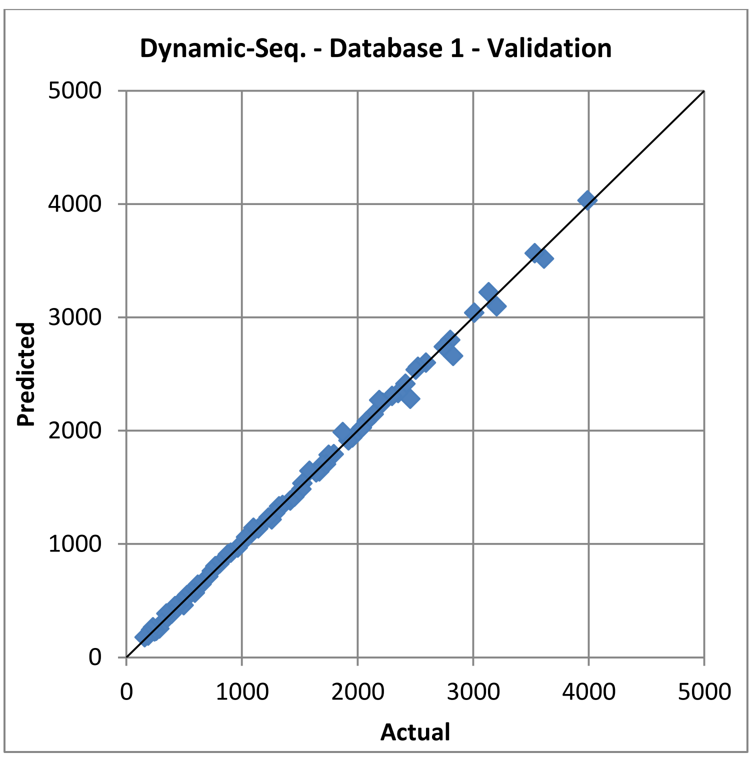

Figure 8-4 Dynamic-Sequential Network Validation Accuracy of Database 1

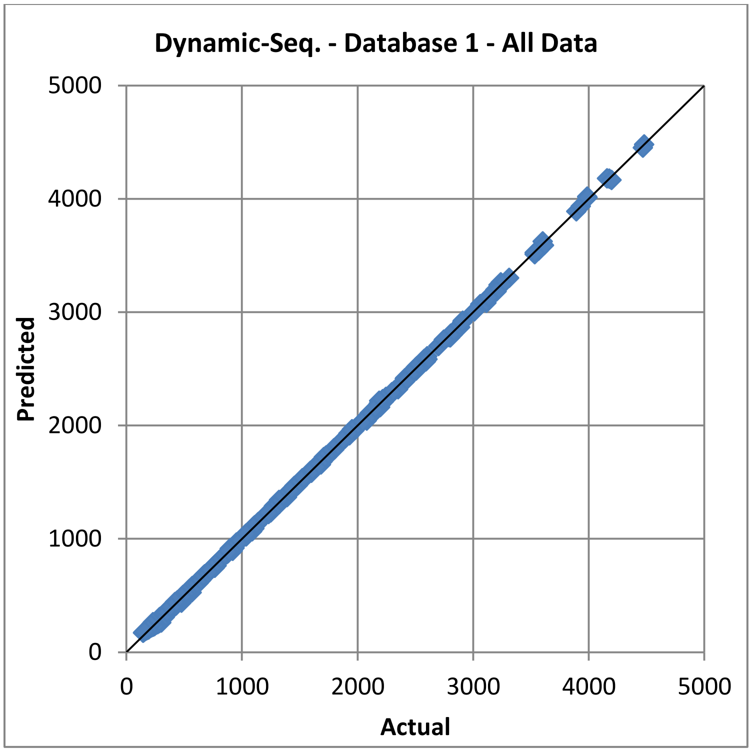

Figure 8-5 Dynamic-Sequential Network All Data Accuracy of Database 1

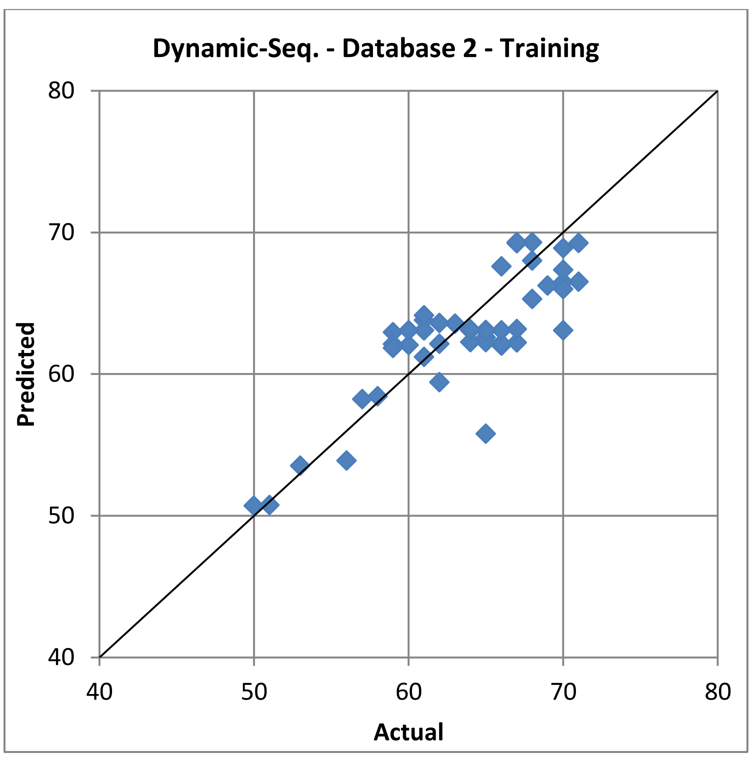

Figure 8-6 Dynamic-Sequential Network Training Accuracy of Database 2

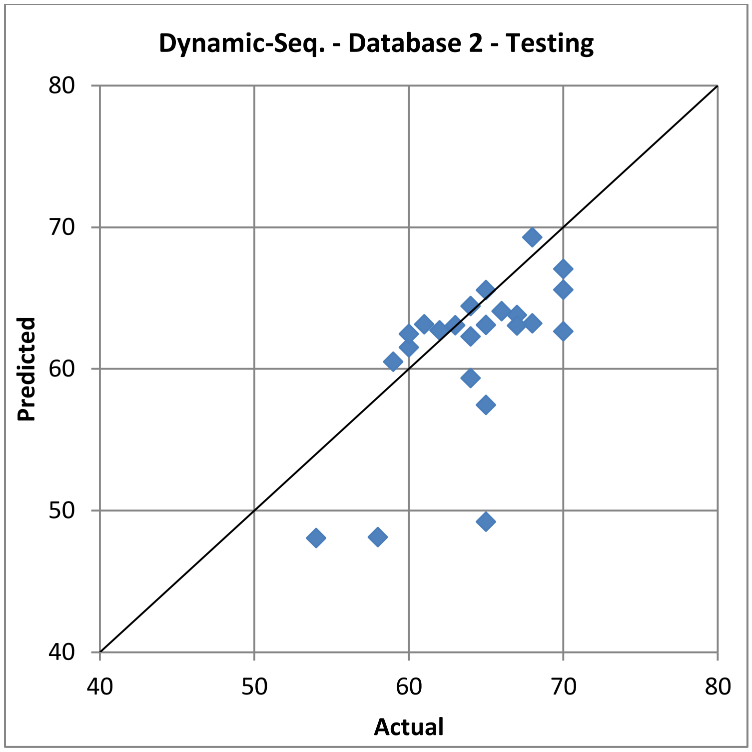

Figure 8-7 Dynamic-Sequential Network Testing Accuracy of Database 2

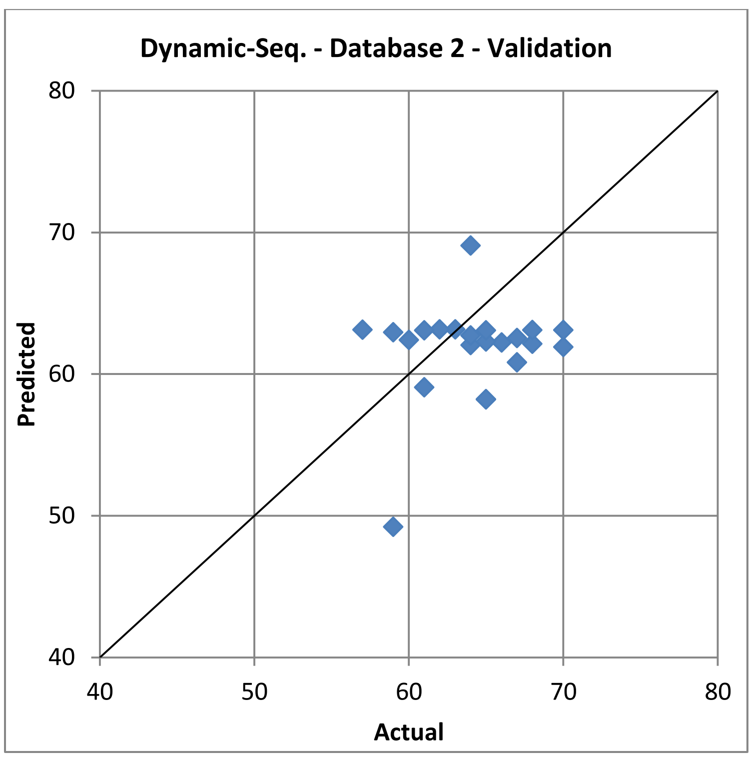

Figure 8-8 Dynamic-Sequential Network Validation Accuracy of Database 2

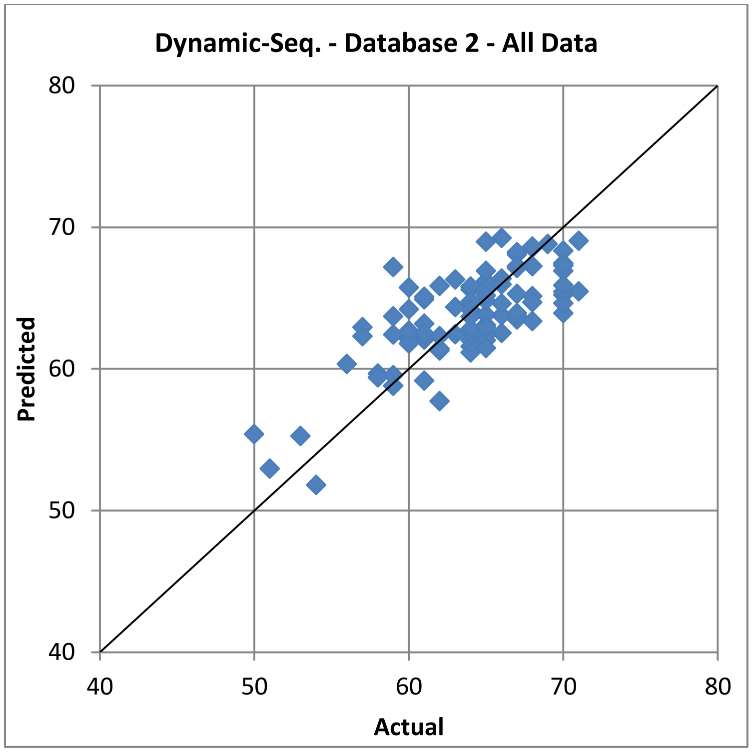

Figure 8-9 Dynamic-Sequential Network All Data Accuracy of Database 2

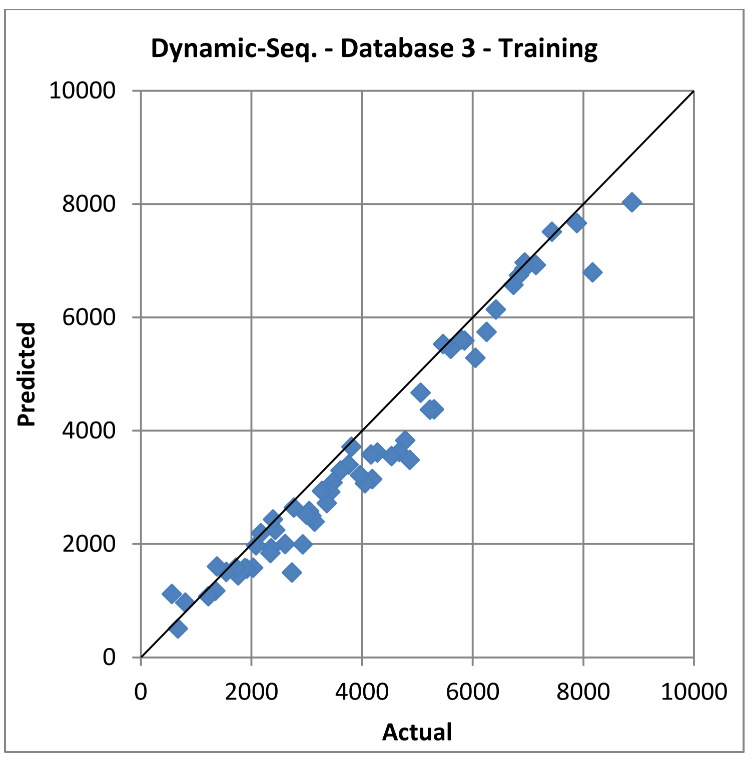

Figure 8-10 Dynamic-Sequential Network Training Accuracy of Database 3

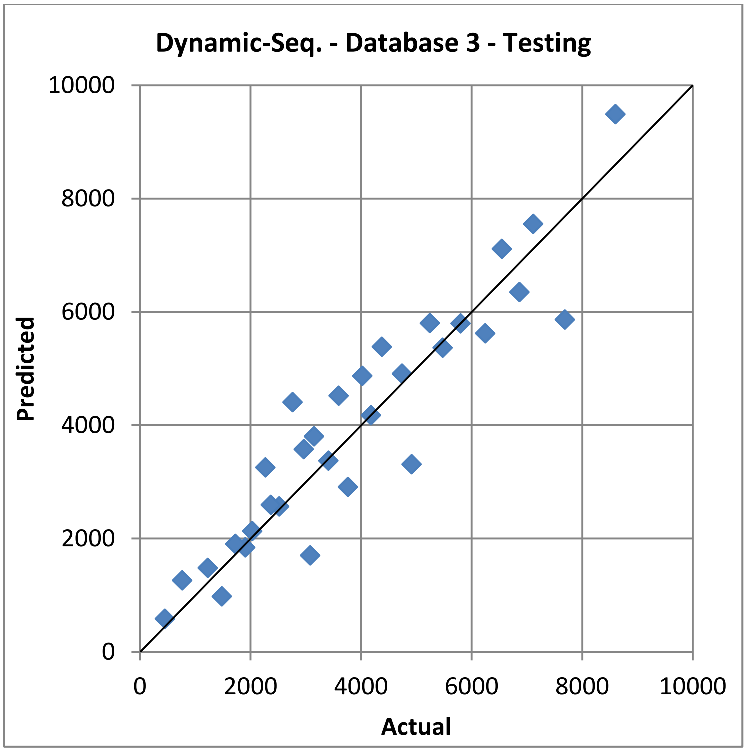

Figure 8-11 Dynamic-Sequential Network Testing Accuracy of Database 3

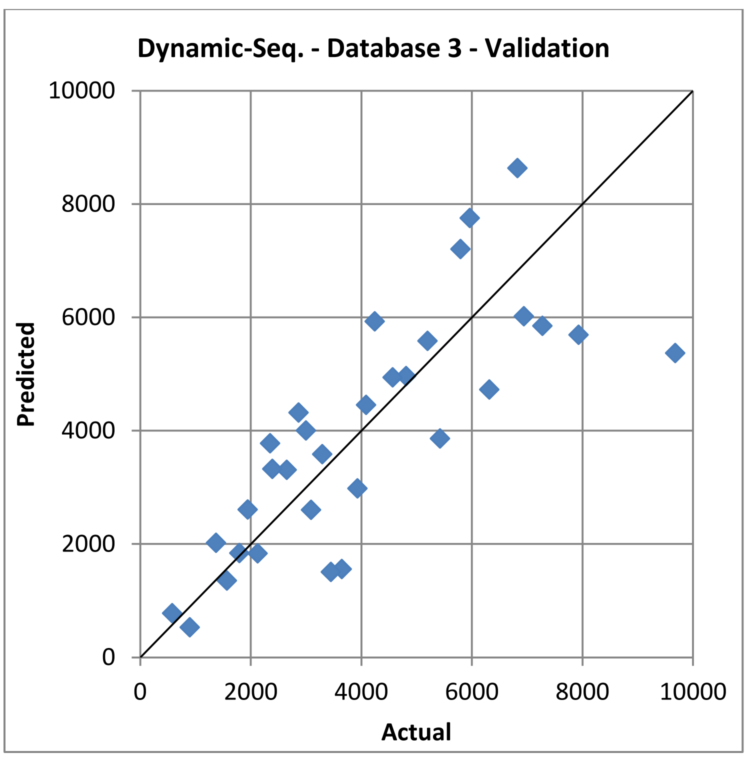

Figure 8-12 Dynamic-Sequential Network Validation Accuracy of Database 3

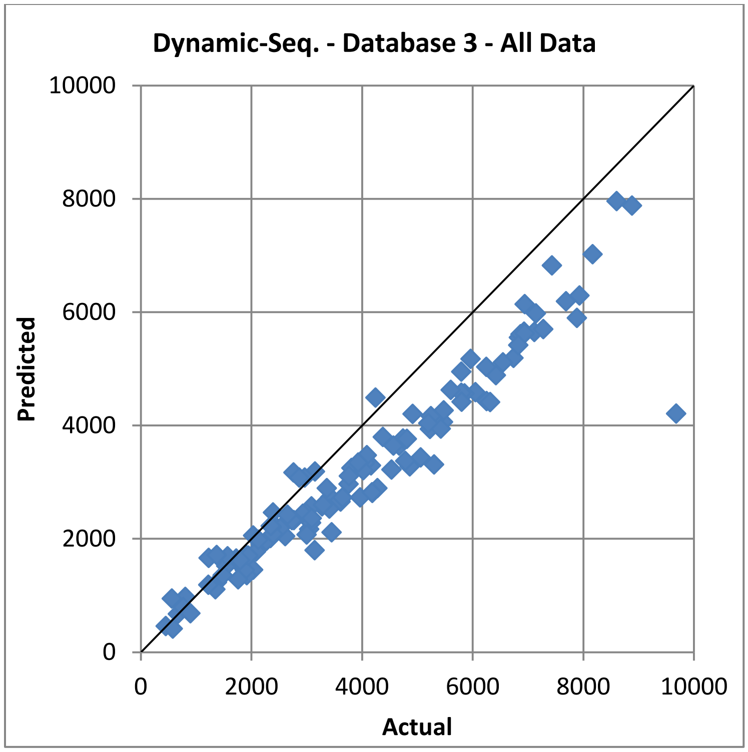

Figure 8-13 Dynamic-Sequential Network All Data Accuracy of Database 3

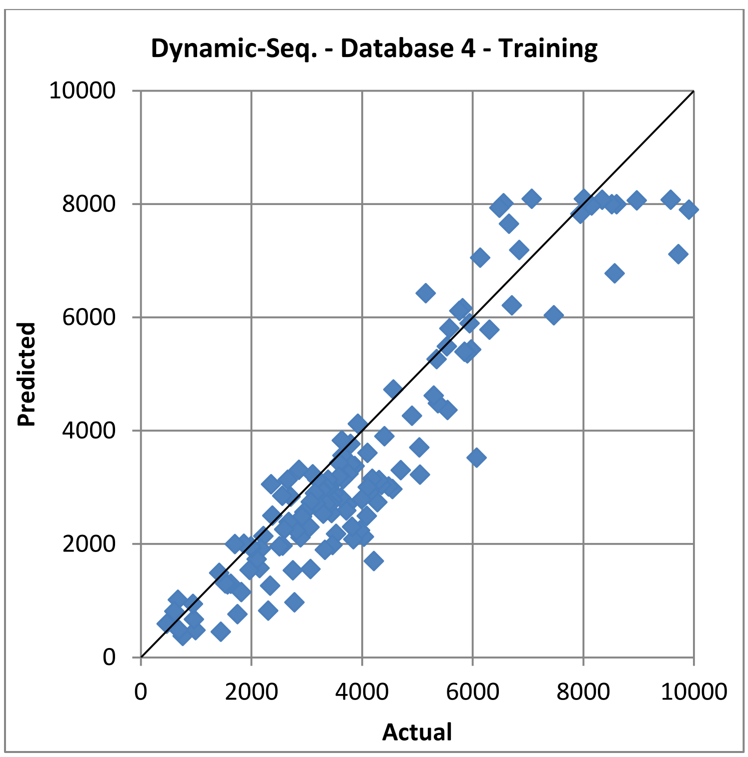

Figure 8-14 Dynamic-Sequential Network Training Accuracy of Database 4

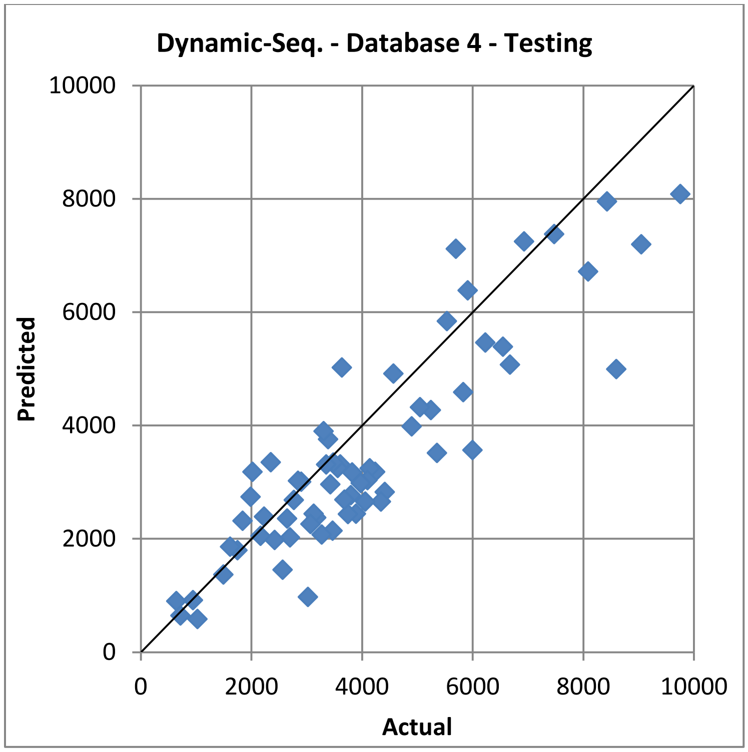

Figure 8-15 Dynamic-Sequential Network Testing Accuracy of Database 4

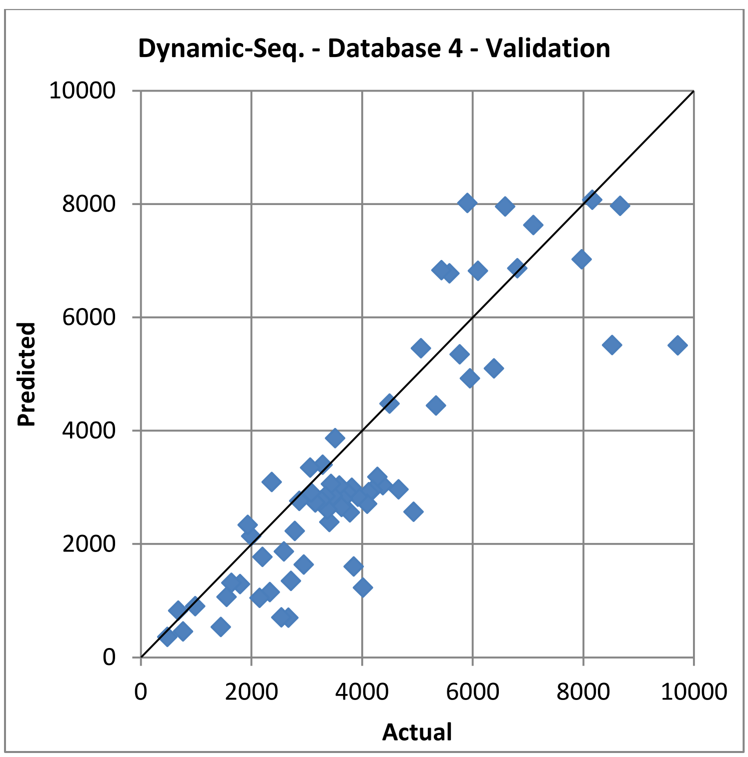

Figure 8-16 Dynamic-Sequential Network Validation Accuracy of Database 4

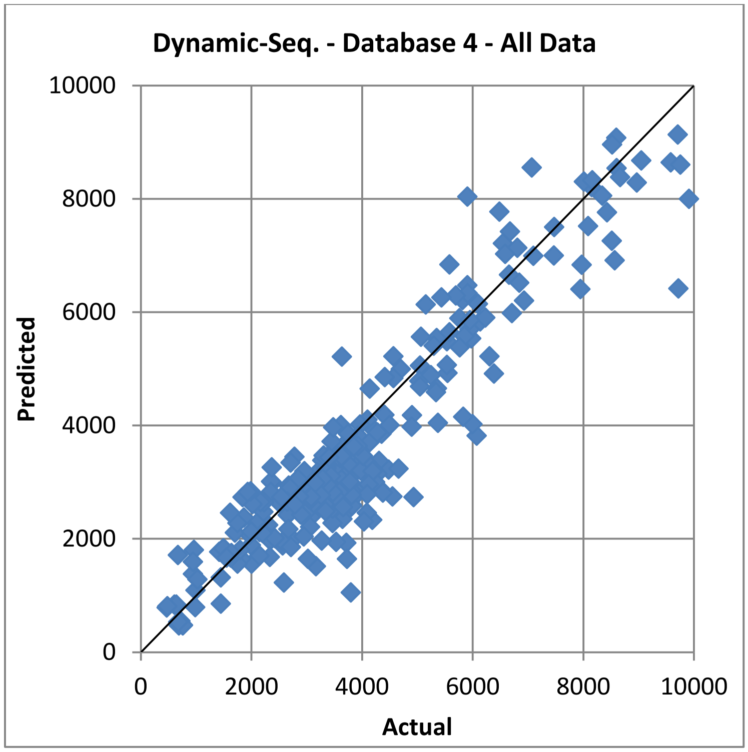

Figure 8-17 Dynamic-Sequential Network All Data Accuracy of Database 4

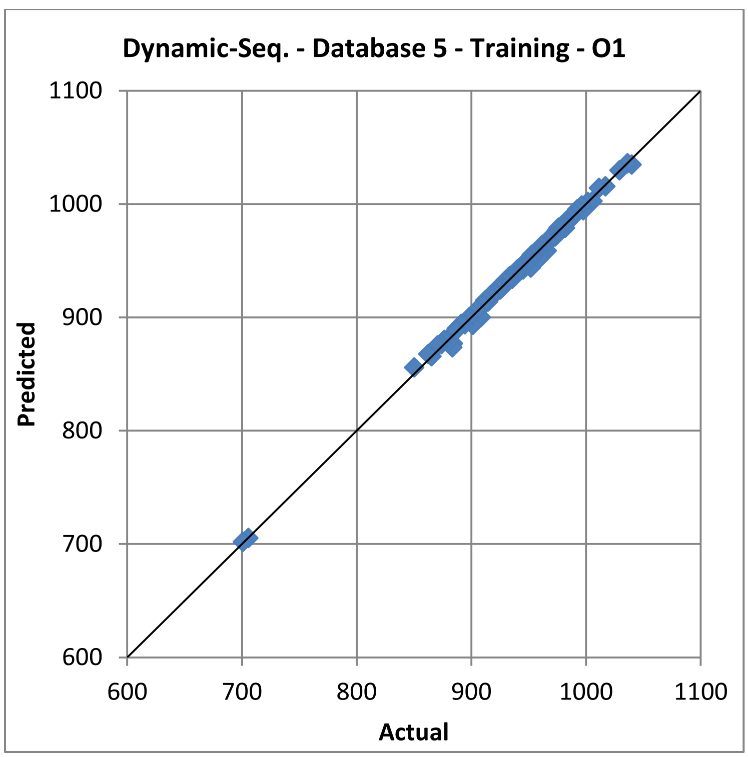

Figure 8-18 Dynamic-Sequential Network Training Accuracy of Database 5, Output 1

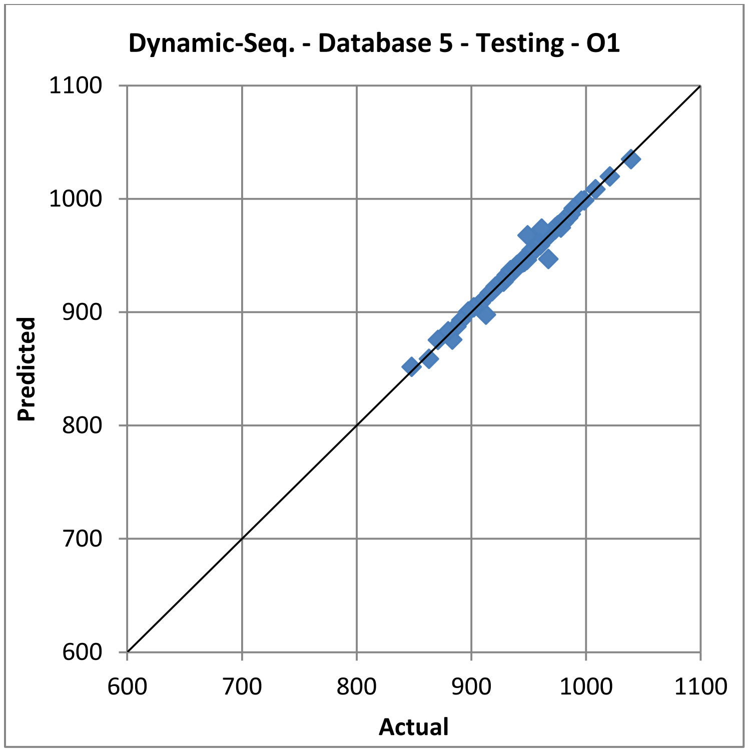

Figure 8-19 Dynamic-Sequential Network Testing Accuracy of Database 5, Output 1

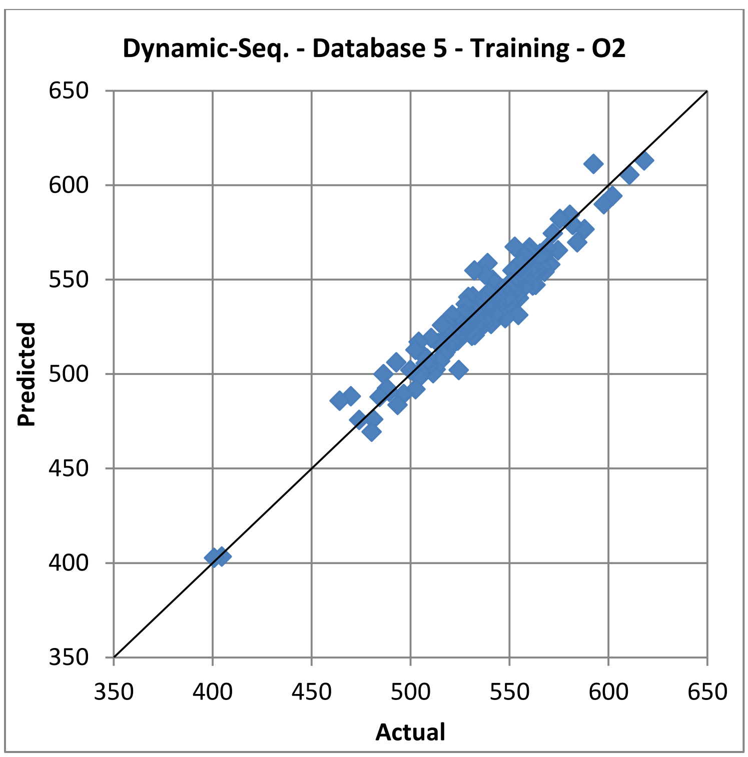

Figure 8-20 Dynamic-Sequential Network Training Accuracy of Database 5, Output 2

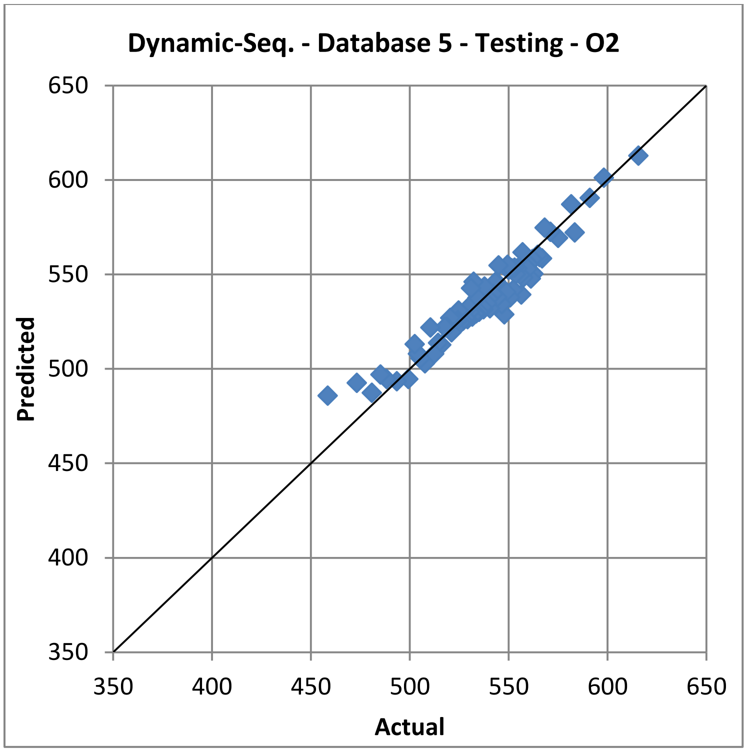

Figure 8-21 Dynamic-Sequential Network Testing Accuracy of Database 5, Output 2

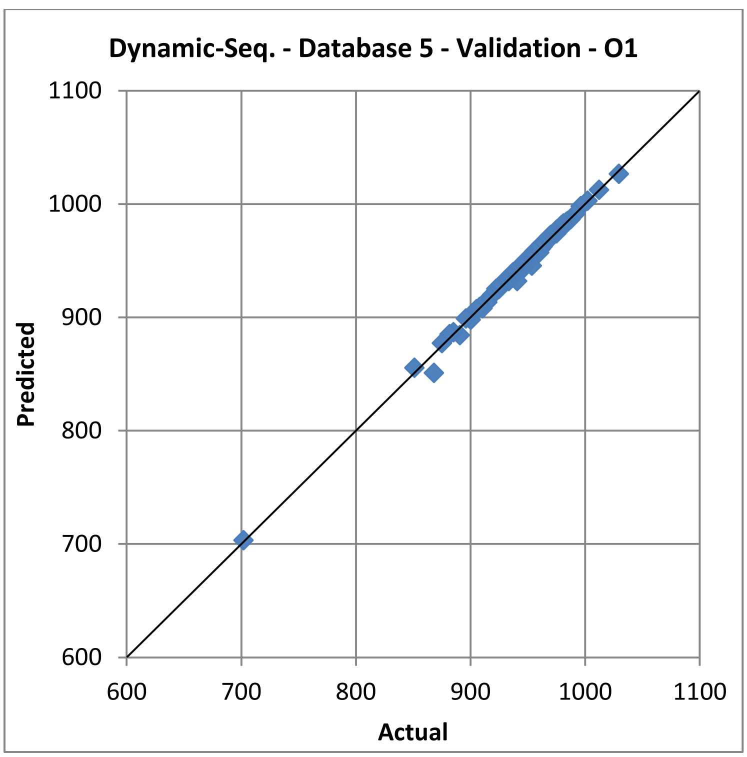

Figure 8-22 Dynamic-Sequential Network Validation Accuracy of Database 5, Output 1

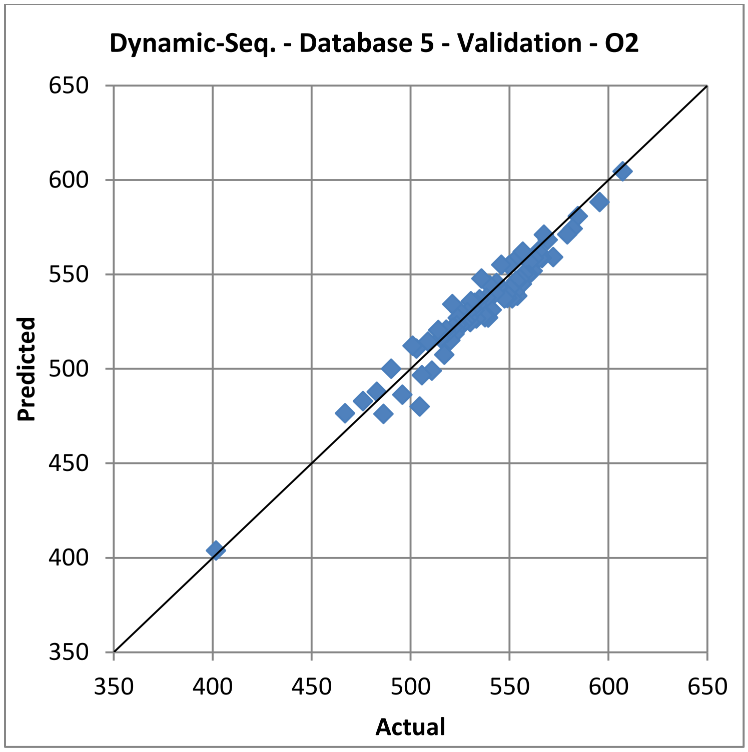

Figure 8-23 Dynamic-Sequential Network Validation Accuracy of Database 5, Output 2

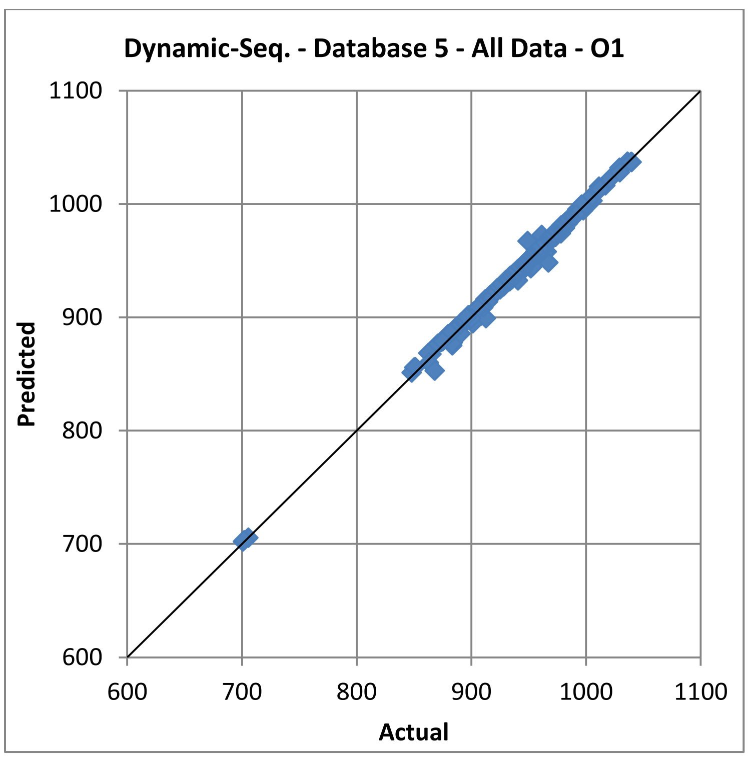

Figure 8-24 Dynamic-Sequential Network All Data Accuracy of Database 5, Output 1

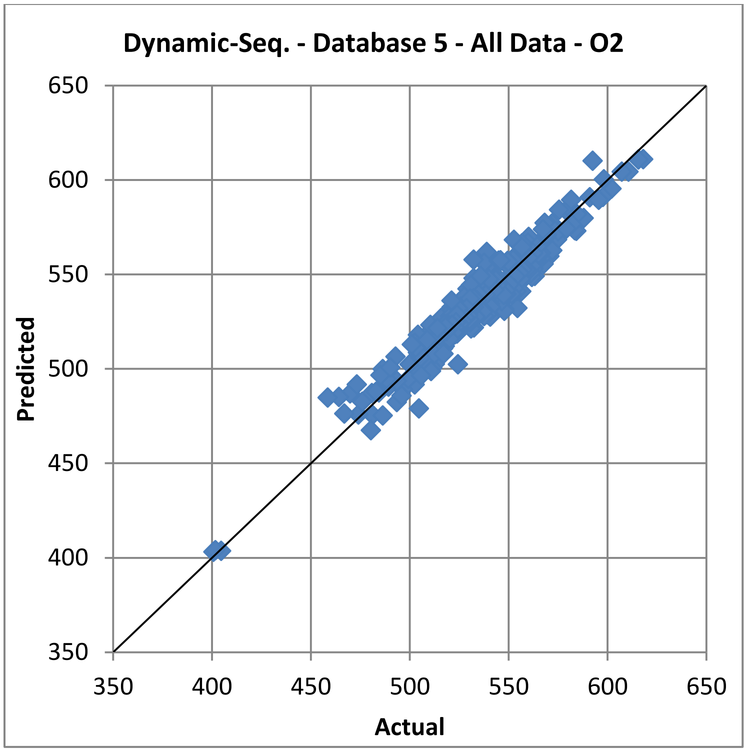

Figure 8-25 Dynamic-Sequential Network All Data Accuracy of Database 5, Output 2

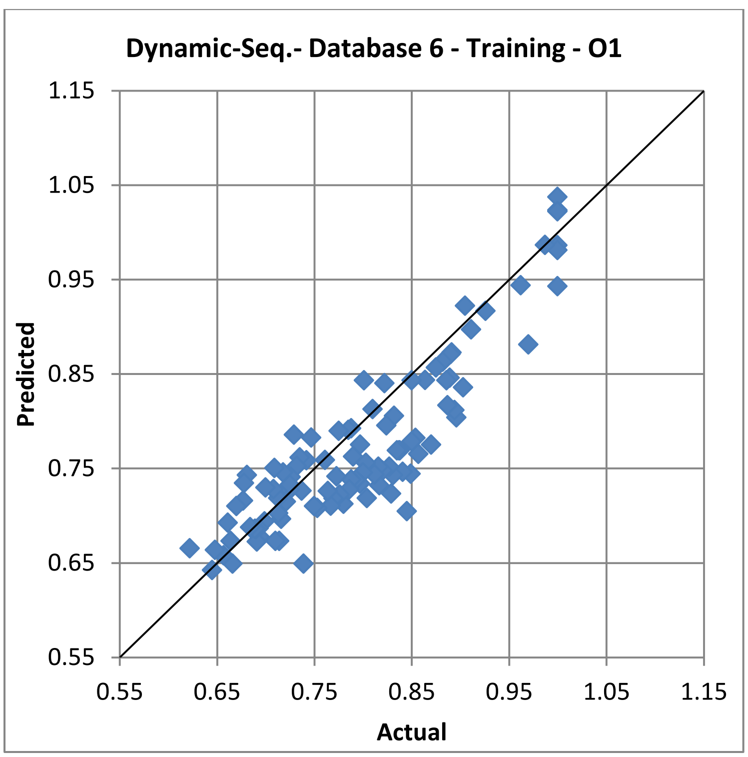

Figure 8-26 Dynamic-Sequential Network Training Accuracy of Database 6, Output 1

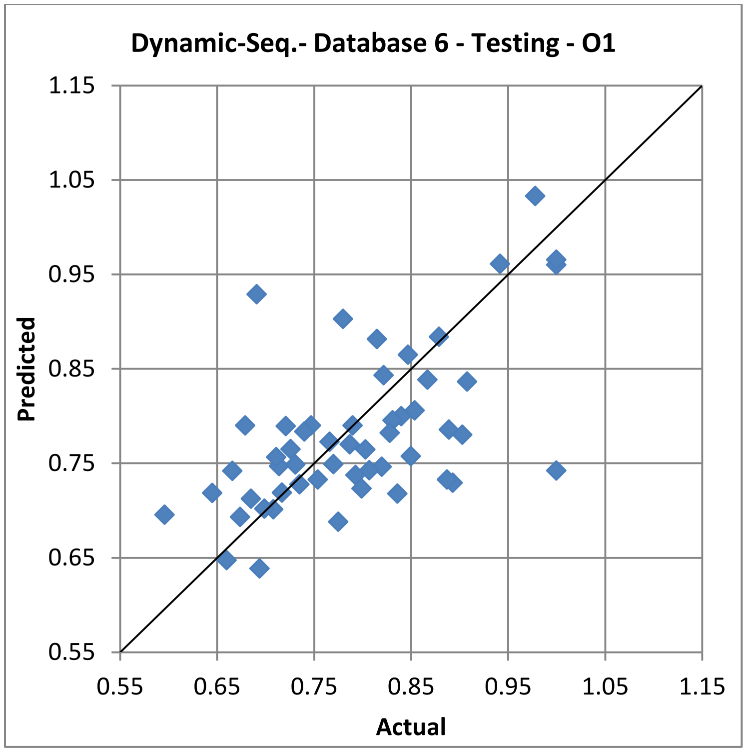

Figure 8-27 Dynamic-Sequential Network Testing Accuracy of Database 6, Output 1

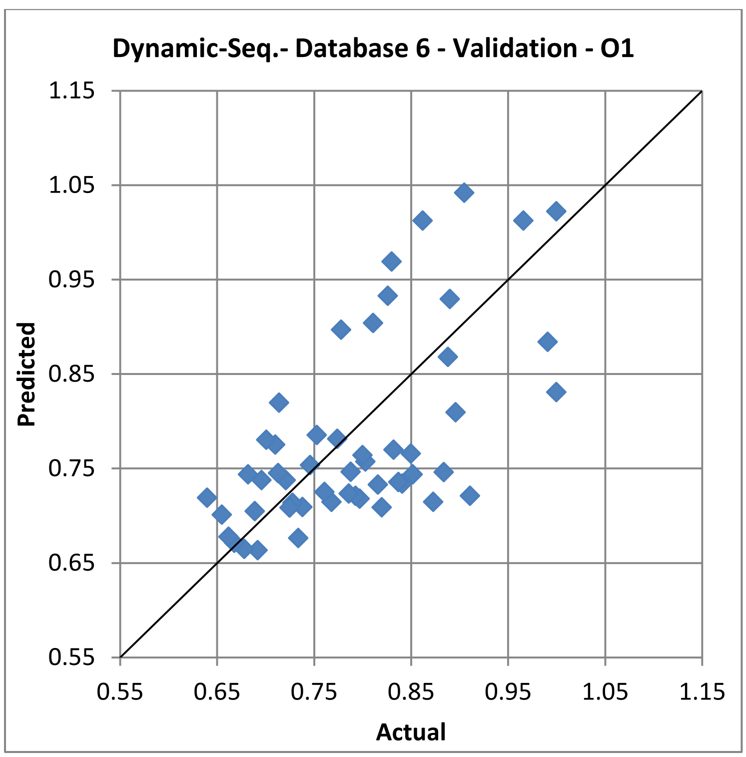

Figure 8-28 Dynamic-Sequential Network Validation Accuracy of Database 6, Output 1

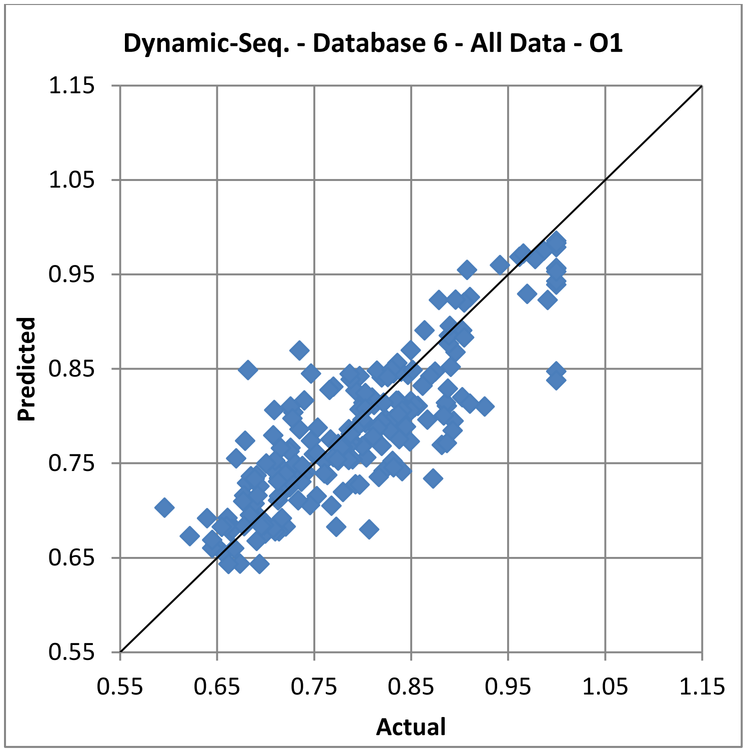

Figure 8-29 Dynamic-Sequential Network All Data Accuracy of Database 6, Output 1

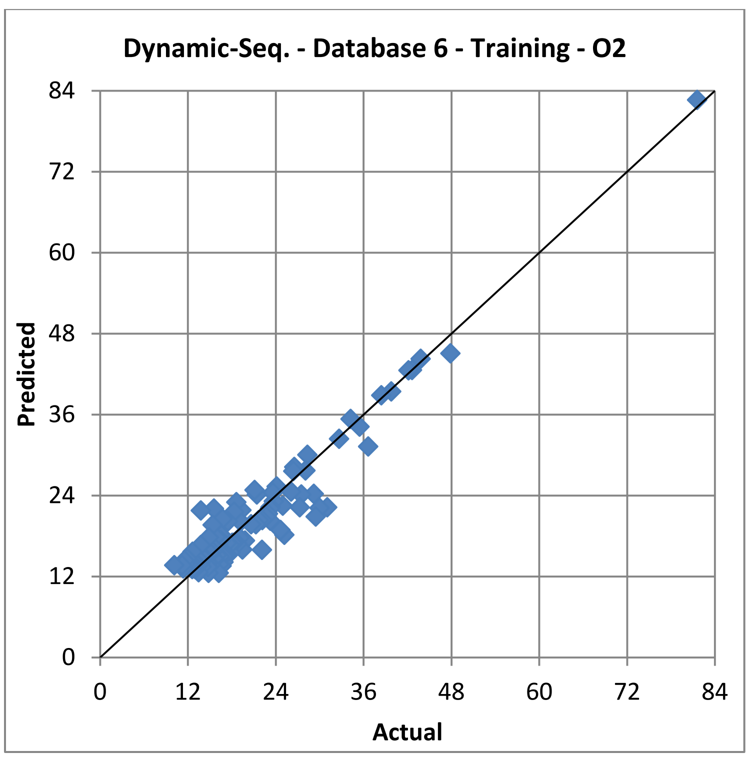

Figure 8-30 Dynamic-Sequential Network Training Accuracy of Database 6, Output 2

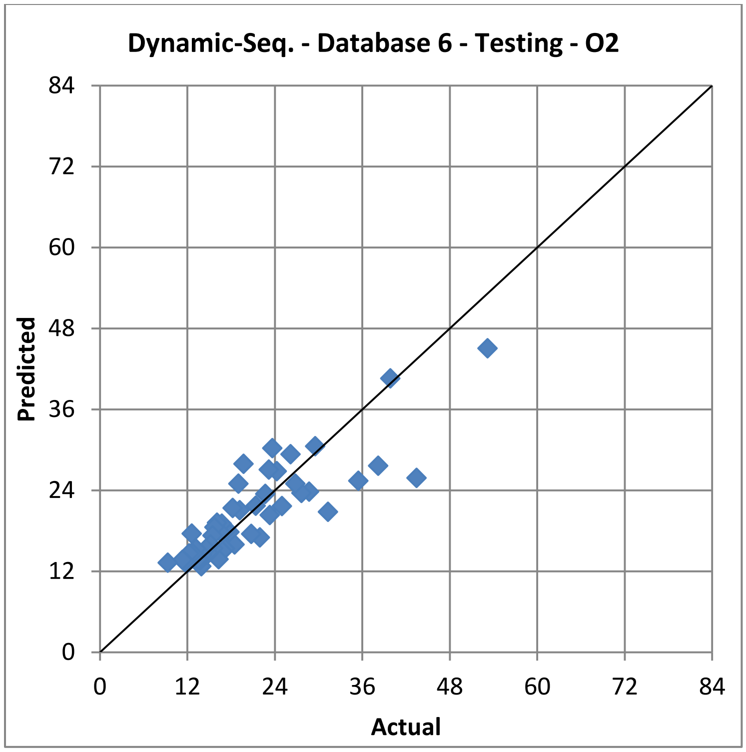

Figure 8-31 Dynamic-Sequential Network Testing Accuracy of Database 6, Output 2

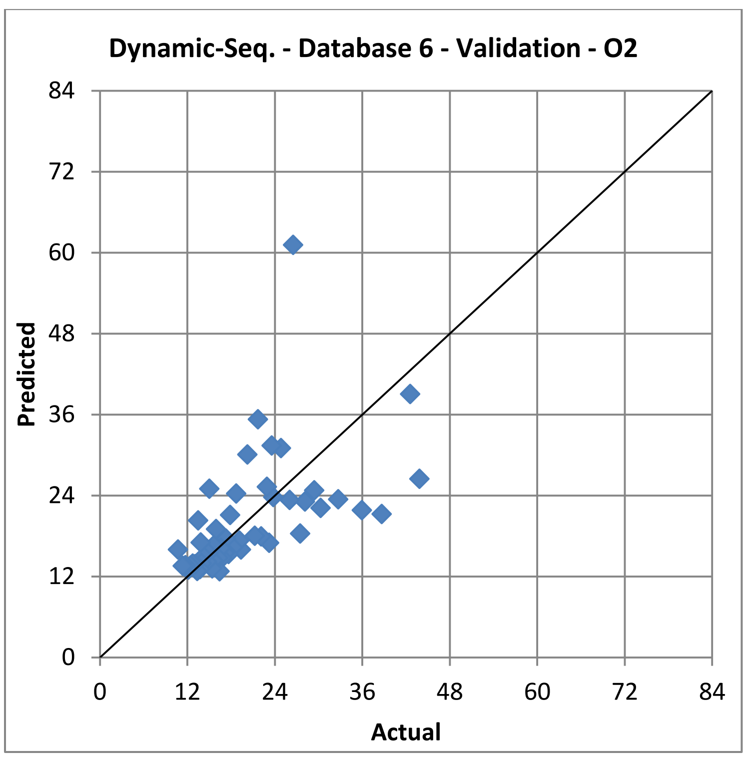

Figure 8-32 Dynamic-Sequential Network Validation Accuracy of Database 6, Output 2

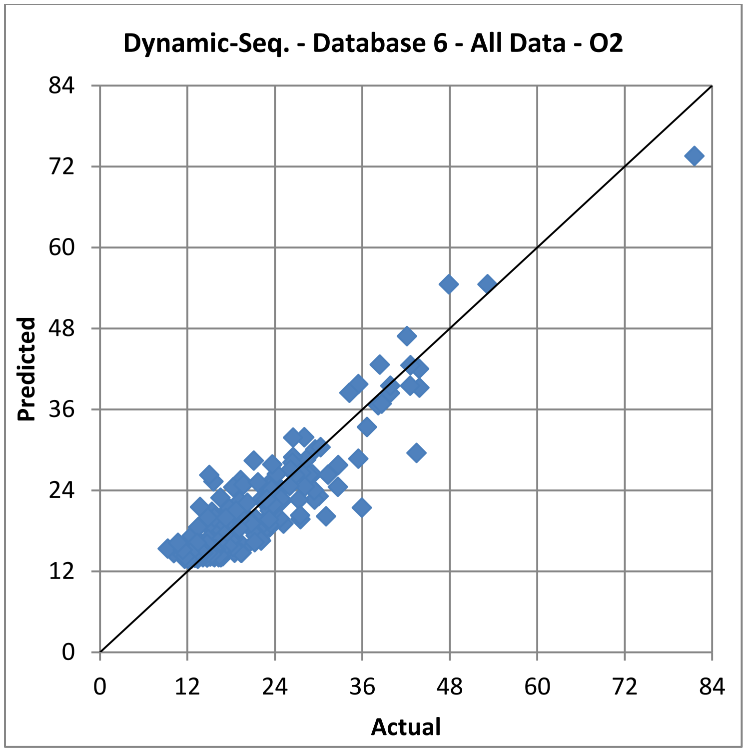

Figure 8-33 Dynamic-Sequential Network All Data Accuracy of Database 6, Output 2

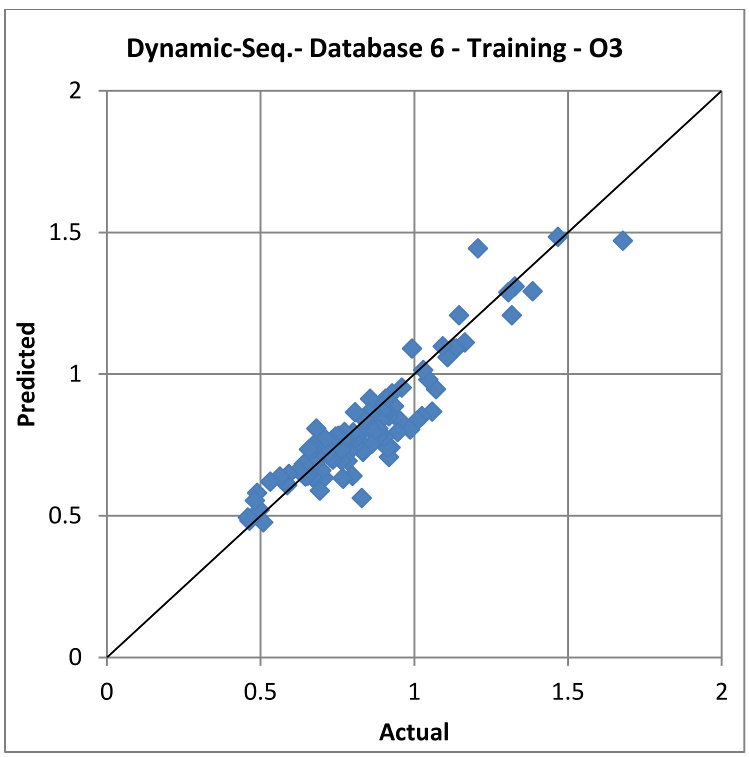

Figure 8-34 Dynamic-Sequential Network Training Accuracy of Database 6, Output 3

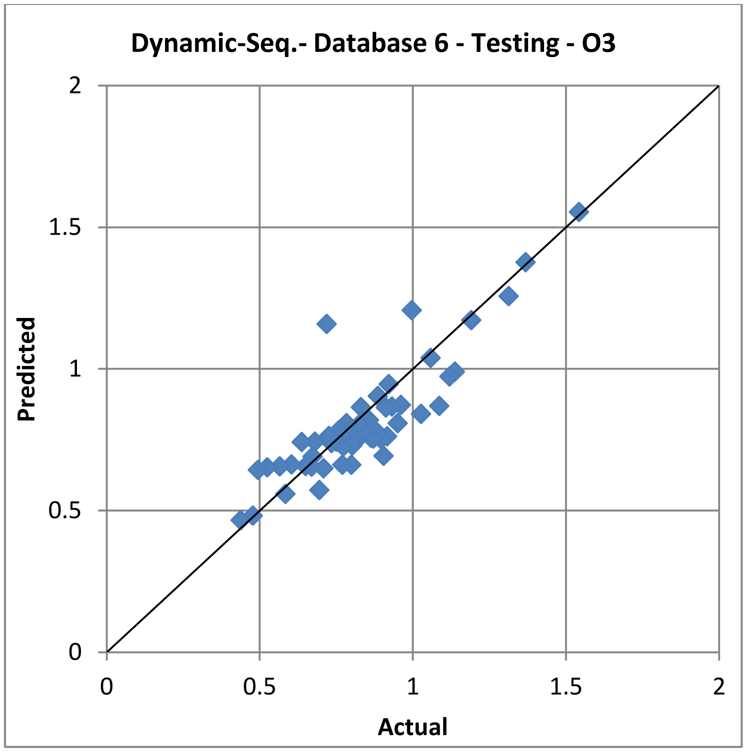

Figure 8-35 Dynamic-Sequential Network Testing Accuracy of Database 6, Output 3

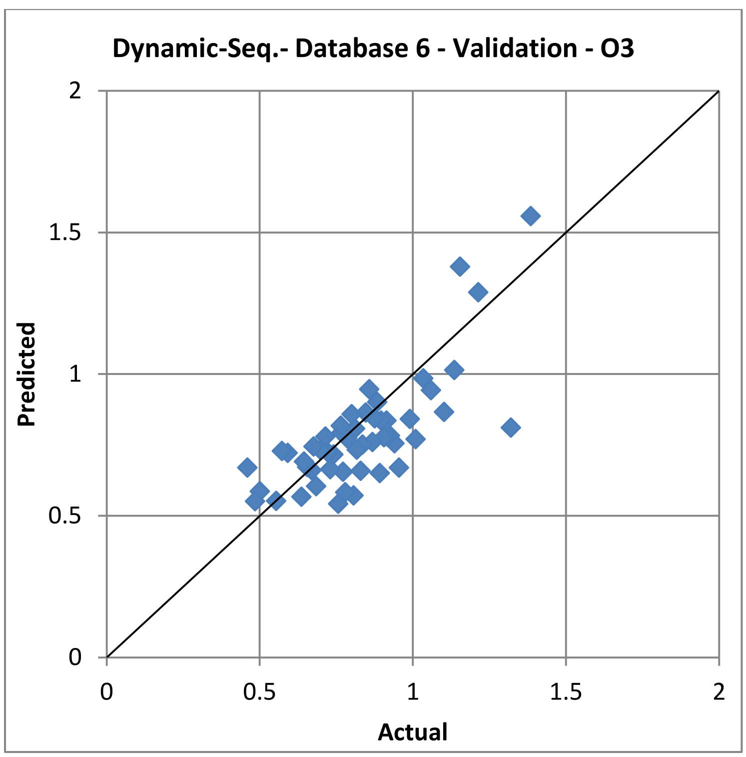

Figure 8-36 Dynamic-Sequential Network Validation Accuracy of Database 6, Output 3

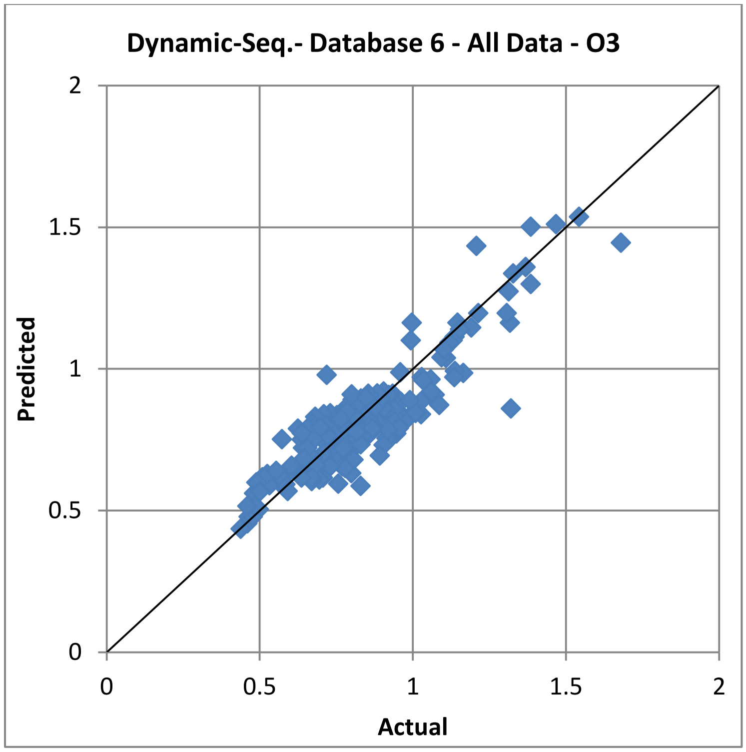

Figure 8-37 Dynamic-Sequential Network All Data Accuracy of Database 6, Output 3

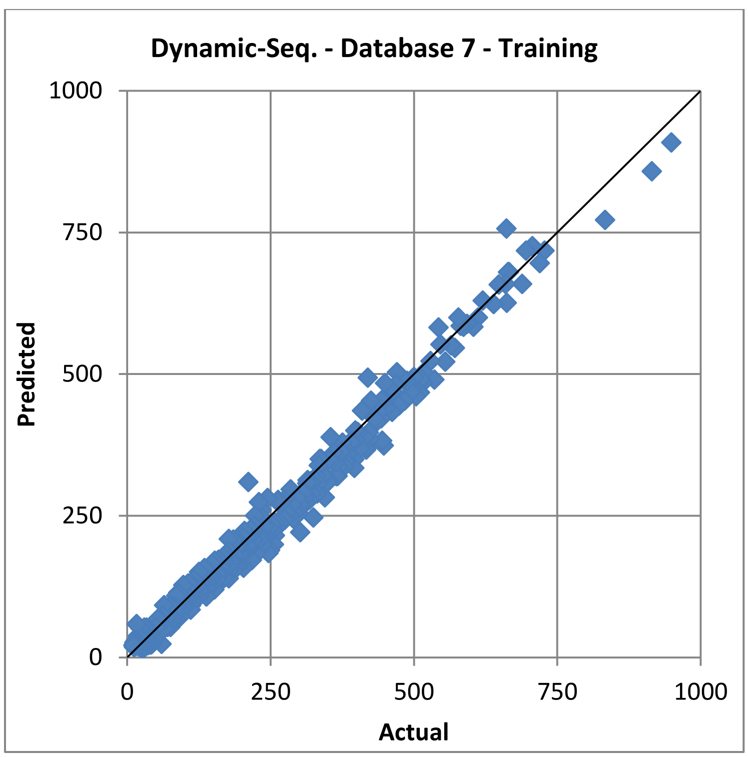

Figure 8-38 Dynamic-Sequential Network Training Accuracy of Database 7

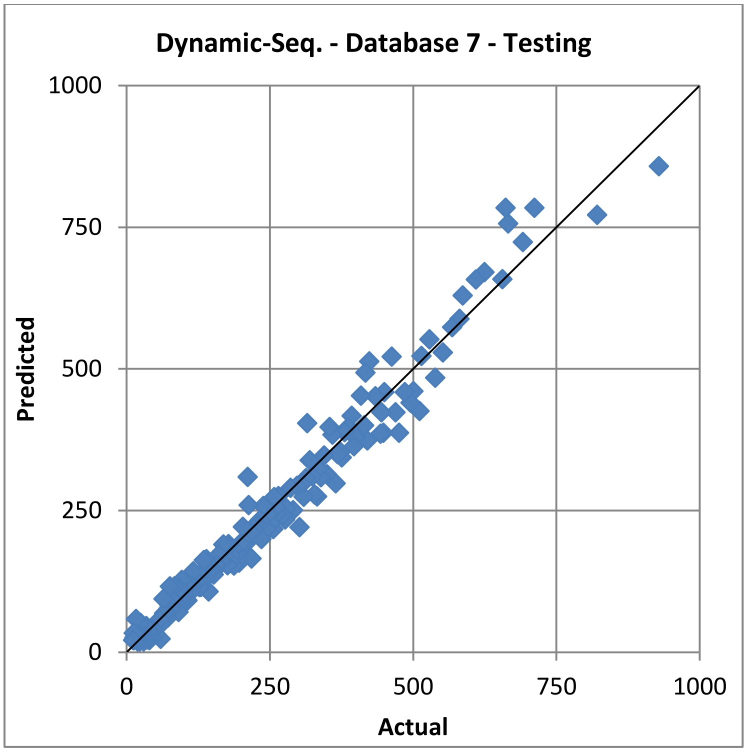

Figure 8-39 Dynamic-Sequential Network Testing Accuracy of Database 7

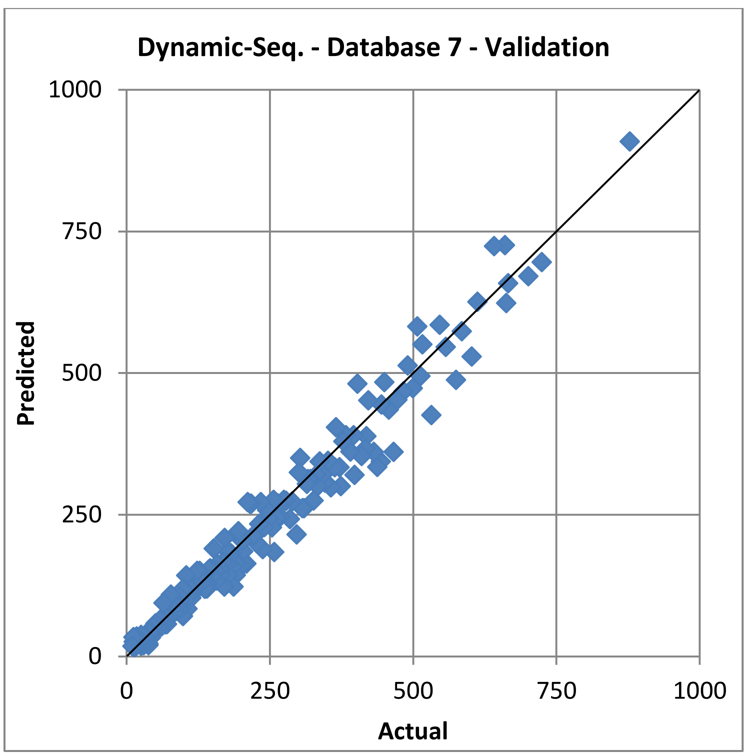

Figure 8-40 Dynamic-Sequential Network Validation Accuracy of Database 7

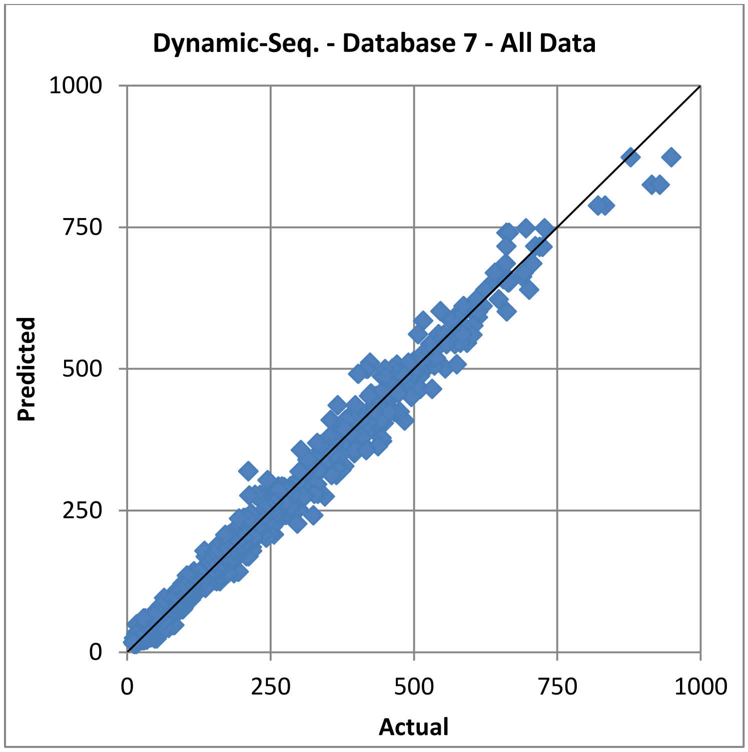

Figure 8-41 Dynamic-Sequential Network All Data Accuracy of Database 7

Table 8-1 Statistical Accuracy of Dynamic-Sequential Network Models for Database 1 to Database 5

		DYNAMIC-SEQUENTIAL NETWORK MODELS					
Accuracy Measures		Database 1	Database 2	Database 3	Database 4	Database 5	
		Output 1	Output 1	Output 1	Output 1	Output 1	Output 2
		8-(1-16)-19600-1	8-(1-3)-20000-1	13-(7-8)-100-1	7 - (1-7)-20000-1	4-(2-4)-16900-1	4-(1-4)-20000-1
TR	MARE	1.632	3.6220	14.322	20.581	0.165	1.278
	R^2	0.9998	0.6681	0.9743	0.8710	0.9967	0.9283
	MRSE	1.2130	0.3839	73.5715	81.6443	0.1901	0.6649
TS	MARE	2.666	5.9117	18.006	21.747	0.213	1.101
	R^2	0.9970	0.3861	0.9126	0.7959	0.9851	0.9283
	MRSE	6.6146	1.0838	135.4948	133.0335	0.4613	0.8500
VAL	MARE	2.923	6.6696	27.787	25.693	0.192	1.197
	R^2	0.9978	0.0443	0.6193	0.7672	0.9952	0.9420
	MRSE	5.3440	1.0599	250.2872	150.2486	0.3266	0.8663
ALL DATA	MARE	1.550	3.6601	19.180	17.653	0.189	1.148
	R^2	0.9998	0.5668	0.9317	0.8703	0.9945	0.9315
	MRSE	0.7781	0.2851	95.9043	49.4487	0.1651	0.4321
FINAL STRUCTURE		**8 - 16 - 1**	**8 - 3 - 1**	**13 - 8 - 1**	**7 - 7 -1**	**4 - 4 - 1**	**4 - 4 - 1**

Table 8-2 Statistical Accuracy of Dynamic-Sequential Network Models for Databases 6 and 7

Accuracy Measures		DYNAMIC-SEQUENTIAL NETWORK MODELS			
		Database 6			**Database 7**
		Output 1	Output 2	Output 3	Output 1
		17-(1-3)-20000-1	17 - (1-2)-20000-1	17 - (4-6)-7100-17	16-(3-6)-20000-1
TR	MARE	4.892	11.622	12.026	14.910
	R^2	0.7882	0.9173	0.6304	0.9855
	MRSE	0.0048	0.2826	0.0131	1.1866
TS	MARE	7.455	14.220	12.322	16.139
	R^2	0.3711	0.7403	0.6792	0.9728
	MRSE	0.0110	0.6284	0.0171	2.1826
VAL	MARE	8.414	21.466	13.470	15.123
	R^2	0.3944	0.3085	0.5789	0.9701
	MRSE	0.0115	1.0644	0.0192	2.2814
ALL DATA	MARE	4.768	13.873	11.711	13.217
	R^2	0.7196	0.8455	0.6419	0.9834
	MRSE	0.0035	0.2500	0.0089	0.8396
FINAL STRUCTURE		**17 - 3 - 1**	**17 - 2 - 1**	**17 - 6 - 17**	**16 - 6 - 1**

Table 8-3 Reduction of Mean Absolute Relative Error (MARE) for seven databases

Database #	**OUTPUT**	**MARE**		
		Static ANN	**Dynamic-Seq.**	**Reduction**
Database 1	Output 1	4.069	1.550	62%
Database 2	Output 1	3.9681	3.6601	8%
Database 3	Output 1	12.719	19.180	-51%
Database 4	Output 1	20.359	17.653	13%
Database 5	Output 1	0.186	0.189	-1%
	Output 2	1.125	1.148	-2%
Database 6	Output 1	5.416	4.768	12%
	Output 2	11.529	13.873	-20%
	Output 3	8.009	11.711	-46%
Database 7	Output 1	12.380	13.217	-7%

Table 8-4 Improvement of Coefficient of Determination (R^2) for seven databases

Database #	OUTPUT	R^2		
		Static ANN	Dynamic-Seq.	Improvement
Database 1	Output 1	0.9984	0.9998	0%
Database 2	Output 1	0.4554	0.5668	24%
Database 3	Output 1	0.9364	0.9317	-1%
Database 4	Output 1	0.8549	0.8703	2%
Database 5	Output 1	0.9944	0.9945	0%
	Output 2	0.9333	0.9315	0%
Database 6	Output 1	0.6612	0.7196	9%
	Output 2	0.8721	0.8455	-3%
	Output 3	0.8377	0.6419	-23%
Database 7	Output 1	0.9831	0.9834	0%

Table 8-5 Reduction of Mean Root Square Error (MRSE) for seven databases

Database #	OUTPUT	MRSE		
		Static ANN	Dynamic-Seq.	Reduction
Database 1	Output 1	2.3740	0.7781	67%
Database 2	Output 1	0.3203	0.2851	11%
Database 3	Output 1	63.7835	95.9043	-50%
Database 4	Output 1	47.9782	49.4487	-3%
Database 5	Output 1	0.1676	0.1651	2%
	Output 2	0.4255	0.4321	-2%
Database 6	Output 1	0.0038	0.0035	9%
	Output 2	0.2276	0.2500	-10%
	Output 3	0.0059	0.0089	-51%
Database 7	Output 1	0.8466	0.8396	1%

Table 8-6 Statistical Performance of the Initial Estimate Configurations, Database 1

Accuracy	Initial Estimate		
	Static ANN	Value "0"	Average
MARE	1.9605	1.9651	1.9534
R^2	0.9995	0.9995	0.9995
MRSE	2.7635	2.5962	2.6430

Table 8-7 Statistical Performance of the Initial Estimate Configurations, Database 2

Accuracy	Initial Estimate		
	Static ANN	Value "0"	Average
MARE	4.8722	4.8722	4.8722
R^2	0.0901	0.0901	0.0901
MRSE	0.8529	0.8529	0.8529

Table 8-8 Statistical Performance of the Initial Estimate Configurations, Database 3

Accuracy	Initial Estimate		
	Static ANN	Value "0"	Average
MARE	31.3910	31.5006	30.8819
R^2	0.3552	0.3515	0.3408
MRSE	347.9069	350.2783	350.2036

Table 8-9 Statistical Performance of the Initial Estimate Configurations, Database 4

Accuracy	Initial Estimate		
	Static ANN	Value "0"	Average
MARE	25.1469	65.1780	29.2794
R^2	0.7586	0.2756	0.6586
MRSE	154.4276	382.1132	156.8112

Table 8-10 Statistical Performance of the Initial Estimate Configurations, Database 5, Output 1

Accuracy	Initial Estimate		
	Static ANN	**Value "0"**	**Average**
MARE	0.1907	0.1907	0.1907
R^2	0.9951	0.9951	0.9951
MRSE	0.3308	0.3308	0.3308

Table 8-11 Statistical Performance of the Initial Estimate Configurations, Database 5, Output 2

Accuracy	Initial Estimate		
	Static ANN	**Value "0"**	**Average**
MARE	1.2140	1.2140	1.2140
R^2	0.9384	0.9384	0.9384
MRSE	0.9089	0.9089	0.9089

Table 8-12 Statistical Performance of the Initial Estimate Configurations, Database 6, Output 1

Accuracy	Initial Estimate		
	Static ANN	**Value "0"**	**Average**
MARE	8.0844	8.0839	8.0844
R^2	0.3659	0.3658	0.3658
MRSE	0.0109	0.0109	0.0109

Table 8-13 Statistical Performance of the Initial Estimate Configurations, Database 6, Output 2

Accuracy	Initial Estimate		
	Static ANN	**Value "0"**	**Average**
MARE	18.4232	18.3283	18.5283
R^2	0.3343	0.3454	0.3256
MRSE	0.9981	0.9831	1.0139

Table 8-14 Statistical Performance of the Initial Estimate Configurations, Database 6, Output 3

Accuracy	Initial Estimate		
	Static ANN	**Value "0"**	**Average**
MARE	13.1380	13.1518	13.1553
R^2	0.5534	0.5514	0.5526
MRSE	0.0208	0.0208	0.0208

Table 8-15 Statistical Performance of the Initial Estimate Configurations, Database 7

Accuracy	Initial Estimate		
	Static ANN	**Value "0"**	**Average**
MARE	11.0321	11.6874	11.3988
R^2	0.9782	0.9659	0.9751
MRSE	1.9365	2.3738	2.0871

CHAPTER 9

9. QUERY METHOD

Advanced systems to collect data and analyze databases became easier with the availability of more powerful computers. Nowadays, the digital revolution made data mining systems and their analysis methods very common. As explored in the previous chapters, computational systems such as Artificial Neural Networks (ANNs) are the preferred advanced systems to extract ultimate information from the databases. Especially in engineering applications, it is very beneficial to utilize these types of advanced systems. However, collecting and analyzing databases can be expensive and time consuming. In addition, very often the values of one or more explanatory variables may be missing. These are incomplete datasets: datasets with missing values. Most data mining algorithms cannot work directly with incomplete datasets. All the ANN approaches explored in this study use backpropagation algorithm, which does not work with incomplete datasets as well. In other words, developed ANN models can generate outputs, if a complete dataset is provided to the models. For this reason, it is necessary to utilize a tool to replace missing inputs of the datasets because missing data is a common occurrence and may have a significant effect on the results that can be drawn from the database.

One of the widely used techniques to deal with a dataset with missing input is the deletion technique, which is simply removing the incomplete dataset from the database. On the other hand, the small database size limits the applicability of deletion technique that reduces the database size even further. This may lead to an inconclusive analysis, because the sample of complete cases may be too small to obtain statistically significant trends. The most common technique for filling in a missing value is mean substitution; replacing missing values with the mean of the variable. The major advantage of the method is its simplicity. However, this method yields biased estimates of variances and covariances (Gheyas et al., 2009). Some of the popular missing data imputation algorithms are EM (Expectation Maximization), MI (Multiple Imputation), MCMC (Markov Chain Monte Carlo), and hot deck MI (Gheyas and Smith, 2009).

Another possible solution for missing input parameters is to develop individual prediction models for each parameter that are optimized on the target input parameter, but this requires lots of time and effort to accomplish. There is a wide variety of methods for handling missing data, which vary a great deal in their mathematical complexity or in time and effort. In addition, the applicability of these algorithms requires advanced knowledge in programming.

Therefore, the simple solution to resolve this issue is introduced in this study: the Query method, a new approach to replace a partially missing dataset. By using the entire database, the closest neighborhood datasets are determined based on Euclidean distances for a newly introduced incomplete dataset. The Query Method tends to maximize the likelihood by finding similar datasets within the closest neighborhood. This method assumes that every data in the database is considered as the center of their neighborhood.

For the case of an incomplete dataset, the closest neighborhood is determined based on Euclidean distances between the normalized incomplete dataset and normalized datasets in the database. It is essential to use normalized values because numerical magnitude of the variables can dominate each other. Normalization has applied based on the minimum and maximum of the variables. In this case, all the variables are normalized between 0 and 1. The normalization process can be expressed as:

$$p_{norm} = \frac{p - p_{min}}{p_{max} - p_{min}} \qquad \text{Eqn. 9.1}$$

Where;

p = Actual value of the parameter

p_{min} =Minimum value of the variable

p_{max} =Maximum value of the variable

In order to implement the Euclidean distance, the Query method is modified by including a missing variable coefficient. With this modification, magnitude of the missing variable is excluded from the calculation. The coefficient value is assigned as “1” if the variable is not missing. If the variable is missing, then the coefficient is assigned a “0” value. New form of the Euclidean distance from p to p_a is defined as:

$$d(p, p_a) = \sqrt{(p_1 - p_{a1})^2 * C_1 + (p_2 - p_{a2})^2 * C_2 + \ldots\ldots\ldots\ldots. + (p_n - p_{an})^2 * C_n} \qquad \text{Eqn. 9.2}$$

Where ;

p = the vector of the incomplete dataset, p= (p_1, p_2, p_3, ……, p_n),

p_a = vector “a” of the complete dataset within the database, p_a=(p_{a1}, p_{a2}, p_{a3}, ……, p_{an}),

C = the vector of variable coefficients, C= (C_1, C_2, $C_{3,\ldots\ldots\ldots}$, C_n).

For example, in Equation 9.2, if the missing variable of the new dataset is p_2 in Equation 9.2, the value of the missing variable coefficient, C_2 is assigned a value of “0” while all the other coefficients are assigned as a value of “1”. N dimensional space is reduced to n-1 dimensional space accordingly. The more missing parameters the dataset has, the less dimensional space the equation considers and the chance to determine the right neighborhood of the corresponding dataset decreases.

By using Equation 9.2, all Euclidean distances between incomplete dataset and every complete dataset in the database are calculated and sorted based on the least distance. The missing parameter is replaced with the average value of the 3 closest datasets. A preliminary study has shown that using the average of 3 datasets with the closest Euclidean distance presented the best possible prediction outcomes. For this reason, the optimum value, as the representative of the neighborhood, is determined based on the closest three datasets. Even though the Query method is able to replace any missing input variable, it can also be used to generate output(s) as well by simply considering the output(s) as a missing variable.

In order to explore the Query method, the seven databases described in Chapter 4 were utilized to verify the Query method predictions. The statistical accuracy measures and graphical comparison plots are presented in the following sections for each database. The Query method application was developed for all seven databases, but it is explored for only the output(s). Accordingly, the results presented in the following sections are only related to the output. In order to develop Query method applications for seven databases, datasets used for training and testing in the previous chapters were utilized to develop the applications, and the validation datasets were similarly used for validation. However, once the statistical accuracy measures and graphical comparison plots were obtained, then the Query method application was re-developed by considering the entire database including validation datasets. This procedure was done to verify the method's performance on the database itself. If the application can find the right neighborhood when the datasets themselves are imputed, then the reliability of the method can be verified properly. As it was done in previous chapters, the accuracy of the method was interpreted based on the statistical accuracy measures such as MARE, R^2, and MRSE.

9.1 Query Method Application Development of Database 1

Two hundred and twenty nine datasets with 8 variables (7 inputs and 1 output) were considered to develop the Query method application. Eight variables of the complete datasets and validation datasets were normalized based on their corresponding minimum and maximum ranges. The actual output values of the validation datasets were only considered to calculate the statistical accuracy measures. Once the variables were normalized, the new form of the Euclidean distance given in Equation 9.2 was calculated between each validation dataset and complete datasets. After sorting the datasets based on the Euclidean distances, the closest three datasets were considered to replace the output. In order to evaluate the replaced values by the Query Method, a model using linear regression analysis approach was developed to compare the results. The first step was to develop the Query method application with 229 datasets and validate it on 71 datasets. The graphical accuracy plot of the validation datasets is shown in Figure 9-1. The statistical accuracy measures of the Query method and regression model are depicted in Table 9-1. The second step was to develop the Query method application with all datasets (i.e. 300 datasets) and validate it by using the same datasets used to develop the application. The graphical accuracy plots for all datasets are given in Figure 9-2. All the corresponding statistical measures of the Query method application for the validation are presented in Table 9-2 along with the statistical accuracy measures of the linear regression model. As can be seen from the plots and tables, Query method has reasonably good results even though the regression model has outperformed the Query method for validation stage. When the entire dataset was included, the accuracy measures were quite improved for Query method while the ones by regression model improved slightly. Consequently, the Query method application for all data performed better than the regression model. It can be inferred that including all datasets improve the accuracy of the method because the Query method relies completely on the available complete datasets.

9.2 Query Method Application Development of Database 2

A database consisting of 100 datasets was used to develop a desired Query method application for Database 2. As noted previously, the datasets are divided into two sets; 78 and 22 datasets for developing the application and validation. An input vector consisting of 7 parameters and an output vector consisting of 1 parameter were considered to develop the Query method application for database 2. A graphical comparison of validation stage between the predicted and the actual is depicted in Figure 9-3. The Query method application for validation stage yielded a mean root square error, $MRSE_{val}$ of 0.7669, mean absolute relative error, $MARE_{val}$ of 4.5964, and coefficient of determination, R_{val}^2 of 0.1731. To evaluate the results by the Query method, a model using linear regression analysis approach was developed. All 100 datasets were also used to develop the Query method application and the regression model and the methods were validated on the same datasets. The graphical comparison of all data stage for the Query method is shown in Figure 9-4 and statistical accuracy measures for all data stage are $MRSE_{all}$ of 0.2919, $MARE_{all}$ of 3.5651%, and R_{all}^2 of 0.5508. The corresponding statistical measures of all the Query method applications and regression models were given in Table 9-1 and Table 9-2. As seen from the statistical measures, the query method successfully predicted the output even though the regression model has outperformed the Query method for the validation stage. It is clear from the results that all data MRSE by the Query method is lower than the all data MRSE by regression model. MARE and R^2 values by the Query method for all data stage are better than those by regression model. As a result, Query method application effectively replaced the output when all datasets were involved in the application development.

9.3 Query Method Application Development of Database 3

Another database from an experimental study was considered to develop the Query method application. A total of 126 datasets were used to develop Query method application for database 3. Ninety five and 31 of total datasets were, respectively, considered for Query method application and validation. Similarly, the same datasets were used to obtain the regression model. An attempt to obtain the application and model for database 3 was initiated with 12 inputs and 1 output. The statistical measures for validation stage are shown in Table 9-1 and the graphical plots of the Query method predictions are depicted in Figure 9-5. As can be perceived from the table, Query method has a MRSE value of 317.5696 while Regression model has 176.1023, which is about 44.54% less. In this case, the regression model has better statistical measures for the validation stage. When all datasets were included to develop the Query method application and the regression model, the statistical measures were improved. However, the improvement by Query method is significantly better than that by the regression model. For example, the MRSE value by the Query method has reduced to a value of 90.4443 while the MRSE value by the regression model has come down to a value of 109.3843. In other words, the Query method application developed with all data has reduced the MRSE value about 71.52% while the regression model developed with all data has reduced the MRSE value about 37.89%. The graphical accuracy measures by the Query method for all data stage can be seen in Figure 9-6. All the corresponding accuracy measures for the Query method application and the regression model are presented in Table 9-2.

9.4 Query Method Application Development of Database 4

To properly develop a Query method application for database 4, a total of 265 datasets; 199 and 66 datasets were, respectively, used for Query method application development and validation. The input vector consisted of 6 parameters and the output vector consisted of 1 parameter. By using 199 datasets, the Query method application was developed and tested on 66 datasets. Similarly, the same datasets were used to develop a regression model and tested. The query method application for validation stage yielded a mean root square error, $MRSE_{val}$ of 124.331, mean absolute relative error, $MARE_{tr}$ of 32.860%, and coefficient of determination, R_{tr}^2 of 0.764, while the regression model has a MRSE value of 166.33330, MARE value of 55.0471, and R^2 of 0.5933. As can be interpreted from the results, the Query method has a lower MRSE and MARE, and higher R^2. Similarly, when all datasets were used, the statistical accuracy measures for the Query method are still better than the regression model. This indicates that the datasets used to develop the Query method application for validation stage had an adequate number of datasets representing the neighborhoods in database 4. Graphical comparisons of validation and all data stages are, respectively, shown in Figure 9-7 and Figure 9-8. The corresponding statistical measures for validation and all data stages are, respectively, given in Table 9-1 and Table 9-2. A good agreement between actual and predicted values by the Query method is evident.

9.5 Query Method Application Development of Database 5

Database 5 has been built by considering 325 datasets. Two hundred and forty four and 81 datasets were, respectively, used for the Query method application and for the validation. As stated before, database 5 has two outputs. Even though other approaches utilized this database twice to arrive at the optimal structure, the Query method utilized the two outputs at once. In order to evaluate the accuracy of the method, a regression model was also developed. However, since the regression model cannot be developed with outputs, two different regression models were used to generate two outputs. The results by the Query method and the regression models for the validation and all data stages are promising for both outputs. However, the accuracy measures for output 1 by the regression model have lower MRSE and

MARE values. A similar situation can be told for output 2 in the validation stage, but when it comes to the all data stage the accuracy measures for only output 2 by the Query method are better than the regression model. Graphical accuracy measures for outputs 1 and 2 in validation stage are given in Figure 9-9 and Figure 9-10. Similarly, graphical accuracy plots for all data stages by the Query method are depicted in Figure 9-11 and Figure 9-12. All the statistical accuracy measures by the Query method and the regression models are depicted in Table 9-1 and Table 9-2. As a result, both of the models can be used efficiently. Nevertheless, the effort to develop two regression models has to be noted. In addition, the models by regression were optimized on the two outputs for this case but when the problem is to place a missing variable, multiple attempts to develop individual models for each variable are necessary. In this case, regression analysis will be a time-consuming method while the Query method is very easy to implement for multiple outputs and missing variables. The Query method can be considered as multifunctional approach compared to other methods even though the prediction accuracy of the method may not be as good.

9.6 Query Method Application Development of Database 6

Another database with highly non-linear behavior and multiple outputs was used to develop a corresponding Query Method application. Two hundred and ten datasets were collected to build database 6 and divided into two sub-databases: 158, and 52 to be used, respectively, for the Query method application and validation. As stated in Chapter 4, database 6 has three outputs that need to be utilized multiple times to develop ANN models or other advanced methods. However, the Query method utilizes all three outputs and inputs together to replace a missing value, or in other words to generate predictions. Databases may have missing variables even though their output(s) are present. In this case, the conventional methods may consider the present output if they are optimized on that specific variable, but because of the effort to develop various models by considering probabilistic cases, it may not be available. The Query method uses every variable provided that belongs to the dataset. For database 6, 16 inputs and 3 outputs were used to develop the Query method application. In order to compare the statistical accuracy measures, a regression analysis was performed three times to obtain three models because the number of output is limited to one in the regression analysis approach. The statistical accuracy measures by the Query method for output 1 in validation and all data stages are given in Figure 9-13 and Figure 9-14. The MRSE value by the Query method for the validation stage, 0.0110, has decreased to 0.0030 for the all data stage, which corresponds to a 72.7% reduction in error while MARE value decreased about 44.4%. For the regression model, the MRSE value for the all data stage improves to a value of 0.0047 from 0.0100, which translates into about 53% reduction in error, while MARE value decreased about 5.8%. Even though R^2 value by the Query method seems to increase as well from 0.349 for validation stage to a value of 0.72 for all data stage, the main criterion, which is MRSE, has a reasonable reduction in error. All the statistical measures for the validation and all data stages by the Query method and regression model can be found in Table 9-1 and Table 9-2. The graphical plots for the output 2 in validation and all data stages by the Query method are shown in Figure 9-15 and Figure 9-16. A good agreement between the actual and the predicted values can be clearly seen in the plots. Similarly, Figure 9-17 and Figure 9-18, which are the plots of output 3 in validation and all data stages, indicate reasonably good correlation between the actual and predicted values. The corresponding accuracy measures of all outputs by the Query method as well as the regression model are depicted in Table 9-1 and Table 9-2.

As can be noted from the tables and all the graphical plots, the Query method application for database 6 was successfully developed even though some of the statistical accuracy measures in validation stage were lower than those by the regression model. Overall comparison of these three outputs has showed that the least MRSE and MARE values were obtained for the output 1 even though R^2 value for output 1 was the least among the three outputs. The Query method for all data stage has outperformed the regression model.

9.7 Query Method Application Development of Database 7

The last database utilized in this chapter to develop Query method application is Database 7, which consists of 792 datasets divided into 594 and 198 datasets for the application and its validation. By considering 15 inputs and 1 output, the desired application was initiated with the validation stage. Five hundred and ninety four datasets were used to develop both the Query method and the regression model and then validated on 198 datasets. The statistical accuracy measures of the validation stage are shown in Table 9-1. The graphical comparison plots of the Query method for the validation stage can be seen in Figure 9-19. The MRSE value by the regression model is less than the one by the Query method. However, the MARE by the Query method is lower than the regression model. Both the Query method and the regression model have improved the accuracy when the application and the model were developed with the entire database. As can be seen from Table 9-2, the Query method has reasonably good statistics such as $MRSE_{val}$= 1.4197, $MARE_{val}$= 10.6884%, and R^2_{val}= 0.9530. Even though the regression model improved the accuracy, its results are not as good as the Query method. Combining all datasets and developing the application improved the model statistics markedly. In this case, MRSE value of 3.6459 for validation was reduced to a value of 1.4197, which can be translated into an 61.1% reduction. The MARE value of 14.5443 for validation stage has gone down to a value of 10.6884, which is about 26.5% reduction, and the R^2 value of 0.9209 for validation increased to a value of 0.9530, that corresponds to a 3.5% increase. In the same order, the regression analysis has shown 46.8% and 3.8% improvements for MRSE and MARE, and -2.3% reduction for the R^2. The plots for all data stage by the Query method is depicted in Figure 9-20 indicate the good correlation between actual and predicted results, even though there seem to be few outliers at the higher end of the plots. All the statistical measures can be evaluated in Table 9-1 and Table 9-2. Also, Table 9-3 presents the changes of the statistical measures between validation stage and all data stage. Negative values indicate that the values deteriorated. As a result, the Query method for database 7 was effectively developed and the statistical accuracy measures are adequate.

9.8 Query Method Utilization

In order to utilize a database to develop a Query method application, all calculations and procedures explained previously must be followed. This may take a lot of time and effort depending on the size of the database and its variables. Even though the Query method can simply be applied to any database, it might be time-consuming to do all the calculations manually. For this reason, to be able to process databases faster and without any calculation errors, an Excel-based application was developed using Visual Basic programing language. As can be seen from Figure 9-21, the datasets were placed into the cells below the blue line, indicated with the arrow, by considering the first column as the numbering. By clicking the “Run Application” button, the program normalizes all the values and then prepares the foundation for the application to calculate the Euclidean distance and sort the datasets. If there is a validation dataset, the Validation cell shown in Figure 9-21 needs to be entered with the desired number. Otherwise the application assumes that there is no validation dataset and uses all datasets for the application. Once it finishes the development, then it generates a user interface which is located in another worksheet called “Program” in the same excel file. The user interface has the cells to enter the incomplete dataset. There is also another button that accomplishes the sorting of the database. All the cells with the calculations placed on another page but every cell is linked to that page. If an incomplete dataset is imputed, then excel functions check to find the missing variable, assigns the coefficients on the other page and all the Euclidean distances between new dataset and all complete datasets in database are simultaneously calculated. However, in order to find the closest datasets, the datasets with the Euclidean distances have to be sorted. To operate this, excel function were used to assign a button on the user interface page. The user interface for database 1 can be seen in Figure 9-22. If there are validation datasets initially imputed on the database page, then statistical analysis can be started by clicking the “Validation” button on the first page, where the datasets were initially placed. This command will copy the validation datasets onto another sheet to start the statistical analysis process. In this process, each variable of the database assumed to be missing and the value is replaced by the Query method. Only one variable at a time is assumed to be missing if there are not multiple outputs involved in the database. The replaced values are imputed into the cells on the same page and the error is calculated by considering the actual value. Once all the datasets are considered, then the statistical accuracy measures, such as MRSE, MARE, and R^2, for each variable including output(s) are calculated in a table and provided to the user. Consequently, placing the desired database and hitting a button to develop the Query method application has made this application very easy to use and apply to any database.

9.9 Concluding Remarks

In this chapter, a new approach to replace partially missing dataset in databases is introduced and utilized on civil engineering databases. This approach is based on determining the Euclidean distance between two vectors whose components are located in the same space and one of which has a missing components. Basically, the Euclidean distance between one vector with a missing component and another vector with complete components is calculated by excluding the missing component in the calculation. In this case, missing (unknown) variable and its matching variable in the same dimension is omitted to obtain a physical distance. As can be seen from the Equation 9.2, the exclusion of the missing variable components is controlled by the missing variable coefficient. So to verify this methodology, seven databases were utilized to develop corresponding Query method applications. To accomplish this, the databases were divided into two sub-datasets, one of which is to develop the Query method application, and the other is to validate the application. Once the statistical accuracy measures were obtained, then all datasets were combined to obtain a Query method application to expand the number of neighborhoods. Moreover, the combined datasets to develop the application were re-used again to validate the application. Similarly, the same datasets used for Query method application development and validation were also utilized to develop the linear regression models to compare the prediction performances.

As can be seen from Figure 9-1 to Figure 9-20 and the statistical accuracy measures presented in Table 9-1 and Table 9-2, all of the Query method applications developed for seven databases performed well. A good trend between predicted and actual values is apparent in the plots. Even though the Query method application for the validation stage did not perform as well as the regression model for most cases, its results can still be considered as reasonable. For most cases, the Query method competed with the regression model results. Once the validation stage was completed, all datasets were used to re-develop the Query method and the regression model by considering all datasets. The performance of the application and the model was, this time, much better. Moreover, the Query method has improved most of its statistical measures dramatically. Table 9-3 presents the performances of the Query method as well as the regression model in terms of percentages. According to Table 9-3, the Query method has

outperformed the regression model in term of the three statistical accuracy measures. Database 5, output 1 is the only case that the Query method could not improve the measures.
In order to develop application that can easily utilize Query method, an Excel-based application was developed as it was explained before. By using Visual basic programing language and Excel functions together, a user-friendly tool in a widely used Excel environment was developed. In order to develop a Query method application for the desired database, this Excel-based application is imputed with the desired database and with one simple click the Query method application is developed. Additionally, the validation of the application can also be performed with the buttons placed on the user-interface of the application. The screen-shots from the Excel-based application are illustrated in Figure 9-21 and Figure 9-22.

Consequently, the Query method was introduced and explored on the seven databases in this chapter. The statistical accuracy measures are very promising. This method can be very handy when there are multiple outputs since it does not require multiple model development. The applicability of this method is not limited to civil engineering databases. It can be used for any database with adequate and reliable components. Databases with lots of datasets are possibly the best candidates for this method because more datasets indicate more neighborhoods, which could mean more accurate data replacements. This method can efficiently be used to replace the missing variables and/or predict outputs. The developed Excel-based application is easy to use and can be applied to any database by anyone without the need for much expert knowledge.

9.10 Figures and Tables

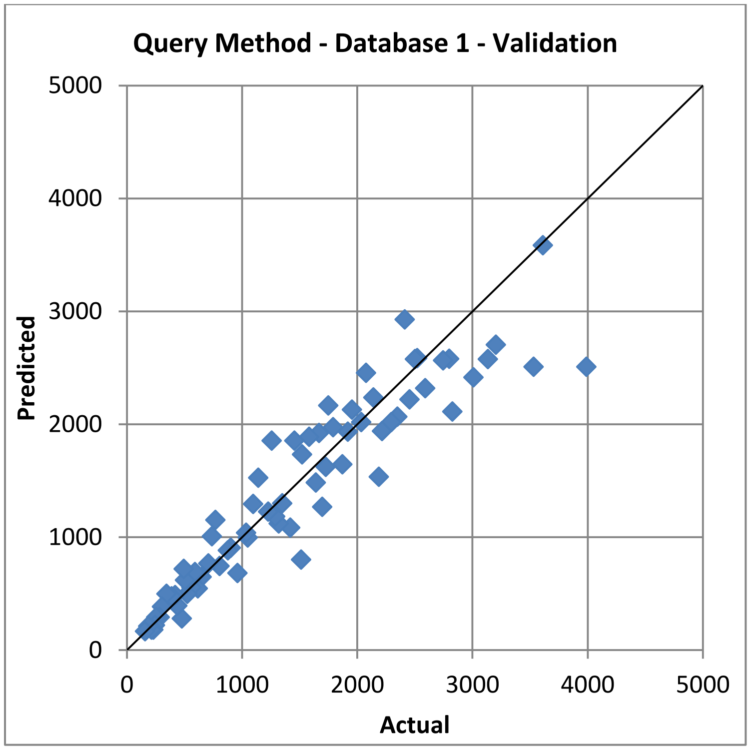

Figure 9-1 Query Method Validation Accuracy of Database 1

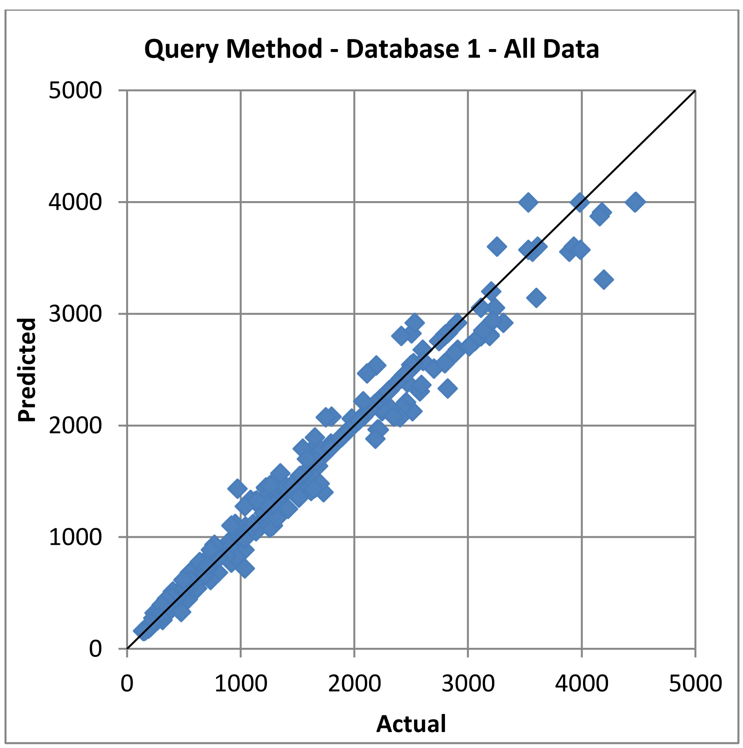

Figure 9-2 Query Method All Data Accuracy of Database 1

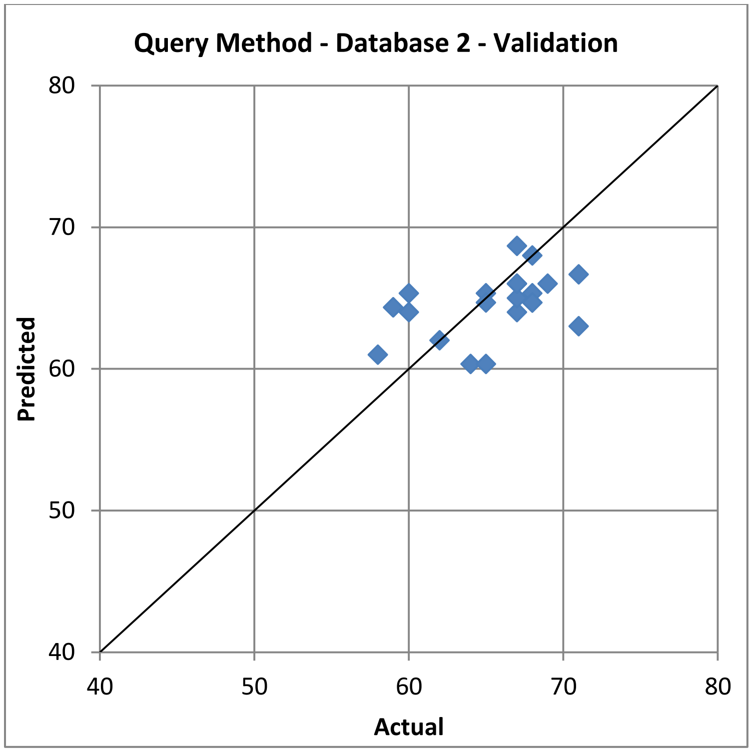

Figure 9-3 Query Method Validation Accuracy of Database 2

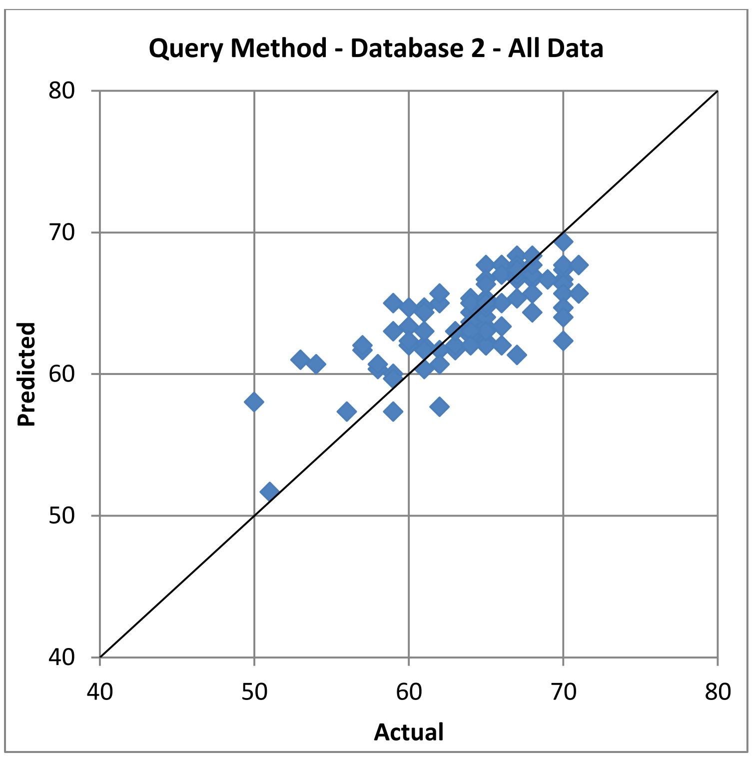

Figure 9-4 Query Method All Data Accuracy of Database 2

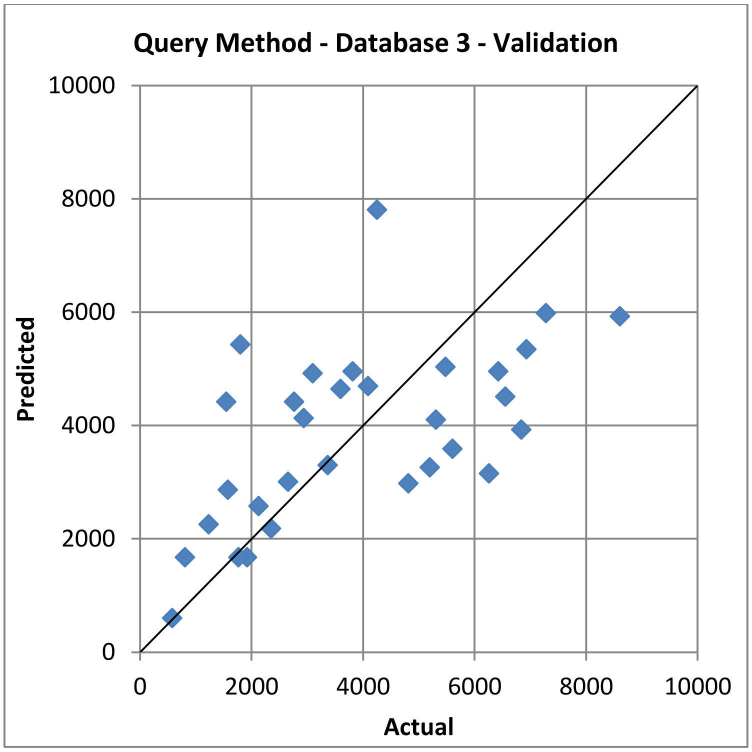

Figure 9-5 Query Method Validation Accuracy of Database 3

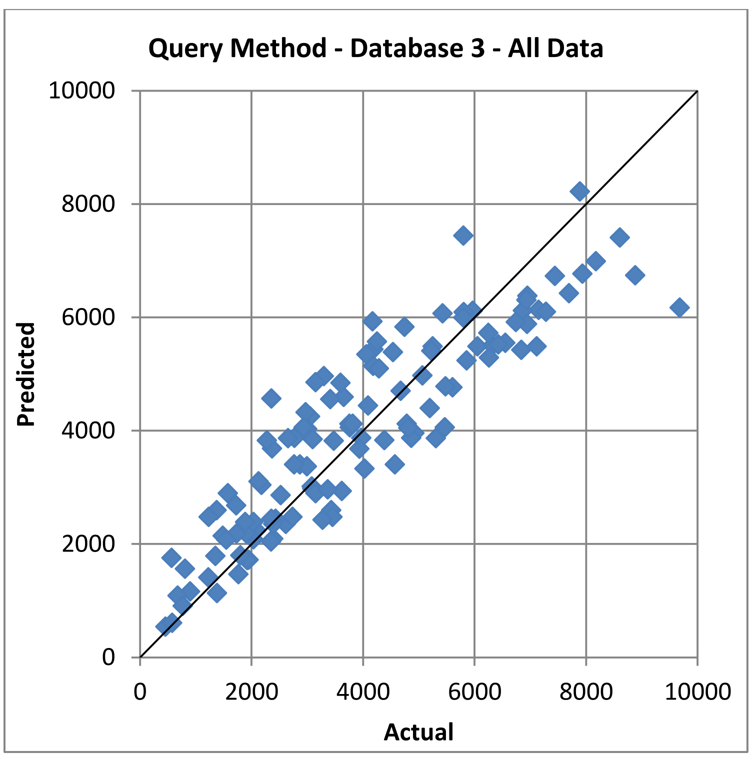

Figure 9-6 Query Method All Data Accuracy of Database 3

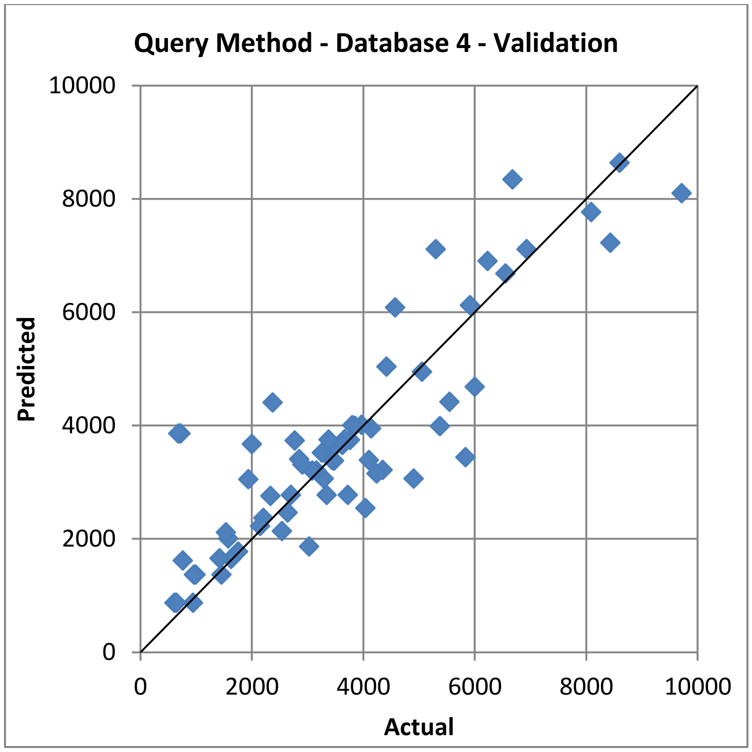

Figure 9-7 Query Method Validation Accuracy of Database 4

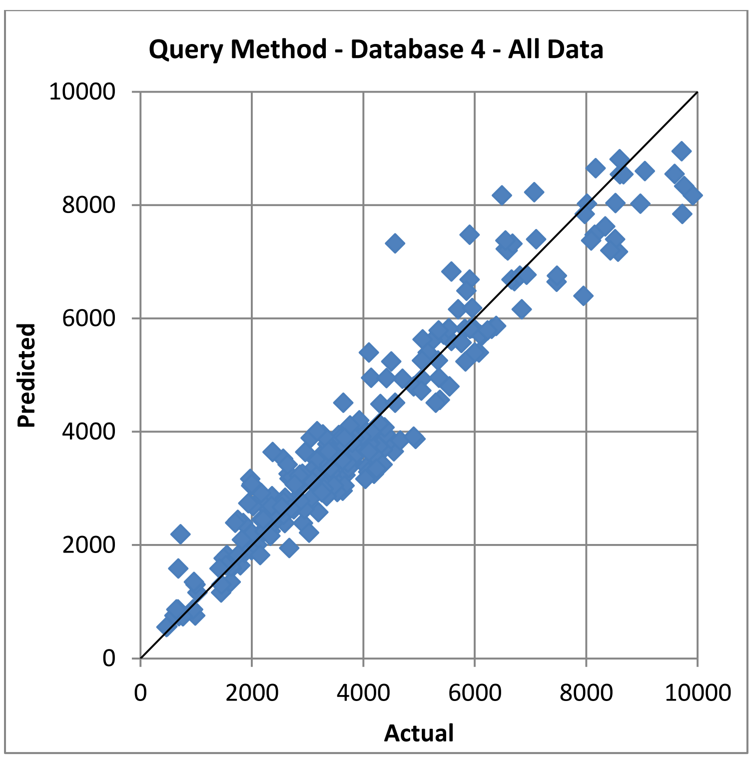

Figure 9-8 Query Method All Data Accuracy of Database 4

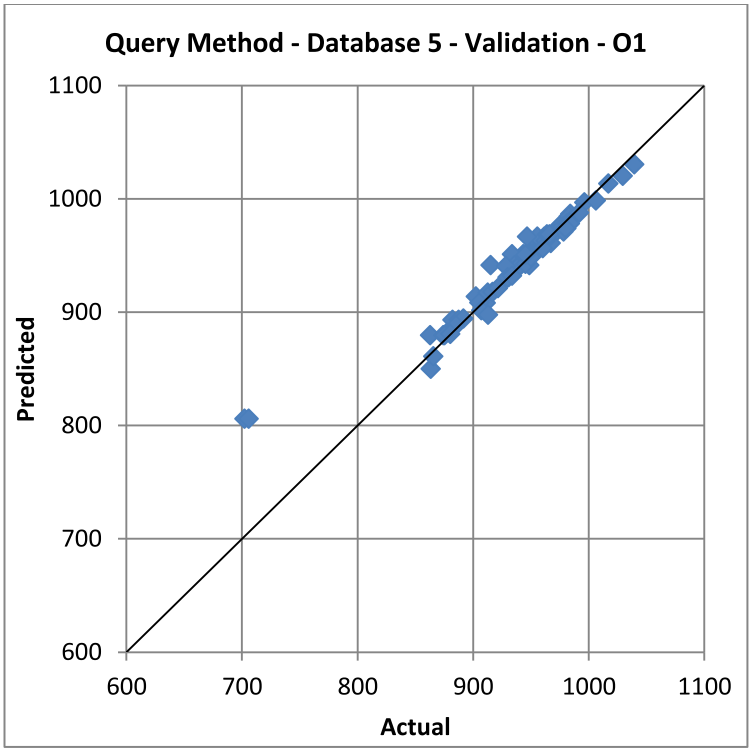

Figure 9-9 Query Method Validation Accuracy of Database 5, Output 1

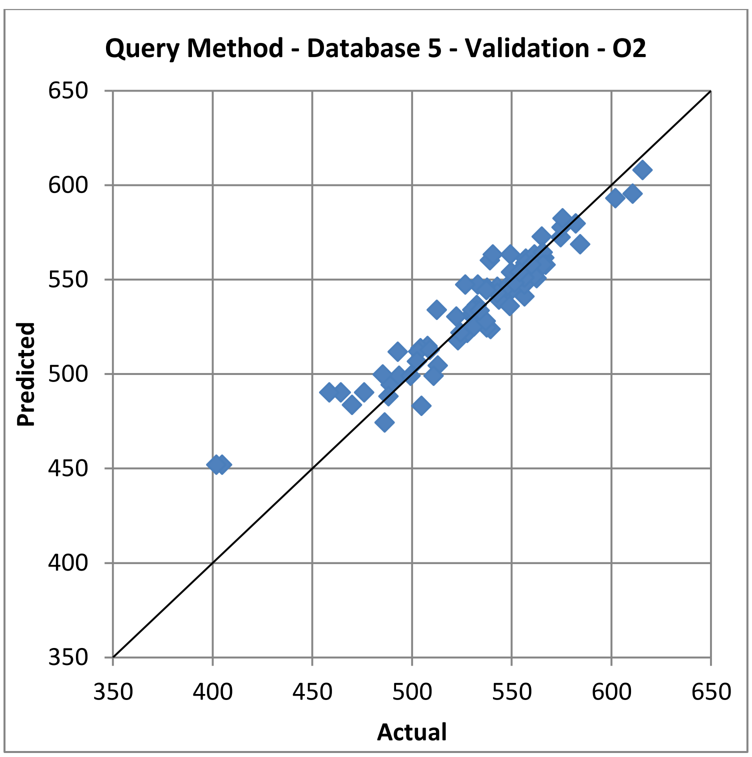

Figure 9-10 Query Method Validation Accuracy of Database 5, Output 2

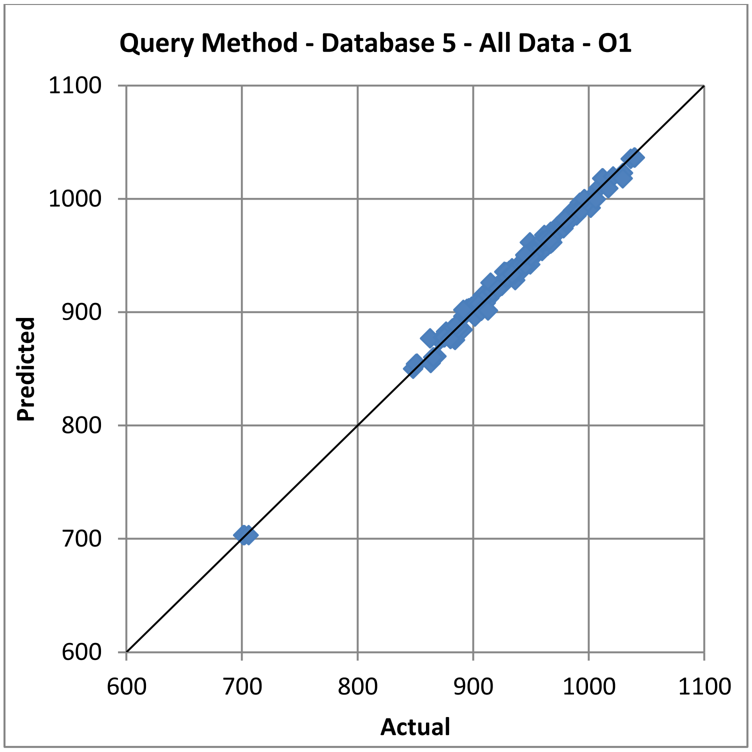

Figure 9-11 Query Method All Data Accuracy of Database 5, Output 1

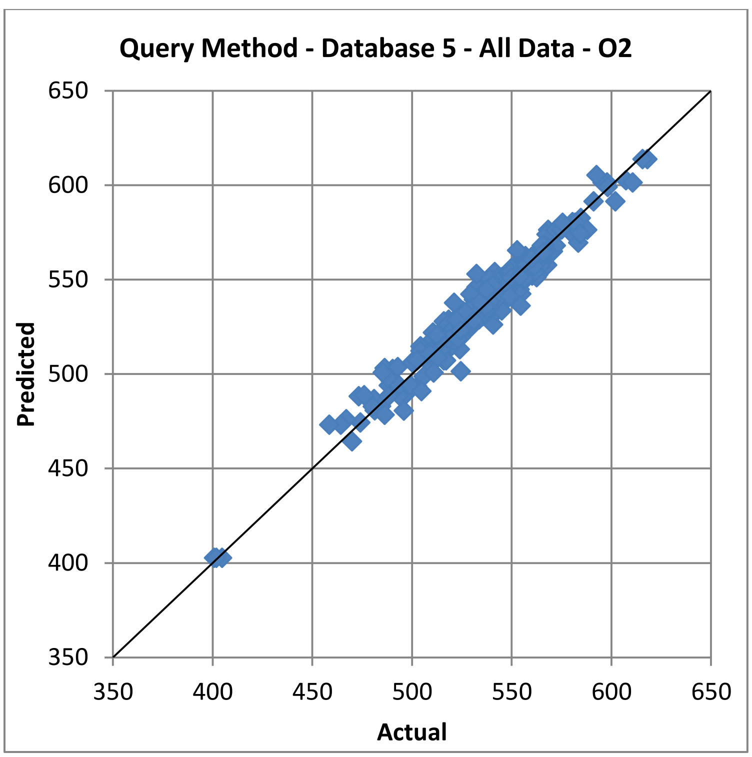

Figure 9-12 Query Method All Data Accuracy of Database 5, Output 2

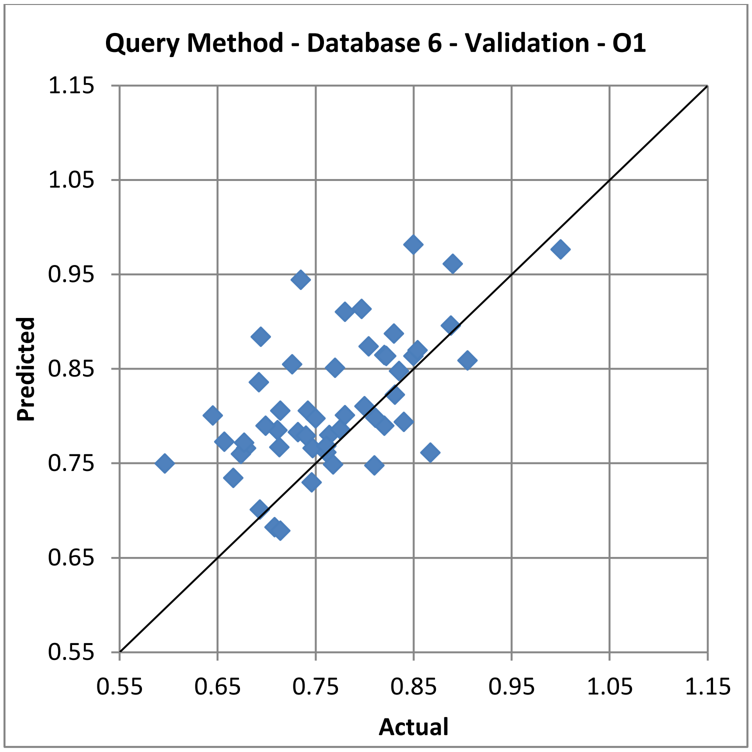

Figure 9-13 Query Method Validation Accuracy of Database 6, Output 1

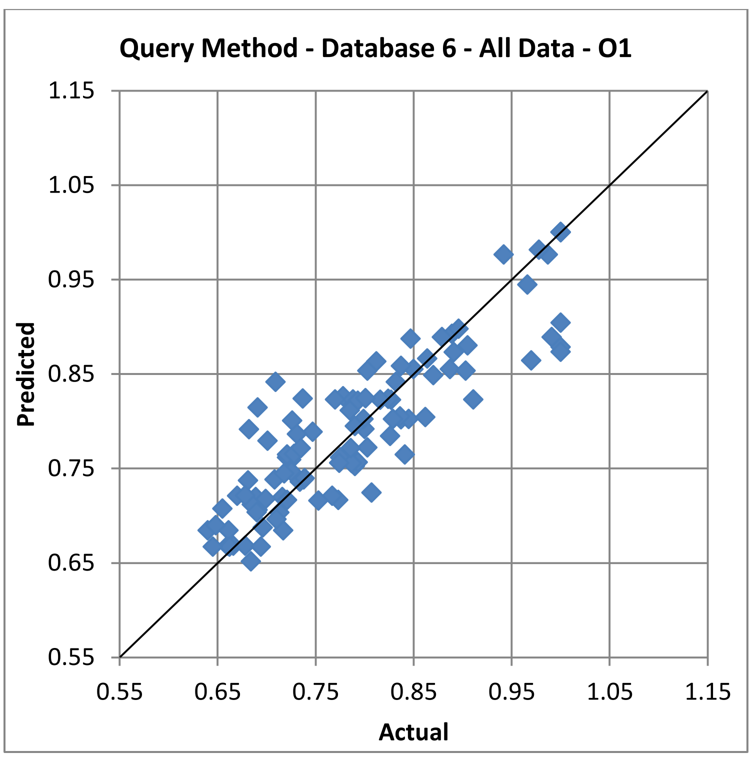

Figure 9-14 Query Method All Data Accuracy of Database 6, Output 1

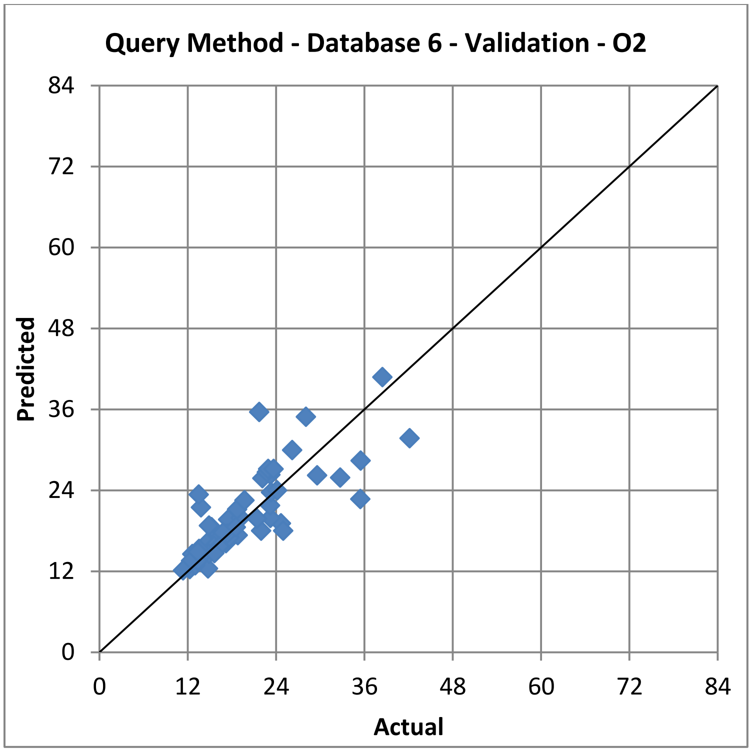

Figure 9-15 Query Method Validation Accuracy of Database 6, Output 2

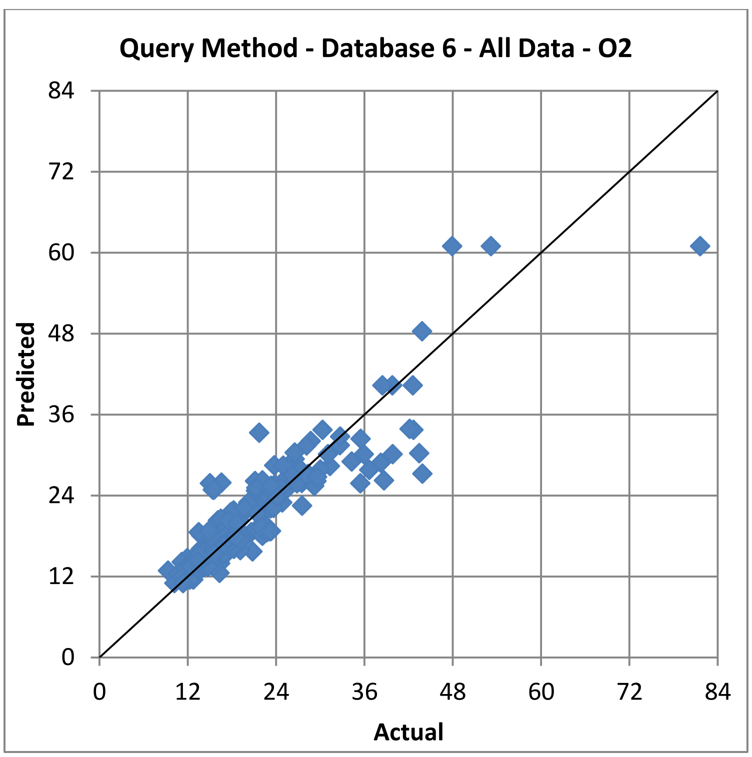

Figure 9-16 Query Method All Data Accuracy of Database 6, Output 2

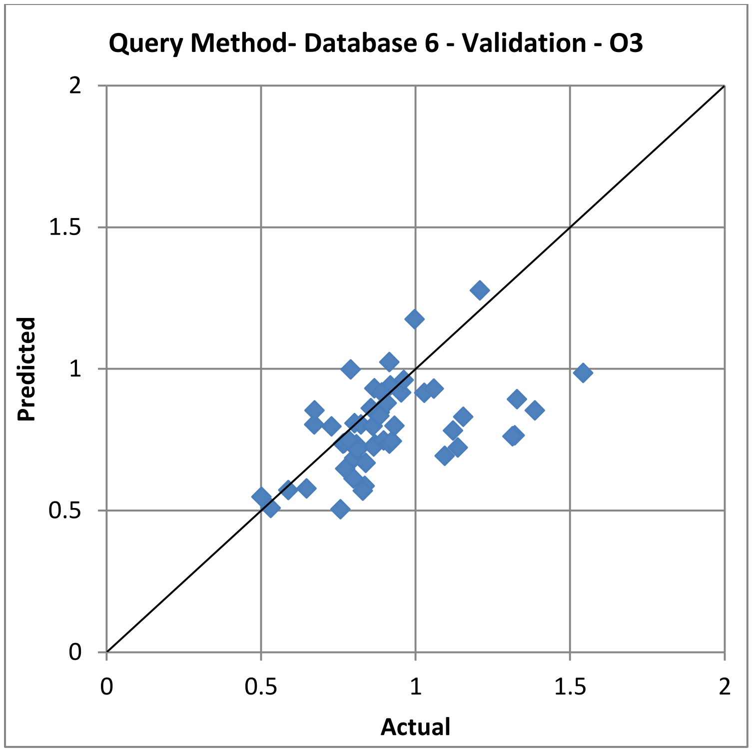

Figure 9-17 Query Method Validation Accuracy of Database 6, Output 3

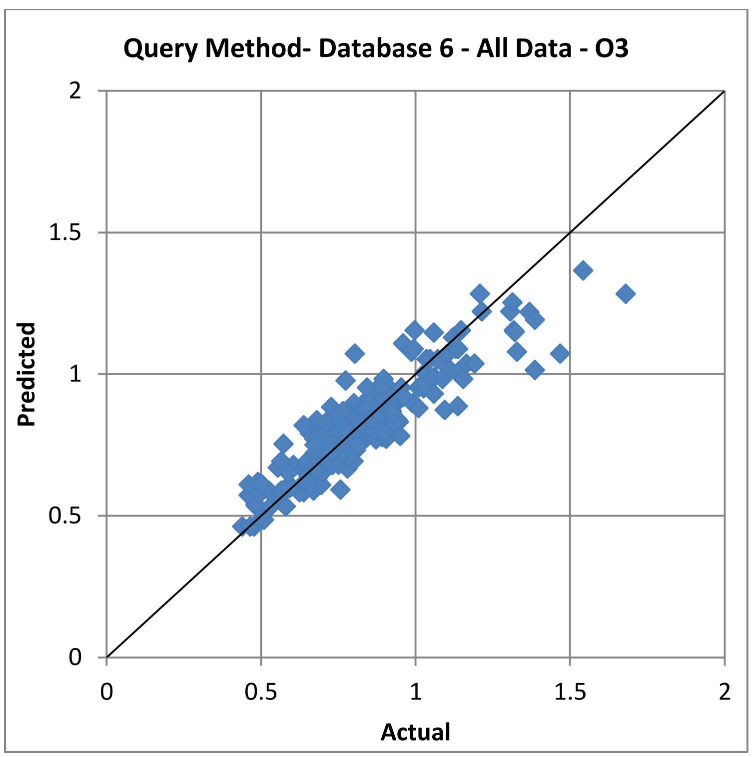

Figure 9-18 Query Method All Data Accuracy of Database 6, Output 3

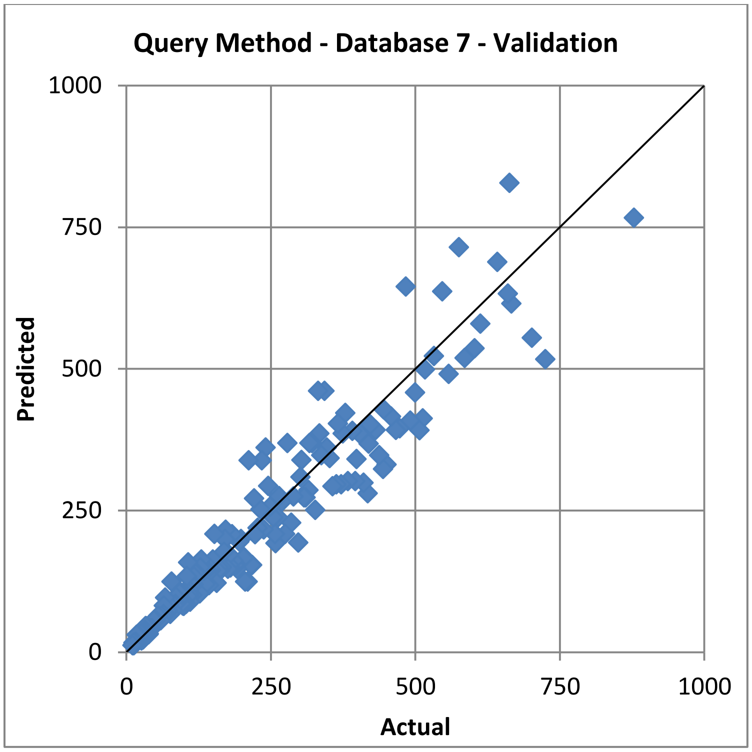

Figure 9-19 Query Method Validation Accuracy of Database 7

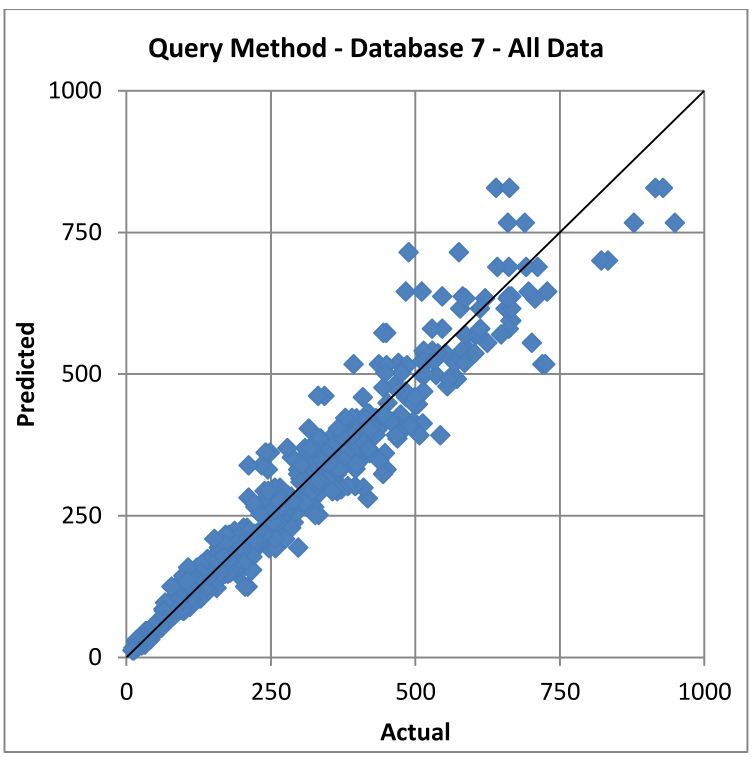

Figure 9-20 Query Method All Data Accuracy of Database 7

QUERY METHOD APPLICATION

# of Inputs =	7		Application	229
# of Output(s) =	1		Validation	71

RUN APPLICATION | VALIDATION

#	X1	X2	X3	X4	X5	X6	X7	Out
1	4.3	1.5	4.4	13	2.5	13	35	144.22
2	**12**	**3.5**	**25**	**45**	**2.4**	**45**	1.6	217.35
3	1	5	10.00	1.50	3.0	45	10	332.2
4	9	11	3.30	1.10	0.8	1.1	1.5	919.3
5	7	11	16.00	2.10	2.4	70	16	953.21
6	11	11	1.00	2.70	2.0	25	0.2	974.18
7	17	11	3.70	0.80	3.7	0.8	2.3	1138.01
8	18	12	29.00	2.60	3.8	2.6	3.1	1277.63
9	6	14	14.00	2.30	1.7	3.8	88	1452.93
10	14	16	1.60	0.20	2.6	19	3.2	2080.48
11	30	18	5.00	4.40	2.0	34	15	2823.61
12	18	20	2.00	21.00	3.0	23	3.6	3239.84
13	10	6	29.00	3.80	45.0	3.8	20	4178.71
14	10	20	13.00	2.20	17.0	2.2	45	4479.13
15	1.7	20	1.80	2.20	17.0	2.2	10	4470.34
16	3.3	6	3.40	3.80	45.0	3.8	25	4159.95

Figure 9-21 Excel-based Query Method Application - Database Replacement Screen shot

Incomplete Dataset

X1	X2	X3	X4	X5	X6	X7	Output
6		19	1.6	1.1	1.6	15	158.33

SORT DATABASE

THE CLOSEST DATASETS

	X1	X2	X3	X4	X5	X6	X7	Out	Euclidean
	6	2.9	19	1.6	1.1	1.6	15	158.33	0
	7	3	20	1.5	1	1.5	16	167.43	0.0511
	5	2.8	18	1.7	1.2	1.7	14	151.13	0.05108
Average	6	2.9	19	1.6	1.1	1.6	15	158.963	

Figure 9-22 Excel-based Query Method Application - User Interface Screen shot

Table 9-1 Statistical Accuracy Measures of the Query Method and Regression Analysis, Validation Datasets

		Query Method			Regression Analysis		
Database #	**Output**	**MARE**	**R^2**	**MRSE**	**MARE**	**R^2**	**MRSE**
Database 1	Output 1	16.1212	0.8791	41.1251	27.8169	0.9466	26.9556
Database 2	Output 1	4.5964	0.1731	0.7669	4.1551	0.3280	0.6884
Database 3	Output 1	45.1334	0.3369	317.5696	33.6932	0.7917	176.1023
Database 4	Output 1	32.8600	0.764	124.331	55.0471	0.5933	166.3330
Database 5	Output 1	0.8540	0.9270	1.9310	0.2418	0.9966	0.3639
	Output 2	1.7740	0.9040	1.4260	1.5859	0.9233	1.2218
Database 6	Output 1	8.5420	0.3490	0.0110	7.3136	0.3080	0.0100
	Output 2	14.5020	0.6140	0.6230	19.2126	0.5527	0.7612
	Output 3	16.0950	0.2410	0.0310	13.3274	0.5486	0.0212
Database 7	Output 1	14.5443	0.9209	3.6459	67.7496	0.8355	5.2258

Table 9-2 Statistical Accuracy Measures of the Query Method and Regression Analysis, All Data

		Query Method			Regression Analysis		
Database #	**Output#**	**MARE**	**R^2**	**MRSE**	**MARE**	**R^2**	**MRSE**
Database 1	Output 1	6.5119	0.9775	9.1164	27.5339	0.9469	13.4313
Database 2	Output 1	3.5651	0.5508	0.2919	4.5952	0.2956	0.3635
Database 3	Output 1	24.0542	0.8293	90.4443	31.4739	0.7460	109.3843
Database 4	Output 1	12.4340	0.9220	35.2290	36.8175	0.6163	77.9634
Database 5	Output 1	1.7740	0.9040	1.4260	1.5859	0.9233	1.2218
	Output 2	0.8480	0.9590	0.3340	1.3239	0.9084	0.4984
Database 6	Output 1	4.7510	0.7200	0.0030	6.8951	0.4731	0.0047
	Output 2	11.0670	0.8220	0.2690	18.1858	0.6419	0.3803
	Output 3	8.3280	0.8060	0.0070	11.5157	0.6732	0.0084
Database 7	Output 1	10.6884	0.9530	1.4197	65.2027	0.8162	2.7822

Table 9-3 Changes of the Statistical Accuracy Measures from Validation to All Data

		Query Method			Regression Analysis		
		MARE	R^2	MRSE	MARE	R^2	MRSE
Database #	**Output#**	**Reduction (%)**	**Improv. (%)**	**Reduction (%)**	**Reduction (%)**	**Improv. (%)**	**Reduction (%)**
Database 1	Output 1	59.6	11.2	77.8	1.0	0.0	50.2
Database 2	Output 1	22.4	218.2	61.9	-10.6	-9.9	47.2
Database 3	Output 1	46.7	146.1	71.5	6.6	-5.8	37.9
Database 4	Output 1	62.2	20.7	71.7	33.1	3.9	53.1
Database 5	Output 1	-107.7	-2.5	26.2	-555.8	-7.4	-235.8
	Output 2	52.2	6.1	76.6	16.5	-1.6	59.2
Database 6	Output 1	44.4	106.3	72.7	5.7	53.6	53.0
	Output 2	23.7	33.9	56.8	5.3	16.2	50.0
	Output 3	48.3	234.4	77.4	13.6	22.7	60.5
Database 7	Output 1	26.5	3.5	61.1	3.8	-2.3	46.8

CHAPTER 10

10. HYBRID DECISION MAKING SYSTEM

10.1 System Components

In the previous chapters, seven databases described in Chapter 4 were utilized to develop a new set of ANN modeling approaches/paradigms along with a new method to tackle partially missing data. For each database, static ANN network was developed in four sequential stages. The predictions generated at the fourth stage were fed into Feedback ANN network, Auto-associative network, and Dynamic-sequential network. Then the model development process was initiated for these approaches to develop new models. In Chapter 9, the Query method, which is a new approach to replace partially missing dataset, was introduced and utilized to develop the application for seven databases. All the statistical measures of the models developed in previous chapter are grouped together in one table for each database and are shown in Table 10-1 to Table 10-10. As can be seen from statistical measures, the performances of the developed models for each database varied. For instance, the best performing network for database 3, database 6 –output 1, -output2, and - output 3, and database 7 was found to be the Feedback ANN network in terms of overall MRSE value. For database 1 and database 2, the best performing network was found to be the Dynamic-sequential network. Similarly, database 4 was modeled best by the Auto-associative network. As also graphically presented in Figure 10-2 to Figure 10-11, the MRSE values by Static ANN network for database 5 - output 1 and –output 2 were the least. All the statistical measures and the graphs proved that the best performing network can vary depending on the characteristics of the database. For this reason, it is necessary to integrate all the models and utilize them through a hybrid decision system before the final decision is made. The schematic diagram of the proposed hybrid decision making system is shown in Figure 10-1. Therefore, the proposed hybrid decision making system has the following components:

1. Static ANN Network

2. Feedback ANN Network
3. Auto-associative Network
4. Dynamic-sequential Network
5. Query Method Application

10.2 System Prediction

As can be seen from the diagram in Figure 10-1, the first part of the diagram questions whether the dataset has a missing variable or not. If there is a missing parameter, then the missing value is replaced by the Query method. Once the complete dataset is obtained, static ANN network generates a prediction by using the input variables. The generated prediction by static ANN network is then fed into Feedback ANN, Auto-associative, and Dynamic-sequential networks. Then, three more predictions are generated from these networks. Additionally, another prediction is obtained from the Query method. As mentioned in Chapter 9, Query method can be used to generate outputs as well. However, the accuracy of the Query method is not as good as the other networks but it is still can be considered in the final decision. It is noteworthy to state that the Query method is developed by using entire database to expand the number of neighborhoods. As static ANN network feeds into the other networks, the prediction by the Query method can also feed into the same networks. However, only static ANN predictions are considered in this study to provide initial estimates because of their high accuracy performance in the model development process. Future studies will look into expanding this study by including Query method output as an initial estimate. Once all four networks are utilized along with the Query method, five predictions are provided to the user. Even though this system is designed for the user to make the final decision utilizing a prediction range, recommended values based on the statistical accuracy measures obtained in the model development process are also provided. Essentially, the user is provided with a prediction range as well as the weighted outputs based on MRSE, MARE, and R^2 values. The weighted output based on the MRSE is calculated as follows:

$$Y_{MRSE} = Y_{Query} \times \frac{1}{\left(\frac{MRSE_{Query}}{\sum MRSE}\right)} + Y_{Static} \times \frac{1}{\left(\frac{MRSE_{static}}{\sum MRSE}\right)} + Y_{Feedback} \times \frac{1}{\left(\frac{MRSE_{Feedback}}{\sum MRSE}\right)} + Y_{Autoasso.} \times \frac{1}{\left(\frac{MRSE_{Autoasso.}}{\sum MRSE}\right)} + Y_{Dynamic} \times \frac{1}{\left(\frac{MRSE_{Dynamic}}{\sum MRSE}\right)} \quad \text{Eqn. 10.1}$$

Similarly, the weighted output based on the MARE values can be calculated as:

$$Y_{MARE} = Y_{Query} \times \frac{1}{\left(\frac{MARE_{Query}}{\sum MARE}\right)} + Y_{Static} \times \frac{1}{\left(\frac{MARE_{static}}{\sum MARE}\right)} + Y_{Feedback} \times \frac{1}{\left(\frac{MARE_{Feedback}}{\sum MARE}\right)} + Y_{Autoasso.} \times \frac{1}{\left(\frac{MARE_{Autoasso.}}{\sum MARE}\right)} + Y_{Dynamic} \times \frac{1}{\left(\frac{MARE_{Dynamic}}{\sum MARE}\right)} \quad \text{Eqn. 10.2}$$

The weighted output based on the Coefficient of determination, R^2 is expressed as:

$$Y_{R^2} = Y_{Query} \times \left(\frac{R^2{}_{Query}}{\sum R^2}\right) + Y_{Static} \times \left(\frac{R^2{}_{static}}{\sum MARE}\right) + Y_{Feedback} \times \left(\frac{R^2{}_{Feedback}}{\sum MARE}\right) + Y_{Autoasso.} \times \left(\frac{R^2{}_{Autoasso.}}{\sum MARE}\right) + Y_{Dynamic} \times \left(\frac{R^2{}_{Dynamic}}{\sum MARE}\right) \quad \text{Eqn. 10.3}$$

Basically, each prediction from the networks is weighted with their model development accuracies and then summed to obtain a weighted output. Equations 10.1, 10.2, and 10.3 represent the weighted outputs based on, respectively, MRSE, MARE, and R^2. Equations 10.1 and 10.2 look different than the Equation 10.3 because of the inverse ratio.

10.3 Utilization

In order to develop a hybrid decision making system for each database in an Excel environment, the connection weights, threshold values and coefficients of the optimal networks, which are described in Chapter 3 were imported into an Excel sheet. The components of each network (i.e. Static ANN, Feedback ANN, Auto-associative, and Dynamic-sequential networks) were imported into an individual worksheet that is linked to an integrated user interface worksheet where all the predictions are shown from all the networks. Developed Query method application for each database was also carried to an individual worksheet in the same excel file. Three examples of the Excel interfaces developed for three databases: Database 1, Database 2, and Database 6 are, respectively, shown in Figure 10-12, Figure 10-13, and Figure 10-14. As can be seen from example interfaces in the figures, two databases have one output and one database has three outputs. By entering the input parameters in the input cells, Excel automatically generates all the predictions by the networks that are embedded in other worksheets. All the predictions are shown under the output section. The only prediction that needs an operation to be generated is the Query method. All the calculations by the Query method are accomplished in other worksheet in the same Excel file. However, datasets have to be sorted based on the least Euclidean distance. This operation can be easily done by using the data sort feature of Excel, but developed worksheets are equipped with Query method button that accomplishes the sorting. The same button is also used for a missing dataset. For example, if an incomplete dataset is entered in one of those interfaces developed in this study, by clicking the Query method button the Excel-based interface sorts the datasets, finds the closest neighborhoods with the closest values, then replaces the missing value. The three closest datasets with the least Euclidean distance are provided to the user on the interface as it can also be seen in Figure 10-12, Figure 10-13, and Figure 10-14. The user can also make the decision to replace the value based the closest datasets because the Query method can find the exact matching datasets with the same input parameters. In this case, the user can manually input the value instead of the Query method application replacing the average of the three closest values, which may diverge from the actual value. Basically, the Query method button located on the interface helps the user to sort the datasets to replace the missing and/or just sort the datasets to be able to use the closest output value. It should be noted that if there is a missing parameter in the dataset, the Excel sheet will not show any predictions. Additionally, the input variables are recommended to be within the applicable range that is placed under the input cells. If the entered input value is out of applicable range, then the models are not valid and Excel shows a warning text below. Moreover, the reflection of the dataset by Auto-associative network is placed right under the input cells to evaluate the Query method input replacements. In this case, the user has options to make the decision to replace the value independently. The user can rely on one of the datasets that the Query method finds, or the average of the three closest values it automatically replaces with. Then the Auto-associative network provides prediction of the data itself (reflections). As can be depicted from the example interfaces, the output section has five outputs, which are Query method, static ANN, ANN-feedback, Auto-associative, and Dynamic-sequential. Once the datasets in the worksheet of Query method application are sorted, then the Query method prediction of the output is automatically updated because all the cells on the interface page work simultaneously and update the values. If there are multiple outputs, such as in Figure 10-14, there are individual output sections with all the predictions for each output. For database 6, three output sections show all the predictions by the networks and the application placed in other worksheets. In order for the user to make the proper decision, statistical accuracy measures of model development stages (i.e. MARE, MRSE, and R^2) are also placed under the output sections. As mentioned previously, by substituting the statistical measures and the predictions in Equations 10.1, 10.2, and 10.3, recommended values are calculated and provided to the user along with the minimum and maximum of the prediction range. As a result, the Hybrid decision making system (HDMS) is produced for seven databases by using the same procedure explained herein.

10.4 Consistency of System Predictions

Once the recommended values are calculated for each output of the seven databases, the accuracy of the recommended values was also evaluated by using the HDMS. All datasets, used to develop the networks and the applications, were imputed in HDMS and the accuracy of the recommended values was calculated for seven databases with their corresponding output(s). As can be evaluated from Table 10-11 to Table 10-20, statistical accuracy measures of the recommended values are in agreement. Only database 1 yielded somewhat different MRSE results while all other databases agreed on the stabilized outputs based on MARE, MRSE, and R^2 values.

As can be seen from all the accuracy measures and the figures, HDMS is a well-designed, multifunctional interface to integrate all the networks. The query method is a reliable solution to handle a missing datasets and very handy to implement through the HDMS. Static ANN, Feedback-ANN, Auto-associative network, and Dynamic-sequential network work simultaneously as parallel systems with no delays through the HDMS. The recommended values help the user to justify the decision. The provided prediction range indicates how far the networks' predictions differ. Consequently, HDMS can be used easily and does not require the user to have prior knowledge of model development.

10.5 Figures and Tables

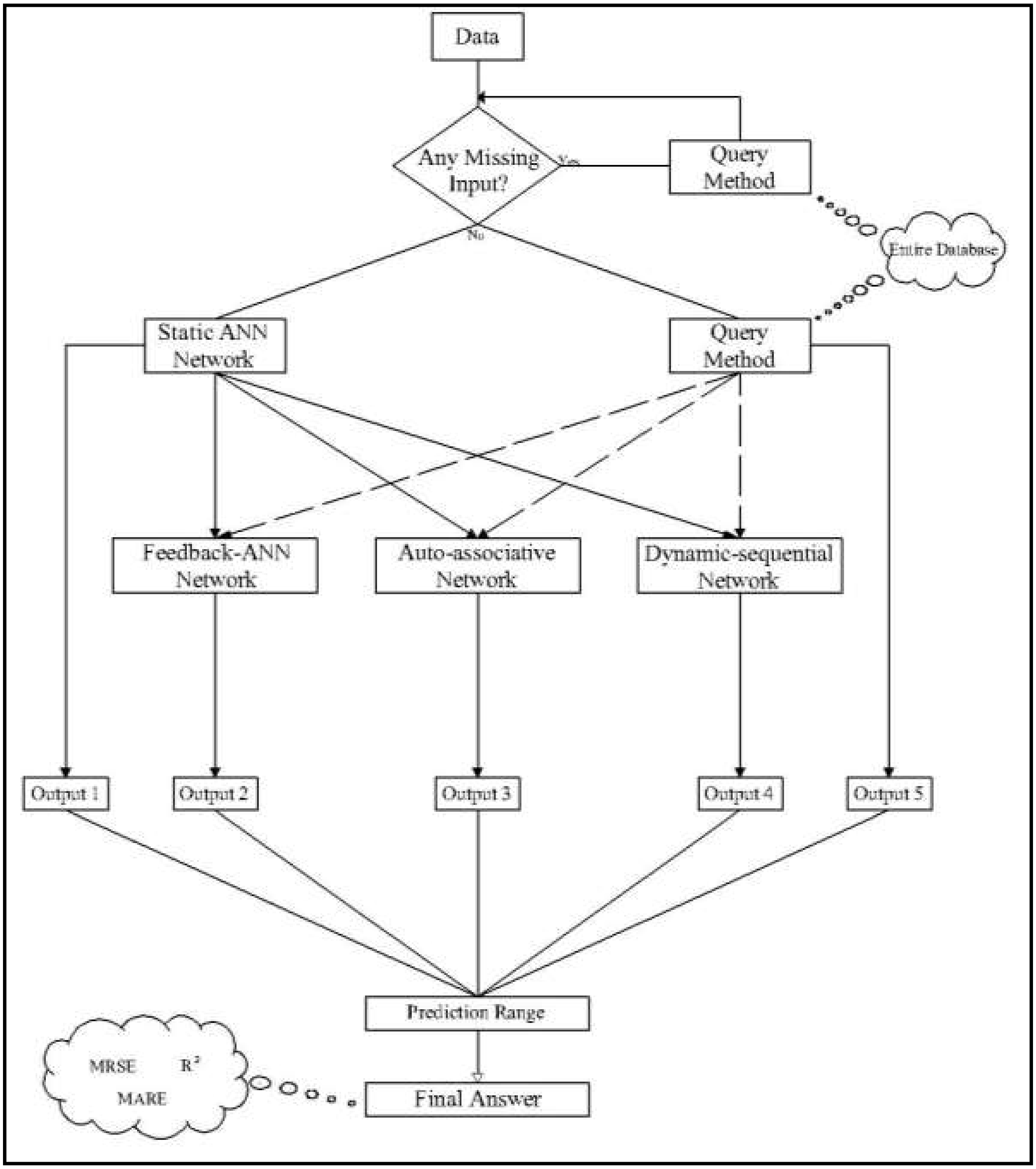

Figure 10-1 Schematic Diagram of the Proposed Hybrid Decision System

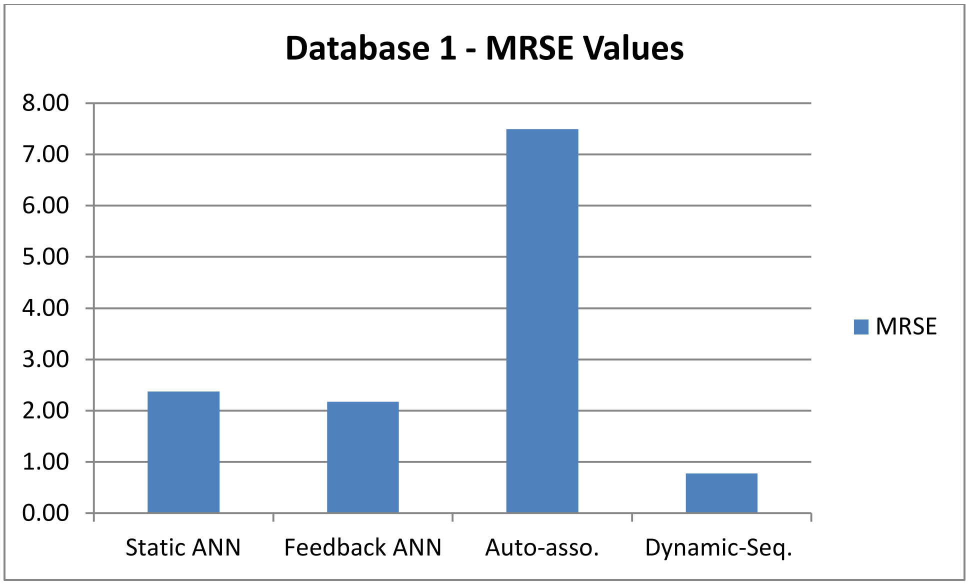

Figure 10-2 Comparison of the networks based on the MRSE values for database 1

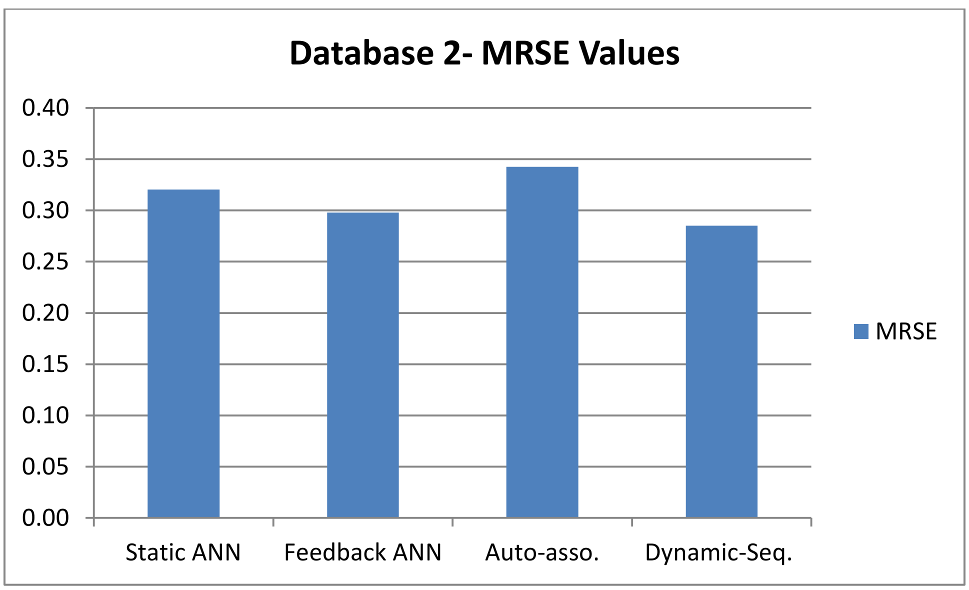

Figure 10-3 Comparison of the networks based on the MRSE values for database 2

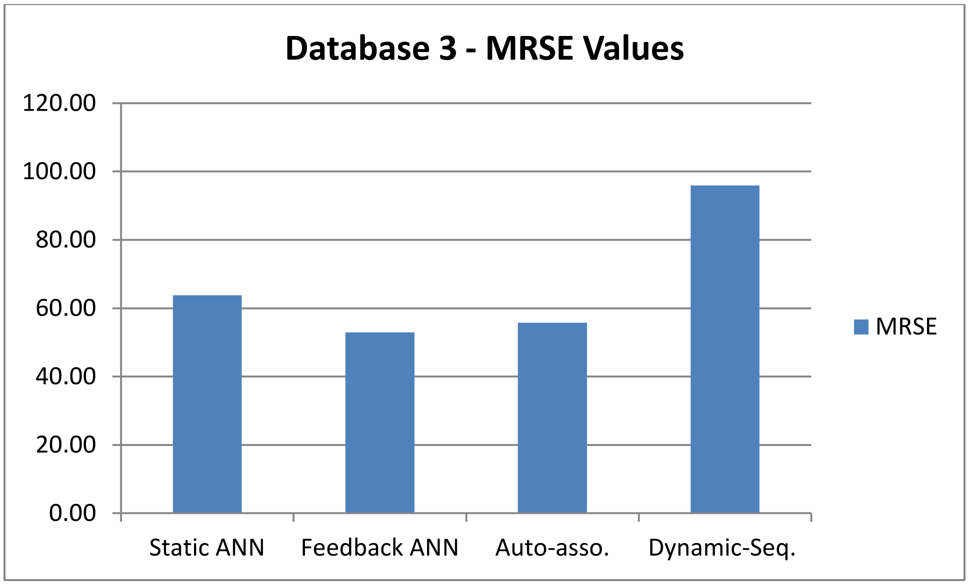

Figure 10-4 Comparison of the networks based on the MRSE values for database 3

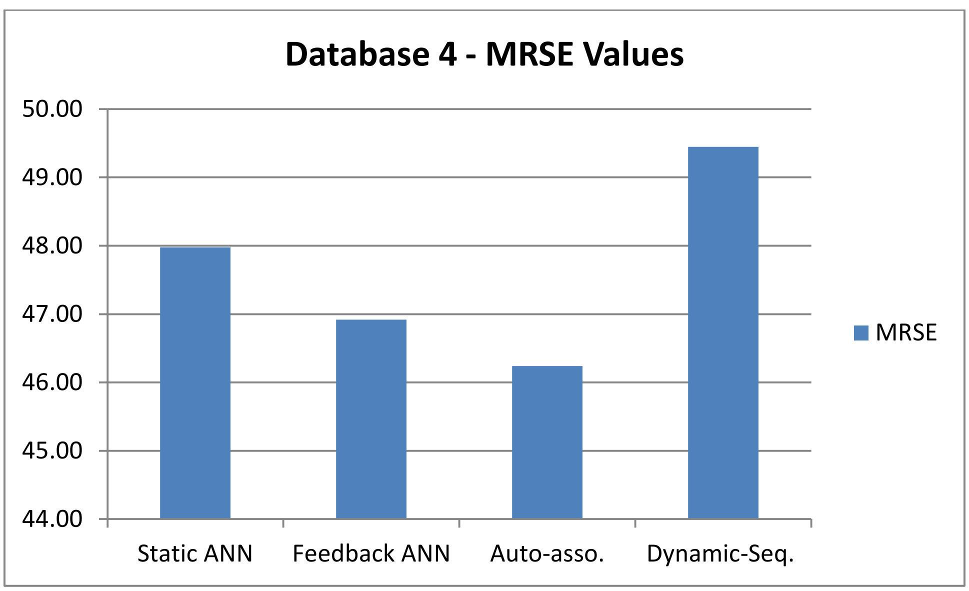

Figure 10-5 Comparison of the networks based on the MRSE values for database 4

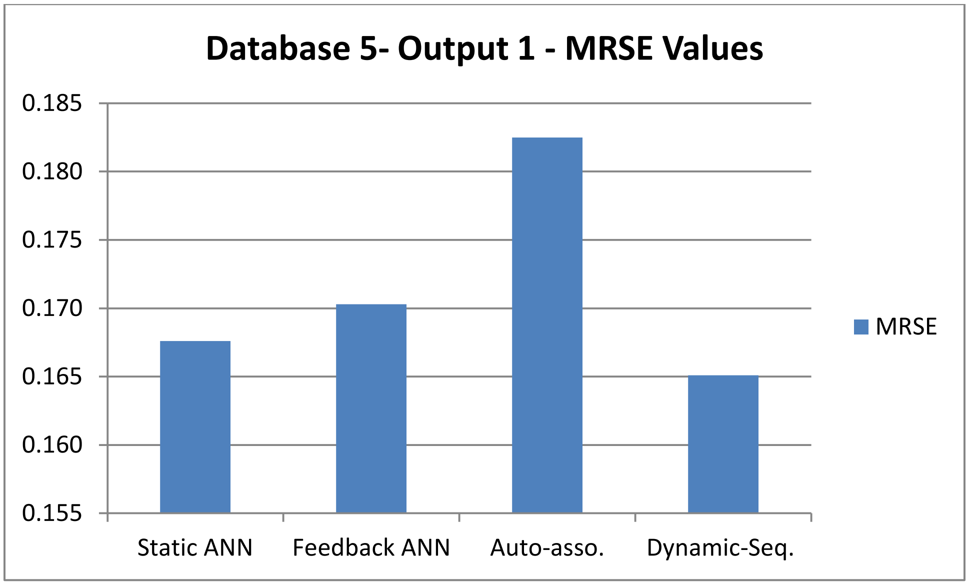

Figure 10-6 Comparison of the networks based on the MRSE values for database 5, Output 1

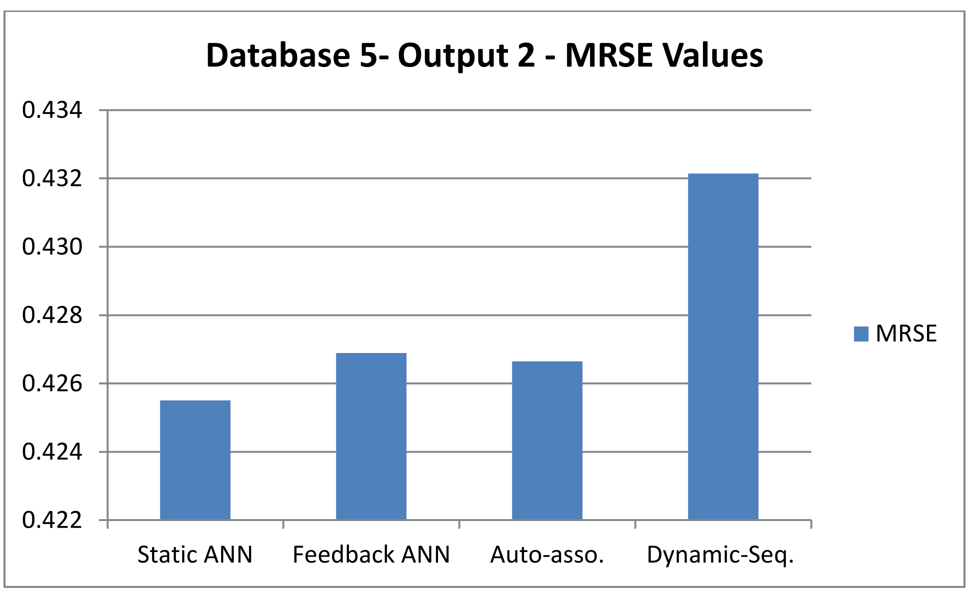

Figure 10-7 Comparison of the networks based on the MRSE values for database 5, Output 2

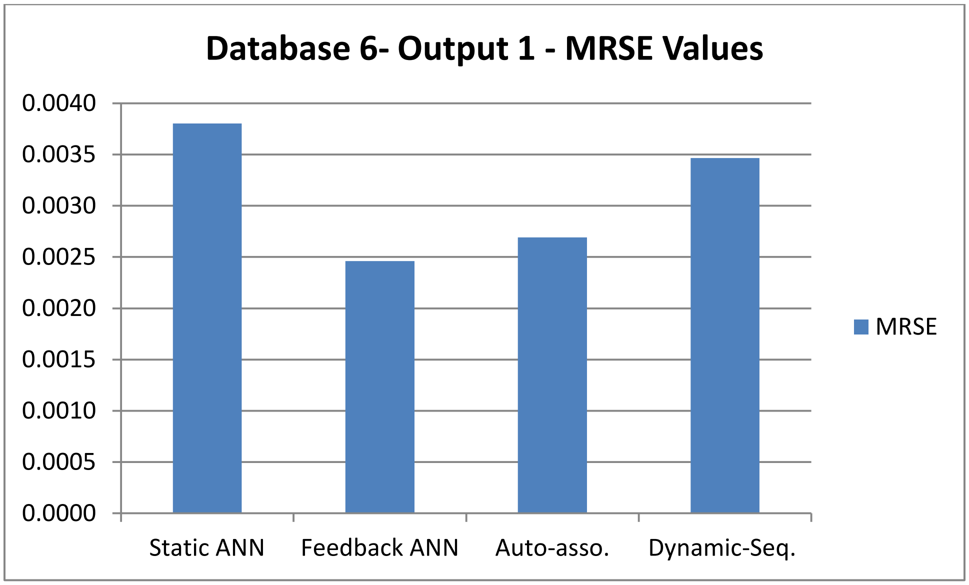

Figure 10-8 Comparison of the networks based on the MRSE values for database 6, Output 1

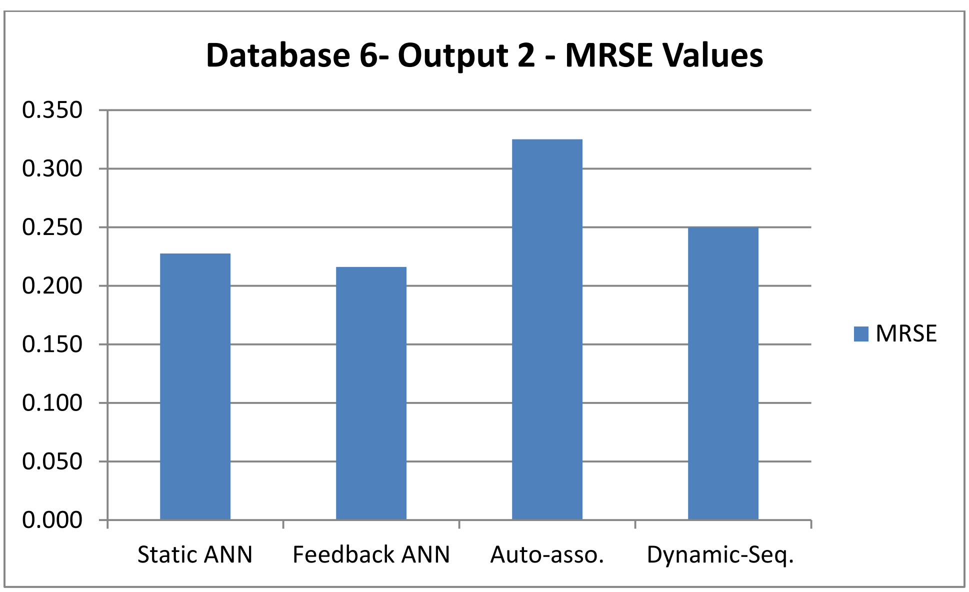

Figure 10-9 Comparison of the networks based on the MRSE values for database 6, Output 2

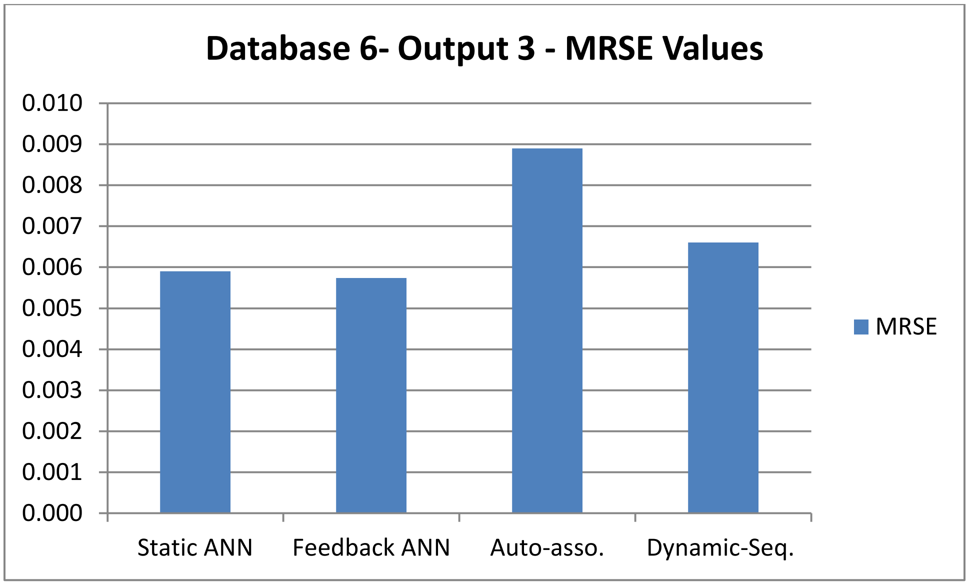

Figure 10-10 Comparison of the networks based on the MRSE values for database 6, Output 3

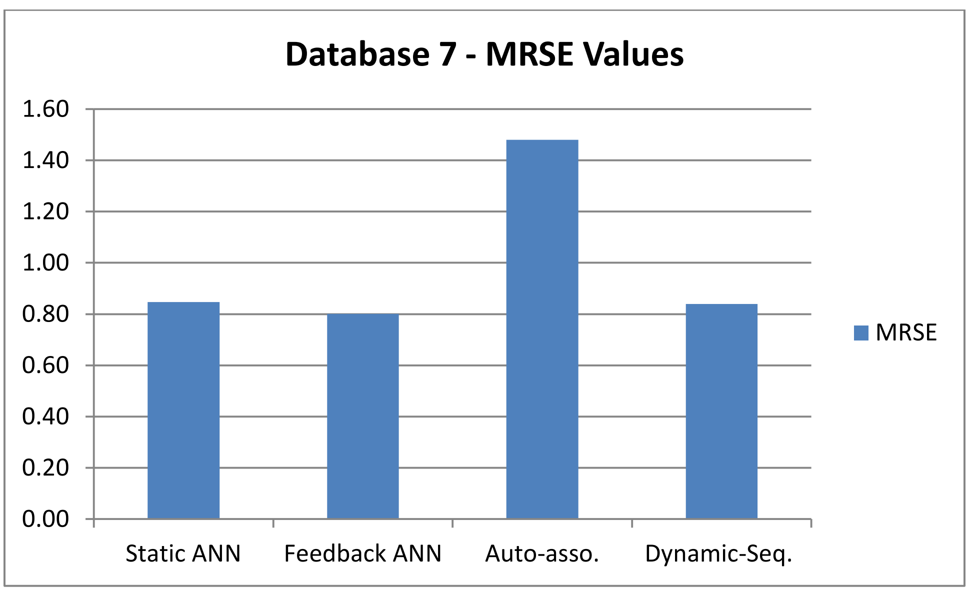

Figure 10-11 Comparison of the networks based on the MRSE values for database 7

INPUTS									
	X1	X2	X3	X4	X5	X6	X7	Actual	
	3.9	12.0	4.0	4.4	15.0	4.4	1.2		
Auto Assoc.	6.1	3.1	19.7	2.6	2.2	3.4	12.7	225.4	
Min	1	1.5	1	0.2	0.8	0.8	0.2	144	
Max	30	20	29	45	45	70	88	4479	

Closest Datasets									
	X1	X2	X3	X4	X5	X6	X7	Output1	Eucli. Dis.
	6	2.9	19	1.6	1.1	1.6	15	158.33	0.00000
	7	3	20	1.5	1	1.5	16	167.43	0.00000
	5	2.8	18	1.7	1.2	1.7	14	151.13	0.00000
Average	2.9	19.0	1.6	1.1	1.6	15.0	159.0	159.0	

OUTPUTS					
	QUERY	STATIC	ANN-FEEDBACK	Auto-Associative	Dynamic-Sequential
	158.96	236.62	212.83	225.42	176.82
MARE	5.5359	4.0690	3.2810	7.2686	1.3499
R2	0.9797	0.9984	0.9986	0.9839	0.9998
MRSE	8.6756	2.3740	2.1754	7.4928	0.7781

QUERY METHOD

Recommended Values		Min	Max
Y_{MARE}	196.060	158.96	236.62
Y_{MRSE}	195.937		
Y_{R2}	202.217		

Figure 10-12 Hybrid Decision Making System Screen-shot for Database 1

INPUTS									
	% pass	AADT	IRI	Surface Width	Shulder Type	Shulder Type	Shulder Width	85th_Perc Speed	
ENTER-->	3.0	150.0	281.0	18.0	0	1	1.0		
Auto Assoc.	16.8	4355.7	103.1	23.9	0	1	5.1	63.4	
Min	0	130	66	18	0	0	1	30	
Max	100	9200	333	24	1	1	10	71	

Closest Datasets									
	X1	X2	X3	X4	X5	X6	X7	Output1	Eucli. Dis.
	17	4200	97	24	0	1	5	65	0.00000
	17	3800	101	24	0	1	5	64	0.04958
	20	4500	78	24	0	1	5	66	0.08404
Average	18.0	4166.7	92.7	24.0	0.0	1.0	5.0	65.0	

85th Perc. Speed PREDICTION					
	QUERY	STATIC	ANN-FEEDBACK	Auto-Associative	Dynamic-Sequential
	65.00	63.87	64.64	63.35	65.60
MARE	3.5650	3.9681	3.6991	4.3912	3.6601
R2	0.5510	0.4554	0.5314	0.3798	0.5668
MRSE	0.2920	0.3203	0.2979	0.3425	0.2851

QUERY METHOD

Recommended Values		Min	Max
Y_{MARE}	64.5	63.35	65.60
Y_{MRSE}	64.5		
Y_{R2}	64.6		

Figure 10-13 Hybrid Decision Making System Screen-shot for Database 2

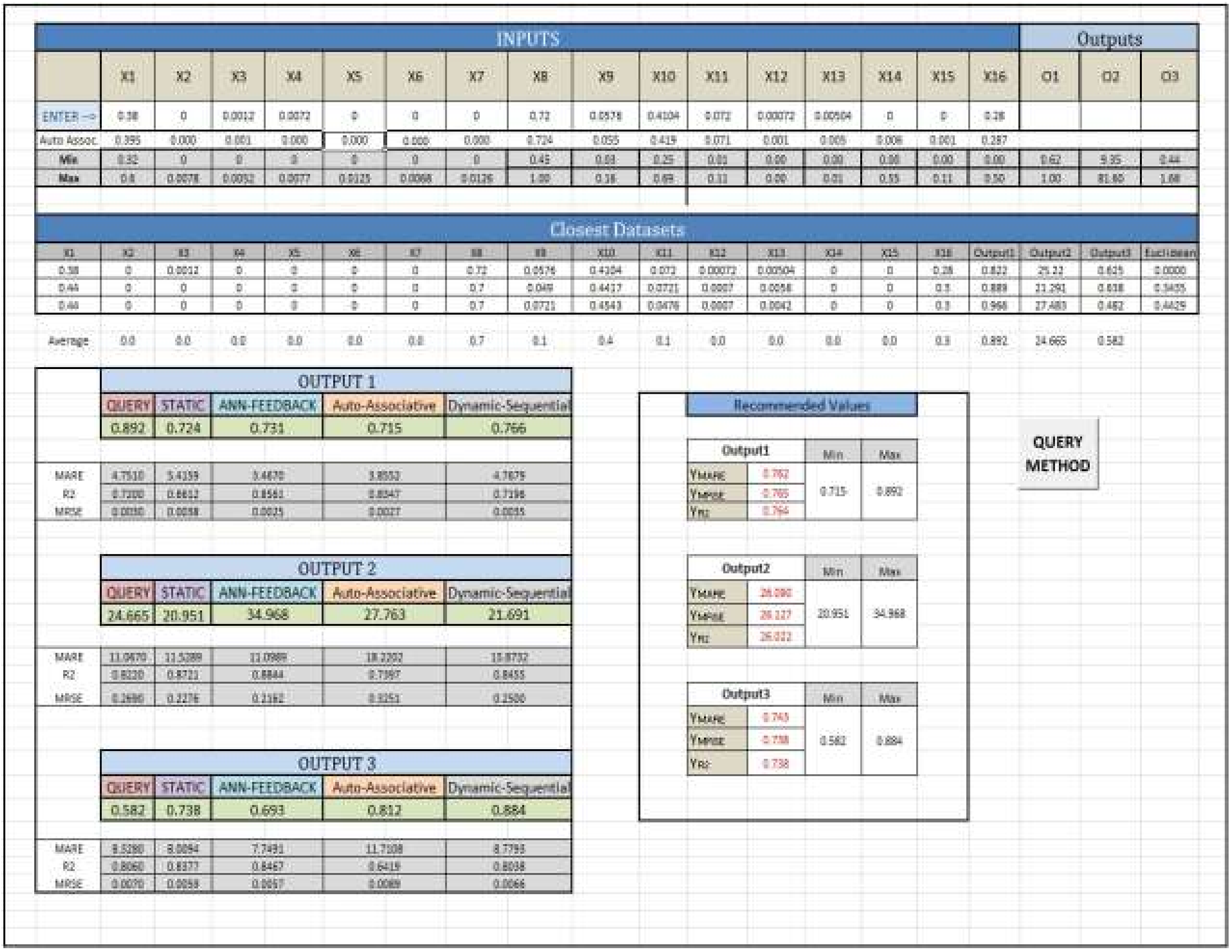

Figure 10-14 Hybrid Decision Making System Screen-shot for Database 6

Table 10-1 Comparison of All the Prediction Models Developed for Database 1

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		7-(8-19)-19500	8-(2-4)-3200-1	8-(3-6)-20000-8	8-(1-16)-19600-1
TR	MARE	2.0280	3.4090	6.2617	1.6319
	R^2	0.9996	0.9985	0.9868	0.9998
	MRSE	1.6151	3.1927	9.6230	1.2130
TS	MARE	2.7410	3.9910	6.9372	2.6656
	R^2	0.9978	0.9986	0.9802	0.9970
	MRSE	5.7671	4.7437	16.5741	6.6146
VAL	MARE	3.0140	4.1160	7.1629	2.9227
	R^2	0.9984	0.9979	0.9882	0.9978
	MRSE	4.5703	5.3192	12.9810	5.3440
All Data	MARE	4.0690	3.2810	7.2686	1.5499
	R^2	0.9984	0.9986	0.9839	0.9998
	MRSE	2.3740	2.1754	7.4928	0.7781
FINAL STRUCTURE		**7 - 19 - 1**	**8 - 4 - 1**	**8 - 6 - 8**	**8 - 16 - 1**

Table 10-2 Comparison of All the Prediction Models Developed for Database 2

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		7-(2-3)-3100-1	8-(3-4)-1100-1	8-(6-6)-3100-8	8-(1-3)-20000-1
TR	MARE	4.0297	4.0208	4.8125	3.6220
	R^2	0.6061	0.6596	0.4795	0.6681
	MRSE	0.4046	0.4059	0.4911	0.3839
TS	MARE	5.9550	5.3680	5.1122	5.9117
	R^2	0.0020	0.2694	0.1798	0.3861
	MRSE	1.0121	0.9944	0.8469	1.0838
VAL	MARE	6.0170	7.1612	6.5996	6.6696
	R^2	0.0078	0.0237	0.0009	0.0443
	MRSE	0.9647	1.1833	1.1071	1.0599
All Data	MARE	3.9681	3.6991	4.3912	3.6601
	R^2	0.4554	0.5314	0.3788	0.5668
	MRSE	0.3203	0.2979	0.3425	0.2851
FINAL STRUCTURE		**7 - 3 - 1**	**8 - 4 - 1**	**8 - 6 - 8**	**8 - 3 - 1**

Table 10-3 Comparison of All the Prediction Models Developed for Database 3

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		12-(2-6)-200-1	13-(1-5)-100-1	13-(7-8)-20000-13	13-(7-8)-100-1
TR	MARE	6.4320	8.8693	12.5441	14.3219
	R^2	0.9916	0.9849	0.9650	0.9743
	MRSE	29.5265	40.5326	58.4512	73.5715
TS	MARE	16.8540	9.5488	16.5935	18.0065
	R^2	0.9406	0.9797	0.9519	0.9126
	MRSE	113.2199	64.9743	103.2161	135.4948
VAL	MARE	15.4386	12.9420	22.1390	27.7868
	R^2	0.7221	0.7766	0.7209	0.6193
	MRSE	211.5120	188.8319	210.3098	250.2872
All Data	MARE	12.7195	9.9850	14.7704	19.1800
	R^2	0.9364	0.9466	0.9342	0.9317
	MRSE	63.7835	52.9530	55.7573	95.9043
FINAL STRUCTURE		**12 - 6 - 1**	**13 - 5 - 1**	**13 - 8 - 13**	**13 - 8 - 1**

Table 10-4 Comparison of All the Prediction Models Developed for Database 4

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		6-(2-7)-20000-1	7 -(2-3)-19900-1	7 -(5-7)-20000-7	7-(1-7)-20000-1
TR	MARE	17.440	20.825	20.557	20.581
	R^2	0.8554	0.8485	0.8539	0.8710
	MRSE	68.4546	70.0604	68.8665	81.6443
TS	MARE	22.372	22.496	21.863	21.747
	R^2	0.8226	0.8369	0.8363	0.7959
	MRSE	107.1671	102.3868	102.4501	133.0335
VAL	MARE	21.604	17.999	20.260	25.693
	R^2	0.7862	0.8626	0.8604	0.7672
	MRSE	118.7498	93.7436	95.4665	150.2486
All Data	MARE	20.359	19.470	18.321	17.653
	R^2	0.8549	0.8613	0.8653	0.8703
	MRSE	47.9782	46.91616	46.2396	49.4487
FINAL STRUCTURE		**6 - 7 - 1**	**7 - 3 - 1**	**7 - 7 - 7**	**7 - 7 -1**

Table 10-5 Comparison of All the Prediction Models Developed for Database 5, Output 1

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		3-(2-4)-19800-1	4-(2-4)-19300-1	4-(4-4)-20000-4	4-(2-4)-16900-1
TR	MARE	0.1784	0.1826	0.2402	0.1651
	R^2	0.9965	0.9963	0.9936	0.9967
	MRSE	0.1973	0.2014	0.2676	0.1901
TS	MARE	0.2275	0.2263	0.2860	0.2127
	R^2	0.9846	0.9841	0.9817	0.9851
	MRSE	0.4684	0.4768	0.5100	0.4613
VAL	MARE	0.2067	0.2052	0.2565	0.1918
	R^2	0.9949	0.9951	0.9925	0.9952
	MRSE	0.3321	0.3303	0.4049	0.3266
All Data	MARE	0.1864	0.1899	0.2061	0.1889
	R^2	0.9944	0.9942	0.9933	0.9945
	MRSE	0.1676	0.1703	0.1825	0.1651
FINAL STRUCTURE		**3 - 4 - 1**	**4 - 4 - 1**	**4 - 4 - 4**	**4 - 4 - 1**

Table 10-6 Comparison of All the Prediction Models Developed for Database 5, Output 2

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		3-(3-4)-19500-1	4-(3-3)-14100-1	4-(4-5)-20000-3	4-(1-4)-20000-1
TR	MARE	1.2049	1.2010	1.1989	1.2775
	R^2	0.9285	0.9293	0.9299	0.9283
	MRSE	0.6420	0.6391	0.6344	0.6649
TS	MARE	1.0555	1.0360	1.0375	1.1009
	R^2	0.9359	0.9345	0.9354	0.9283
	MRSE	0.8316	0.8311	0.8233	0.8500
VAL	MARE	1.1796	1.1379	1.1397	1.1969
	R^2	0.9379	0.9425	0.9420	0.9420
	MRSE	0.8464	0.8179	0.8142	0.8663
All Data	MARE	1.1251	1.1294	1.1324	1.1477
	R^2	0.9333	0.9329	0.9329	0.9315
	MRSE	0.4255	0.4269	0.4266	0.4321
FINAL STRUCTURE		**3 - 4 - 1**	**4 - 3 - 1**	**4 - 5 - 4**	**4 - 4 - 1**

Table 10-7 Comparison of All the Prediction Models Developed for Database 6, Output 1

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		16-(3-3)-5000-1	17-(1-2)-10100-1	17-(1-7)-20000-17	17-(1-3)-20000-1
TR	MARE	5.2591	2.9933	3.4857	4.8917
	R^2	0.7130	0.9050	0.8766	0.7882
	MRSE	0.0053	0.0029	0.0033	0.0048
TS	MARE	7.3715	4.3274	4.4339	7.4545
	R^2	0.4081	0.7823	0.7841	0.3711
	MRSE	0.0102	0.0062	0.0061	0.0110
VAL	MARE	7.3367	4.5943	4.7644	8.4136
	R^2	0.3851	0.7331	0.7112	0.3944
	MRSE	0.0105	0.0066	0.0068	0.0115
All Data	MARE	5.4159	3.4670	3.8552	4.7679
	R^2	0.6612	0.8561	0.8347	0.7196
	MRSE	0.0038	0.0025	0.0027	0.0035
FINAL STRUCTURE		**16 - 3 - 1**	**17 - 2 - 1**	**17 - 7 - 17**	**17 - 3 - 1**

Table 10-8 Comparison of All the Prediction Models Developed for Database 6, Output 2

Accuracy Measures		Static ANN	Feedback ANN	Auto-associative	Dynamic-Sequential
		16-(1-3)-13000-1	17-(2-3)-15300-1	17-(6-7)-20000-17	17-(1-2)-20000-1
TR	MARE	11.0376	10.7310	17.3192	11.6224
	R^2	0.9202	0.9239	0.7564	0.9173
	MRSE	0.2761	0.2690	0.4821	0.2826
TS	MARE	13.9424	11.6815	17.4655	14.2198
	R^2	0.7554	0.8372	0.7311	0.7403
	MRSE	0.6057	0.5004	0.6290	0.6284
VAL	MARE	19.0559	16.6108	20.1883	21.4662
	R^2	0.4636	0.6032	0.5817	0.3085
	MRSE	0.8268	0.7122	0.7514	1.0644
All Data	MARE	11.5289	11.0989	18.2202	13.8732
	R^2	0.8721	0.8844	0.7397	0.8455
	MRSE	0.2276	0.2162	0.3251	0.2500
FINAL STRUCTURE		**16 - 3 - 1**	**17 - 3 - 1**	**17 - 7 - 17**	**17 - 2 - 1**

Table 10-9 Comparison of All the Prediction Models Developed for Database 6, Output 3

Accuracy Measures		**Static ANN**	**Feedback ANN**	**Auto-associative**	**Dynamic-Sequential**
		16-(1-3)-19400-1	17-(1-3)-3100-1	17-(4-6)-7100-17	17-(1-2)-18100-1
TR	MARE	6.5114	7.1362	12.0262	7.9631
	R^2	0.9066	0.8854	0.6304	0.8561
	MRSE	0.0065	0.0075	0.0131	0.0086
TS	MARE	10.6633	9.2528	12.3217	10.0551
	R^2	0.7678	0.8178	0.6792	0.7535
	MRSE	0.0151	0.0131	0.0171	0.0155
VAL	MARE	13.5152	10.3996	13.4702	13.2073
	R^2	0.5444	0.6862	0.5789	0.5665
	MRSE	0.0201	0.0164	0.0192	0.0203
All Data	MARE	8.0094	7.7491	11.7108	8.7793
	R^2	0.8377	0.8467	0.6419	0.8038
	MRSE	0.0059	0.0057	0.0089	0.0066
FINAL STRUCTURE		**16 - 3 - 1**	**17 - 3 - 1**	**17 - 6 - 17**	**17 - 2 - 1**

Table 10-10 Comparison of All the Prediction Models Developed for Database 7

Accuracy Measures		**Static ANN**	**Feedback ANN**	**Auto-associative**	**Dynamic-Sequential**
		15-(4-7)-7900-1	16-(4-5)-5200-1	16-(7-8)-20000-16	16-(3-6)-20000-1
TR	MARE	12.5600	11.7337	30.2491	14.9102
	R^2	0.9834	0.9850	0.9660	0.9855
	MRSE	1.2149	1.1518	1.8678	1.1866
TS	MARE	14.8755	13.2057	32.9600	16.1394
	R^2	0.9735	0.9780	0.9533	0.9728
	MRSE	2.1657	1.9500	3.1479	2.1826
VAL	MARE	15.0643	11.4591	33.1301	15.1229
	R^2	0.9750	0.9816	0.9397	0.9701
	MRSE	2.0286	1.7721	3.5617	2.2814
All Data	MARE	12.3796	11.5041	31.2433	13.2173
	R^2	0.9831	0.9848	0.9553	0.9834
	MRSE	0.8466	0.8011	1.4805	0.8396
FINAL STRUCTURE		**15 -7 - 1**	**16 - 5 - 1**	**16 - 8 - 16**	**16 - 6 - 1**

Table 10-11 Accuracy of the Recommended Values by HDMS for Database 1

	MARE	MRSE	R^2
Y_{MARE}	3.005	2.753	0.999
Y_{MRSE}	2.500	1.899	0.999
Y_{R^2}	4.556	5.090	0.998

Table 10-12 Accuracy of the Recommended Values by HDMS for Database 2

	MARE	MRSE	R^2
Y_{MARE}	4.6563	0.3713	0.4856
Y_{MRSE}	4.6190	0.3683	0.4876
Y_{R^2}	4.6806	0.3733	0.4886

Table 10-13 Accuracy of the Recommended Values by HDMS for Database 3

	MARE	MRSE	R^2
Y_{MARE}	11.2050	55.8135	0.9478
Y_{MRSE}	11.5240	55.2836	0.9475
Y_{R^2}	11.7784	57.1080	0.9472

Table 10-14 Accuracy of the Recommended Values by HDMS for Database 4

	MARE	MRSE	R^2
Y_{MARE}	16.1337	40.2666	0.8992
Y_{MRSE}	16.3475	40.6301	0.8973
Y_{R^2}	16.7516	41.3247	0.8934

Table 10-15 Accuracy of the Recommended Values by HDMS for Database 5, Output 1

	MARE	MRSE	R^2
Y_{MARE}	0.1787	0.1546	0.9952
Y_{MRSE}	0.1783	0.1537	0.9953
Y_{R^2}	0.1777	0.1529	0.9953

Table 10-16 Accuracy of the Recommended Values by HDMS for Database 5, Output 2

	MARE	MRSE	R^2
Y_{MARE}	1.0755	0.4092	0.9384
Y_{MRSE}	0.9626	0.3691	0.9499
Y_{R^2}	0.9871	0.3776	0.9475

Table 10-17 Accuracy of the Recommended Values by HDMS for Database 6, Output 1

	MARE	MRSE	R^2
Y_{MARE}	4.9452	0.0034	0.7168
Y_{MRSE}	4.9337	0.0034	0.7181
Y_{R^2}	4.9293	0.0034	0.7184

Table 10-18 Accuracy of the Recommended Values by HDMS for Database 6, Output 2

	MARE	MRSE	R^2
Y_{MARE}	44.6121	0.6265	0.6683
Y_{MRSE}	45.8312	0.6433	0.6581
Y_{R^2}	43.2074	0.6085	0.6769

Table 10-19 Accuracy of the Recommended Values by HDMS for Database 6, Output 3

	MARE	MRSE	R^2
Y_{MARE}	13.3865	0.0092	0.7416
Y_{MRSE}	13.1793	0.0091	0.7527
Y_{R^2}	13.1313	0.0091	0.7521

Table 10-20 Accuracy of the Recommended Values by HDMS for Database 7

	MARE	MRSE	R^2
Y_{MARE}	9.1724	0.7439	0.9872
Y_{MRSE}	10.3117	0.7652	0.9864
Y_{R^2}	11.2015	0.7928	0.9855

CHAPTER 11

11. SUMMARY, CONCLUSIONS, and RECOMMENDATIONS

11.1 SUMMARY

In this study, new modeling methodologies for all engineering/scientific prediction systems were proposed and tested on the seven databases described in Chapter 4. These methodologies comprise a new set of ANN modeling approaches along with a new method to replace missing values in datasets. Because ANNs approach is a powerful function approximation computational technique capable of mapping and capturing the relationships within databases, it has been widely used by many researchers. The most widely used ANN approach type is the static ANN network using the backpropagation training algorithm.

Seven databases were utilized to develop each static ANN models at the first stage of this research. As mentioned in Chapter 5, the static ANN models were developed in four sequential stages. In the first stage, the ANN architectures were determined based on problem characteristics. This step also includes classifying the datasets as training, testing or validation sets. In the second stage, the networks were trained and tested on the experimental data to obtain the optimum number of hidden nodes and iterations for the ANN architecture determined in stage one. In the third stage, the best performing networks obtained from the second stage were validated on the validation database. In the fourth stage, the best performing networks obtained in the second stage (with known hidden nodes and training iterations) were retrained on all experimental data to improve the prediction accuracy. Essentially these four sequential stages were repeated for each output of the seven databases.

Then, a new ideology of ANN approach is introduced in Chapter 6 that considers predictions from static ANN networks as initial estimates to develop newer models. These new network models are called Feedback ANN networks because of the feedback input from static ANN network. Similarly, the same static ANN predictions were utilized by the Auto associative network in Chapter 7 to develop new models. In this case, the inputs were also projected at the output layer as well. The input layer consisted of the inputs and the initial estimate from the static ANN network. The output layer similarly had the same number of nodes consisting of number of inputs and the actual output(s). In this case, Auto associative network is optimized on not only output, but also on inputs as well. The dataset predicts itself as well as it predicts the output. This multifunctional nature of this network can be employed in the case of missing data. The idea of providing input reflections is to help the user in justifying the values used to replace the missing data.

Chapter 8 introduced another ANN approach, Dynamic-sequential network that relies on static ANN predictions as well. However, training order of the Dynamic-sequential network is different than other ANN approaches. It still uses the backpropagation algorithm, but the training order of the datasets was changed by replicating each dataset 5 times. Accordingly, each dataset was used in training 5 times during each epoch. This approach helps the network in extracting more knowledge from the available datasets. Chapter 9 presented a new method to handle missing variables in datasets. The new method based on Euclidean distance was utilized for all seven databases. Because of the time restriction, only the output variable was assumed to be missing and assessed accordingly. To compare the results by the Query method, linear regression analysis approach was used for each database.

The last chapter presented a system that integrates all the ANN modeling approaches as well as the Query method application to replace missing values in datasets. Basically, all the developed networks for each database were integrated in one system called hybrid decision making system (HDMS). This system employed the input dataset and the generated predictions by all networks developed in this study. Moreover, the Query method application was also integrated into the same system to tackle missing values in datasets. The output predictions of the Query method were also considered in the final decision. All four ANN approaches and the Query method were integrated together to design a HDMS for each database. Moreover, recommended (weighted) values based on the noted statistical accuracy measures of the developed models involved were calculated and further validated.

11.2 CONCLUSIONS

Based on the results presented in the previous chapters, the notable set of conclusions are listed here:

1. The developed static ANN network models for seven databases produced output values that are very close to the actual values. The statistical accuracy measures indicated that the static ANN models were able to map linear and, most importantly, highly nonlinear processes. In other words, comparison between observed data and static ANN model predictions indicated that the developed ANN model has efficiently characterized the relevant phenomena. Therefore, the developed ANN models can reliably be used for future prediction tasks.
2. The new ANN approach, Feedback ANN network, was introduced and tested on the seven databases successfully. Feedback ANN network approach has improved the statistical accuracy measures of the models developed with static ANN approach. This method has improved not only output prediction accuracy, but also the optimal network architecture by employing less complicated internal structure and training iterations in most cases. Highly nonlinear correlations between inputs and output(s) were nicely captured by the Feedback ANN network approach.
3. The Auto-associative network approach was introduced and verified successfully using civil engineering databases. The prediction accuracy of the Auto-associative network models were based on the predicted outputs only even though the input parameters were predicted as well. The obtained statistical accuracy measures indicate that the output predictions may not be as accurate as those predicted by other ANN approaches but the predictions are still considerably good. However, it is important to mention that these networks were optimized on the outputs and inputs together. So it was expected to have higher errors than other approaches because these networks have to meet additional requirements. This approach can be considered as an identity recognition network that can optimize missing values within the input as well as the output vectors.

4. Dynamic-sequential network is another new ANN approach introduced and validated successfully in this study. By utilizing the best predictions from static ANN network approach, Dynamic-sequential network was able to develop efficient models even though some databases with high linearity did not respond well due to the excellent mapping capability of the static ANN approach. The least prediction errors for several databases were attained when using this type of network. Additional study to verify the stability of the network's predictions has showed that any value used as an initial estimate was ultimately optimized by the Dynamic-sequential network. Once the model is developed and optimized by Dynamic-sequential network approach using the initial estimates from the static ANN network, even misleading entries will not impact the final predictions as long as the values provided for input vectors are valid and within the applicable training range.
5. The Query method, a new method to replace missing values in datasets, aspects were introduced and tested in this study. The prediction accuracy of the Query method application has indicated good agreement between the actual and projected output values. The Query method applications have outperformed the linear regression-based models that were developed for comparison purposes. It can be inferred from the results obtained that the bigger-size the database, the higher the prediction accuracy measures. This is due to the fact theses databases have more neighborhoods that this method can use to query from. The Query method is an easy and multifunctional method that does not require any training like the ANN cases. In this study, an Excel-based application was developed to generate a Query method application for each desired database. With this application, this method can be simply applied to any database. The Excel-based application has an added feature to validate additional datasets. This feature allows the user to validate, and as more datasets become available, as well as update the Query method application to improve its statistical accuracy measures.
6. All the developed models via ANN modeling approaches are integrated into one hybrid system: a system that utilizes all four ANN approaches (i.e., static ANN, Feedback ANN,

Auto-associative, and Dynamic-sequential network) and the Query method application. This system is called Hybrid Decision Making System (HDMS) and provides all the predictions by the developed networks as well as the closest datasets in the database. The developed HDMS excel-based applications, for all seven databases, are user-friendly and can easily be used by anyone not having high level of modeling knowledge. In this case, a missing value within a dataset can nicely be handled by the application. Accordingly, the user is immediately updated with the closest datasets, the missing value is replaced in the cell, and all predictions are generated simultaneously. The multi-functionality and easiness of the Query method makes this method a powerful, rapid, and low cost alternative to logically replace missing values within datasets.

Based on the previously stated conclusions, this study has showed that all new modeling approaches have performed efficiently and their performance is very promising. Even though Feedback ANN, Auto-associative, and Dynamic-sequential networks are dependent on the initial estimates from the static ANN network for both developing the models and utilizing the models to generate outputs, each method still has its own characteristics as the overall evaluation process has indicated. The Query method application is very handy for those who are always dealing with databases with missing values. The results by the Query method are very realistic since the method searches for the most similar datasets within the database. The interface for the HDMS is a user-friendly Excel-based application that can easily be utilized. Accordingly, HDMSs can be a viable solution for many quality management problems by eliminating the unnecessary future experiments and their associated costs.

This research has successfully contributed new modeling methodologies that can be used in all engineering/scientific prediction systems. The new methods supported by reported quantified evidence allow researchers to consider using these computational techniques in their scientific and engineering modeling endeavors.

11.3 RECOMMENDATIONS

Even though the statistical accuracy measures of the ANN models presented in this study are reasonably acceptable, the significance of the developed ANN models is limited to the utilized databases. The significance of the new ANN approaches relies on many factors. This study has shown that the new ANN approaches have performed well on the selected seven engineering databases. However, the accuracy measures for other databases may vary. For this reason, it is recommended to utilize these approaches on other databases with various characteristics. The Auto-associative network approach is very new to civil engineering systems. Accordingly, the verification of this approach needs to be investigated further by including the statistical accuracy measures of the input variables and their optimization. Similarly, it is necessary to broaden the applicability of the Dynamic-sequential network approach by utilizing other databases with different characteristics and sizes. Additionally, the Query method application was only considered to replace outputs. Accordingly, the results for the input parameters can be different. Future studies should look into expanding this research by validating other parameters (i.e., input variables). Query method can also be used to pre-process the incomplete datasets before the ANN models are developed, if there are any, because it is important to include more datasets in the model development stages to obtain statistically significant trends. Future studies should also focus on improving the user interface of the hybrid decision making system by including iterations through the feedback networks and enhancing their graphical presentations.

CHAPTER 12

12. REFERENCES

Ali, H. E. (2000). Neuronet-based liquefaction potential assessment and stress-strain behavior simulation of sandy soils. *PhD dissertation*, Kansas State University, Manhattan, KS.

Ascough, J., Maier, H. R., Ravalico, J. K., & Strudley, M. (2008). Future research challenges for incorporation of uncertainty in environmental and ecological decision-making. *Ecological Modelling, 219*(3), pp. 383-399.

Basheer I. A. (1998). Neuromechanistic-based modeling and simulation of constitutive behavior of fine-grained soils. *PhD dissertation*, Kansas State University, Manhattan, KS.

Basso, A., & Kunt, M. (1992). Auto-associative neural networks for image compression. *European Transactions on Telecommunications and Related Technologies, 3*(6), pp. 593-598.

Bishop, J. M., Mitchell, R. J., & Minchinton, P. R. (1992). Auto-associative digital neural network for grey level data. *International Conference on Artificial Neural Networks, ICANN 92,* Brighton, England. *, Vol. 1,* pp. 639-642.

Cybenko, G. (1989). "Approximation by superposition of a sigmoid function. Mathematics of Control", *Signals and Systems*, 2(4), pp. 303-314.

Daszykowski, M., Walczak, B., & Massart, D. L. (2003). A journey into low-dimensional spaces with Auto-associative neural networks. *Talanta, 59*(6), pp. 1095-1105.

Desjardins, G., Proulx, R., & Godin, R. (2006). An auto-associative neural network for information retrieval. *International Joint Conference on Neural Networks,* Vancouver, BC, Canada, pp. 3492-3498.

Fausett, L. (1994). Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice-Hall, Inc., Upper Saddle River, NJ.

Funahashi, K. I. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks,* 2(3), pp. 183-192.

Ghaboussi, J., Garret, J. H., & Wu, X. (1991). Knowledge-Based modeling of material behavior with neural networks. *Journal of Engineering Mechanics, 117*(1), pp. 132-153.

Ghaboussi, J., J. H. Garrett, J., & Wu, X. (1990). Material modeling with neural networks. *Proceedings of NUMETA 90 - the International Conference on Numerical Methods in Engineering: Theory and Applications,* Swansea, England. *,* pp. 701-717.

Ghaboussi, J. (1994). Neural network based modeling in geomechanics. *1*, pp. 153-153.

Gheyas, I. A., & Smith, L. S. (2009). Reconstruction of cross-sectional missing data using neural networks. In *Engineering applications of neural networks,* pp. 28-34. Springer Berlin Heidelberg.

Gucunski, N., & Krstic, V. (1996). Backcalculation of pavement profiles from spectral-analysis-of-surface-waves test by neural networks using individual receiver spacing approach. *Transportation Research Record,* (1526), pp. 6-13.

Güneyisi, E., Gesoğlu, M., Özturan, T., & Özbay, E. (2009). Estimation of chloride permeability of concretes by empirical modeling: Considering effects of cement type, curing condition and age. *Construction and Building Materials, 23*(1), pp. 469-481.

Hadidi, R., Gucunski, N., & Maher, A. (2007). Application of probabilistic approach to the solution of inverse problems in nondestructive testing and engineering geophysics. *Proceedings of SAGEEP, 1*, pp. 156-165.

Hand, C. (2001). Deep Auto-associative networks. *2001 International Conference on Artificial Intelligence, Las Vegas, NV.* Retrieved from http://hdl.handle.net/2014/12928

Hartman, E. J., Keeler, J. D., and Kowalski, J. M. (1990). Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation*, 2(2), pp. 210-215.

Haykin, S. (1999). Neural networks: *A Comprehensive Foundation*, Second Edition, Prentice Hall, Upper Saddle River, New Jersey.

Herz J., Krogh A., & Palmer R. (1991). *Introduction to the Theory of Neural Computation*, Volume 1, Addison-Wesley Publishing Company, Redwood City, California.

Hines, J. W. (1998). Use of Auto-associative neural networks for signal validation. *Journal of Intelligent Robotic Systems, 21*(2), pp. 143-54.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. National Academy of Science*, USA, Vol. 79, pp. 2554-2558.

Hornik, K., Stinchcombe, M., & White, H. (1989) "Multilayer feedforward networks are universal approximators," *Neural Networks,* 2 (5), pp. 359-366.

Hsu, K., Gupta, H. V., & Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research, 31*(10), 2517-2530.

Khoshhal, A. (2010). Artificial neural network modeling of apple drying process. *Journal of Food Process Engineering, 33*(1), pp. 298-313.

Kramer, M. A. (1992). Auto-associative neural networks. *Computers & Chemical Engineering, 16*(4), pp. 313-328.

Kramer, M. (1991). Nonlinear principal component analysis using Auto-associative neural networks. *AIChE Journal, 37*(2), pp. 233-243.

Krose, B., & Smagt, P. (1993). An introduction to neural networks. Retrieved from http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/torres/livres/ book-neuro-intro.pdf

Lav, A. H., Goktepe, A. B., & Lav, M. A. (2009). Backcalculation of flexible pavements using soft computing. *Intelligent and soft computing in infrastructure systems engineering* Springer Berlin Heidelberg, pp. 67-106.

Levine, D. S. (2002). Neural network modeling. *Stevens' Handbook of Experimental Psychology.*

Little, R. J., & Su, H. (1987). Missing data adjustments for partially scaled variables. *Proceedings of the Section on Survey Research Methods,* pp. 644-649.

Marseguerra, M., & Zoia, A. (2005). The Auto-associative neural network in signal analysis: II. application to on-line monitoring of a simulated BWR component. *Annals of Nuclear Energy, 32*(11), pp. 1207-23.

Meier, R. (1994). Backcalculation of flexible pavement moduli using artificial neural networks. *Transportation Research Record,* (1448), pp. 75-82.

Najjar, Y. M., & S. Mryyan. (2009). ANN-Based Profiling: Data Importance. *Intelligent Engineering Systems through Artificial Neural Networks,* 19, pp. 155-162.

Najjar, Y. M., & Ali, H. E. (1999). Simulating the stress-strain behavior of Nevada sand by ANN. In Proceedings of the 5th US National Congress on Computational Mechanics (USACM), Boulder, Colorado.

Najjar, Y. M., Ali, H. E., & Basheer, I. A. (1999). On the use of neuronets for simulating the stress-strain behavior of soils. *In Numerical Models in Geomechanics, Proceedings of the 7th Int. Symp. on Numerical Models in Geomechanics* Graz, Austria. pp. 657-662.

Najjar, Y. M., & Basheer, I. A. (1996). Stress-strain modeling of sands using artificial neural networks. [Disscussion, ASCE] *Journal of Geotechnical Engineering, 122*(11), pp. 949-951.

Najjar, Y. M., Basheer, I. A., & McReynolds, R. (1996). Neural modeling of Kansas soil swelling. *Transportation Research Record: Journal of the Transportation Research Board, 1526*(1), pp. 14-19.

Najjar, Y. M., Basheer, I. A., & McReynolds, R. L. (1997). Modeling the durability of aggregate used in concrete pavement construction: A neuro-reliability-based approach, Final Report no.: KS-97-3, Kansas Department of Transportation.

Najjar, Y., Huang, C., & Yasarer, H. (2009). Neuronet-based soil chemical stabilization model. *Contemporary Topics in Ground Modification, Problem Soils, and Geo-Support,* pp. 393-400.

Najjar, Y. M., Stokes, R. W., & Russell, E. R. (2000). Setting speed limits on Kansas two-lane highways: Neuronet approach. Transportation Research Record: *Journal of the Transportation Research Board*, 1708(1), pp. 20-27.

Najjar, Y. M., & McReynold, R. (2003). Modeling the Joint Deterioration Behavior of PCC Kansas Pavements via Dynamic ANN Approach, ASCE Geotechnical Special Publication # 123**,** pp. 209-222.

Nazarian, S. (2004). Neural networks for rapid reduction interpretation of spectral analysis of surface waves results. *Transportation Research Record,* (1868), pp. 150-155.

Riding, K. A., Poole, J. L., Folliard, K. J., Juenger, M. C., & Schindler, A. K. (2012). Modeling hydration of cementitious systems. *ACI Materials Journal, 109*(2), pp. 225-234.

Roth, G. (2007). Decision making in systems engineering: The foundation. *Collaborative Technologies and Systems, 2007. CTS 2007. International Symposium on,* pp. 236-246.

Rumelhart, D. E., & McClelland, J. L. (1986). Parallel distributed processing: explorations in the microstructure of cognition. Vol. 1, *MIT Press*, Cambridge, MA.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representation by back-propagating errors. *Nature*, 323(6088), pp. 533-536.

Sohn, H., Allen, D. W., Worden, K., & Farrar, C. R. (2005). Structural damage classification using extreme value statistics. *Journal of dynamic systems, measurement, and control*, 127(1), pp. 125-132.

Tutumluer, E. (1998). Neural network modeling of anisotropic aggregate behavior from repeated load triaxial tests. *Transportation Research Record,* (1615), pp. 86-93.

Wu, X., & Ghaboussi, J. (1995). *Neural network based material modeling. Report* No. SRS-599, Department of Civil Engineering, University of Illinois Urbana-Champaign, IL.

Yasarer, H., & Najjar, Y. (2010). Permeability prediction model for concrete mixes used in Kansas PCC pavements*. Transportation Research Board 91st Annual Meeting,* Report# 12-3457.

Yasarer, H. (2010). Characterizing the permeability of concrete mixes used in transportation applications: A neuronet approach. *MS Thesis*, Kansas State University.

Zupan, J., & Gasteiger, J. (1993). *Neural Networks for Chemists: An Introduction*, VCH publishers, New York, NY.

Najjar, Y. M., & Zhang, X. (2000). Characterizing the 3D Stress-Strain Behavior of Sandy Soils: A Neuro-Mechanistic Approach. *ASCE Geotechnical Special Publications*, No. 96, pp. 43-57.