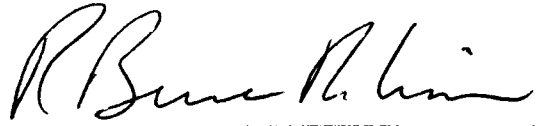


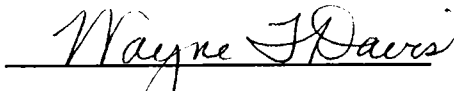
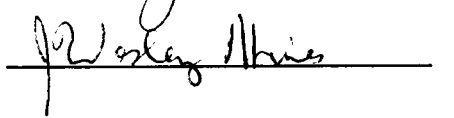
To the Graduate Council,

I am submitting herewith a thesis written by Yeeming Leong entitled "Artificial Neural Network for the Evaluation of Uranium Hexafluoride Cylinders." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Environmental Engineering.



R. Bruce Robinson, Major Professor

We have read this thesis and
recommend its acceptance:

Accepted for the Council:



Associate Vice Chancellor and
Dean of The Graduate School

ARTIFICIAL NEURAL NETWORKS FOR THE EVALUATION OF URANIUM HEXAFLUORIDE CYLINDERS

**A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville**

**Yeeming Leong
December 1997**

DEDICATION

To my lord and savior

Jesus Christ

To my beloved parents

Mr. and Mrs. Leong Chee Hoe

To my Grandma

and

last but not least

my dearest brother

Soo Ming

ACKNOWLEDGEMENT

I am most grateful to my major professor, Dr. R. Bruce Robinson, for providing me with the opportunity, encouragement, technical guidance, constructive criticism, as well as the freedom to work on this research project.

In addition, I would like to thank Dr. Bradford Lyon for providing all the technical guidance and resources needed for the completion of this project.

Furthermore, I would like to extend my appreciation for the technical advice and constructive criticism given by my thesis committee members, Dr. J. Wesley Hines and Dr. Wayne Davis.

Support for this research was provided by the UTK-DOE Educational Program in Environmental Restoration and Waste Management under contract number DE-AC05-84OR21400, subcontract 11B-99732C. Technical support provided by Lockheed-Martin Energy System Inc. is also much appreciated.

Abstract

The United States Department of Energy (DOE) currently manages approximately 50,000 carbon steel cylinders containing more than 1 billion pounds of depleted uranium hexafluoride (UF₆), a byproduct of uranium enrichment activities. These cylinders are currently located at outdoor storage yards, exposed to the atmosphere. Knowledge about these cylinders' minimum wall thickness will help decision-makers in deciding the most feasible method to manage and/or dispose of the cylinders. In this project, neural networks were trained to predict the cylinders' minimum wall thickness and to categorize them according to their wall thickness. In addition, a genetic algorithm/neural network (GA/NN) model was trained to do the same task. The genetic algorithm was used to search for combinations of variables that would provide best neural network training results. Prediction results obtained from neural network training and the GA/NN model were compared to those from multiple linear regression. The data set used in the network training was from the UF₆ Cylinder Location, Inspection, and Measurement System (UCLIM) database. The neural network and GA/NN training results showed that the UCLIM database information was not sufficient to predict the cylinder wall thickness or categorize the cylinders according to their minimum wall thickness.

Executive Summary

The United States Department of Energy (DOE) currently manages in excess of one billion pounds of depleted uranium hexafluoride (UF₆) that is stored in approximately 50,000 carbon steel cylinders. The depleted uranium hexafluoride was a byproduct of uranium enrichment activities carried out by the DOE at three Gaseous Diffusion Plants (GDP) located in Paducah, Kentucky (PGDP), Portsmouth, Ohio (PORTS), and Oak Ridge, Tennessee (K-25).

These carbon steel cylinders are currently located at outdoor storage yards of the three GDP sites where they are constantly exposed to atmospheric conditions. Because of the ongoing exposure to the elements, the cylinders are gradually deteriorating. Decision-makers will decide the ultimate fate of these cylinders, whether they will be reused or disposed of. Knowledge about the cylinders' physical conditions will aid in the judgement and decision-making on whether certain activities such as transportation and lifting are feasible.

This project is a follow-up to the previous work done by Lyon and Stumb (1996), and Monclova-Ortiz (1996) in the C3 Project. The purpose of this project is to make use of existing information to categorize the cylinders' physical conditions. Specifically it is desired to identify cylinders that have a minimum wall thickness less than 250 mils, which is the American National Standards Institute (ANSI)/American Society of Mechanical Engineers (ASME) boiler and pressure vessel code standard. This project uses the information of a central data repository for UF₆ cylinders called the UF₆ Cylinder Location, Inspection, and Measurement System (UCLIM) database.

In their efforts to categorize the physical conditions of the cylinders, Lyon and Stumb (1996) performed standard statistical analyses on the cylinders including linear regression in an attempt to determine any connections between the measured minimum wall thickness data and the UCLIM data set. In their research, each yard was analyzed separately and mixed results were achieved. The percentage of cylinders correctly classified ranged from a high of 100% to a low of 52% across the yards.

Monclova-Ortiz (1996) used an artificial neural network to classify the same data set used by Lyon and Stumb. Using the so-called best model search method, he found the best combination of variables that were used as input variables for the neural network. He reported that it was possible to train a single neural network to classify all the UF₆ cylinders at the different yards at the same time, as opposed to analyzing cylinder conditions in different yards separately.

This project is a further analysis of the possibility of using an artificial neural network to predict the minimum wall thickness and/or to categorize the cylinders in terms of their wall thickness. A new data set with additional information coupled with the existing information from the UCLIM database will be used as input variables. The motivation behind the use of an artificial neural network is because it is capable of handling nonlinear data sets that may pose problems to standard linear regression methods.

The analyses conducted on the data set are listed below:

- The UCLIM data set was coded into two formats. Linear coding was used in the first format in which nonnumeric variables were replaced by numeric values in ascending order. The second data set used the so-called “one-of-N” coding

method for enumerated data that have no natural ranking.

- Multiple linear regression analysis was performed on the 61 encoded variables from the UCLIM database.
- Correlation coefficients between the variables and the minimum wall thickness were computed. Variables with the highest correlation values were retained as input variables for neural network training.
- Four groups of networks were trained. The first two network groups were trained to predict minimum wall thickness. The other two groups of networks were trained to categorize the cylinders into whether they have minimum wall thickness greater or less than 250 mils.
- A GA/NN model was used to predict the minimum wall thickness and to categorize the cylinders' conditions.
- A sorting program written in the *Visual Basic* language was used to compare cylinders with minimum wall thickness less than 250 mils to the rest of the samples. The number of cylinders with wall thickness greater than 250 mils that closely resembled the target cylinder was recorded.

For the prediction of minimum wall thickness, the highest R^2 value was obtained from the MLR analysis using the whole one-of-N data set ($R^2 = 0.4060$). The R^2 value of the MLR analysis using the training set's regression coefficients yielded a lower value (0.292). The best neural network R^2 value was 0.2501 while the best GA/NN model R^2 value was 0.3875.

One possible reason the GA/NN model yielded R^2 values better than those of the neural network training was a better way of selecting the input variables compared to the

method used for the neural network training. The reason the networks yielded lower R^2 values may be because the networks were trained to achieve the minimum root mean squared error, and not to achieve the highest R^2 values. Therefore, comparing the performance of a neural network to the MLR analysis based on R^2 values might put the neural network at a disadvantage.

For the classification of cylinders, again, the highest R^2 value was obtained from the MLR analysis using the whole one-of-N data set ($R^2 = 0.2115$). When only the training set was used to obtain the regression coefficients, the R^2 value from MLR analysis was very close to the values produced by neural network training and the GA/NN model. The best values managed by the MLR analysis, neural network training, and GA/NN model were 0.041, 0.0449, and 0.0482 respectively.

Based on the results gathered from the above mentioned analyses, it was concluded that the 61 variables from the UCLIM database were unable to provide enough information to either deterministically predict the cylinders' minimum wall thickness or to categorize the cylinders based on their minimum wall thickness. This conclusion was supported by the consistent results obtained from the multiple linear regression analysis, neural network training, and the GA/NN modeling. The conclusion was further strengthened by the findings of many cylinders that were almost identical but were classified into two different categories (greater or less than 250 mils). Statistical approaches for estimating the probabilities of cylinder wall thickness may produce more meaningful results but were beyond the scope of this project.

It is clear that the existing information in the UCLIM database was not sufficient to predict the UF₆ cylinders' wall thickness and/or to categorize them according to their

wall thickness. However, it should be noted that the UCLIM database was initially developed as a management tool to record and maintain a history of the UF₆ cylinders. It was not expected to serve as a predictive tool for the categorization of the cylinders. Gathering more data or information may not even help solve the problem. Therefore, it is recommended that the UCLIM database not be used to deterministically predict cylinder minimum wall thickness and/or categorize the cylinders according to their minimum wall thickness. This recommendation only applies when one intends to predict an absolute wall thickness value. It may not apply if it is desired to analyze the cylinders using statistical distribution.

TABLE OF CONTENTS

1 Introduction	1
2 Neural Networks and Genetic Algorithms	5
2.1 Introduction to Neural Networks	5
2.1.1 Neural Network Architecture	7
2.1.1.1 Multi-Layer Perceptrons	9
2.1.1.2 Learning Rule	10
2.1.1.2.1 Backpropagation Algorithm	12
2.1.1.2.2 Levenberg-Marquardt Algorithm	14
2.1.2 Building a Network	16
2.1.3 Data Set Preparation	17
2.1.3.1 Training and Testing Sets Preparation	17
2.1.3.2 Input Data Scaling	18
2.1.4 Network Architecture Selection.....	19
2.1.4.1 Number of hidden neurons and hidden layers	19
2.1.4.2 Transfer Functions.....	23
2.1.4.3 Training Algorithm (Learning Rule)	24
2.2 Introduction to Genetic Algorithms	24
2.2.1 Genetic Algorithm Operators	26
3 Literature Review	30
3.1 Neural Network Applications in Environmental Engineering	30
3.2 Neural Networks for Prediction and/or Classification	33
3.3 Hybrid Neural Network and Genetic Algorithm Systems	35
3.3.1 Supportive Hybrid Systems	35
3.3.1.1 Data Preparation by Genetic Algorithm.....	35
3.3.1.2 Other Applications of Supportive Hybrid Systems	37
4 UCLIM Database Encoding	39
4.1 UCLIM Database	39
4.2 Additional Information for Data Set	40
4.3 Data Encoding	42
4.4 Output Encoding	46

5 Prediction and Classification Results of the UF₆ Cylinders	47
5.1 Prediction by Multiple Linear Regression	47
5.2 Correlation Coefficients of Variables	50
5.3 Neural Network Training	56
5.3.1 MATLAB Training File	56
5.3.2 Neural Network Training Results	60
5.3.2.1 Prediction of Minimum Wall Thickness Using Original Codes	61
5.3.2.2 Classification of Cylinders Using Original Codes	68
5.3.2.3 Prediction of Minimum Wall Thickness Using One-of-N Codes	76
5.3.2.4 Classification of Cylinders Using One-of-N Codes	76
5.4 Genetic Algorithm/Neural Network (GA/NN) Training	87
5.4.1 Prediction of Minimum Wall Thickness	91
5.4.2 Classification of Cylinders	96
5.5 Summary of Results	104
5.6 Data Sorting and Comparison Using <i>Visual Basic</i>	106
6 Conclusions and Recommendations	111
6.1 Summary and Conclusion	111
6.2 Recommendations	113
References	115
Appendices	122
Vita	148

LIST OF TABLES

Table 2.1 Example of String Fitness Values.....	27
Table 3.1 Published Works on Supportive Hybrid Systems.....	38
Table 4.1 Defect Codes Found in the UCLIM Database.....	41
Table 4.2 61 Variables Used as Neural Network Input Variables.....	43
Table 5.1 Identical One-of-N Variables.....	48
Table 5.2 R and R^2 Values for the MLR Analysis.....	48
Table 5.3 Correlation Coefficients Between Variables and a) Actual Wall Thickness; b) Categorized Cylinders.....	51
Table 5.4 Minimum Wall Thickness: Correlation Coefficients Between 21 variables.....	54
Table 5.5 Categorized Cylinders: Correlation Coefficients Between 22 variables.....	55
Table 5.6 Neural Network Training Results for Prediction of Wall Thickness Using Original Codes.....	62
Table 5.7 Neural Network Training Results for Classification of Cylinders Using Original Codes.....	69
Table 5.8 Neural Network Training Results for Prediction of Wall Thickness Using One- of-N Codes.....	77
Table 5.9 Neural Network Training Results for Classification of Cylinders Using One - in-N Codes.....	83
Table 5.10 Training Parameters Used in <i>Predict's</i> GA/NN training (I).....	88
Table 5.11 Training Parameters Used in <i>Predict's</i> GA/NN training (II).....	89
Table 5.12 GA Selected Input Variables and Their Data Transformation (I).....	92
Table 5.13 GA/NN Model Linear Correlation and Error Values (I).....	92
Table 5.14 GA Selected Input Variables and Their Data Transformation (II).....	97
Table 5.15 GA/NN Model Linear Correlation and Error Values (II).....	97
Table 5.16 Summary of Results for Prediction of Minimum Wall Thickness.....	105
Table 5.17 Summary of Results for the Classification of Cylinders.....	105
Table 5.18 Comparison Between Cylinders With Similar Input Variable Values (I).....	108
Table 5.19 Comparison Between Cylinders With Similar Input Variable Values (II).....	109
Table 5.20 Number of Cylinders (>250 mils) with Error < 3.....	110

LIST OF FIGURES

Figure 2.1 Multiple-Input Neuron.....	6
Figure 2.2 Log-Sigmoid Transfer Function.....	8
Figure 2.3 Hyperbolic Tangent Sigmoid Transfer Function.....	8
Figure 2.4 Typical MLP Architecture.....	11
Figure 2.5 A Fitness-based Roulette Wheel	27
Figure 2.6 Example of the Crossover Operation	28
Figure 5.1 MLR Analysis for Minimum Wall Thickness.....	49
Figure 5.2 MLR Analysis for Categorized Cylinders.....	49
Figure 5.3 NN Trial Training Using Perfect Function.....	58
Figure 5.4 Session 1-1.1c: RMSE.....	63
Figure 5.5 Session 1-1.1c: Prediction Results for Training Set.....	63
Figure 5.6 Session 1-1.1c: Training Set Prediction Error.....	64
Figure 5.7 Session 1-1.1c: Prediction Results for Test Set.....	64
Figure 5.8 Session 1-1.1c: Test Set Prediction Error.....	65
Figure 5.9 Session 1-1.1h: RMSE	65
Figure 5.10 Session 1-1.1h: Prediction Results for Training Set.....	66
Figure 5.11 Session 1-1.1h: Training Set Prediction Error.....	66
Figure 5.12 Session 1-1.1h: Prediction Results for Test Set.....	67
Figure 5.13 Session 1-1.1h: Training Set Prediction Error.....	67
Figure 5.14 Session 2-1.2a: RMSE.....	70
Figure 5.15 Session 2-1.2a: Prediction Results for Training Set.....	70
Figure 5.16 Session 2-1.2a: Training Set Prediction Error.....	71
Figure 5.17 Session 2-1.2a: Prediction Results for Test Set.....	71
Figure 5.18 Session 2-1.2a: Test Set Prediction Error.....	72
Figure 5.19 Session 2-3.1a: RMSE.....	73
Figure 5.20 Session 2-3.1a: Prediction Results for Training Set.....	74
Figure 5.21 Session 2-3.1a: Training Set Prediction Error.....	74
Figure 5.22 Session 2-3.1a: Prediction Results for Test Set.....	75
Figure 5.23 Session 2-3.1a: Test Set Prediction Error.....	75
Figure 5.24 Session 3-1.1d: RMSE	78
Figure 5.25 Session 3-1.1d: Prediction Results for Training Set.....	78
Figure 5.26 Session 3-1.1d: Training Set Prediction Error.....	79
Figure 5.27 Session 3-1.1d: Prediction Results for Test Set.....	79
Figure 5.28 Session 3-1.1d: Test Set Prediction Error.....	80
Figure 5.29 Session 3-1.1f: RMSE	80
Figure 5.30 Session 3-1.1f: Prediction Results for Training Set	81
Figure 5.31 Session 2-3.1a: Training Set Prediction Error.....	81
Figure 5.32 Session 2-1.2a: Prediction Results for Test Set.....	82
Figure 5.33 Session 3-1.1f: Test Set Prediction Error	82
Figure 5.34 Session 4-3.1b: RMSE	84
Figure 5.35 Session 4-2.1b: Prediction Results for Training Set.....	85

Figure 5.36 Session 4-3.1b: Training Set Prediction Error.....	85
Figure 5.37 Session 4-3.1b: Prediction Results for Test Set.....	86
Figure 5.38 Session 4-3.1b: Test Set Prediction Error.....	86
Figure 5.39 Session 5-1: Prediction Results for Training Set.....	93
Figure 5.40 Session 5-1: Training Set Prediction Error.....	93
Figure 5.41 Session 5-2: Prediction Results for Training Set.....	94
Figure 5.42 Session 5-2: Training Set Prediction Error.....	94
Figure 5.43 Session 5-3: Prediction Results for Training Set.....	95
Figure 5.44 Session 5-3: Training Set Prediction Error.....	95
Figure 5.45 Session 6-1: Prediction Results for Training Set.....	98
Figure 5.46 Session 6-1: Training Set Prediction Error.....	98
Figure 5.47 Session 6-1: Prediction Results for Test Set.....	99
Figure 5.48 Session 6-1: Test Set Prediction Error.....	99
Figure 5.49 Session 6-2: Prediction Results for Training Set.....	100
Figure 5.50 Session 6-2: Training Set Prediction Error.....	100
Figure 5.51 Session 6-2: Prediction Results for Test Set.....	101
Figure 5.52 Session 6-2: Test Set Prediction Error.....	101
Figure 5.53 Session 6-3: Prediction Results for Training Set.....	102
Figure 5.54 Session 6-3: Training Set Prediction Error.....	102
Figure 5.55 Session 6-3: Prediction Results for Test Set.....	103
Figure 5.56 Session 6-3: Test Set Prediction Error.....	103

1 Introduction

The United States Department of Energy (DOE) currently has in storage depleted uranium hexafluoride (UF_6) in excess of 1 billion pounds (Lyon and Stumb, 1996). The depleted uranium hexafluoride was a byproduct of uranium enrichment activities that were carried out at three Gaseous Diffusion Plants (GDP) located in Kentucky, Ohio, and Tennessee. The Oak Ridge Gaseous Diffusion (K-25) site in Oak Ridge, Tennessee was the first of the three GDP's to start operation in the mid 1940's. In the early 1950's, two other plants were built with one in Paducah, Kentucky (PGDP) and the other in Portsmouth, Ohio (PORTS) (Stumb, 1996).

During the uranium enrichment process, the uranium-235 (U^{235}) isotope's concentration would be increased from its naturally occurring assay (normal weight percent concentration) of approximately 0.711% to higher concentrations that would allow its use in nuclear power reactors and nuclear weapons (Stumb, 1996, Lyon and Stumb, 1996). The depleted UF_6 is stored in about 50,000 10- and 14-ton carbon steel cylinders that are typically 4 feet wide in diameter, 10 to 12 feet long with wall thickness of 5/16 inch (Rao et al., 1994). Many of these cylinders were designed in accordance with the American Society of Mechanical Engineers (ASME) Boiler and Pressure Vessel Code for unfired pressure vessels (Rao et al, 1994).

These cylinders are currently located at outdoor storage yards of the three GDP sites where they are constantly exposed to atmospheric conditions. The cylinders are typically stacked in two-high rows. Cylinders on the bottom row are supported on

contoured wooden saddles while cylinders on the top row rest in the valleys between bottom row cylinders. Because of the ongoing exposure to the elements, many cylinders are facing serious corrosion problems (Rao et al., 1994; Stumb, 1996). Decision-makers are interested in knowing as much about the conditions of these cylinders as possible. Knowledge about the cylinders' physical conditions will aid in the judgement and decision-making on whether certain activities such as transportation and lifting are feasible.

The purpose of this project is to make use of existing information to categorize the cylinders' physical conditions. Specifically it is desired to identify cylinders that have wall thickness less than 250 mils, which is the American National Standards Institute (ANSI)/ASME boiler and pressure vessel code standard. This project uses the information of a central data repository for UF₆ cylinders called the UF₆ Cylinder Location, Inspection, and Measurement System (UCLIM) database. The database serves as "a management tool to record and maintain a history of the DOE-owned UF₆ cylinders" (Lyon and Stumb, 1996).

There has been previous work done in the efforts to categorize the conditions of the cylinders. Lyon and Stumb (1996) performed standard statistical analyses on the cylinders in the first attempt to determine any connections between the measured minimum wall thickness data and the UCLIM data set. In their analysis, they used two different linear regression methods to categorize cylinders into two separate groups – a) cylinders with minimum wall thickness less than 250 mils; and b) cylinders with minimum wall thickness greater than 250 mils. Both regression methods, backward elimination and stepwise regression, resulted in a linear model that used a certain subset

of the input variables. The only difference was how the selection of variable subset was performed by the two methods. In Lyon and Stumb's research, each yard was analyzed separately and mixed results were achieved. The percentage of cylinders correctly classified ranged from a high of 100% to a low of 52% across the yards. However, for the yard with a 100% classification rate, Lyon and Stumb pointed out that the regression model using backward elimination had more variables compared to the data points. Therefore that particular regression fit might not be suitable for other cylinders.

Monclova-Ortiz (1996) used an artificial neural network to classify the same data set used by Lyon and Stumb. Using the so-called best model search method, he found the best combination of variables that were used as input variables for the neural network. He reported that it was possible to train a single neural network to classify all the UF₆ cylinders at the different yards at the same time, as opposed to analyzing cylinder conditions in different yards separately.

This project is a further analysis of the possibility of using an artificial neural network to predict the minimum wall thickness and/or to categorize the cylinders in terms of their wall thickness. A new data set with additional information coupled with the existing information from the UCLIM database will be used as input variables. The motivation behind the use of an artificial neural network is that it is capable of handling a nonlinear data set that may pose problems to standard linear regression methods.

Artificial neural networks are loosely based on biological metaphors. They can be viewed as function approximators that have highly interconnected simple computational devices as network building blocks (Hagan, Demuth, and Beale, 1996). A network will search for the relationship between a set of input variables and the corresponding output

variables by fine-tuning a set of parameters called weights and biases (Swingler, 1996).

For this project, a genetic algorithm will be used to help search for the best combinations of variables that will increase neural network accuracy and training time by reducing the dimensionality of the input variables. Genetic algorithms are search techniques that mimic the behaviors of natural selections and the way biological systems adapt to their environments (Goldberg, 1989; Rogers, Dowla, and Johnson., 1995).

Genetic algorithms are capable of finding an optimal local solution and avoid the need to do an exhaustive global search (Medsker, 1995). This feature is very useful in situations where the optimization problems are extremely complex and the search space is large.

Concepts of neural networks and genetic algorithms will be discussed in Chapter 2. Chapter 3 presents the literature review of three areas related to this project: applications of neural networks in environmental engineering, the use of neural networks as prediction tools and/or classifiers, and applications of hybrid genetic algorithm/neural network (GA/NN) systems. Chapter 4 illustrates the data encoding of the UCLIM data set. Chapter 5 presents the training and testing results of four groups of networks and their comparisons to the multiple linear regression analysis. Finally, Chapter 6 presents the conclusions and recommendations for this project.

2 Neural Networks and Genetic Algorithms

2.1 Introduction to Neural Networks

Artificial neural networks, or simply called neural networks (NN), and genetic algorithms (GA) are both loosely based on biological metaphors. Neural networks are used to model biological neural systems, which are much more complex. Although an artificial neural network has a much simpler structure and far fewer neurons than a human brain's neural system, they do share two important similarities:

- Both rely on highly interconnected simple computational devices that serve as network building blocks;
- The networks' function is determined by the connections between neurons (Hagan, Demuth, and Beale, 1996).

A neural network can be viewed as a function approximator. By fine-tuning a set of parameters known as weights ($w_1, w_2, \dots, w_i$) and biases (b), the network searches for the relationship between a given set of input variables ($p_1, p_2, \dots, p_i$) and output variables ($a_1, a_2, \dots, a_i$) (Blum, 1992; Dowla and Rogers, 1995; Hagan, Demuth, and Beale, 1996; Swingler, 1996). The relationship between a number of inputs and a single neuron is shown in Figure 2.1 (Hagan, Demuth, and Beale, 1996).

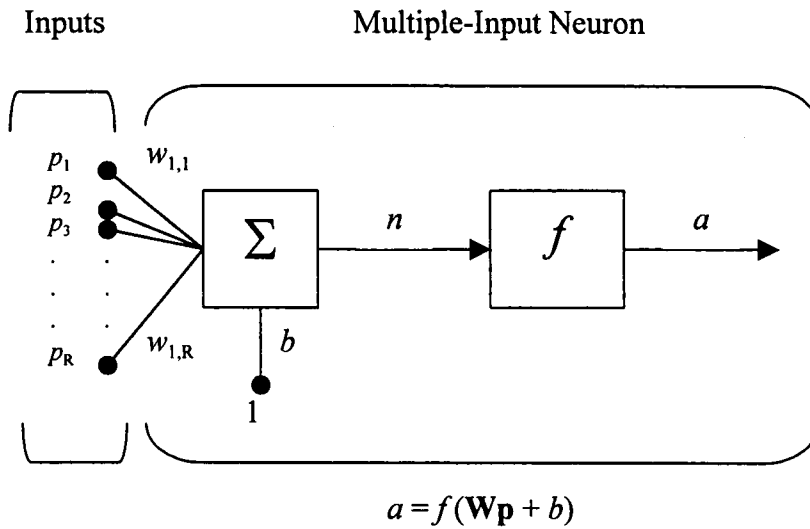


Figure 2.1 Multiple-Input Neuron

The individual inputs $p_1, p_2, \dots, p_i$ are multiplied by the corresponding weights $w_1, w_2, \dots, w_i$ to form $w_i p_i$. The net input n is the summation of the weighted inputs and the bias b .

$$n = w_1 p_1 + w_2 p_2 + \dots + w_i p_i + b \quad (2.1)$$

The inputs and weights can be represented by an input matrix $\mathbf{p}$ and a weight matrix $\mathbf{W}$.

The matrix form of Equation 2.1 is thus:

$$n = \mathbf{W}\mathbf{p} + b \quad (2.2)$$

The relationship between the input and the output is established when the transfer function f produces the scalar neuron output a based on n :

$$a = f(\mathbf{W}\mathbf{p} + b) \quad (2.3)$$

There are several transfer functions commonly used in neural network applications; for example: hard limit, symmetric hard limit, linear, log-sigmoid, and hyperbolic tangent sigmoid (Hagan, Demuth, and Beale, 1996). The main reason multi-layer networks can map nonlinear functions is because of the use of nonlinear transfer

functions. Those that are more popular in nonlinear mapping applications are the log-sigmoid (*logsig*) and the hyperbolic tangent sigmoid (*tansig*) functions. Equation 2.4 and Figure 2.2 show the log-sigmoid transfer function and the relationship between the input and the output.

$$a = \frac{1}{1 + e^{-n}} \quad (2.4)$$

Equation 2.5 and Figure 2.3 show the hyperbolic tangent sigmoid transfer function and the relationship between the input and the output.

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (2.5)$$

The y-axes show the limited output ranges of the two functions with respect to their input ranges. Both functions can take any input value between plus and minus infinity. The log-sigmoid function limits the output to a range between 0 and 1 while the hyperbolic tangent sigmoid function output range is limited to between -1 and 1. Input values outside the range between -2 and 2 will only change the output slightly. Multi-layer networks trained by the back-propagation algorithm commonly use the log-sigmoid function partly because the function is differentiable (Hagan, Demuth, and Beale, 1996).

2.1.1 Neural Network Architecture

As mentioned above, the highly interconnected simple computational devices that serve as the network building blocks are called neurons. The architecture of a neural network refers to the arrangements of neurons within a network. There are three general

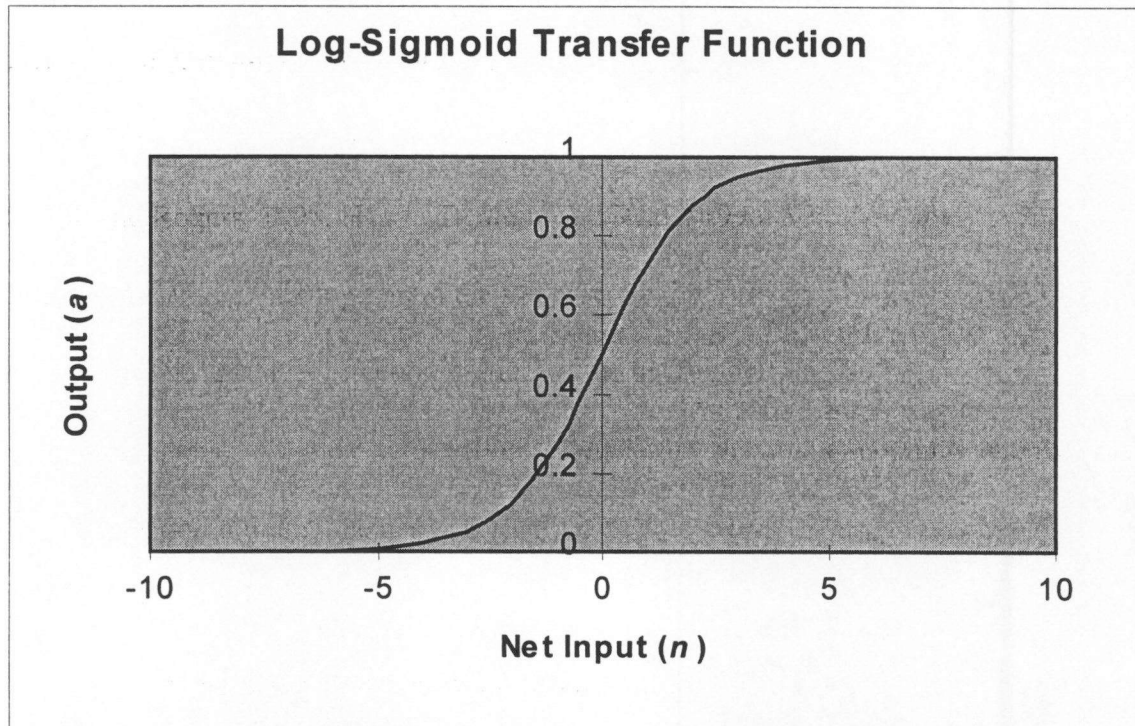


Figure 2.2 Log-Sigmoid Transfer Function

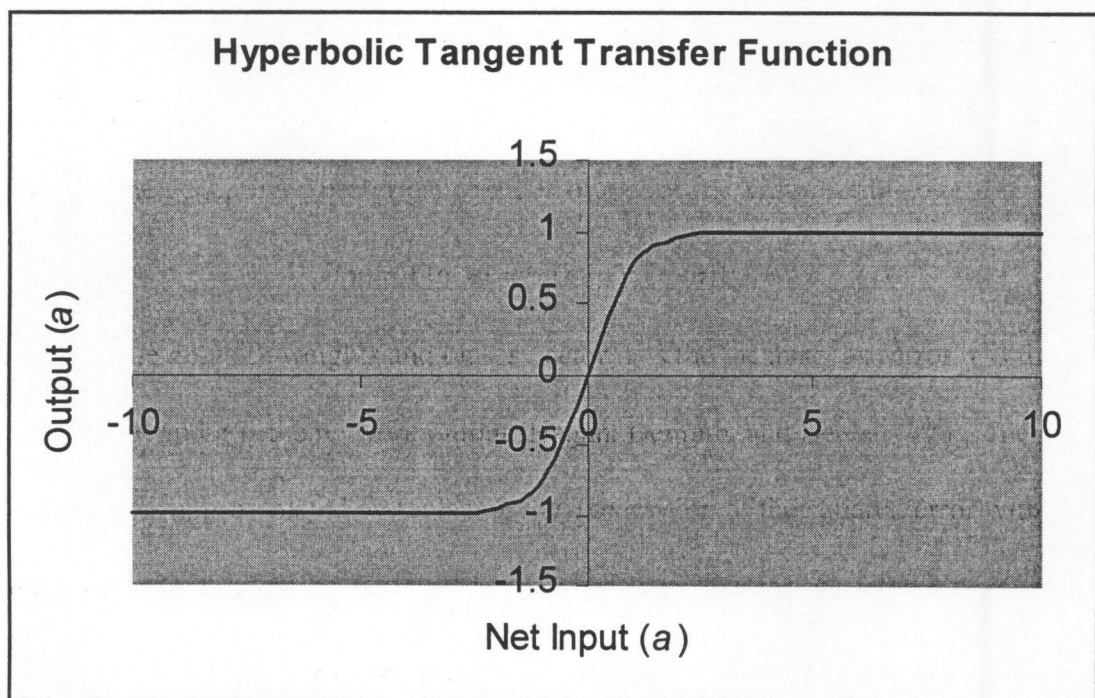


Figure 2.3 Hyperbolic Tangent Sigmoid Transfer Function

categories of network architectures: feed-forward networks, competitive networks, and recurrent associative memory networks (Hagan, Demuth, and Beale, 1996). Examples of feed-forward networks are the simple perceptron and the multi-layer feed-forward network. Competitive networks are characterized by the Hamming network while recurrent networks are represented by the Hopfield network. The multi-layer feed-forward network or the multi-layer perceptron (MLP) with as few as one hidden layer and a nonlinear transfer function has proved to be a universal function approximator (Lee and Lam, 1995; Haykin, 1994). It is the type of network employed in this research; thus it will be discussed in detail in this chapter.

2.1.1.1 Multi-Layer Perceptrons

In MLPs, the output is computed directly from the input with no feedback. The basic architecture of the network usually consists of an input layer, an output layer, and one or more layers called hidden layers connecting the input and output layers. The inputs are processed by neurons in the first hidden layer to obtain outputs that serve as new inputs for subsequent hidden layers. The process continues until the final hidden layer produces output values for the output layer.

Figure 2.4 shows the basic architecture of a typical MLP. The network has two hidden layers as well as an input layer and an output layer. The input layer consists of m input variables, p_i , and each variable is linked to all the hidden neurons in the first hidden layer. The symbol $w_{r,m}$ denotes the weights and a refers to the output. The number of input variables and outputs are usually constrained by the problem that the network is

trained to solve. On the other hand, the numbers of hidden neurons and transfer functions are user-defined. Normally, the same transfer function is used on all hidden neurons in a particular hidden layer. The network will have fewer hidden neurons than input variables when “the hidden layer is required to extract features from the inputs in order to generalize or reduce the dimensionality of the data set”, as shown in Figure 2.4 (Swingler, 1996).

2.1.1.2 Learning Rule

In order to let a network learn to map inputs to outputs, a learning rule or algorithm is implemented. A learning rule is the procedure that a network uses to adjust its weights and biases (Hagan, Demuth, and Beale, 1996). There are three broad categories of learning rules: supervised learning, unsupervised learning, and reinforcement learning (Hagan, Demuth, and Beale, 1996; Masters, 1993).

Unsupervised learning rule modifies weights and biases based on inputs only, with no regards to the outputs. Reinforcement learning is both supervised and unsupervised in the sense that the algorithm is given a network performance grade instead of the specific desired output.

A network employing supervised learning is provided with a training set with examples of inputs to the network and their corresponding target outputs. During each training cycle (i.e. epoch), the weights and biases are modified to move the calculated outputs closer to the target outputs. It usually requires more than one epoch to achieve the mapping of input variables onto the desired outputs. Examples of supervised learning

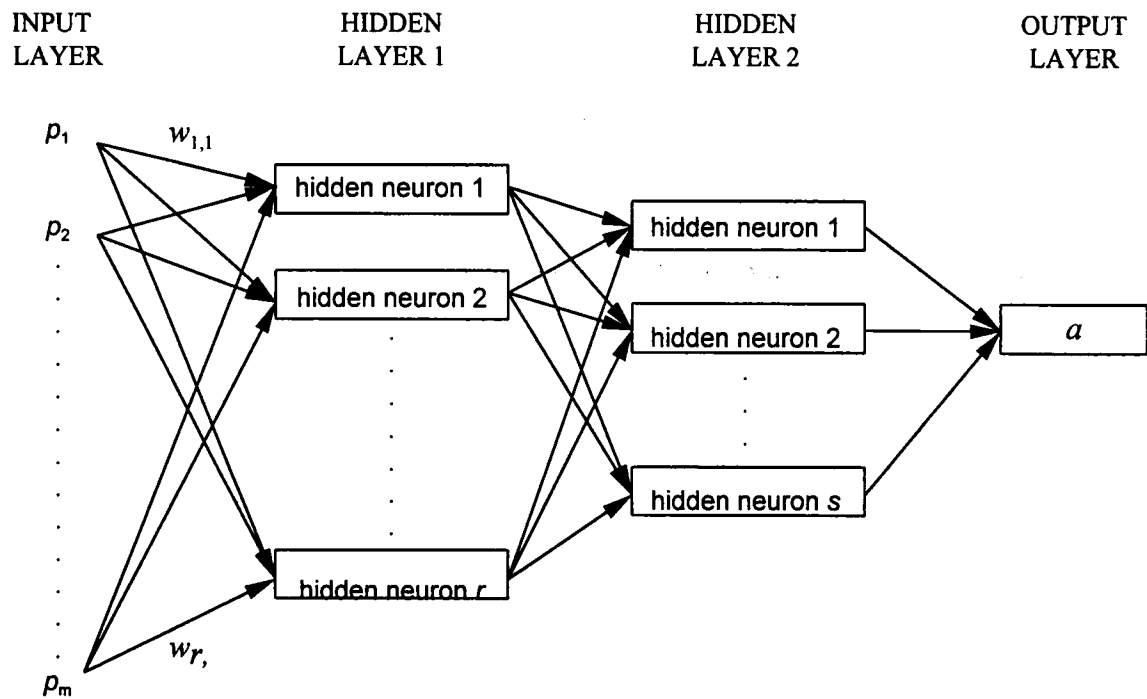


Figure 2.4 Typical MLP Architecture

rules are the simple perceptron, backpropagation, and Levenberg-Marquardt algorithms. The backpropagation algorithm is widely used in multi-layer perceptron training (Blum, 1992; Hagan, Demuth, and Beale, 1996; Medsker, 1995).

A network employing supervised learning is provided with a training set with examples of inputs to the network and their corresponding target outputs. During each training cycle (i.e. epoch), the weights and biases are modified to move the calculated outputs closer to the target outputs. It usually requires more than one epoch to achieve the mapping of input variables onto the desired outputs. Examples of supervised learning rules are the simple perceptron, backpropagation, and Levenberg-Marquardt algorithms. The backpropagation algorithm is widely used in multi-layer perceptron training (Blum,

1992; Hagan, Demuth, and Beale, 1996; Medsker, 1995).

2.1.1.2.1 Backpropagation Algorithm

The backpropagation algorithm is based on gradient descent update method (Dowla and Rogers, 1995; Hagan, Demuth, and Beale, 1996; Masters, 1993). A multi-variate function's gradient refers to the steepest "uphill" direction (Masters, 1993). A small step in the gradient direction produces "the maximum increase in the function" while a tiny step in the opposite direction results in the function's maximum decrease. In backpropagation training, the network training error is the multi-variate function. The algorithm computes the error function's gradient and lets the function move in the direction of the negative gradient until it reaches the minimum error.

The mean square error is a commonly used quantitative performance measure, called the performance index, of networks trained by backpropagation. The backpropagation algorithm, whereby derivatives are processed from the network's last layer to the first, adjusts the weights and biases to reduce the mean square error. Equation 2.6 shows the generalized performance index of a network with multiple outputs:

$$F(\mathbf{x}) = E[\mathbf{e}^T \mathbf{e}] = E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})] \quad (2.6)$$

where $\mathbf{x}$ is the network weights and biases vector, $\mathbf{e}$ is the mean square error, $\mathbf{t}$ is the target output, and $\mathbf{a}$ is the network output (Hagan, Demuth, and Beale, 1996). The mean square error is approximated by replacing the expectation of the squared error with the squared error at iteration k :

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k) \quad (2.7)$$

Equations 2.8 and 2.9 show the basic weight and bias update rule for the steepest descent algorithm:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m} \quad (2.8)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m} \quad (2.9)$$

where superscript m is the network layer, subscripts i and j are the i -th neuron and j -th input of layer m , and α is the learning rate. Using the chain rule, the derivatives in Equations 2.8 and 2.9 are simplified:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a^{m-1} \quad (2.10)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \quad (2.11)$$

In matrix form, the weight update rule using approximate steepest descent becomes:

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (2.12)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad (2.13)$$

Backpropagation has a few limitations. First of all, choosing the right learning rate for a nonlinear network is a difficult task (Demuth and Beale, 1994). In order to enable stable learning, a small learning rate has to be used. If the learning rate is too large, there is a danger of the network repeatedly overstepping the error minimum without converging. In other words, the network will always move in the direction of the error minimum. However, the large steps caused by a large learning rate will make the network

miss the error minimum. On the other hand, if the learning rate is too small, the network may take an extremely long time to train (Masters, 1993).

Furthermore, the algorithm does not guarantee convergence to the global minimum (Swingler, 1996). Demuth and Beale (1994) noted that so far algorithms that avoid all local minima have yet to be created. They suggested that a reasonable approach is to provide the network with higher degrees of freedom to ensure that there will be local minima with sufficiently low error values. One can add more layers and/or neurons to increase the degrees of freedom. However, this may lead to poor generalization by the network.

The problems of convergence to the global minima aside, there are several ways to improve the backpropagation algorithm. The two general categories to improve the algorithm are the heuristic techniques, which include the use of a variable learning rate and momentum, and the standard numerical optimization techniques, which include the conjugate gradient algorithm and the Levenberg-Marquardt algorithm (Hagan and Menhaj, 1994; Hagan, Demuth, and Beale, 1996).

2.1.1.2.2 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm is a modification of Newton's method. It is designed to minimize functions that are sums of squares of other nonlinear functions (Hagan, Demuth, and Beale, 1996). The algorithm approximates the steepest descent algorithm when it is far from the error minimum and becomes the Gauss-Newton method when near the error minimum (Demuth and Beale, 1994). In order to minimize the error function $F(\mathbf{x})$ with respect to the parameter vector $\mathbf{x}$, Newton's method would be:

$$\Delta \mathbf{x} = -[\nabla^2 F(\mathbf{x})]^{-1} \nabla F(\mathbf{x}) \quad (2.14)$$

where $\nabla^2 F(\mathbf{x})$ is the Hessian matrix and $\nabla F(\mathbf{x})$ is the gradient. If $F(\mathbf{x})$ is a sum of squares function:

$$F(\mathbf{x}) = \sum_{i=1}^N e_i^2(\mathbf{x}) = \mathbf{e}^T(\mathbf{x}) \mathbf{e}(\mathbf{x}) \quad (2.15)$$

then the gradient written in matrix form is:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x}) \mathbf{e}(\mathbf{x}) \quad (2.16)$$

while the Hessian matrix form is:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \quad (2.17)$$

where $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{x})}{\partial x_1} & \frac{\partial e_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial e_2(\mathbf{x})}{\partial x_1} & \frac{\partial e_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\mathbf{x})}{\partial x_1} & \frac{\partial e_N(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (2.18)$$

and $\mathbf{S}(\mathbf{x})$ is:

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N e_i(\mathbf{x}) \nabla^2 e_i(\mathbf{x}) \quad (2.19)$$

Assuming $\mathbf{S}(\mathbf{x}) \approx 0$, the Hessian matrix can be approximated as:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) \quad (2.20)$$

Substituting Equations 2.16 and 2.20 to Equation 2.14, the Gauss-Newton method is found:

$$\Delta \mathbf{x} = -[\mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x})]^{-1} \mathbf{J}^T(\mathbf{x}) \mathbf{e}(\mathbf{x}) \quad (2.21)$$

The Levenberg-Marquardt algorithm is obtained by modifying Equation 2.21:

$$\Delta \mathbf{x} = -[\mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) + \mu \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}) \mathbf{e}(\mathbf{x}) \quad (2.22)$$

where μ is a scalar parameter. When a step increases the function $F(\mathbf{x})$, μ is multiplied by some factor $\nu > 1$; the algorithm approaches steepest descent when μ is large. If $F(\mathbf{x})$ decreases, μ is divided by ν and the algorithm becomes Gauss-Newton (Hagan and Menhaj, 1994; Hagan, Demuth, and Beale, 1996). Since the Gauss-Newton method is faster and more accurate near an error minimum, the algorithm aims to shift towards Gauss-Newton as soon as possible. Therefore, μ decreases with each successful step and only increases when the error increases (Demuth and Beale, 1994).

The major drawback of the Levenberg-Marquardt algorithm is the large memory required to store the approximate Hessian matrix $\mathbf{J}^T \mathbf{J}$. The matrix has a size of $n \times n$, “where n is the number of parameters (weights and biases) in the network”; typically, a few thousand parameters is the upper limit for n (Hagan, Demuth, and Beale, 1996). Networks with a few hundred weights have been found to be more efficient using the Levenberg-Marquardt algorithm to the conjugate gradient method (Hagan and Menhaj, 1994). Because of its fast convergence speed compared to the standard backpropagation algorithm, the Levenberg-Marquardt algorithm is used in this research.

2.1.2 Building a Network

In order to build a neural network to perform certain tasks such as function approximations and classifications, the network needs to be trained and then tested. During training, the network learns to perform the desired tasks by employing certain

learning rules or training algorithms. Training stops when the error, i.e., the difference between the network outputs and the target outputs, is deemed acceptable by the user. The network is then tested by a test set to see if it could correctly predict the output. A useful network must be able to produce the correct output vector for an input vector that it has not been trained on (Demuth and Beale, 1994; Dowla and Rogers, 1995). This ability is called generalization.

2.1.3 Data Set Preparation

Prior to training, preparation of the data set has to be carried out. This procedure is very important because the data set will determine how well a network will perform. Data preparation consists of two main components:

- Training and testing sets preparation
- Input data scaling

2.1.3.1 Training and Testing Sets Preparation

The preparation of data sets includes selecting training and testing sets that are representative of the population and then encoding the data into a format that the network recognizes and can train on. In order for the network to generalize well, the training set should have a wide distribution of values that includes the maximum and minimum input and output values. Poor generalization will result if input variables outside the training data range are presented to the network.

2.1.3.2 Input Data Scaling

The scaling of input data normalizes the values of the input variables to fit within a common fixed range. Scaling is important because many times a multi-variable data set has variables that cover significantly different ranges. For example, one variable may range from 10 to 1000 while another from 0.02 to 0.05. The former will have a greater training effect on the network because its error has a greater magnitude (Swingler, 1996). Scaling will equalize the importance of the variables (Masters, 1993).

Another reason for input data scaling is related to the nonlinear transfer functions. As shown in Figures 2.2 and 2.3, the log-sigmoid and hyperbolic tangent sigmoid functions have a limited useful input range. The gradients or slopes of the functions decrease sharply when the input falls outside the range approximately between 2 and -2. Training speed is directly related to the gradient; with small gradients, the update of weights and biases slow down significantly. Furthermore, premature saturation may occur. Premature saturation is a situation whereby training stops prematurely when large internal activations caused by large input magnitudes are mapped to small gradients in the log-sigmoid and hyperbolic tangent sigmoid functions.. Scaling the input is a good way to avoid premature saturation and to increase training speed. Two scaling methods that are commonly used in neural network applications are the linear scaling and the Z-score (mean-centered-unit-variance scaling) (Swingler, 1996; Masters, 1993). Linear scaling normally scales inputs to the range between 0.1 and 0.9. Its use is recommended when the input data spreads over its range evenly without the presence of outliers (Swingler, 1996).

The Z-score scaling method may be used when the variables have an approximately normal distribution (Masters, 1993). Equation 2.23 shows how a variable's *Z-score* is calculated:

$$Z = \frac{x - \mu}{\sigma} \quad (2.23)$$

All variables are centered on the mean and scaled to a unit variance.

2.1.4 Network Architecture Selection

An important procedure during network training is the selection of network architecture. The neural network design process has an inherent nature of trial-and error. The user usually has to choose the initial network architecture although there is a multitude of options. Numerous methods have been proposed to help the user get started. If training is not successful or the result from testing the network is not satisfactory, the user needs to modify the network architecture until the network performs the desired mapping. Choosing the network architecture involves selecting the:

- Number of hidden layers and hidden neurons
- Transfer functions
- Training algorithm (learning rule)

2.1.4.1 Number of hidden neurons and hidden layers

According to Swingler (1996), there is usually a trade-off between accuracy and generalization ability when one tries to optimize the number of hidden neurons. With more hidden neurons, the network will have higher degrees of freedom and will usually

train well. However, the problem of overfitting may arise when too many hidden neurons are used. Overfitting happens when the network is equipped with excessive information processing ability (too many weights) that it will start learning irrelevant details of individual inputs rather than the data set's basic structure (Masters, 1993; Dowl and Rogers, 1995). The network memorizes the training data perfectly but fails to generalize when provided with new test data.

There is not a single universal way to determine the number of hidden neurons needed by a network (Masters, 1993; Swingler, 1995; Sarle, 1997). Researchers have proposed various methods to estimate the number of hidden neurons based on a wide variety of approaches. Nevertheless, there is always a trial-and-error factor involved in these proposed methods. In order to ensure the ability of a network to generalize when faced with new data, the network should have fewer parameters than data points in the training (Dowl and Rogers, 1995; Hagan, Demuth, and Beale, 1996). Onoda (1995) suggested a statistical approach that optimizes hidden neurons by analyzing the learning error and the generalization error of the network. Li, Chow, and Yu (1995) proposed an estimation theory to determine the sufficient number of hidden neurons for higher-order feed-forward networks.

Chow and Teeter (1994) found that for XOR and AND logic problems, using the weight decay methodology can consistently minimize the number of hidden neurons in back-propagation algorithms that use the hyperbolic tangent transfer function. By adding a weight decay term, weights that do not contribute to the input/output mapping process will decay to zero. Engelbrecht and Cloete (1996) used a pruning algorithm based on

sensitivity analysis to prune a trained feed-forward network's input and hidden layers to reduce the number of neurons.

According to Swingler (1996), Baum and Haussler presented a simple estimation method based on the number of weights (w). Assuming a lower error limit e in which $0 < e \leq 1/8$, the number of hidden neurons required (h) is approximated by

$$h \geq \left(\frac{w}{e} \right) \quad (2.24)$$

Swingler (1996) also suggested that the number of hidden neurons (h) should never be greater than twice the number of input neurons:

$$h \leq 2i + 1 \quad (2.25)$$

Masters (1993) proposed a general guideline called the geometric pyramid rule that states that most network architectures resemble the shape of a pyramid. The number of neurons gradually decreases from the input layer to the output layer by following a geometric progression. Therefore, the number of hidden neurons h in a one-hidden-layer network with m inputs and n outputs is:

$$h = \sqrt{mn} \quad (2.26)$$

For a two-hidden-layer network, the number of neurons in each layer is approximated by:

$$r = \sqrt[3]{\frac{n}{m}} \quad (2.27)$$

$$h_1 = mr^2 \quad (2.28)$$

$$h_2 = mr \quad (2.29)$$

in which h_1 and h_2 represent the number of neurons in the first and second hidden layers

respectively. Masters cautioned that these approximations only apply in situations that have many inputs and few outputs. The equations may underestimate the required number of hidden neurons when the problem is highly complex and there are very few inputs and outputs. Conversely, the number of neurons may be overestimated if the network is dealing with a simple problem with many inputs and outputs.

In many cases, the techniques proposed by researchers are either limited to specific situations or they are too general and only serve best as a starting point when choosing the number of hidden neurons. Sarle (1997) noted that choosing the network architecture simply based on the number of inputs and outputs is totally unreliable; the network topology is largely affected by the degree of noise and the complexity of the function the network intends to learn.

The best approach according to Masters (1993) would be a time-consuming trial-and-error method that Engelbrecht and Cloete (1996) refer to as the network construction algorithm. He suggested starting training and testing with the least number of neurons and recording the network performance. The user could start with two hidden neurons if a good guess of minimum number of neurons is impossible. Meanwhile, the output errors can serve as the performance criteria. Then, the number of hidden neurons is increased slightly and the network is trained and tested again. The procedure is repeated until the error is deemed acceptable or when no significant improvement is gained.

Sarle (1997) had pointed out that the best way to avoid overfitting is to have ample training data. Since this is sometimes impossible or too time-consuming, the user should always use the minimum required number of hidden neurons in order to avoid overfitting.

One common method that is used to alleviate overfitting is cross-validation (Murray, 1995; Sarle, 1997). Cross-validation works by testing the network immediately after each training epoch (cycle) with a separate test set. As training proceeds, the errors of both the training and validation sets will decrease. The error of the validation set will reach a minimum and start to increase when training becomes excessive and the network starts to overfit. Stopping training at this point will prevent overfitting (Haykin, 1994).

Similar to determining the number of hidden neurons, there is no universal rule in deciding the number of hidden layers. Funahashi (1989) proved that the one-hidden-layer feed-forward network is a universal function approximator. However, Chester (1990) pointed out that “Funahashi assume[d] that the function to be approximated is given and that an unlimited number of neurons are available”, assumptions that are impractical in real life applications. Chester added that the neurons in a one-hidden-layer network interact with each other globally; therefore, improving an approximation at one point without worsening another is difficult. With two hidden layers, the neurons will have an easier time adjusting weights and biases for better approximation performances. In summary, two hidden layers are recommended if only a few neurons are needed in each layer compared to having thirty or forty neurons in a single-hidden-layer network.

2.1.4.2 Transfer Functions

As mentioned earlier, the log-sigmoid (*logsig*) and the hyperbolic tangent sigmoid (*tansig*) functions are more commonly used in nonlinear mapping applications. Either transfer function can be used in the hidden layer as long as the nonlinear function is well

approximated. The transfer function for the output layer usually depends on the desired output range (Sarle, 1997). The use of the identity or linear function is sufficed if the target outputs do not have bounded range. If target outputs are bounded, bounded transfer functions such as the log-sigmoid and hyperbolic tangent sigmoid functions are recommended. The log-sigmoid function should be used when the desired output ranges from 0 to 1 while the hyperbolic tangent function works when the output ranges from -1 to 1.

2.1.4.3 Training Algorithm (Learning Rule)

As mentioned in Section 2.2.2.2, the Levenberg-Marquardt algorithm is used in this research because it converges much faster compared to the standard backpropagation algorithm. When it is far from the error minimum, the algorithm utilizes the gradient descent convergence. It becomes the rapid Gauss-Newton method when near the error minimum.

2.2 Introduction to Genetic Algorithms

Genetic algorithms are searching techniques that mimic the behaviors of natural selections and the way biological systems adapt to their environments (Goldberg, 1989; Medsker, 1995; Rogers, Dowla, and Johnson, 1995). Genetic algorithms have four distinctive characteristics compared to other more conventional optimization and search techniques:

- Genetic algorithms operate on a coding of the data set rather than the data

themselves;

- Genetic algorithms work with populations of patterns rather than a single pattern;
- Genetic algorithms use objective function information and not derivatives;
- Genetic algorithms are based on probabilistic transition rules and not deterministic rules (Goldberg, 1989).

First of all, the data set has to be coded as a bit string of fixed length, called the parameter string, over some finite alphabet (Goldberg, 1989; Holland, 1992). Secondly, compared to other optimization methods that move from a single point in the search space to another using certain transition rules, genetic algorithms start with a population of strings and continue to generate new populations of strings in order to search for the solution. Third, genetic algorithms search for better string structures based on payoff (or objective function; or fitness) values associated with individual strings. Strings that produce higher pay-offs are favored. This is different from other optimization methods that use auxiliary information to guide their search (for example, gradient techniques such as the backpropagation algorithm use derivatives). Finally, instead of using deterministic methods, genetic algorithms make use of probabilistic transition rules in their search for solutions (Bäck, 1995).

The coded *strings* of genetic algorithms are similar to *chromosomes* in biological systems. Each coded bit string can be viewed as a possible solution to the problem. Chromosomes are made up of *genes*, which may have a certain value called *allele*;

strings, on the other hand, consist of *features* or *detectors* that take on certain *values*. The total genetic package called *genotype* in natural systems is analogous to the total string package called *structure* in the artificial genetic algorithms. In biological systems, the organism or individual created through the interaction between the genotype and the environment is called the *phenotype*. In artificial genetic algorithms, structures are decoded to yield a particular *point* in the solution space (Goldberg, 1989).

2.2.1 Genetic Algorithm Operators

Genetic algorithms produce new generations of strings using operators that mimic natural selections. Three of the most basic genetic algorithm operators are:

- Reproduction
- Crossover
- Mutation (Rogers, Dowla, and Johnson, 1995; Holland, 1992; Bäck, 1996)

The reproduction operator ranks strings according to their objective function values or the fitness scores (Goldberg, 1989; Rogers, Dowla, and Johnson, 1995). Strings that are highly fit have greater chances of contributing to one or more offsprings in the next generation. A classical way of implementing the reproduction operator is by using the biased roulette wheel (Goldberg, 1989; Whitley, 1989). Figure 2.5 shows a roulette wheel in which each string in the population has a slot that is sized according to its relative fitness shown in Table 2.1.

Table 2.1 Example of String Fitness Values

String	Rank	Fitness
A	1	40
B	2	30
C	3	15
D	4	10
E	5	5

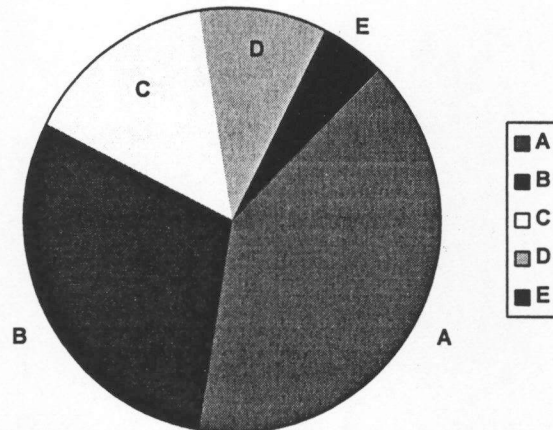


Figure 2.5 A Fitness-based Roulette Wheel

Every time a new generation is produced, strings with higher fitness will have a greater probability of being selected for reproduction. The reproduction operator will create an exact duplicate of a string that is selected and enter it into a mating pool to undergo further genetic operations.

The crossover operator oversees the sexual production of an offspring by crossing two parents. There are two common ways of performing crossover. First, the strings that are entered in the mating pool are mated at random (Goldberg, 1989). The second way is by randomly selecting a position along the parent strings and cutting the strings at the same point. Two offsprings are created by swapping the front parts of the parent strings while the rear portions remain intact. Figure 2.6 illustrates this crossover method:

Parent A: 001001 | 11001111

Parent B: 111010 | 01010101

Offspring 1: 111010 11001111

Offspring 2: 001001 01010101

Figure 2.6 Example of the Crossover Operation

The randomly selected crossover point is independent of the actual information carried by the bits. A single parameter can be coded in more than one bit; a crossover may actually cut the bits into two separate parts and produce a new value different from either parent.

Both the reproduction and crossover operate on the higher-payoff or the “target” regions of the solution space (Holland, 1992). Successive generations of offsprings produced by the operators will increasingly dominate the target regions. While reproduction produces offsprings reminiscent to their parents, the crossover operation serves to test new parts of target regions instead of testing the same string repeatedly in successive generations (Holland, 1992).

The mutation operator protects the mating pool from being dominated by highly fit strings. Since reproduction and crossover favor strings with high fitness, the mating pool faces the danger of losing certain less promising but potentially useful genetic information (Rogers, Dowla, and Johnson, 1995). Mutation in genetic algorithms is carried out by simply altering a string position’s value from 0 to 1 and vice versa. Goldberg (1989) pointed out that mutation is a secondary operator and a mutation rate of

one per every thousand bits (positions) is a common value.

3 Literature Review

This chapter focuses on three main areas that are related with the scopes of this study; they are:

- Neural network applications in environmental engineering;
- Neural networks used as prediction and/or classification tools;
- Hybrid neural network and genetic algorithm (GA/NN) systems.

3.1 Neural Network Applications in Environmental Engineering

Currently, many of the applications of neural networks in the field of environmental engineering have focused on the area of hydrology and hydraulics (Garrett, Ranjithan, and Eheart, 1992; Buch, Mazumdar, and Pandey, 1993; Karunanithi et al., 1994; Grubert, 1995; Rogers, Dowla, and Johnson, 1995). Other possible environmental-related applications are decision-making on environmental policies, clean-up and funding levels for the remedial investigation/feasibility study (RI/FS), and initial clean-up stage and remediation methods for Superfund sites (Schmuller, 1990).

Garrett, Ranjithan, and Eheart (1992) used a feed-forward network to predict the level of costs to treat a contaminated underground plume. A standard random field generator called COVAR was initially used to generate a large set of transmissivity fields. Using extraction well pumping rate as the direct measure of cost, the network was then trained to learn the relationship between the spatially distributed transmissivity fields and the levels of cost. Costly pumping patterns would then be used in the design

aspects of a groundwater management model that would work even with uncertainty in the hydraulic conductivity parameter. A reliable design with reduced computational effort would be ensured with the use of the costly pumping patterns.

Rogers, Dowla, and Johnson (1995) went a step further by using a hybrid neural network and genetic algorithm system to find cost-effective groundwater remediation techniques for a Superfund site. Initially, multi-layer perceptrons were trained to predict three outcomes in which administrators are interested: a) whether the contamination would be contained by a certain pumping pattern; b) the total mass of VOCs that a pumping pattern could extract over fifty years; and c) how much a certain pattern would cost. The genetic algorithm, on the other hand, was used to search for optimal patterns that satisfied all three components, i.e., containing the contamination by extracting the most VOC mass with the least costs. The genetic algorithm search was based on a fitness function defined as the equally-weighted sum of the three outcomes. The genetic algorithm's task was to evolve and locate patterns with the highest fitness functions.

Krovvidy and Wee (1990) applied neural networks in another environmental engineering area: wastewater treatment system. They used a Hopfield network to aid in the design of a wastewater treatment system that would meet various treatment objectives. In the initial design phase, namely the analysis phase, they used an inductive learning algorithm (ID3) to extract knowledge rules from a database containing the treatability information of various compounds through different treatment processes. The extracted knowledge rules were represented by unit process descriptions that would later be used to generate treatment options and sequences in the synthesis phase. In the synthesis phase, a Hopfield network was used to select the sequence of treatment

methods that would achieve the target concentration levels based on the input concentrations of several compounds.

Schmuller (1990) suggested the use of neural networks in the decision-making process of the remediation of hazardous waste sites. He cited the abundant data available about these sites and the inherent information in the data as the primary reasons that neural networks are favored. He suggested using the network input layer to represent a waste site's characteristics such as volume of contamination, type of contaminants, and the contaminated media. Executive decisions on the methods and level of clean-up would be used as the network outputs.

Grubert (1995) used feed-forward backpropagation networks to predict flow conditions that occurred during interfacial mixing in stratified estuaries. Initially, he trained a network to separate stable flows from unstable ones by using boundary friction factor and densimetric Froude number as inputs. Based on the experience and results from the first training, he trained two more networks that were more complex. One would differentiate stable thermal flows from unstable thermal flows while the other would separate stable saline flows from unstable ones.

Karunanithi et al. (1994) trained a neural network based on cascade-correlation algorithm to predict flows at an ungaged site on the Huron River, Michigan. The network's prediction results were compared to that of the most commonly used analytic nonlinear power model. The comparison showed that the network was able to adapt its complexity to match changes in flow history and develop a much more complex model to handle varying flow data compared to the nonlinear power model.

Buch, Mazumdar, and Pandey (1993) used a one-hidden-layer feed-forward

network to predict glacier melt runoffs in a Himalayan valley compared its prediction results to another more conventional method, the Energy-Balance Model. The results obtained from the neural network prediction showed an exceptional performance by the network (a correlation coefficient of 0.9998 between the measured and simulated runoff values). However, the network is suspected to have suffered the problem of excessive degree of freedom since thirty hidden neurons were used to predict about twenty-five input patterns. Furthermore, the researchers did not indicate the use of any verification set to guard against overfitting. Nevertheless, this experiment shows that the application of neural network in environmental engineering is only limited by our imagination.

3.2 Neural Networks for Prediction and/or Classification

Neural networks have been used as prediction tools and/or classifiers in various fields including engineering, sciences, manufacturing, and businesses (Deng and Haykin, 1990; Ramani et al., 1990; Shyu, Libert, and Mann, 1990; Kessler and Kessler, 1995; Smith and Longstaff, 1995). Selected examples are presented below:

Kessler and Kessler (1995) utilized a Kohonen self-organization map and an interpolation method to predict the thickness of transparent surface layers on metals even in the range below 100 nm. Furthermore, by combining the use of diffuse reflectance spectroscopy and a Kohonen map, they were able to predict the corrosion behavior of carbon steel used for car production.

In an attempt to classify textures, Smith and Longstaff (1995) mapped a binary input vector to a binary output vector using a single layer feed-forward network. Each output node represented a cluster in the input vectors' probability distribution function.

The nodes were trained in order rather than simultaneously, which was similar to the hidden node training of Fahlman's Cascade Correlation algorithm.

Deng and Haykin (1990) used a multi-layer neural network to distinguish between types of signals detected by radar system. A modified version of the backpropagation algorithm that combined the learning rate adaptation method with the gradient reuse method was used as the neural classifier. The network achieved close to 90% success rate in classifying the target aircraft, weather, bird, and ground. The researchers pointed out that preprocessing and post-processing procedures based on prior information about the input data further enhanced network performance.

Vrckovnik, Chung, and Carter (1990) classified impulse radar waveforms from asphalt-covered bridge decks using a multi-layer network trained with the backpropagation algorithm. They utilized Principal Components Analysis (PCA) to reduce the input data's dimensionality and this produced excellent classification rates that ranged between 95.6% and 100%. The network supposedly was able to produce accurate estimates of the specific deck structure and would eliminate the need for taking core samples from the bridge deck.

Ramani et al. (1990) employed the use of a neural network to distinguish fish from debris and classify fish based on sonar echoes. Single-hidden-layer networks were trained using the backpropagation algorithm and preliminary results were encouraging. The researchers were hopeful that in the future a neural-network-based active sonar system would replace current netting practices for fish stocks assessment and impact studies.

3.3 Hybrid Neural Network and Genetic Algorithm Systems

Applications of hybrid neural network and genetic algorithms (GA/NN) have become increasingly popular in the field of artificial intelligence. These hybrid applications are either supportive (i.e. they are used sequentially) or collaborative (i.e. they are used simultaneously) (Schaffer, Whitley, and Eshelman, 1992). Currently, most of the applications have been focused on using genetic algorithms to improve neural network training rather than vice versa (Medsker, 1995). Since the GA/NN model used in this project belongs to the supportive hybrid system, the following section will only focus on literature reviews of the supportive systems.

3.3.1 Supportive Hybrid Systems

Several areas where supportive hybrid systems have been successfully applied are:

- Selection of data or feature sets or transformation of feature space by genetic algorithm for neural network training;
- Selection of neural network learning rule or parameters by genetic algorithm;
- Evolving neural network structures using genetic algorithm and other evolution techniques;
- Neural network analysis by a genetic algorithm.

3.3.1.1 Data Preparation by Genetic Algorithm

In their attempts to train neural networks capable of monitoring accident scenarios at a nuclear power plant, Guo and Uhrig (1992) used a genetic algorithm to search for optimal combinations of input variables from hundreds of plant processing variables.

They noted that the genetic algorithm's performance was strongly affected by the fitness function. The optimal combinations of input variables managed to reduce neural network training time, improve recall accuracy, and reduce the number of input variables for successful network training.

Brill, Brown, and Martin (1992) proposed two genetic search techniques to analyze training data for a Kohonen network. First, they used an approximate evaluation method called nearest-neighbor classifier to evaluate the data set. Second, they presented another method called the training set sampling whereby only a portion of the training set is used during any given evaluation. They managed to save training time of the Kohonen network.

Romaniuk (1994) partitioned initial data sets using genetic algorithm to produce a training set that was more suited for neural network training. Coded strings have binary values of 1 or 0 to represent the presence or absence of a particular training parameter in a training set.

Fukumi, Omatu, and Nishikawa (1995) used a genetic algorithm with partial fitness to reduce the required inputs for coin recognition by a multi-layer network. They coded the input signals, Fourier spectra, into genetic strings and divided each string into several parts. The partial fitness of these parts is evaluated and partial selection and crossover are conducted based on the fitness value. Strings with high fitness values are retained while less fit strings would be eliminated to reduce input size.

Using a genetic algorithm, Kelly and Davis (1991) searched for a data set's rotations and each attribute's scaling parameters that would improve the training of a K nearest neighbors (KNN) neural network classifier. The genetic algorithm selected the

rotation and scaling factors to minimize the data set's intraclass differences and maximize interclass differences.

Chang and Lipmann (1992) also successfully reduced the number of input variables for a KNN classifier for complex speech recognition tasks from 153 to 33 using a genetic algorithm. Furthermore, they used the genetic algorithm to create new inputs that were polynomial functions of the original inputs for the KNN classifier.

Wilson (1990) applied the genetic algorithm into a different use in his hybrid system. The genetic algorithm was used to evolve feature predicates for a neural network trained to solve six-bit Boolean multiplexer problems. The logical conjunct of some of the features was computed by the predicates, which also served as the inputs to a perceptron. The perceptron was trained on random input strings for each chromosome by using the average score over the entire training period of 100 epoch.

3.3.1.2 Other Applications of Supportive Hybrid Systems

Another popular application of supportive hybrid systems involves the use of genetic algorithm to search for neural network learning rules and/or parameters. Examples of using genetic algorithms to search for initial weights, learning rate, momentum, and even learning algorithm are works by Belew, McInerney, and Schraudolph (1990); Chalmers (1990); Harp, Samad, and Guha (1989); and Schaffer, Caruana, and Eshelman (1990). Table 3.1 lists some of the published works on supportive hybrid systems.

Table 3.1 Published Works on Supportive Hybrid Systems

AUTHOR(S)	YEAR	RESEARCH/PUBLICATION
Belew, McInerney, and Schraudolph	1992	Evolving networks: using the genetic algorithm with connectionist learning
Chalmers	1990	The evolution of learning: an experiment in genetic connectionism
Fukuda, Ishigami, Arai, and Shibata	1993	Structure organization of fuzzy neural networks using genetic algorithms
Harp, Samad, and Guha	1989	Towards the genetic synthesis of neural networks
Janson and Frenzel	1993	Training product unit neural networks with genetic algorithms
Jockusch and Ritter	1994	Self-organizing maps: local competition and evolutionary optimization
Kitano	1993	Neurogenetic learning: an integrated method of designing and training neural networks using genetic algorithms
Saha and Christensen	1994	Genetic design of sparse feed-forward neural networks
Schaffer, Caruana, and Eshelman	1990	Using genetic search to exploit the emergent behavior of neural networks
Yoon, Holmes, Langholz, and Kandel	1994	Efficient genetic algorithms for training layered feed-forward neural networks

4 UCLIM Database Encoding

4.1 UCLIM Database

The UF₆ Cylinder Location, Inspection, and Measurement System (UCLIM) database contained information about the UF₆ cylinders located at the three DOE Gaseous Diffusion Plant (GDP) sites in Kentucky, Ohio, and Tennessee. The system serves as a management tool to record and maintain a history of these UF₆ cylinders.

The database contained the following information:

- a. the plant site where the cylinder was stored;
- b. for some cylinders that were moved more than once, where they were previously stored and when the move occurred;
- c. the drawing number, model type, size, steel material, and wall gauge for the cylinder;
- d. whether the cylinder had a skirt;
- e. whether the cylinder had an ASME (American Society of Mechanical Engineers) stamp; the stamp indicated that the cylinder was classified as an ASME pressure vessel;
- f. the cylinder's purchase order, the dates it was ordered and delivered, the vendor and the cost;
- g. the depleted uranium assay in the cylinder, the tare weight and net weight of the cylinder;

h. visual inspection codes referred to as defect codes that were grouped into four categories:

- *A codes* – general conditions of the cylinder
- *B codes* – conditions of the cylinder's body contact points
- *C codes* – the cylinder's valve ends
- *D codes* – the cylinders plug ends.

Table 4.1 shows the four types of defect codes found in the UCLIM database while Appendix 4.1 shows the complete list of variables stored in the UCLIM database.

4.2 Additional Information for Data Set

The data file used in this research project included additional information that was previously not available in the UCLIM database. These 10 new variables were:

- a. maximum wall thickness of the cylinder (mils);
- b. the current site of a cylinder;
- c. the current yard the cylinder was located at;
- d. the yard name of the cylinder's location;
- e. the row the cylinder was located on;
- f. the cylinder's nominal thickness (mils);
- g. the year the cylinder was delivered;
- h. the year the cylinder thickness was measured;
- i. the age of the cylinder when measured;
- j. the maximum pit depth measured on the surface of the cylinder;

Table 4.1 Defect Codes Found in the UCLIM Database

Defect Code Type	Defect Code	Comment
General Condition of Cylinder	A01	Hole in cylinder
	A02	Visible leakage/contamination on cylinder or ground
	A03	Bulge
	A04	Gouge
	A05	Dent
	A06	Bent stiffening ring
	A07	Misalignment: lugs should be horizontal
	A08	Holes, dents, or gouges in channel stiffening ring
	A09	Several local pitting
	A10	Heavy scale on cylinder or ground
	A11	Stacking of incompatible cylinders
	A12	Longitudinal weld in 6 o'clock position
Cylinder Body Contact Point	B01	Body ground contact
	B02	Stiffening ring ground contact
	B03	Water/cylinder contact from poor yard drainage
	B04	Dent caused by lifting lug contact
	B05	Lifting lug contact
	B06	Improper upper tier support
	B07	Wood saddle: cracking, chipping, corroding, cracking, or sinking
	B08	Concrete saddles: cracking, chipping, corroding, cracking, or sinking
	B09	No saddles
	B10	Debris between saddle and cylinder
Valve End of Cylinder	C01	Evidence of contamination on valve
	C02	Bent valve body
	C03	Bent skirt
	C04	Scale in skirt: depth > 1 inch
	C05	Skirt in ground contact
	C06	Weep hole in skirt plugged
	C07	Valve bagged
	C08	Valve end not accessible
	C09	Teflon tape on valve threads
	C10	Packing nut
	C11	Improper packing nut installation
	C12	Port cap missing/cracked
	C13	Bent or sheared valve stems
	C14	Plug in place of valve
	C15	Bent valve protector
	C16	I.D. plate missing
	C17	I.D. plate loose/cracked welds
	C18	New I.D. plate attached
	C19	Misalignment of valve
Plug End of Cylinder	D01	Evidence of contamination on plug
	D02	Bent or damaged plug
	D03	Bent skirt
	D04	Scale in skirt: depth > 1 inch
	D05	Skirt in ground contact
	D06	Weep hole in skirt plugged
	D07	Plastic sheet covering plug area
	D08	Plug end not accessible
	D09	Teflon tape on plug threads
	D10	Plug rusted/non-standard shape
	D11	Plug seal or tack welded

The difference between maximum wall thickness and the maximum pit depth was the minimum wall thickness, which was the subject of interest in this research project. The minimum wall thickness would be used as the output for the training of a neural network to predict the wall thickness. The data set had the information of 1077 cylinders gathered from eight yards – one at K-25 (yard name K-1066-K), six at PGDP (yard names C-745-B, C-745-C, C-745-F, C-745-G, C-745-K, C-745-L), and one at PORTS. The 10 listed variables from this data set along with the existing UCLIM database variables made up the input variables used in this project. Altogether 61 variables were used as initial input variables and are listed in Table 4.2.

Of the 1077 cylinders, only 912 cylinders were used as input patterns for the neural networks. The other 165 cylinders were not used due to various reasons. Some of the cylinders were not found in the UCLIM database while others either had incomplete information or had suspicious thickness values. The suspicious thickness values referred to some cylinders that had thickness values less than the maximum pit values, which was not possible. Cylinders that were not used and their reason for dismissal from the final data set are listed in Appendix 4.2.

4.3 Data Encoding

In order to use the data set for neural network training, the variables had to be encoded into numeric format. Table 4.2 shows the 61 input variables with all their original formats and their equivalent numerical codings. For this project, two coding schemes were used to code the UCLIM database variables. The tersely named “original codes” refers to the first of the two coding schemes. In this format, UCLIM

Table 4.2 61 Variables Used as Neural Network Input Variables

Variable	Comments/UCLIM Database Values	Original Encoding*	One-of-N Encoding**
Cylinder ID	A unique no. that identifies each cylinder	-	-
Yard	A B C F G K L	1 2 3 4 5 6 7	yes
YardName	C-745-B C-745-C C-745-F C-745-G C-745-K C-745-L K-1066-K PORTS	1 2 3 4 5 6 7 8	yes
Row	T B	1 2	yes
Age When Measured	5 to 40	no change	no
Plant	1 2 3	no change	yes
Modifier	BLANK 4B 4G	0 1 2	yes
Drawing	D-S-10137 REV 2 D-S-10137 REV 5 E-M-14863-A E-M-14863-A REV 1 E-M-14863-A REV 6 E-M-14863-A REV 7 E-M-14863-A REV 11 E-M-14863-A REV 13 E-M-14863-A REV 14 E-M-14863-A REV 16 E-S-10313-D E-S-10313-D REV 0 E-S-10313-D REV 1 E-S-10313-F REV 0 E-S-10313-F REV 2 E-S-10313-F REV 3 E-S-12292-A REV 0 E-S-12292-A REV 1 E-S-12292-A REV 2 E-S-12292-A REV 3 E-S-12292-A REV 4 E-S-12292-A REV 8 E-S-12292-A REV 9 E-S-12292-A REV 10	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24	no
Model	48G 48O 48OM 48T	1 2 3 4	yes
Size	10 14	no change	yes
Material	A285 A516	1 2	yes
Skirt	Y N	1 0	yes
ASME_Stamp	Y N	1 0	yes

* note: no change -- the UCLIM database values were used

** note: yes -- variables were coded as one-of-N codes

no -- variables were not coded as one-of-N codes and were kept as the original codes

"" -- variables were not coded at all and not used as inputs

Table 4.2 (continued)

Variable	Comments/UCLIM Database Values	Original Encoding	One-of-N Encoding
PO	71-P-25550 WCP-31725 71-P-43430 71-P-55600 71-P-64234 71-P-73100A 15-P-81978-1 15-H-94273-1 8V-62570-1 IV-88544 8V-14073-1 87G-30711V 87G-43928V 95G-60000V 88G-91545V 88G-17922V 88G-53277V 88P-27640V 88G-53148V 28G-30609V 25G-08844V 60G-87300V 60G-K3097V 61P-JL377V 61P-JL891V	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	no
ORD_yymmdd	560910 570628 581008 591216 601021 611004 620709 631010 680901 711102 730110 730810 740301 740925 760305 761008 771020 790701 800801 820401 840801 860701 870501 890201 900301	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	no
Vendor	DTC TSI MWC JBB RMA WCO WJS	1 2 3 4 5 6 7	yes
Cost	308 to 998	no change	no change
Del_Plt	2 3	no change	yes
Assay	0.001815 to 0.007111	no change	no change
Tare	2316 to 2863	no change	no change
Net	20543 to 28583	no change	no change
Defect Codes: A02, A04, A05, A06, A07, A08, A09, A10, A11, A12, B01, B02, B03, B04, B05, B06, B07, B09, B10, C01, C02, C03, C04, C06, C07, C09, C10, C11, C13, C14, C15, C16, C17, C19, D02, D03, D08, D09, D10, X00	1 if condition exists, 0 if not	no change	no change

database variables not in numeric formats were simply replaced by numeric values in ascending order. The “Original Encoding” column of Table 4.2 shows the numerical codes equivalent to the UCLIM database values. Some variables had numerical values and were simply used as input values for the neural network training without additional encoding. Examples for this kind of variables were *Plant*, *Size*, *Cost*, and the *Assay*, *Tare*, and *Net Weights*. Other variables such as *Yard* and *Yard Name* were coded as numerical equivalents of their original codes.

The second coding scheme used the so-called “one-of-N” coding method for enumerated data that have no natural ranking (NeuralWorks Predict, 1995). An example of enumerated data is marital status. Consider “single,” “married,” and “divorced;” one should not assume “single” is closer to “married” than it is to “divorced” and code them as 1, 2, and 3 respectively. The best way to code marital status is therefore using the one-of-N encoding where the single parameter is replaced by three parameters for whether one is single, married, or divorced, as shown below:

Marital Status	Linear Encoding	One-of-N Encoding
Single	1	1-0-0
Married	2	0-1-0
Divorced	3	0-0-1

An important reason enumerated data should be coded as one-of-N codes is when a particular variable is scaled into a range suitable for neural network input, the variable’s predictive power could be eliminated or reduced (NeuralWorks Predict, 1995).

4.4 Output Encoding

The variable used as the output for neural network training was based on the cylinder's minimum wall thickness measured with handheld instrument. The neural network trained to predict minimum wall thickness simply used the measured thickness values as output values. For the network trained to categorize cylinders, the cylinders were classified into two categories – a) cylinders with minimum wall thickness greater than 250 mils, and b) cylinders with minimum wall thickness less than 250 mils.

Cylinders in the first group were represented by the numerical value of 0.95 while those in the second group were represented by 0.05. The reason to use 0.95 and 0.05 was the *sigmoidal* transfer function *logsig* had lower and upper limits of 0 and +1. Using *logsig* as the output transfer function ensured that the network output would fall between 0 and +1.

5 Prediction and Classification Results of the UF₆ Cylinders

5.1 Prediction by Multiple Linear Regression

Before neural networks and GA/NN models were used to predict and classify the UF₆ cylinders' thickness, multiple linear regression (MLR) was performed. MLR analysis can easily be performed by many commercial spreadsheet softwares. In this research, *Corel Quattro Pro 6.0* was used to perform the MLR analysis. The prediction and classification results by neural networks and GA/NN models would later be compared to results predicted by the MLR analysis.

For the one-of-N data set, it was discovered that some of the variables were exactly identical in the one-of-N coding. Because of the similarities, one of the two identical variables had to be removed from the data set in order to perform the MLR analysis; otherwise, the problem of a singular matrix would occur. These variables include *Yard-7* (originally *Yard-L*), *Yard Name-6* (originally *Yard C-745-L*), *Modifiers-0*, *-4B*, and *-4G*, *Del_Plt-2*, *Del_Plt-3*, *Models-48G*, *-48O*, *-48OM*, and *-48T*, and *Sizes-10* and *-14*. Table 5.1 shows the relationship between these variables. The variables in the first column are directly related to the variables in the third column. For example, *Yard-7* is exactly identical to *Yard Name-6*. On the other hand, the number of cylinders with *Del_Plt-3* classification is equal to the total number of cylinders with *Modifiers-4B*, and *-4G* classifications. Table 5.2 lists the R^2 values obtained from the MLR analysis while Figures 5.1 and 5.2 show the MLR prediction results.

Table 5.1 Identical One-of-N Variables

Variable that was kept	No. of cylinders having this parameter	Duplicate variable that was discarded	No. of variables having this parameter
Yard Name 6	3	Yard 7	3
Del_Plt 2	465	Modifier (blank)	465
Del_Plt 3	447	Modifier (4B)	219
		Modifier (4G)	228
Size 10	245	Model 4 (48T)	245
Size 14	667	Models 1 (48G)	77
		Models 2 (48O)	119
		Models 3 (48OM)	471

Table 5.2 R and R^2 Values for the MLR Analysis

	Original Codes		One-of-N Codes	
	R^2	R	R^2	R
Minimum Wall Thickness	0.396	0.629	0.406	0.637
Categorized Cylinders	0.204	0.452	0.212	0.460

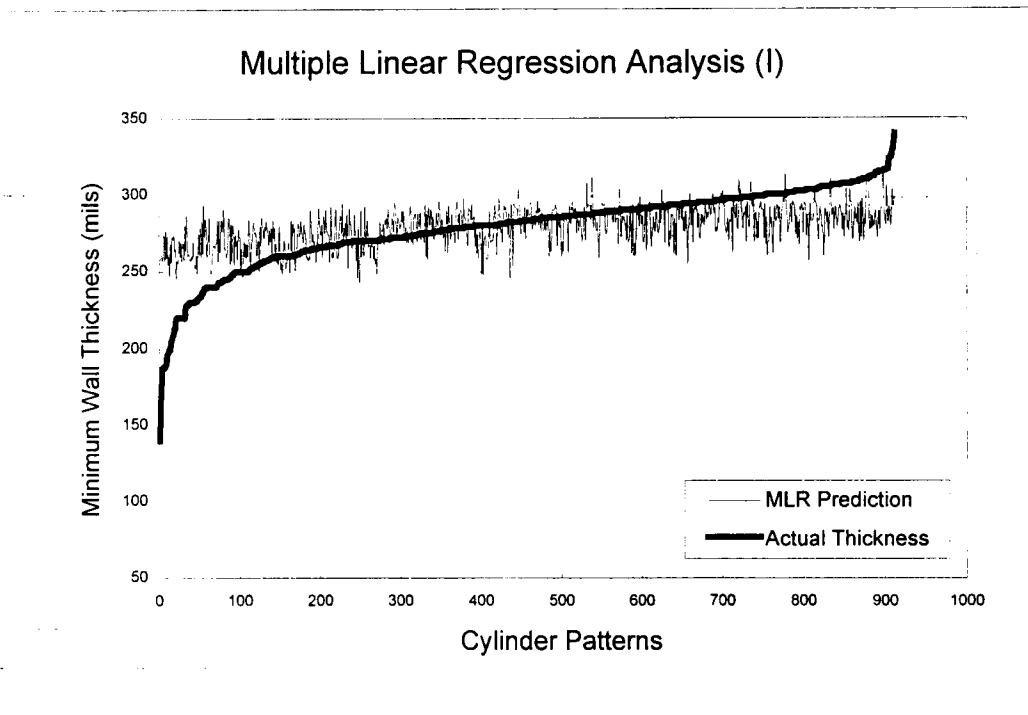


Figure 5.1 MLR Analysis for Minimum Wall Thickness

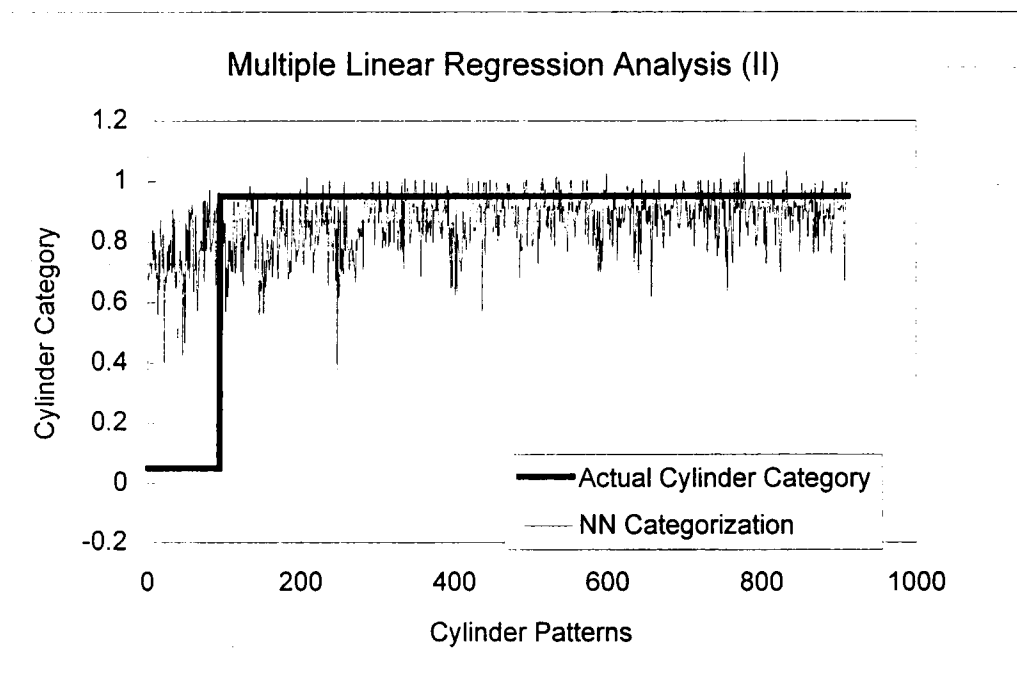


Figure 5.2 MLR Analysis for Categorized Cylinders

5.2 Correlation Coefficients of Variables

Following the MLR analysis, the correlation coefficients between the variables and the cylinder thickness as well as the categorized cylinders were computed. The main reason to calculate the correlation values was to identify variables that are highly correlated with the cylinder thickness or the categorized cylinders. Variables with the highest correlation would later be used as input variables for neural network training. This is because with fewer variables, *MATLAB*, the programming language used for neural network training, would be able to train a neural network in a reasonable period of time.

Table 5.3 shows the correlation coefficient values of the 61 UCLIM variables. 21 highest correlated variables were identified as possible input variables for the neural network trained to predict actual thickness. On the other hand, 22 variables were identified for the neural network trained to categorize the cylinders. These variables have the highest absolute correlation coefficient values and are listed in bold-italic font type.

The decision to use only the first 21 variables for the actual wall thickness group and 22 for the categorized cylinders was because neural network training time by *MATLAB* would increase significantly if too many variables were used. It should be noted that although these 21 and 22 variables were selected as potential input variables based on their correlation with the actual wall thickness or the categorized cylinders, none of the variables were actually highly correlated with either the wall thickness or the categorized cylinders. This was evidenced in the highest correlation values (R between

Table 5.3 Correlation Coefficients Between Variables and a) Actual Wall Thickness; b) Categorized Cylinders

a) Actual Wall Thickness	R	Absolute R	b) Categorized Cylinders	R	Absolute R
<i>Plant</i>	<i>0.2844</i>	<i>0.2844</i>	<i>Plant</i>	<i>0.2285</i>	<i>0.2285</i>
<i>Cost</i>	<i>0.2694</i>	<i>0.2694</i>	<i>Yard</i>	<i>-0.1778</i>	<i>0.1778</i>
<i>Row</i>	<i>-0.2636</i>	<i>0.2636</i>	<i>A08</i>	<i>-0.1764</i>	<i>0.1764</i>
<i>Tare</i>	<i>0.2409</i>	<i>0.2409</i>	<i>Del_Plt</i>	<i>0.1709</i>	<i>0.1709</i>
<i>ORD_yymmdd</i>	<i>0.2343</i>	<i>0.2343</i>	<i>Cost</i>	<i>0.1647</i>	<i>0.1647</i>
<i>Assay</i>	<i>0.2333</i>	<i>0.2333</i>	<i>ORD_yymmdd</i>	<i>0.1609</i>	<i>0.1609</i>
<i>B07</i>	<i>-0.2318</i>	<i>0.2318</i>	<i>Modifier</i>	<i>0.1604</i>	<i>0.1604</i>
<i>Yard</i>	<i>-0.2278</i>	<i>0.2278</i>	<i>Tare</i>	<i>0.1601</i>	<i>0.1601</i>
<i>A08</i>	<i>-0.2239</i>	<i>0.2239</i>	<i>Cylinder</i>	<i>0.1567</i>	<i>0.1567</i>
<i>Material</i>	<i>0.2229</i>	<i>0.2229</i>	<i>B07</i>	<i>-0.1558</i>	<i>0.1558</i>
<i>B02</i>	<i>-0.2129</i>	<i>0.2129</i>	<i>Row</i>	<i>-0.1521</i>	<i>0.1521</i>
<i>Age When Measured</i>	<i>-0.2127</i>	<i>0.2127</i>	<i>Age When Measured</i>	<i>-0.1441</i>	<i>0.1441</i>
<i>D09</i>	<i>-0.1849</i>	<i>0.1849</i>	<i>A12</i>	<i>-0.1367</i>	<i>0.1367</i>
<i>Cylinder</i>	<i>0.1756</i>	<i>0.1756</i>	<i>B02</i>	<i>-0.1348</i>	<i>0.1348</i>
<i>Del_Plt</i>	<i>0.1682</i>	<i>0.1682</i>	<i>Assay</i>	<i>0.1330</i>	<i>0.1330</i>
<i>Modifier</i>	<i>0.1552</i>	<i>0.1552</i>	<i>C14</i>	<i>-0.1128</i>	<i>0.1128</i>
<i>D08</i>	<i>0.1530</i>	<i>0.1530</i>	<i>D08</i>	<i>0.1079</i>	<i>0.1079</i>
<i>A12</i>	<i>-0.1430</i>	<i>0.1430</i>	<i>B03</i>	<i>-0.1070</i>	<i>0.1070</i>
<i>C09</i>	<i>-0.1320</i>	<i>0.1320</i>	<i>Skirt</i>	<i>-0.0941</i>	<i>0.0941</i>
<i>A09</i>	<i>0.1255</i>	<i>0.1255</i>	<i>Material</i>	<i>0.0910</i>	<i>0.0910</i>
<i>C14</i>	<i>-0.1196</i>	<i>0.1196</i>	<i>C07</i>	<i>-0.0873</i>	<i>0.0873</i>
<i>Vendor</i>	<i>0.1019</i>	<i>0.1019</i>	<i>A07</i>	<i>-0.0847</i>	<i>0.0847</i>
<i>C07</i>	<i>-0.0988</i>	<i>0.0988</i>	<i>D09</i>	<i>-0.0765</i>	<i>0.0765</i>
<i>C17</i>	<i>-0.0835</i>	<i>0.0835</i>	<i>YardName</i>	<i>0.0758</i>	<i>0.0758</i>
<i>A04</i>	<i>0.0834</i>	<i>0.0834</i>	<i>PO</i>	<i>0.0723</i>	<i>0.0723</i>
<i>Skirt</i>	<i>-0.0806</i>	<i>0.0806</i>	<i>A09</i>	<i>0.0697</i>	<i>0.0697</i>
<i>B03</i>	<i>-0.0693</i>	<i>0.0693</i>	<i>C09</i>	<i>-0.0664</i>	<i>0.0664</i>
<i>A11</i>	<i>-0.0679</i>	<i>0.0679</i>	<i>A04</i>	<i>0.0632</i>	<i>0.0632</i>
<i>PO</i>	<i>0.0634</i>	<i>0.0634</i>	<i>B10</i>	<i>-0.0617</i>	<i>0.0617</i>
<i>ASME_Stamp</i>	<i>0.0632</i>	<i>0.0632</i>	<i>Vendor</i>	<i>0.0535</i>	<i>0.0535</i>
<i>A10</i>	<i>0.0606</i>	<i>0.0606</i>	<i>X00</i>	<i>0.0505</i>	<i>0.0505</i>
<i>B10</i>	<i>-0.0534</i>	<i>0.0534</i>	<i>A11</i>	<i>-0.0494</i>	<i>0.0494</i>
<i>B06</i>	<i>0.0521</i>	<i>0.0521</i>	<i>C04</i>	<i>0.0485</i>	<i>0.0485</i>
<i>A07</i>	<i>-0.0518</i>	<i>0.0518</i>	<i>D10</i>	<i>-0.0447</i>	<i>0.0447</i>
<i>B04</i>	<i>0.0490</i>	<i>0.0490</i>	<i>C06</i>	<i>0.0390</i>	<i>0.0390</i>
<i>C03</i>	<i>0.0457</i>	<i>0.0457</i>	<i>C17</i>	<i>-0.0371</i>	<i>0.0371</i>

Table 5.3 (continued)

a) Actual Wall Thickness	R	Absolute R	b) Categorized Cylinders	R	Absolute R
A05	0.0429	0.0429	A10	0.0356	0.0356
C04	0.0404	0.0404	Model	0.0344	0.0344
Model	-0.0398	0.0398	Drawing	0.0333	0.0333
C16	0.0376	0.0376	C16	0.0296	0.0296
D10	-0.0355	0.0355	B06	0.0296	0.0296
C06	0.0345	0.0345	B04	0.0294	0.0294
A02	0.0290	0.0290	A06	0.0274	0.0274
B01	-0.0226	0.0226	C13	0.0257	0.0257
C02	0.0219	0.0219	ASME_Stamp	0.0243	0.0243
X00	0.0201	0.0201	C19	0.0237	0.0237
Size	-0.0193	0.0193	C02	0.0224	0.0224
C15	0.0175	0.0175	C03	0.0224	0.0224
C13	0.0174	0.0174	D02	0.0194	0.0194
C01	-0.0165	0.0165	Size	-0.0162	0.0162
C11	0.0150	0.0150	A05	0.0159	0.0159
Net	-0.0132	0.0132	D03	0.0158	0.0158
Drawing	-0.0117	0.0117	B09	0.0158	0.0158
YardName	0.0115	0.0115	C11	0.0158	0.0158
A06	-0.0097	0.0097	C01	-0.0126	0.0126
C19	-0.0069	0.0069	A02	0.0112	0.0112
D03	0.0053	0.0053	B05	0.0078	0.0078
C10	-0.0053	0.0053	Net	-0.0073	0.0073
B09	-0.0033	0.0033	C15	0.0071	0.0071
B05	-0.0011	0.0011	C10	0.0012	0.0012
D02	-0.0001	0.0001	B01	0.0007	0.0007

actual thickness and *Plant* = 0.2844 and $R^2 = 0.081$; *R* between categorized cylinders and *Plant* = 0.2285 and $R^2 = 0.052$).

Next, the correlation coefficients between each of the 21 and 22 candidate variables were calculated. This step was carried out to further reduce the number of input variables for neural network training. If two variables were found to be highly correlated with each other, only one would be used as input. This would eliminate the redundant information carried by two similar variables and speed up the convergence of neural network training. Tables 5.4 and 5.5 show the correlation coefficients between the 21 and 22 variables respectively.

As shown in Table 5.4, the variable *Plant* was highly correlated to the variables *Del_Plt* ($R = 0.8965$) and *Modifier* ($R = 0.8134$). Since *Plant* was the highest correlated variable to minimum wall thickness, it was retained as an input variable while the other two variables were removed from the list. Another reason *Del_Plt* was removed was because it was highly correlated to *Yard* ($R = -0.8732$), another variable relatively highly correlated to the minimum wall thickness. Since the variable *Cost* was highly correlated with *ORD_yymmdd* ($R = 0.8868$) and *Age When Measured* ($R = -0.8783$), the two variables were removed because *Cost* was better correlated with the wall thickness. Since the variable *Cylinder* essentially only represented the identity of each individual cylinder, it was not included as an input variable, although one should note that it was highly correlated with both *ORD_yymmdd* and *Age When Measured*. In short, four variables: *Cylinder*, *Modifier*, *ORD_yymmdd* and *Age When Measured* were eliminated from both the minimum wall thickness and the categorized cylinder groups leaving 17 and 18 input variables for each group respectively.

Table 5.4 Minimum Wall Thickness: Correlation Coefficients Between 21 Variables

	Min. Wall Thickness (mils)	Plant	Cost	Row	Tare	ORD yymddd	Assay	B07	Yard	A08	Material	B02	Age When Measured	D09	Cylinder Del Pit Modifier	D08	A12	C09	A09	C14
Min Wall Thickness (mils)	1.0000																			
Plant	0.2844	1.0000																		
Cost	0.2694	0.2636	1.0000																	
Row	-0.2636	0.0510	0.0601	1.0000																
Tare	0.2409	-0.0290	0.4474	-0.0937	1.0000															
ORD yymddd	0.2343	0.0517	0.8868	0.0212	0.6999	1.0000														
Assay	0.2333	0.1989	0.5674	-0.0077	0.3781	0.6285	1.0000													
B07	-0.2318	-0.2578	-0.1228	0.2879	-0.0368	-0.0514	-0.0383	1.0000												
Yard	-0.2278	-0.8812	-0.2988	-0.0462	0.0653	-0.0995	-0.0514	0.2564	1.0000											
A08	-0.2239	-0.2768	-0.2465	0.0565	-0.2966	-0.2017	-0.0970	0.2706	0.2727	1.0000										
Material	-0.2129	0.1405	0.7678	0.0667	0.2274	0.6364	0.6045	-0.0748	-0.3064	-0.0871	1.0000									
B02	-0.2129	-0.1811	-0.0884	0.2879	0.0103	-0.0011	0.0200	0.6029	0.2217	0.2003	-0.0849	1.0000								
Age When Measured	-0.2127	-0.0186	-0.8783	-0.0243	-0.7041	-0.9878	-0.6082	0.0496	0.0734	0.1987	-0.6248	0.0092	1.0000							
D09	-0.1849	-0.6181	-0.3630	-0.1208	0.0603	-0.1349	-0.1745	0.2258	0.5176	0.2395	-0.1623	0.1844	0.1388	1.0000						
Cylinder	0.1756	-0.0565	0.6636	-0.0428	0.7353	0.8695	0.4803	-0.0354	0.0422	-0.2728	0.3785	0.0181	-0.8655	-0.0009	1.0000					
Del Pit	0.1682	0.8965	0.1868	0.0572	-0.0881	-0.0284	0.0906	-0.3024	-0.8732	-0.3164	0.1273	-0.3082	0.0433	-0.5396	-0.0934	1.0000				
Modifier	0.1552	0.8134	0.3366	0.0496	0.2215	0.2558	0.3137	-0.2767	-0.8376	-0.2870	0.2309	-0.2796	-0.2439	-0.4749	0.1853	0.9072	1.0000			
D08	0.1530	0.5470	-0.0365	0.0116	-0.1378	-0.1990	-0.1993	-0.2055	-0.3911	-0.2159	-0.1524	-0.2103	0.2059	-0.4021	-0.1778	0.5922	0.4485	1.0000		
A12	-0.1430	0.0654	-0.3395	-0.0263	-0.6986	-0.5700	-0.3872	0.0816	-0.0198	0.4621	-0.1761	0.0213	0.5622	-0.1041	-0.6436	0.1088	-0.1305	0.1758	1.0000	
C09	-0.1320	-0.1564	-0.5147	-0.0631	-0.4180	-0.5548	-0.4013	0.1146	0.1753	0.1612	-0.2551	0.1024	0.5717	0.3826	-0.4657	-0.0736	-0.2805	0.0667	0.3685	1.0000
A09	0.1265	-0.2537	0.0868	-0.1100	0.2369	0.2122	0.0732	0.1323	0.4291	0.1337	-0.0027	0.2381	-0.1882	0.2075	0.2498	-0.4846	-0.4317	-0.2216	-0.1529	0.0271
C14	-0.1196	0.0618	-0.1298	0.1507	-0.2170	-0.1959	-0.0805	0.0439	-0.0466	0.0338	-0.0723	0.0539	0.1919	-0.1112	-0.2239	0.0185	-0.0535	0.0108	0.1612	-0.0770
																				1.0000

Table 5.5 Categorized Cylinders: Correlation Coefficients Between 22 Variables

	Categorized Cylinders	Plant	Yard	A08	Del Pit	Cost	ORD yymmdd	Modifier	Tare	Cylinder	B07	Row	Age When Measured	A12	B02	Assay	C14	D08	B03	Skirt	Material	C07	A07
Categorized Cylinders	1.0000																						
Plant	0.2285	1.0000																					
Yard	-0.1778	-0.8812	1.0000																				
A08	-0.1764	-0.2768	0.2727	1.0000																			
Del Pit	0.1709	0.8965	-0.8732	-0.3164	1.0000																		
Cost	0.1647	0.2536	-0.2988	-0.2455	0.1868	1.0000																	
ORD yymmdd	0.1609	0.0517	-0.0895	-0.2017	-0.0284	0.8868	1.0000																
Modifier	0.1604	0.8134	-0.8376	-0.2870	0.9072	0.3366	0.2558	1.0000															
Tare	0.1601	-0.0250	0.0653	-0.2966	-0.0881	0.4474	0.6989	0.2215	1.0000														
Cylinder	0.1567	-0.0565	0.0422	-0.2728	-0.0934	0.6536	0.8695	0.1883	0.7353	1.0000													
B07	-0.1558	-0.2578	0.2564	0.2706	-0.3024	-0.1239	-0.0514	-0.2767	-0.0368	-0.0354	1.0000												
Row	-0.1521	0.0510	-0.0462	0.0565	0.0572	0.0601	0.0212	0.0496	-0.0937	-0.0428	0.2979	1.0000											
Age When Measured	-0.1441	-0.0186	0.0734	0.1987	0.0433	-0.8763	-0.9878	-0.2439	-0.7041	-0.8655	0.0496	-0.0243	1.0000										
A12	-0.1367	0.0654	-0.0198	0.4621	0.1088	-0.3395	-0.5700	-0.1305	-0.6386	0.0816	-0.0263	0.5622	1.0000										
B02	-0.1348	-0.1811	0.2217	0.2003	-0.3082	-0.0884	-0.0011	-0.2796	0.0103	0.0181	0.6029	0.2879	0.0092	0.0213	1.0000								
Assay	0.1330	0.1989	-0.3927	-0.0970	0.0906	0.5674	0.6285	0.3137	0.3781	0.4803	-0.0383	-0.0077	-0.6092	-0.3872	0.0200	1.0000							
C14	-0.1128	0.0618	-0.0466	0.0338	0.0185	-0.1298	-0.1959	-0.0535	-0.2170	-0.2239	0.0439	0.1507	0.1919	0.1612	0.0539	-0.0805	1.0000						
D08	0.1079	0.5470	-0.3911	-0.2159	0.5922	-0.0365	-0.1990	0.4485	-0.1378	-0.1778	-0.2055	0.0116	0.2059	0.1758	-0.2103	-0.1883	0.0108	1.0000					
B03	-0.1070	-0.0036	0.0174	0.0666	-0.0255	-0.0615	-0.0526	-0.0072	-0.0243	-0.0747	0.1550	0.0805	0.0547	0.0655	0.1167	-0.0285	0.1954	0.0042	1.0000				
Skirt	-0.0941	0.2847	-0.2558	0.0351	0.3541	-0.3111	-0.6713	0.0060	-0.7673	-0.7081	-0.0781	0.0693	0.6551	0.6499	-0.1424	-0.4249	0.2234	0.3526	0.0602	1.0000			
Material	0.0910	0.1405	-0.3064	-0.0871	0.1273	0.7678	0.6364	0.2309	0.2274	0.3785	-0.0748	0.0667	-0.6248	-0.1761	-0.0849	0.5045	-0.0723	-0.1924	-0.0220	-0.1858	1.0000		
C07	-0.0873	-0.1302	0.0928	-0.0214	-0.0651	-0.0254	-0.0679	-0.0590	-0.0729	-0.0690	0.0893	0.0325	0.0484	0.1018	-0.0209	-0.0484	-0.0178	-0.0444	-0.0054	0.0965	-0.0179	1.0000	
A07	-0.0847	-0.2993	0.3163	0.1198	-0.3311	-0.0149	0.1026	-0.2471	0.1723	0.1163	0.0962	-0.0784	-0.0896	-0.0997	0.1136	-0.0086	-0.0829	-0.2303	0.0055	-0.2426	-0.0427	0.0710	1.0000

Using the 17 and 18 input variables and the training set, which consisted of one third of the data set, multiple linear regression again was performed so that their R^2 values were more comparable with the neural network and GA/NN R^2 values. The y intercept and the regression coefficients calculated from the training set were used to calculate the R^2 values of the whole data set. The R^2 values for the whole data set was 0.292 for the 17 variables (prediction of minimum wall thickness) and 0.041 for the 18 variables (classification of cylinders).

5.3 Neural Network Training

Four groups of neural networks were trained and tested. The first two groups of networks were trained to predict minimum cylinder thickness using the 17 variables that had the highest correlation coefficients with the cylinder thickness. The difference between these two groups was that one was trained on the original linear codes while the other was trained on the one-of-N codes. The other two groups were trained to categorize the conditions of the cylinders with one group using the original linear codes and the one-of-N codes for the other group. In this case, the cylinders' "conditions" referred to whether the cylinders had minimum wall thickness greater or less than 250 mils.

5.3.1 MATLAB Training File

The programming language *MATLAB for Windows* version 4.2c.1 was used in the neural network training. First, a *MATLAB* function file (from here on referred to as the M-file) was written to carry out the desired functions for neural network training. This M-file used the Levenberg-Marquardt algorithm as the learning algorithm and was

modified from a similar function file generously provided by Dr. J. Wesley Hines. The M-file allowed the user to select the following training options:

- Input data scaling: MCVU (Z-score) or linear (between +0.1 and +0.9) scaling;
- Number of hidden layers: 1 to 3 layers;
- Number of hidden neurons in each layer;
- Transfer function in each layer;
- Sum-squared error (SSE) goal; and
- Maximum training epochs.

Before the actual data file was used in the training of a neural network, a test file was created to make sure that the M-file performed the training process correctly. The 17 input variables selected for the prediction of minimum wall thickness were used to create a perfect function in which the output value was directly calculated from these variables. The equation used to calculate the output value is shown below:

$$a_p = \frac{((P + C) \times R)}{ORD} + (T \times A) - (B07 + Y + A08 + M + B02 + D09 + D08 + A12 + C09 + A09 + C14) \quad (5.1)$$

where $P = \text{Plant}$, $C = \text{Cost}$, $R = \text{Row}$, $ORD = ORD_yymmdd$, $T = \text{Tare}$, $A = \text{Assay}$, $Y = \text{Yard}$, and $M = \text{Material}$; defect codes symbols remained unchanged.

A network with one hidden layer and twelve hidden neurons was used. Figure 5.3 shows the test result when the network was tested. It shows that the network can produce reasonably good result when predicting the made-up output values. With the assurance that the M-file worked correctly, the next step was to proceed with the network training sessions to predict minimum wall thickness and to categorize the cylinders.

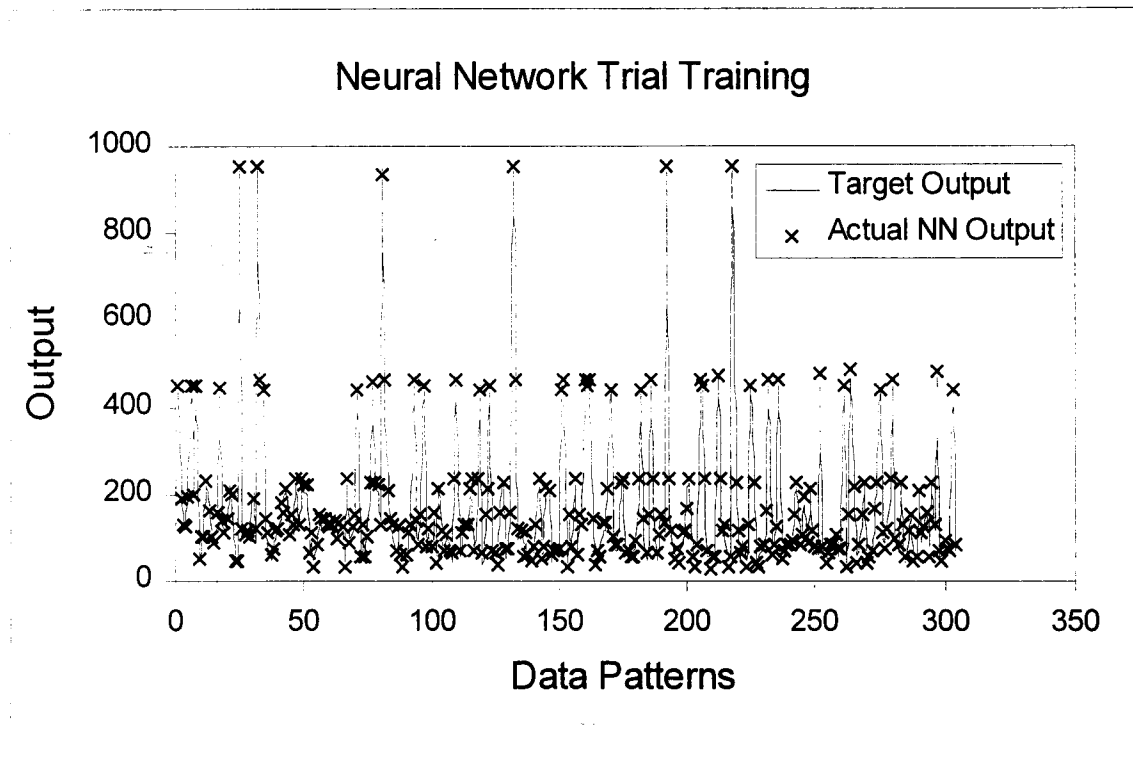


Figure 5.3 NN Trial Training Using Perfect Function

The network architecture was selected using the trial-and-error method suggested by Masters (1993) in Chapter 2. This was done by starting training and testing with the least number of hidden layers and neurons and slowly increasing the network size. Network structures used for each training process are shown in the following sections.

The linear transfer function was used as the output function for networks predicting wall thickness while the log-sigmoid (*logsig*) output transfer function was used for networks classifying the cylinders. On the other hand, hyperbolic tangent sigmoid (*tansig*) and *logsig* functions were used for the hidden layers to ensure that the networks could handle nonlinear mapping.

The maximum training epochs varied significantly from one network training to another. All networks started with a maximum number of 20 epochs. When overfitting

occurred, the number of training epochs would be reduced to approximately the number when overfitting started to occur and the network was retrained. If the network did not reach the error goal after the maximum training epoch, the number of maximum training epochs were increased provided the network error continued to decrease steadily while overfitting did not occur.

The default training parameters (*tp*) used by *MATLAB*'s Levenberg-Marquardt algorithm included:

- *tp*(1) – Training progress's display frequency, default = once every 25 epochs
- *tp*(2) – Maximum training epochs, default = 1000 epochs
- *tp*(3) – Sum-squared error (SSE) goal, default = 0.02
- *tp*(4) – Minimum gradient, default = 0.0001
- *tp*(5) – Initial value for μ , default = 0.001
- *tp*(6) – Multiplier for increasing μ , default = 10
- *tp*(7) – Multiplier for decreasing μ , default = 0.1
- *tp*(8) – Maximum value for μ , default = $1e10$

Initially, default values were used for *tp*(4) through *tp*(8) while *tp*(1) through *tp*(3) were varied according to various training situations. For the prediction of minimum wall thickness, the SSE goal was set at 30400 with a root-mean-squared error (RMSE) goal of ± 10 for each of the 304 cylinders in the training data. For the classification of cylinders, the SSE goal was 3.04 with a RMSE goal of ± 0.1 for each of the 304 cylinders. The minimum gradient of 0.0001 was further lowered down to $1e-20$ because error gradient was too small to continue training in many cases. The minimum gradient had to be

lowered to prevent some of the networks from stopping training prematurely after several epochs.

5.3.2 Neural Network Training Results

Four groups of networks were trained. The first network group was trained on the 17 linearly coded variables to predict minimum wall thickness. The second network group was trained to predict the wall thickness with the same 17, this time one-of-N encoded, variables mentioned above. The other two groups of networks were trained to categorize the cylinders into whether they have minimum wall thickness greater or less than 250 mils using the 18 selected variables. Again, one group was trained with the original linearly coded data while the other group with the one-of-N codes. As mentioned in Chapter 4, the one-of-N codes were used because they represented more appropriately enumerated data that had no natural ranking. The prediction capabilities of these neural networks would be compared to that by the MLR analysis.

All four network groups initially started with one hidden layer and two hidden neurons. The network architecture would be increased slowly when the network did not converge to reach the error goal. The maximum number of hidden neurons in a particular hidden layer was 30. Beyond that, the computer would either run out of memory quickly or the training process would slow down significantly. Although two hidden layers are widely recognized as the universal function approximator (see Chapter 2), up to three hidden layers were used for this project because none of the one- and two-hidden layer networks reached the desired error goal. Each architecture was trained three times and the weights and biases that produced the best network prediction results were retained. The

best weights were determined based on the RMSE and R^2 values of the training and test sets.

5.3.2.1 Prediction of Minimum Wall Thickness Using Original Codes

Table 5.6 shows the results of the network trained to predict minimum wall thickness using the original codes. Notice that none of the training sessions reached the error goal. The network RMSE would become stable at about 20 to 25. Training sessions that produced the best results were Sessions 1-1.1c, 1-1.1d, 1-1.1h, 1-1.2d, and 1-3.4a. Session 1-1.1d had the highest training set R^2 value – 0.3327 while Session 1-1.1h had the highest test set R^2 value – 0.2385. The low R^2 values indicated that the networks were not producing satisfactory prediction results.

Figures 5.4 – 5.8 show the plots of RMSE, training set and test set prediction results for Session 1-1.1c. The plots show that the network trained in Session 1-1.1c was incapable of producing prediction values in a gradually increasing pattern similar to the sorted actual thickness values. The predicted values varied mainly between 260 and 280 mils while not showing any consistent trend that would show the effectiveness of the network in predicting wall thickness.

Figures 5.9 – 5.13 show the prediction results for Session 1-1.1h, which were similar to those of Session 1-1.1c. Figures 5.10 and 5.12 show that the network had difficulty predicting the minimum wall thickness of both the training and test set. The only significant difference between the prediction results of the two was that the predicted values of Session 1-1.1h varied between 240 and 300 mils.

Table 5.6 Neural Network Training Results for Prediction of Wall Thickness using Original Codes

NUMBER OF INPUT VARIABLES: 17
 TRAINING SET FILE NAME: 17ver.m
 TARGET RMSE: 10
 TARGET SSE: 30400

TRAINING SESSION	INPUT SCALING	NUMBER OF HIDDEN LAYERS	HIDDEN LAYER 1 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 1	HIDDEN LAYER 2 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 2	HIDDEN LAYER 3 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 3	OUTPUT LAYER TRANSFER FUNCTION	MAXIMUM TRAINING EPOCHS	ACTUAL TRAINING EPOCHS	NETWORK TRAINING SSE	NETWORK TRAINING RMSE	TEST SET SSE	TEST SET RMSE	TRAINING SET CORREL. COEFF. (R)	TRAINING SET CORREL. R ²	TEST SET CORREL. COEFF. (R)	TEST SET R ²	REMARKS
1-1.1a	MCUV	1	sigmoid	2	NA	NA	NA	NA	linear	20	20	1.78E+05	24.2	1.75E+05	24.0	0.2113	0.0446	0.1780	0.0317	No overtraining, flat after 12th epoch
1-1.1b	MCUV	1	sigmoid	4	NA	NA	NA	NA	linear	20	20	1.57E+05	22.7	1.65E+05	23.3	0.3947	0.1558	0.3056	0.0934	No overtraining, flat after 6th epoch
1-1.1c	MCUV	1	sigmoid	7	NA	NA	NA	NA	linear	10	10	1.48E+05	21.9	1.43E+05	21.7	0.4686	0.2170	0.4603	0.2119	No overtraining, flat after 8th epoch
1-1.1d	MCUV	1	sigmoid	12	NA	NA	NA	NA	linear	20	20	1.24E+05	20.2	1.51E+05	22.3	0.5766	0.3327	0.4237	0.1795	No overtraining, flat after 10th epoch
1-1.1e	linear	1	tansig	16	NA	NA	NA	NA	linear	20	20	1.27E+05	20.4	1.55E+05	22.6	0.5644	0.3185	0.4064	0.1652	No overtraining, flat after 10th epoch
1-1.1f	linear	1	tansig	8	NA	NA	NA	NA	linear	20	20	1.42E+05	21.6	1.61E+05	23.0	0.4861	0.2363	0.3437	0.1181	No overtraining, flat after 6th epoch
1-1.1g	linear	1	tansig	16	NA	NA	NA	NA	linear	4	4	2.12E+05	26.4	2.38E+05	28.0	0.2402	0.0577	0.1157	0.0134	No overtraining, flat after 7th epoch
1-1.1h	MCUV	1	tansig	16	NA	NA	NA	NA	linear	20	20	1.28E+05	20.3	1.39E+05	21.4	0.6722	0.3274	0.4884	0.2383	No overtraining, flat after 7th epoch
1-1.1i	MCUV	1	sigmoid	20	NA	NA	NA	NA	linear	20	20	1.32E+05	20.8	1.55E+05	22.8	0.5397	0.2913	0.3956	0.1565	No overtraining, flat after 16th epoch
1-1.1j	MCUV	1	sigmoid	8	NA	NA	NA	NA	linear	20	20	1.33E+05	20.9	1.58E+05	22.9	0.5340	0.2852	0.3796	0.1441	No overtraining, flat after 18th epoch
1-1.2d	MCUV	1	sigmoid	12	NA	NA	NA	NA	linear	50	50	1.32E+05	20.9	1.44E+05	21.8	0.5365	0.2878	0.4571	0.2089	No overtraining, flat after 30th epoch
1-2.1a	MCUV	2	sigmoid	2	tansig	1	NA	NA	linear	20	4	1.83E+05	24.5	1.81E+05	24.4	0.1296	0.0168	0.0403	0.0016	No overtraining, flat after 2nd epoch
1-2.1b	linear	2	tansig	4	sigmoid	2	NA	NA	linear	20	4	1.85E+05	24.7	1.77E+05	24.1	0.0674	0.0045	0.1397	0.0195	No overtraining, flat after 2nd epoch
1-2.1c	MCUV	2	sigmoid	6	tansig	4	NA	NA	linear	20	20	1.78E+05	24.2	1.75E+05	24.0	0.1998	0.0399	0.1774	0.0315	No overtraining, flat after 3rd epoch
1-2.1d	MCUV	2	sigmoid	10	tansig	7	NA	NA	linear	50	50	1.66E+05	23.4	1.76E+05	24.1	0.3287	0.1080	0.1907	0.0364	No overtraining, flat after 6th epoch
1-3.2b	MCUV	3	tansig	5	sigmoid	4	sigmoid	3	linear	50	50	1.81E+05	24.4	1.77E+05	24.1	0.1640	0.0269	0.1426	0.0203	No overtraining, flat after 5th epoch
1-3.3b	MCUV	3	sigmoid	8	tansig	6	sigmoid	4	linear	50	50	1.78E+05	24.2	1.70E+05	23.8	0.2069	0.0428	0.2427	0.0589	No overtraining, flat after 6th epoch
1-3.4a	MCUV	3	tansig	12	sigmoid	9	tansig	6	linear	150	150	1.53E+05	22.4	1.46E+05	21.9	0.4237	0.1795	0.4340	0.1884	No overtraining, flat after 35th epoch

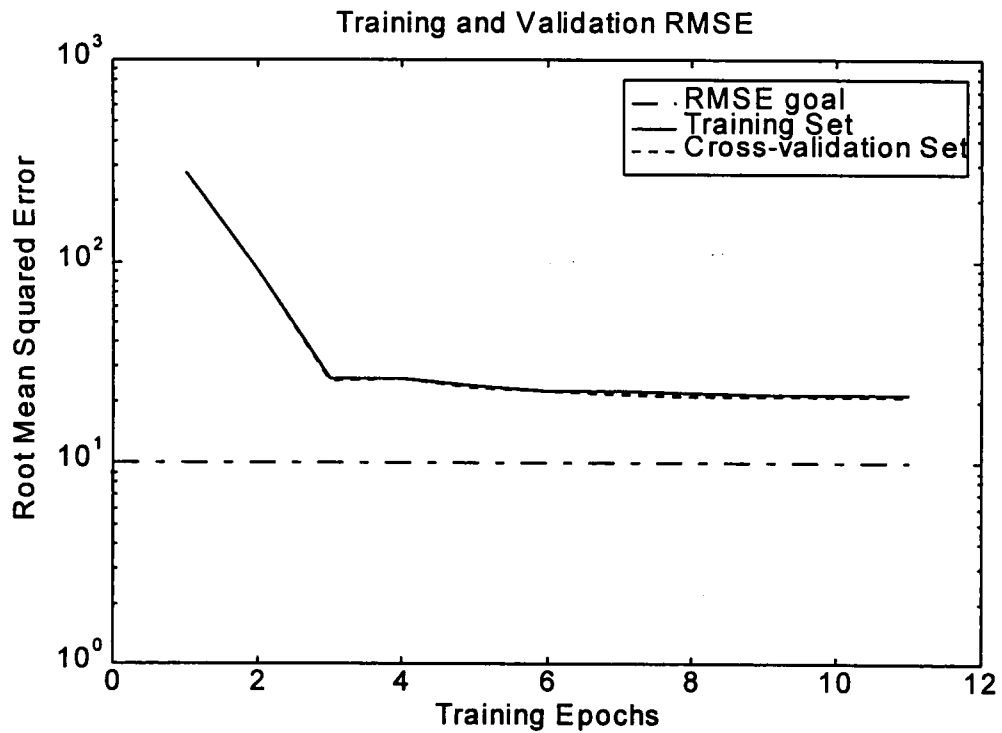


Figure 5.4 Session 1-1.1c: RMSE

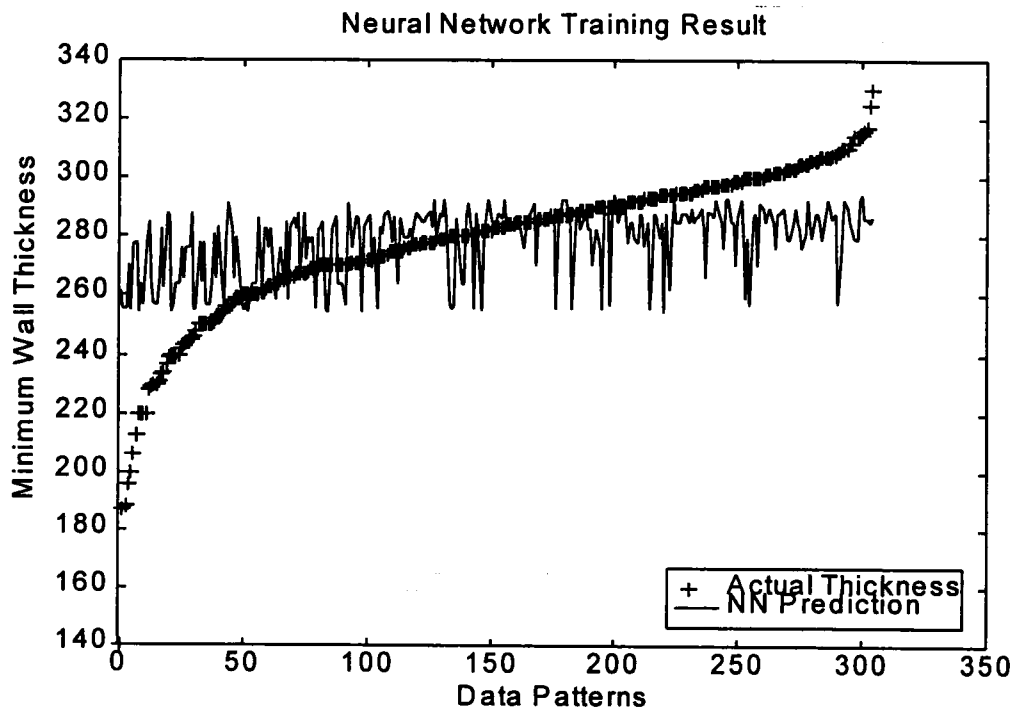


Figure 5.5 Session 1-1.1c: Prediction Results for Training Set

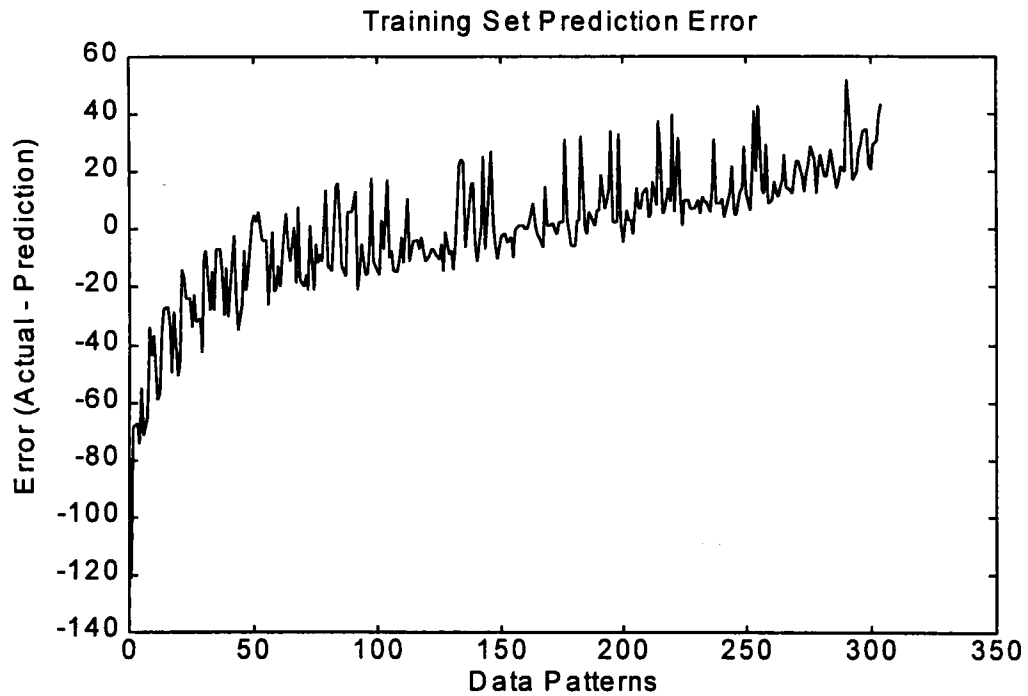


Figure 5.6 Session 1-1.1c: Training Set Prediction Error

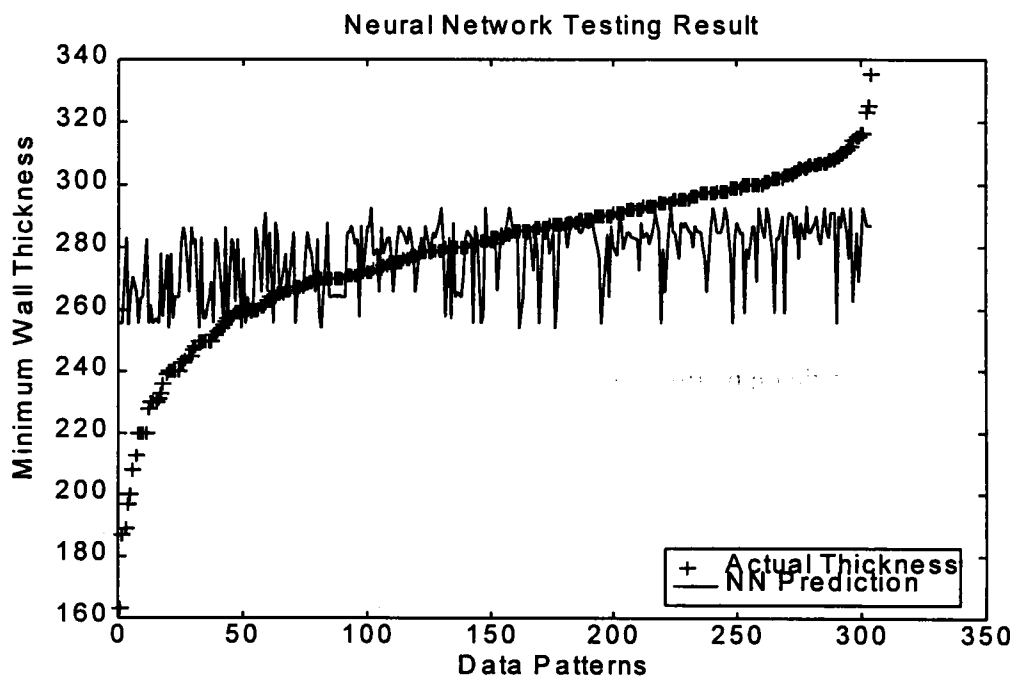


Figure 5.7 Session 1-1.1c: Prediction Results for Test Set

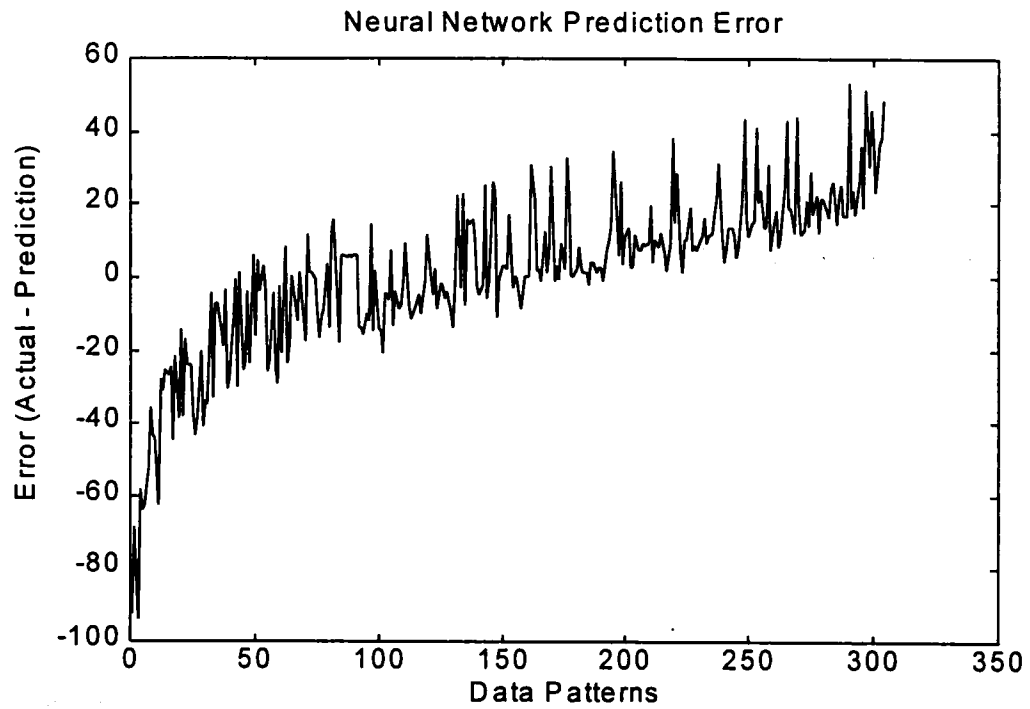


Figure 5.8 Session 1-1.1c: Test Set Prediction Error

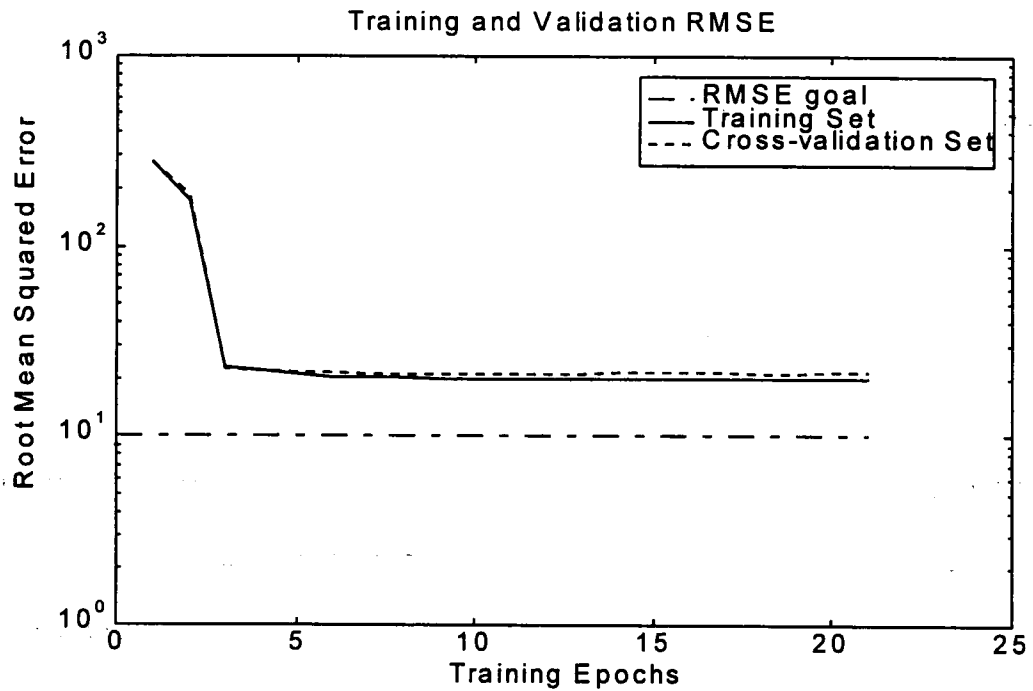


Figure 5.9 Session 1-1.1h: RMSE

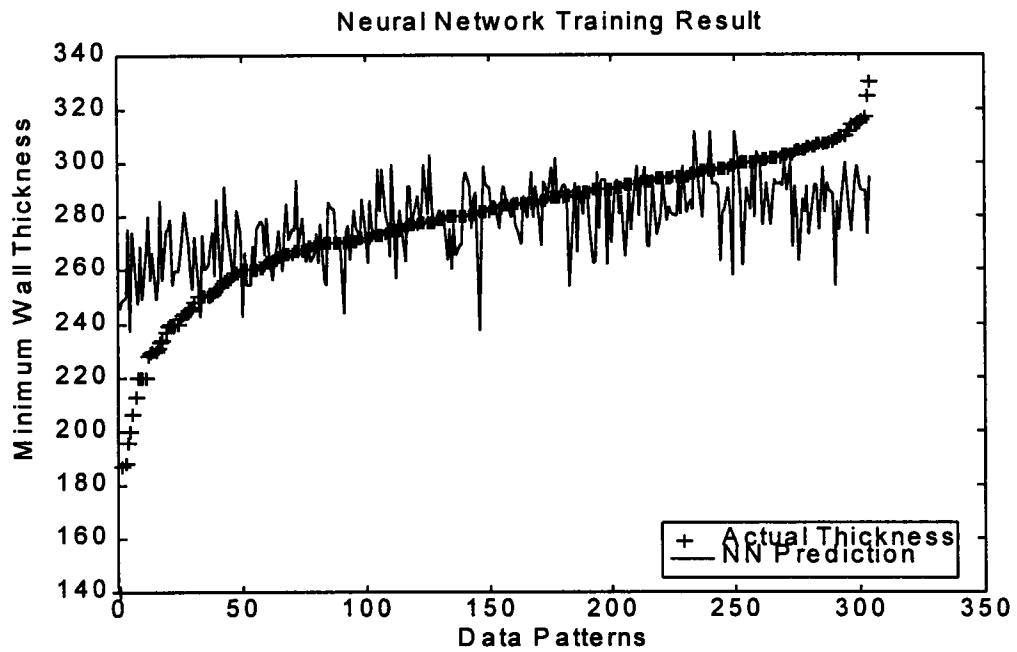


Figure 5.10 Session 1-1.1h: Prediction Results for Training Set

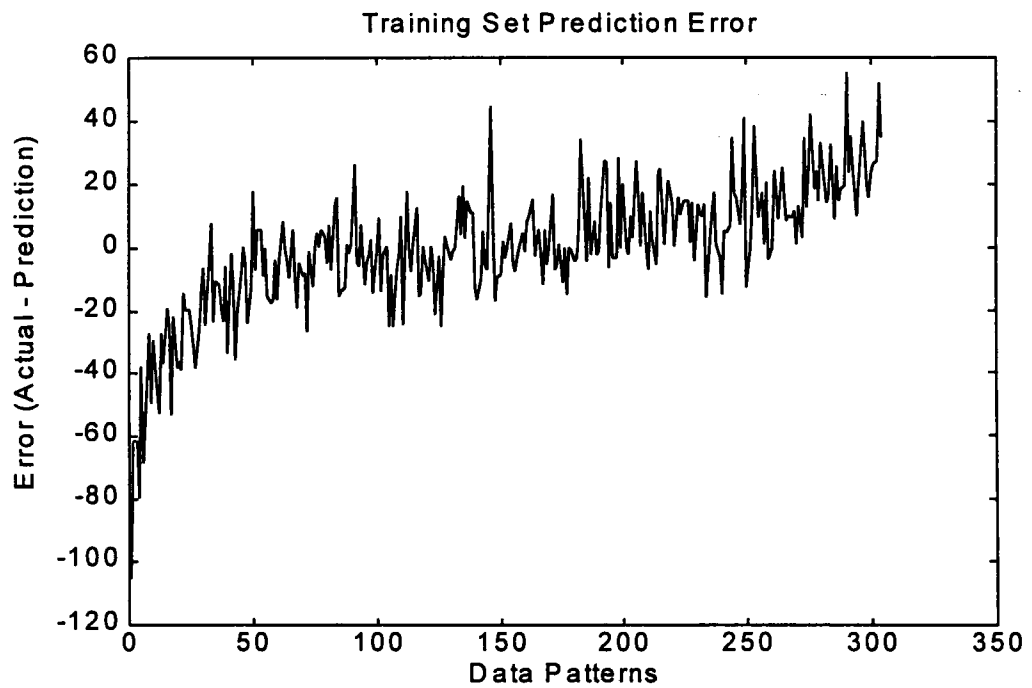


Figure 5.11 Session 1-1.1h: Training Set Prediction Error

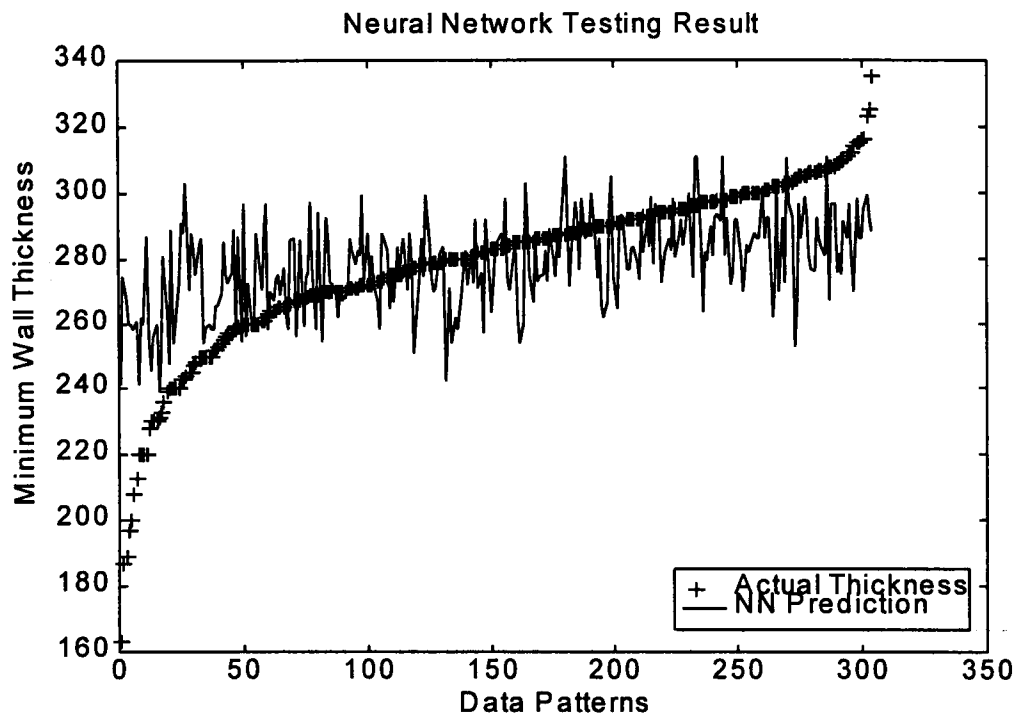


Figure 5.12 Session 1-1.1h: Prediction Results for Training Set

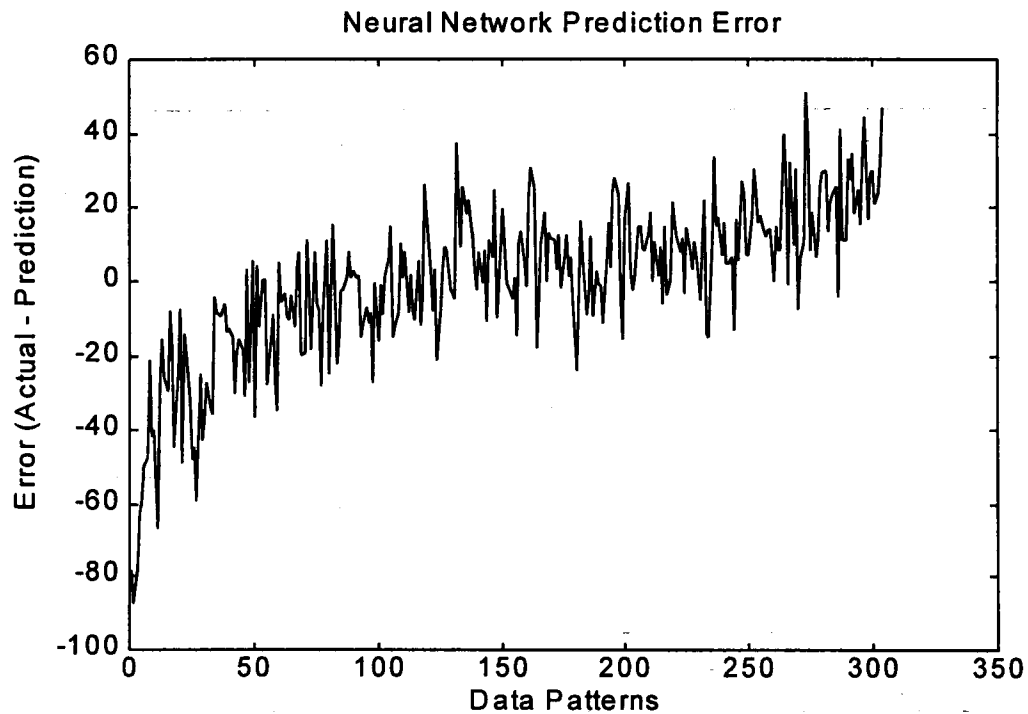


Figure 5.13 Session 1-1.1h: Training Set Prediction Error

5.3.2.2 Classification of Cylinders Using Original Codes

Table 5.7 shows the results of the network trained to classify the cylinders using the original codes. Many training sessions actually reached the target error when given enough training epochs; however, overtraining occurred in all cases and resulted in poor prediction when tested with the test set. Therefore, the training epochs were limited in some cases to as low as 4 or 6 epochs. Training sessions that produced the best results were Sessions 2-1.2a, 2-1.4a, 2-1.5a, and 2-3.1a. Other sessions might have higher training set R^2 values but they were the result of overtraining and produced very bad results when tested with the test set. Among the four training sessions, Session 2-1.2a had the highest training set R^2 value – 0.4285 while Session 2-1.1c recorded the highest test set R^2 value – 0.0944.

Figures 5.14 – 5.18 show the plots of RMSE, training set and test set classification results for Session 2-1.2a. Although the target values for the results shown in Figures 5.15 and 5.17 were 0.95 (designating cylinders with thickness greater than 250 mils) and 0.05 (designating cylinders with thickness less than 250 mils), the value predicted by the network would not always reach these two values. Additional analysis was conducted on the values returned by the network in which a value greater than 0.6 was considered to be equivalent to 0.95. On the other hand, if the network returned a value less than 0.4, it was considered to be equivalent to 0.05. If the value was between 0.4 and 0.6, the network was considered to be indecisive.

Using the above approach for Session 2-1.2a, of the training set's 31 cylinders that were less than 250 mils, the network wrongly classified 17 of them as cylinders with thickness greater than 250 mils (a *false negative* or **FN** prediction). A false negative

Table 5.7 Neural Network Training Results for Classification of Cylinders Using Original Codes

NUMBER OF INPUT VARIABLES: 18
 TRAINING SET FILE NAME: 10var.m
 TARGET RMSE: 0.1
 TARGET SSE: 3.04

TRAINING SESSION	INPUT SCALING	NUMBER OF HIDDEN LAYERS	HIDDEN LAYER 1 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 1	HIDDEN LAYER 2 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 2	HIDDEN LAYER 3 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 3	OUTPUT LAYER TRANSFER FUNCTION	MAXIMUM TRAINING EPOCHS	ACTUAL TRAINING EPOCHS	NETWORK TRAINING SSE	NETWORK TRAINING RMSE	TEST SET SSE	TEST SET RMSE	TRAINING SET CORREL. COEFF. (R)	TRAINING SET R ²	TEST SET CORREL. COEFF. (R)	TEST SET R ²	REMARKS
2-1.1a	MCUV	1	sigmoid	5	NA	NA	NA	NA	sigmoid	20	20	10.35	0.185	34.8326	0.338	0.7389	0.5460	0.0641	0.0041	network overtrained @ 5th epoch
2-1.1b	MCUV	1	sigmoid	5	NA	NA	NA	NA	sigmoid	6	6	17.58	0.240	23.1874	0.276	0.4712	0.2220	0.1948	0.0379	slight overtrain @ 5th epoch
2-1.2a	MCUV	1	sigmoid	3	NA	NA	NA	NA	sigmoid	20	20	12.86	0.206	27.2484	0.299	0.5546	0.4285	0.2119	0.0449	slight overtrain @ 5th epoch
2-1.2b	MCUV	1	sigmoid	3	NA	NA	NA	NA	sigmoid	20	4	20.57	0.260	22.7502	0.274	0.3108	0.0966	0.1763	0.0318	no overtraining
2-1.3b	MCUV	1	sigmoid	1	NA	NA	NA	NA	sigmoid	6	6	24.27	0.283	27.2515	0.299	0.2282	0.0521	0.111	0.0123	no overtraining
2-1.4a	linear	1	tansig	2	NA	NA	NA	NA	sigmoid	20	20	14.61	0.219	27.1901	0.289	0.5932	0.3519	0.1321	0.0175	slight overtrain @ 7th epoch
2-1.4b	linear	1	tansig	2	NA	NA	NA	NA	sigmoid	6	6	17.17	0.238	24.4672	0.284	0.4919	0.2420	0.1988	0.0385	slight overtrain @ 4th epoch
2-1.4c	linear	1	tansig	2	NA	NA	NA	NA	sigmoid	1000	1000	9.81	0.180	48.8809	0.401	0.7733	0.5980	-0.0481	0.0023	slight overtrain @ 10th epoch
2-1.5a	linear	1	sigmoid	10	NA	NA	NA	NA	sigmoid	20	20	4.96	0.128	39.9251	0.362	0.8875	0.7877	0.0841	0.0071	slight overtrain @ 2nd epoch
2-2.2a	linear	2	tansig	2	sigmoid	1	NA	NA	sigmoid	20	20	16.65	0.234	25.3872	0.289	0.5114	0.2615	0.1342	0.0180	slight overtrain @ 6th epoch
2-2.2b	linear	2	tansig	2	sigmoid	1	NA	NA	sigmoid	20	20	12.21	0.200	24.2305	0.282	0.6773	0.4587	0.1483	0.0220	slight overtrain @ 5th epoch
2-3.1a	MCUV	3	sigmoid	3	tansig	2	tansig	1	sigmoid	200	200	14.40	0.218	27.0472	0.298	0.6015	0.3618	0.1805	0.0326	slight overtrain @ 10th epoch
2-3.1b	linear	3	tansig	1	sigmoid	1	tansig	1	sigmoid	20	20	16.35	0.232	25.0492	0.287	0.5245	0.2751	0.1499	0.0225	slight overtrain @ 10th epoch
2-3.1c	linear	3	tansig	1	sigmoid	1	tansig	1	sigmoid	10	10	17.11	0.237	24.1341	0.282	0.492	0.2421	0.1754	0.0308	slight overtrain @ 5th epoch

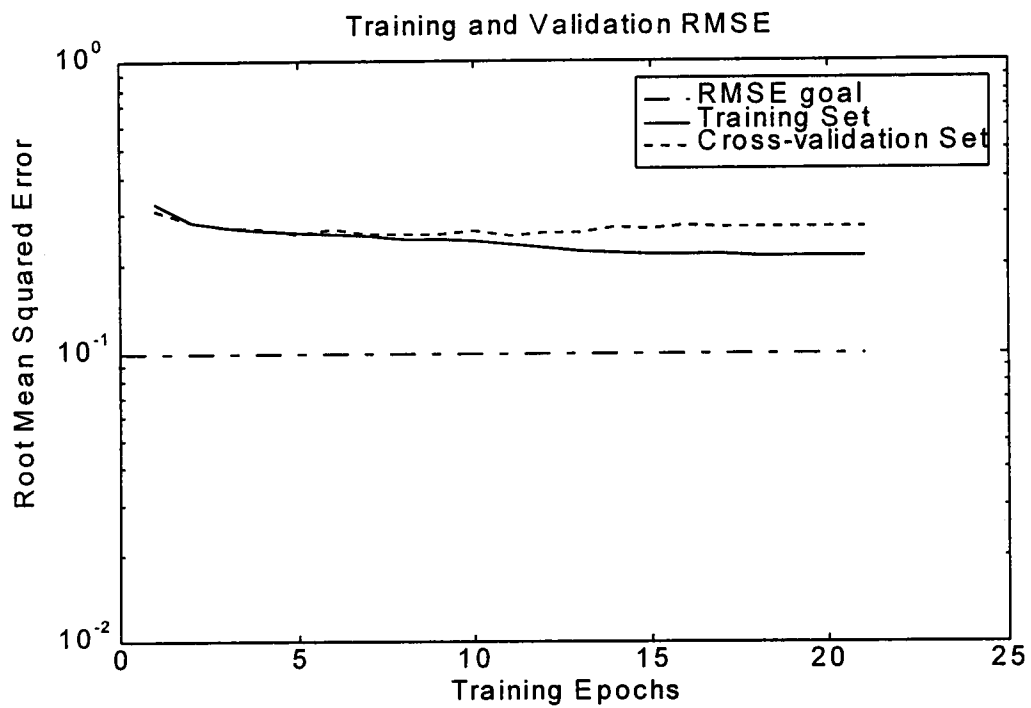


Figure 5.14 Session 2-1.2a: RMSE

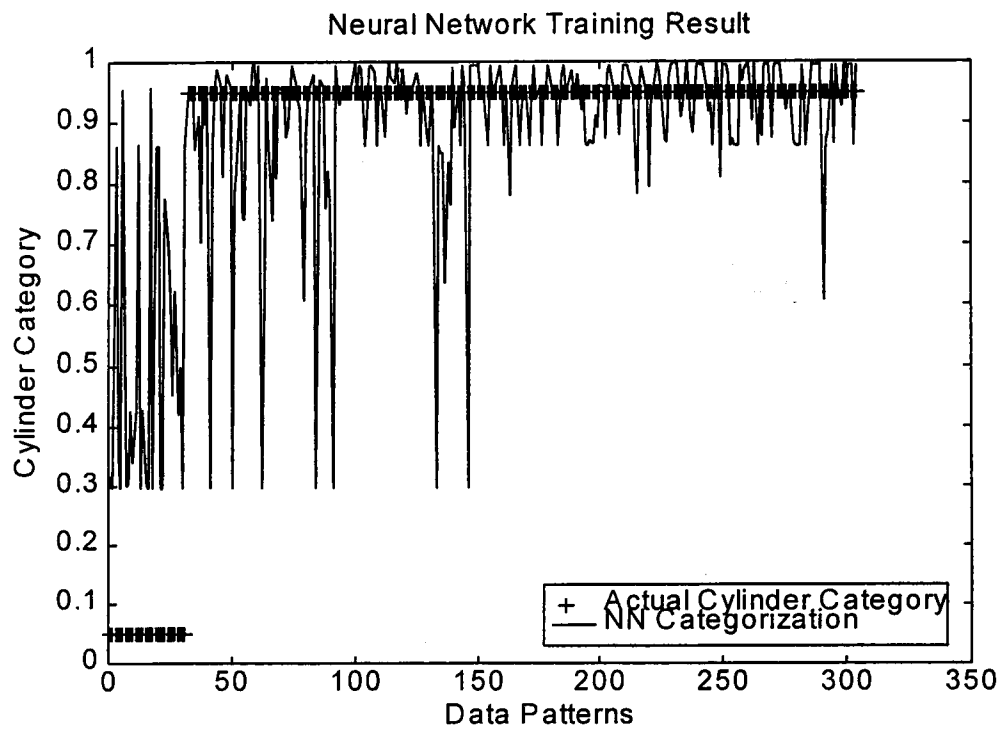


Figure 5.15 Session 2-1.2a: Prediction Results for Training Set

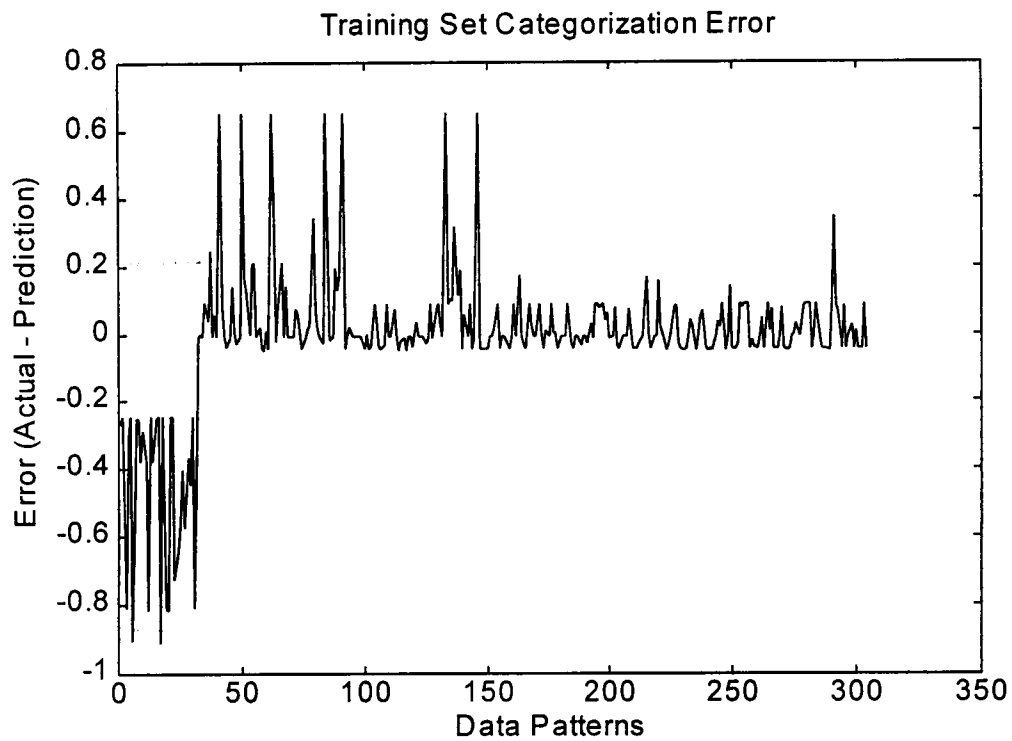


Figure 5.16 Session 2-1.2a: Training Set Prediction Error

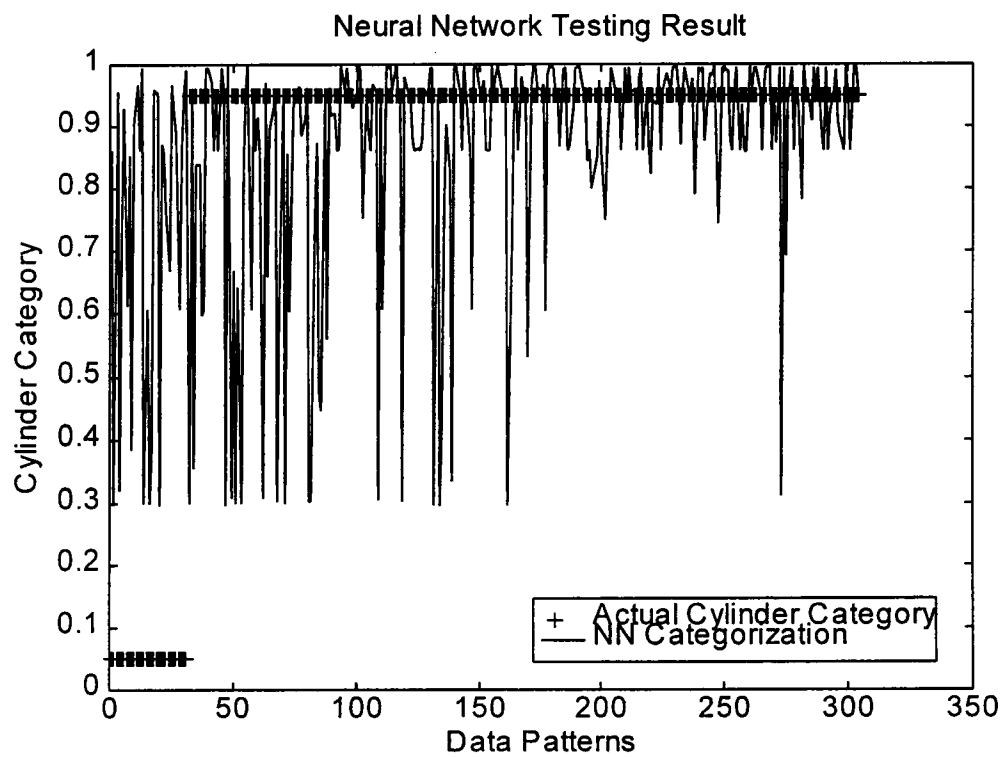


Figure 5.17 Session 2-1.2a: Prediction Results for Test Set

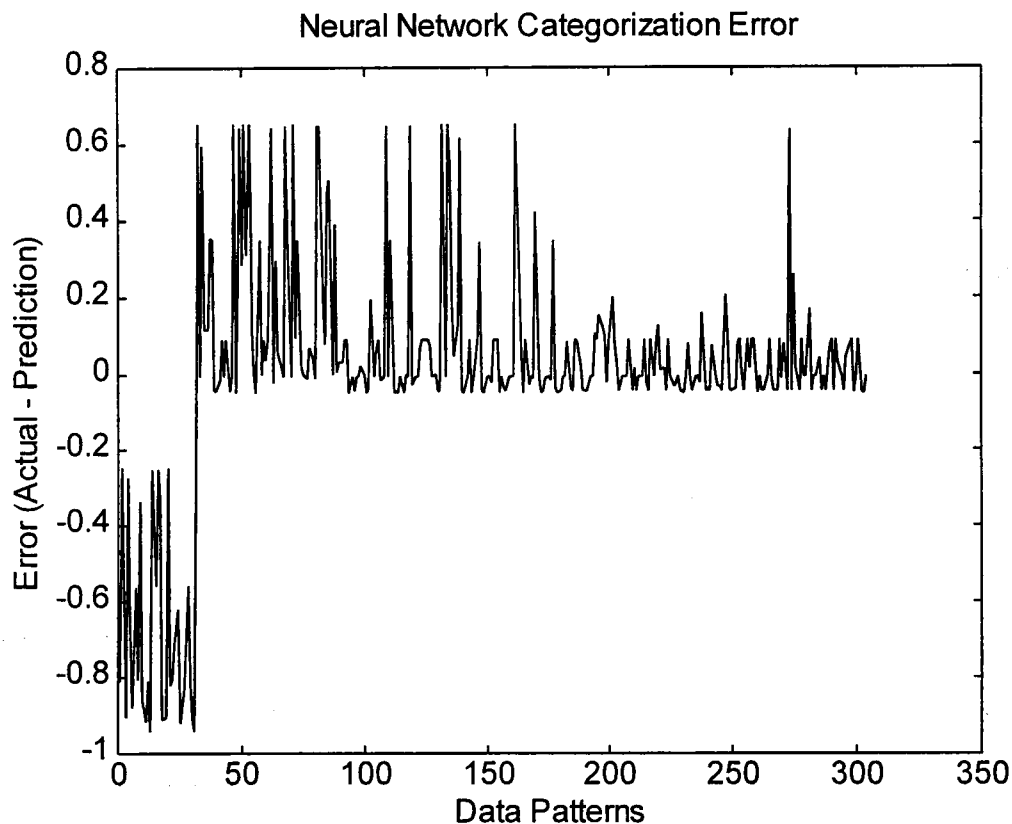


Figure 5.18 Session 2-1.2a: Test Set Prediction Error

prediction was undesirable because it allowed a cylinder with thickness less than 250 mils to go undetected. On the other hand, only 7 of the 273 cylinders with a thickness greater than 250 mils were wrongly classified as less than 250 mils (a *false positive* or **FP** prediction).

As for the test set, a total of 48 cylinders were classified wrongly with 24 having **FN** classification while the other half the **FP** classification. Based on the large number of **FN** classification, it was evident that the network had a tendency to classify cylinders in the greater-than-250-mils category. This was probably because 89.9% of the cylinders used as inputs were thicker than 250 mils. Therefore, although 84.2% (both true positive **TP** and true negative **TN**) of the test set were classified correctly, the network was actually doing a far from satisfactory job since 77.4% (24 out of 31) of the cylinders that

were supposed to be classified as less than 250 mils were wrongly classified.

Figures 5.19 – 5.23 show the plots of RMSE, training set and test set classification results for Session 2-3.1a. When the weights of Session 2-3.1a was tested using the training set, 17 FN classifications were recorded while 7 FP classifications were produced. The test set had 49 cylinders classified incorrectly with 25 FN classifications and 24 FP classifications. That means 83.9% of the test set was correctly predicted. However, similar to Session 2-1.2a, 80.6% of the cylinders less than 250 mils were classified incorrectly. This shows that the network was not capable of differentiating between cylinders greater or less than 250 mils.

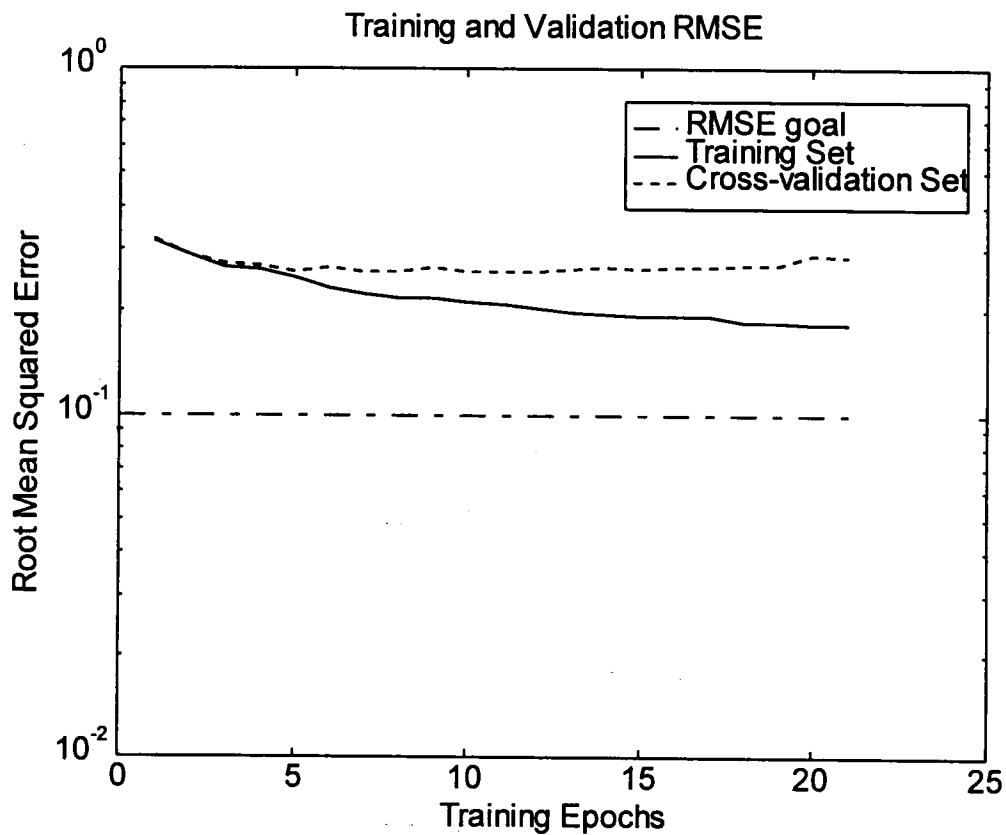


Figure 5.19 Session 2-3.1a: RMSE

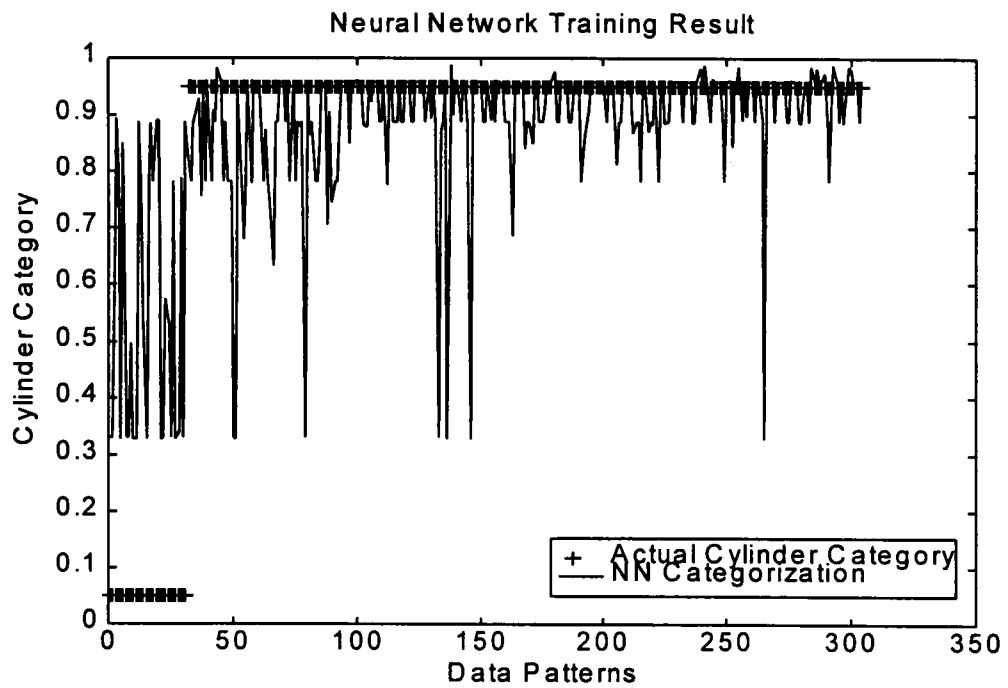


Figure 5.20 Session 2-3.1a: Prediction Results for Training Set

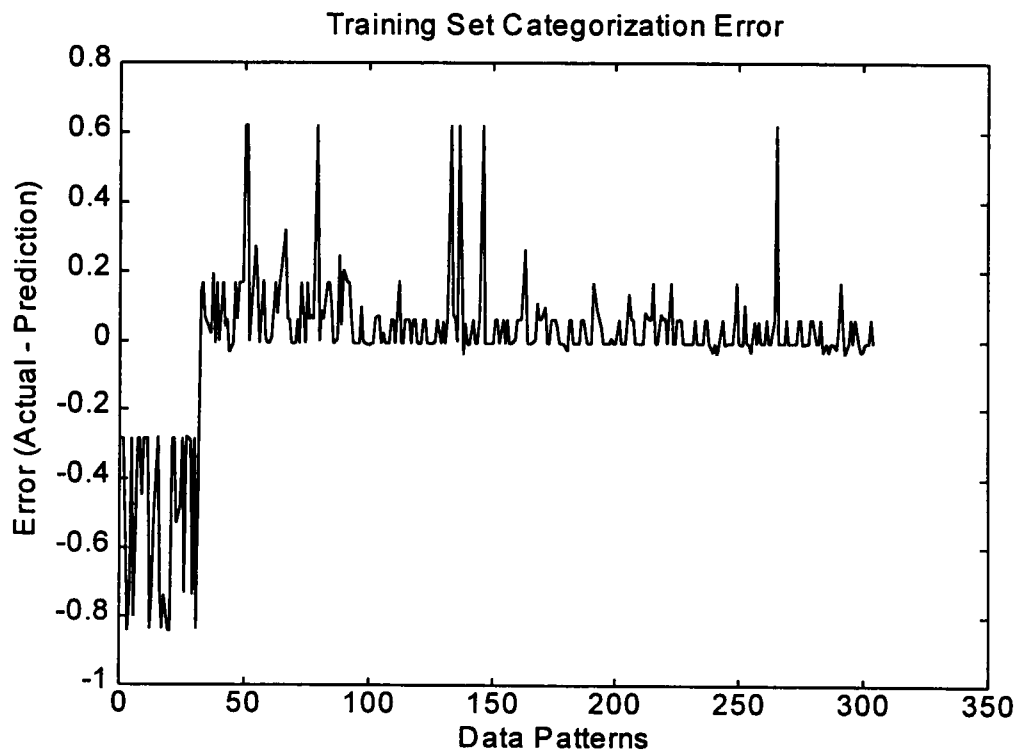


Figure 5.21 Session 2-3.1a: Training Set Prediction Error

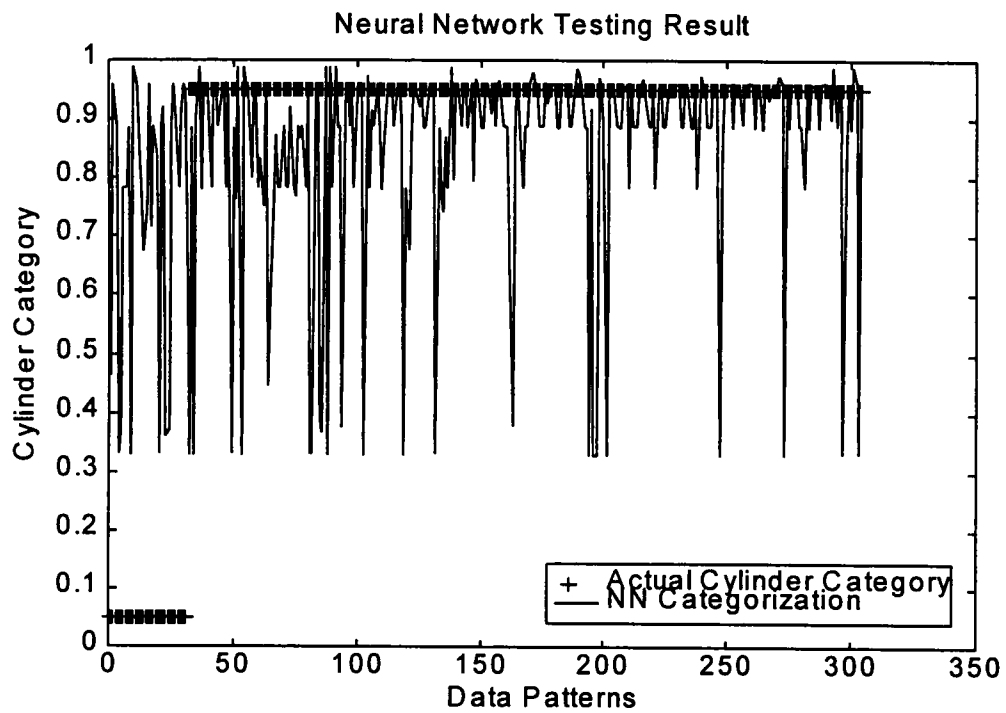


Figure 5.22 Session 2-3.1a: Prediction Results for Test Set

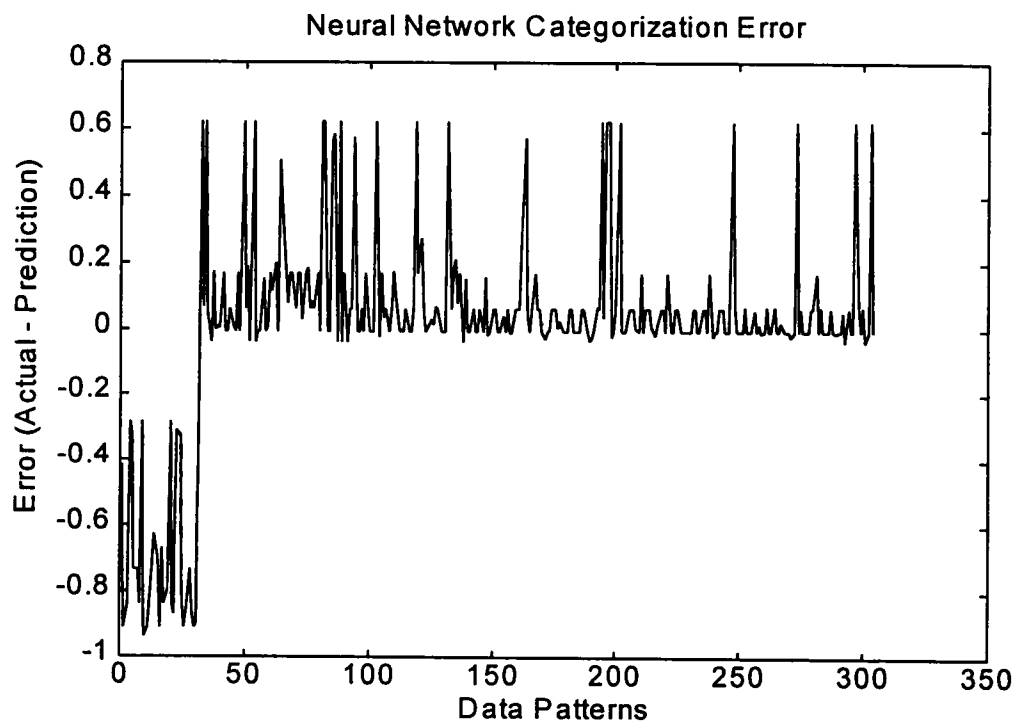


Figure 5.23 Session 2-3.1a: Test Set Prediction Error

5.3.2.3 Prediction of Minimum Wall Thickness Using One-of-N Codes

Table 5.8 shows the results of the network trained to predict minimum wall thickness using the one-of-N codes. As in the case where original codes were used, none of the training sessions reached the error goal. The lowest network RMSE achieved was about 20 while overtraining occurred the only time the RMSE fell as low as 17.5 (Session 3-1.1d). Sessions 3-1.1d, 3-1.1f, 3-2.1b, and 3-2.1c had the best prediction results. Even though overtraining occur, Session 3-1.1d still had the highest training set R^2 value – 0.5014 as well as test set R^2 – 0.2501. These values are slightly higher compared to the ones achieved by the training sessions using original codes. However, the improvement was too small to consider the networks developed using the one-of-N codes were useful as predicting tools of minimum wall thickness.

Figures 5.24 – 5.28 show the plots of RMSE, training set and test set prediction results for Session 3-1.1d. The plots show that the network was producing mostly random prediction values that were meaningless. The predicted values for the test set varied mainly between 240 and 300 mils.

Figures 5.29 – 5.33 show the plots of Session 3-1.1f which produces the next best test set R^2 value – 0.2227. This network was only slightly overtrained but like the network developed in Session 3-1.1d, it too could not satisfactorily predict the cylinder minimum thickness.

5.3.2.4 Classification of Cylinders Using One-of-N Codes

Table 5.9 shows the results of the network trained to classify the cylinders using the One-of-N codes. Similar to the network training using original codes, the network started

Table 5.8: Neural Network Training Results for Prediction of Wall Thickness using One-of-N Codes

NUMBER OF INPUT VARIABLES: 17, coded in 1-p-N format
TRAINING SET FILE NAME: N17year.m

TARGET RMSE: 10
TARGET SSE: 30400

TRAINING SESSION	INPUT SCALING	NUMBER OF HIDDEN LAYERS	HIDDEN LAYER 1 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 1	HIDDEN LAYER 2 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 2	HIDDEN LAYER 3 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 3	OUTPUT LAYER TRANSFER FUNCTION	MAXIMUM TRAINING EPOCHS	ACTUAL TRAINING EPOCHS	NETWORK TRAINING SSE	NETWORK TRAINING RMSE	TEST SET SSE	TEST SET RMSE	TRAINING SET CORREL. COEFF. (R)	TRAINING SET R ²	TEST SET CORREL. COEFF. (R)	TEST SET R ²	REMARKS
3-1-1a	MCUV	1	sigmoid	3	NA	NA	NA	NA	linear	50	30	1.50E+05	22.2	1.48E+05	22.2	0.4415	0.1949	0.4148	0.1721	Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 3.289474e-018
3-1-1b	MCUV	1	sigmoid	5	NA	NA	NA	NA	linear	50	50	1.51E+05	22.3	1.56E+05	22.7	0.4312	0.1859	0.3704	0.1372	No overtraining
3-1-1c	MCUV	1	tansig	15	NA	NA	NA	NA	linear	100	28	1.29E+05	20.6	1.54E+05	22.5	0.5550	0.3080	0.4166	0.1736	Error gradient is too small;
3-1-1d	linear	1	tansig	20	NA	NA	NA	NA	linear	20	20	9.27E+04	17.8	1.44E+05	21.8	0.7081	0.8014	0.8001	0.2601	overtraining after 10th epoch.
3-1-1e	MCUV	1	tansig	8	NA	NA	NA	NA	linear	100	20	1.48E+05	22.1	1.74E+05	23.9	0.4488	0.2014	0.2694	0.0728	Slightly overtrained
3-1-1f	MCUV	1	sigmoid	30	NA	NA	NA	NA	linear	100	20	1.18E+05	19.7	1.49E+05	22.1	0.6038	0.3546	0.4719	0.2227	No overtraining, list after 5th epoch
3-2-1a	MCUV	2	sigmoid	4	tansig	2	NA	NA	linear	50	5	1.80E+05	24.3	1.74E+05	23.9	0.1748	0.0306	0.1847	0.0341	Slightly overtrained after 5th epoch
3-2-1b	MCUV	2	sigmoid	15	tansig	10	NA	NA	linear	100	100	1.32E+05	20.8	1.61E+05	23.0	0.6389	0.2904	0.3689	0.1339	Slightly overtrained after 10th epoch
3-2-1c	linear	2	tansig	8	sigmoid	4	NA	NA	linear	50	50	1.48E+05	22.1	1.53E+05	22.4	0.4488	0.2012	0.3933	0.1847	No overtraining, list after 4th epoch
3-2-1d	linear	2	tansig	18	sigmoid	15	NA	NA	linear	10	10	1.75E+05	24.0	1.67E+05	23.4	0.2428	0.0590	0.2785	0.0776	No overtraining, list after 6th epoch
3-2-1e	linear	2	tansig	25	sigmoid	10	NA	NA	linear	25	12	1.61E+05	23.0	1.69E+05	23.6	0.3649	0.1332	0.2710	0.0734	No overtraining, list after 6th epoch
3-3-1a	linear	3	tansig	17	sigmoid	12	tansig	8	linear	20	20	1.49E+05	22.2	1.60E+05	24.3	0.4432	0.1964	0.2006	0.0402	No overtraining, list after 15th epoch
3-3-1b	MCUV	3	sigmoid	18	tansig	13	sigmoid	9	linear	150	8	1.86E+05	24.7	1.93E+05	24.5	0.0467	0.0022	-0.0891	0.0079	No overtraining, list after 5th epoch
3-3-1c	MCUV	3	tansig	20	sigmoid	15	sigmoid	10	linear	40	40	1.48E+05	22.1	1.57E+05	22.7	0.4525	0.2048	0.3534	0.1321	No overtraining, list after 7th epoch

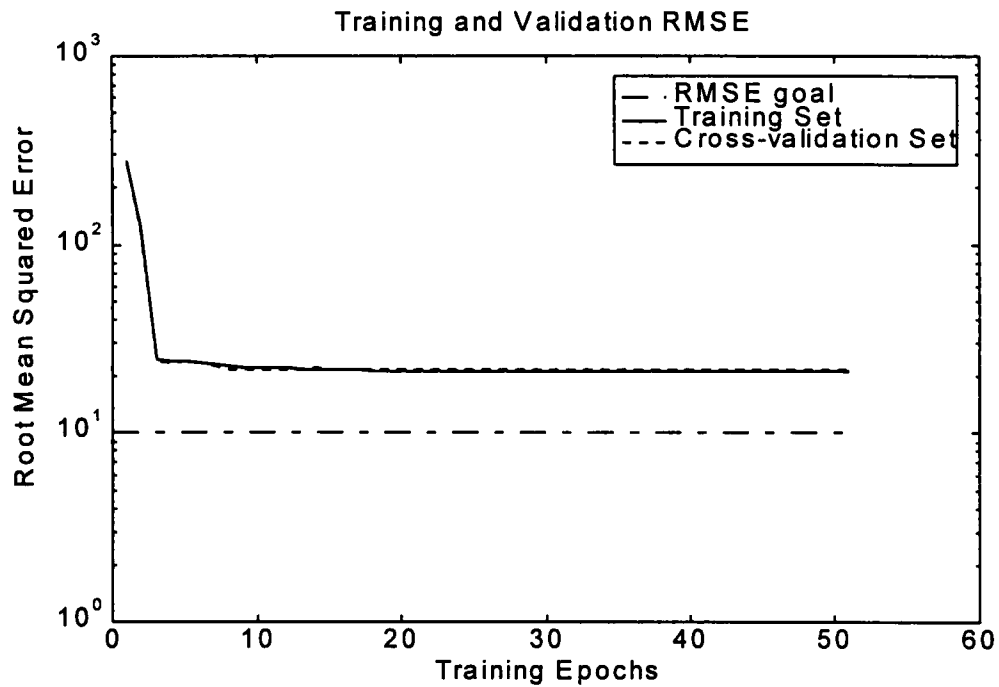


Figure 5.24 Session 3-1.1d: RMSE

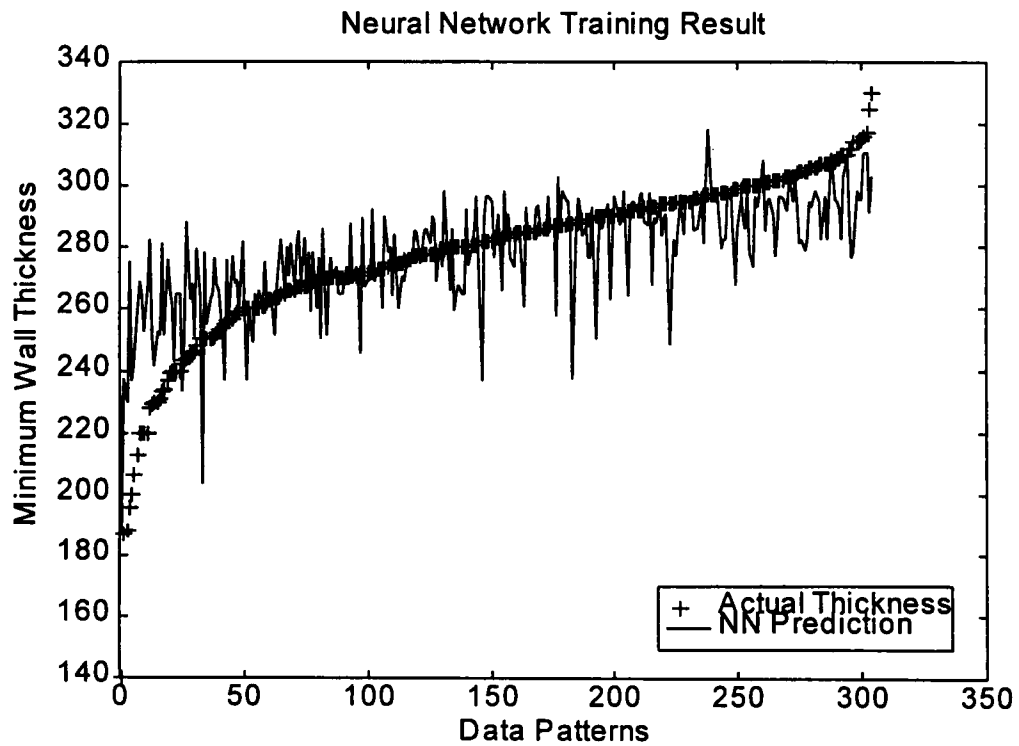


Figure 5.25 Session 3-1.1d: Prediction Results for Training Set

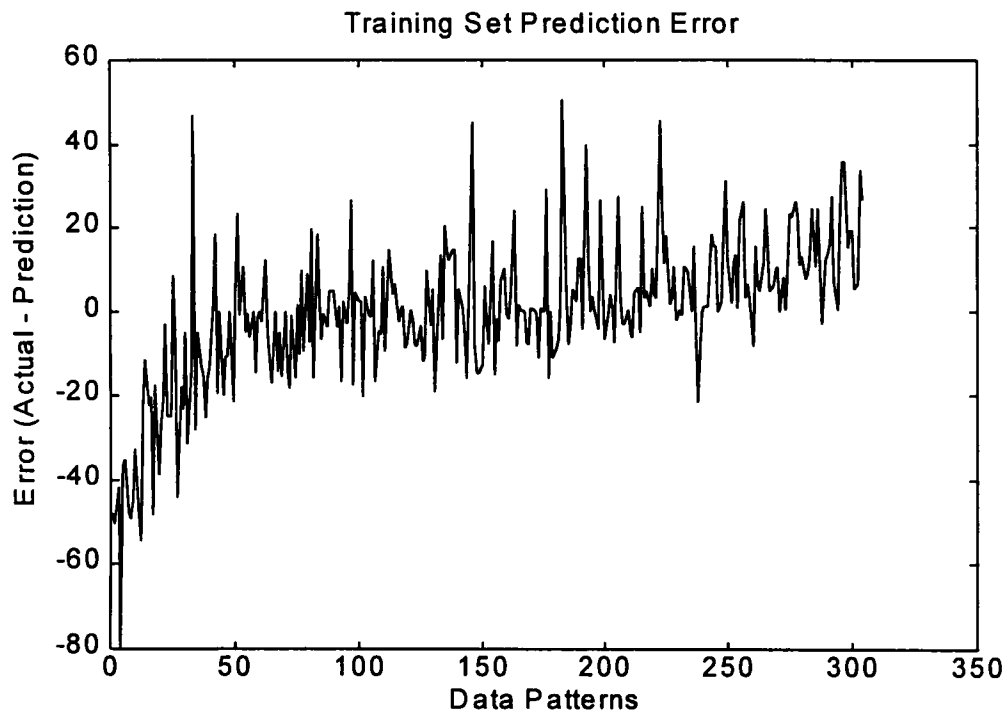


Figure 5.26 Session 3-1.1d: Training Set Prediction Error

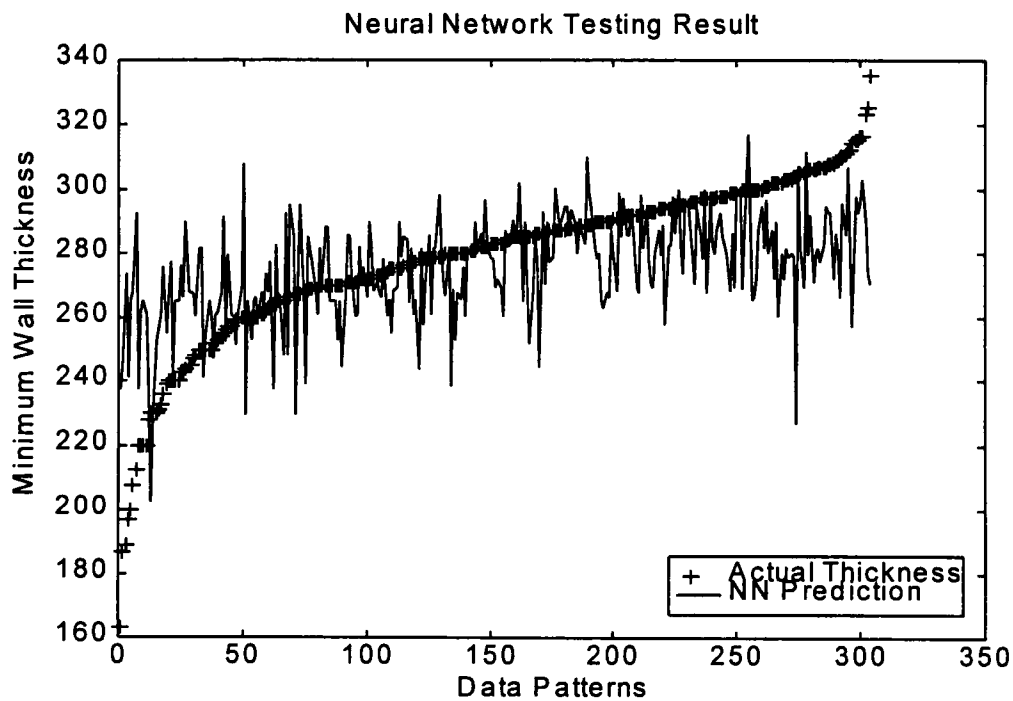


Figure 5.27 Session 3-1.1d: Prediction Results for Test Set

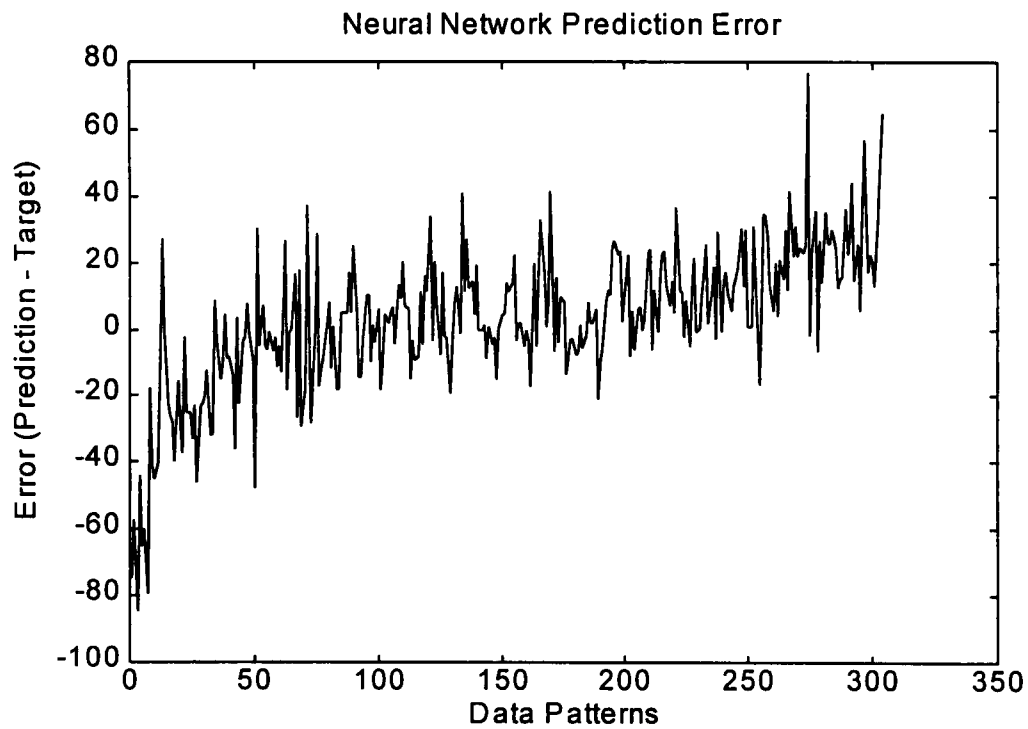


Figure 5.28 Session 3-1.1d: Test Set Prediction Error

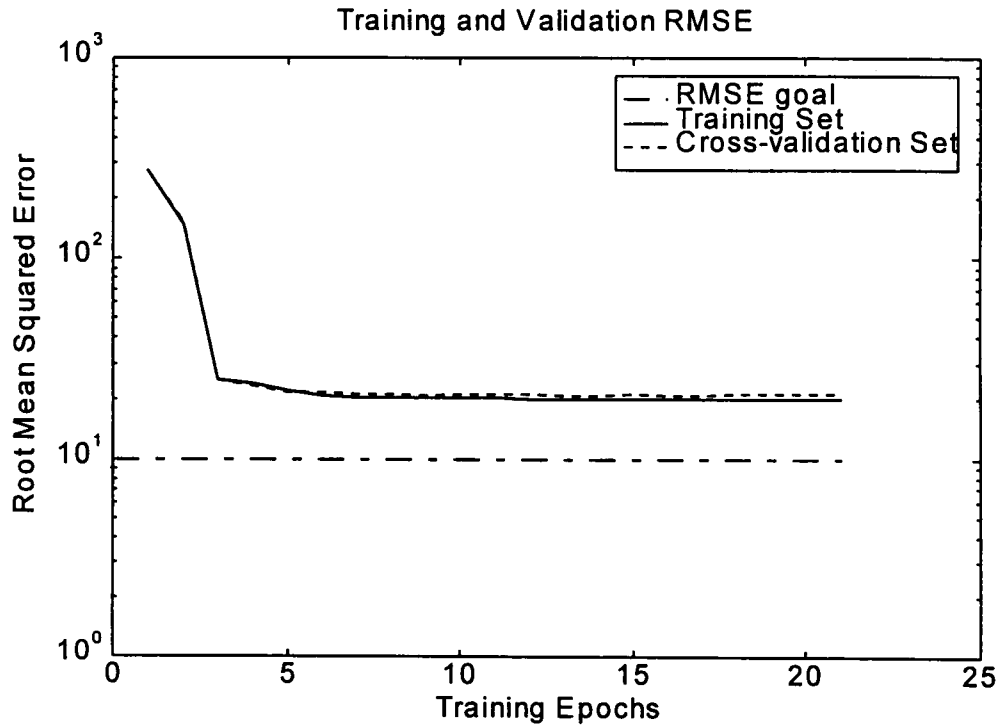


Figure 5.29 Session 3-1.1f: RMSE

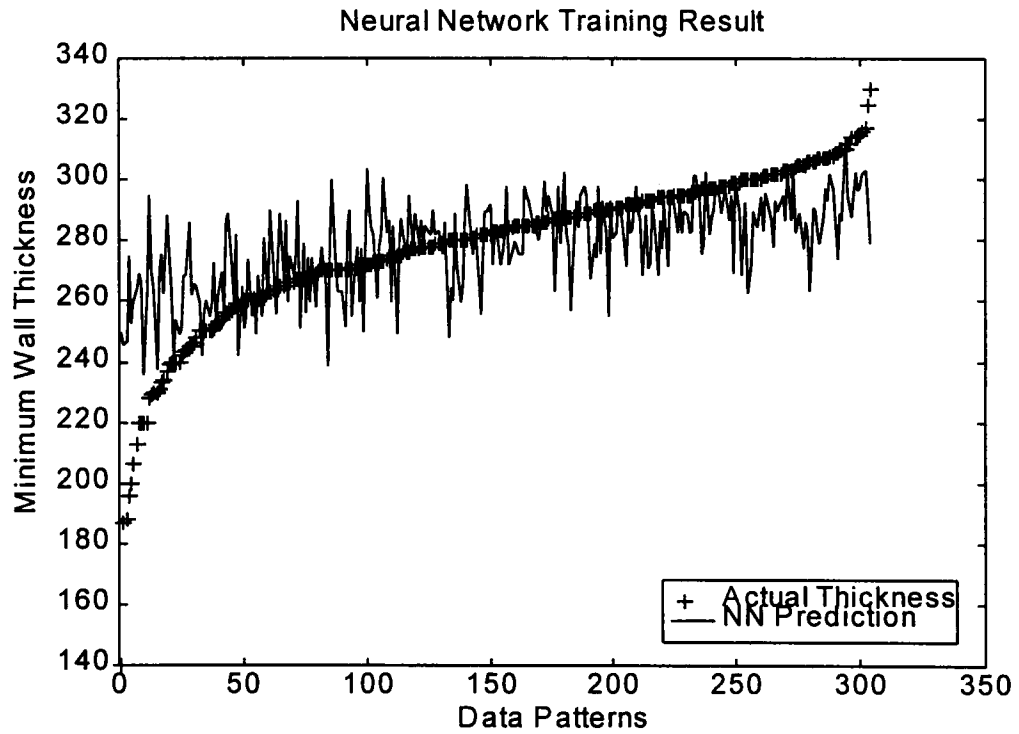


Figure 5.30 Session 3-1.1f: Prediction Results for Training Set

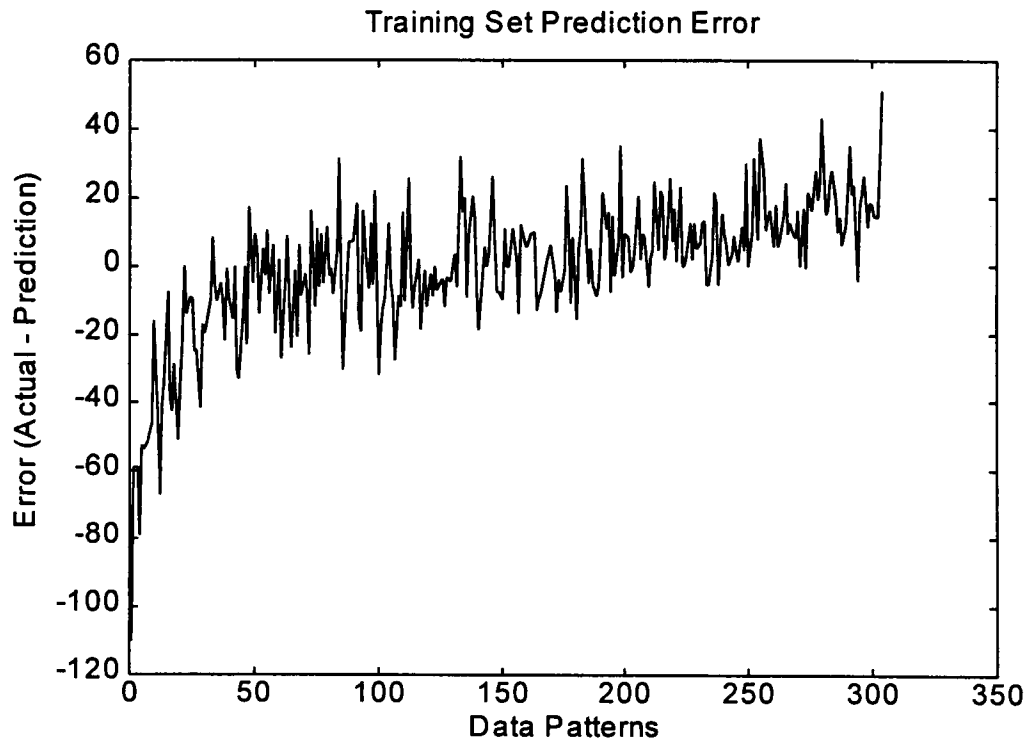


Figure 5.31 Session 2-3.1a: Training Set Prediction Error

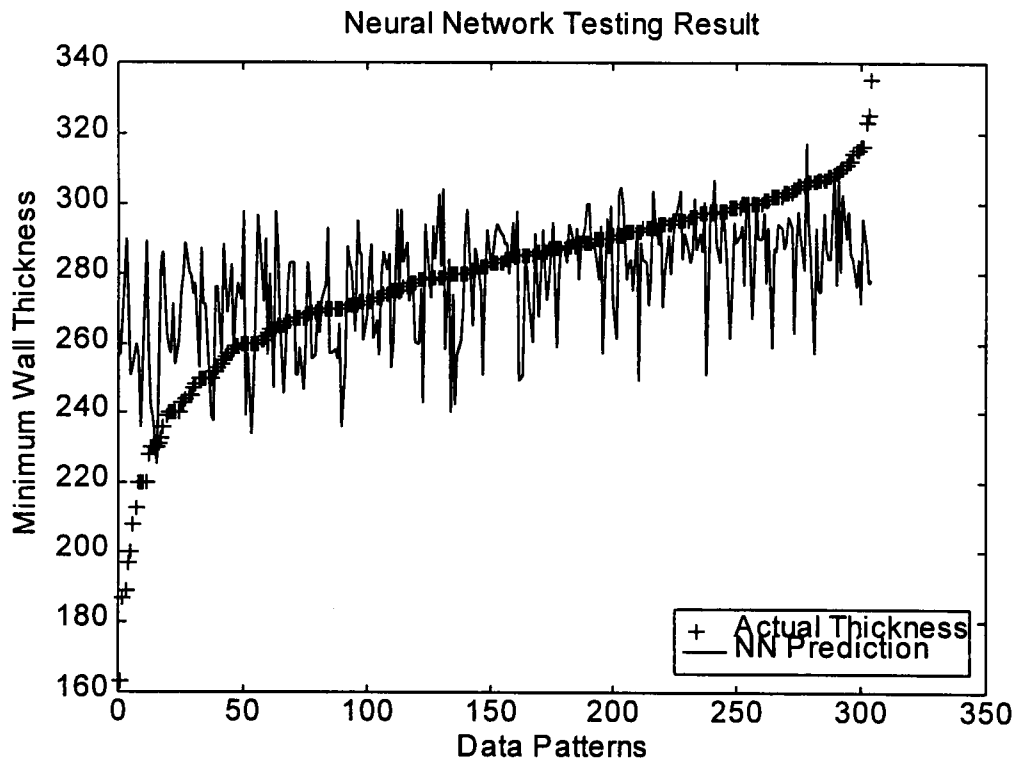


Figure 5.32 Session 2-1.2a: Prediction Results for Test Set

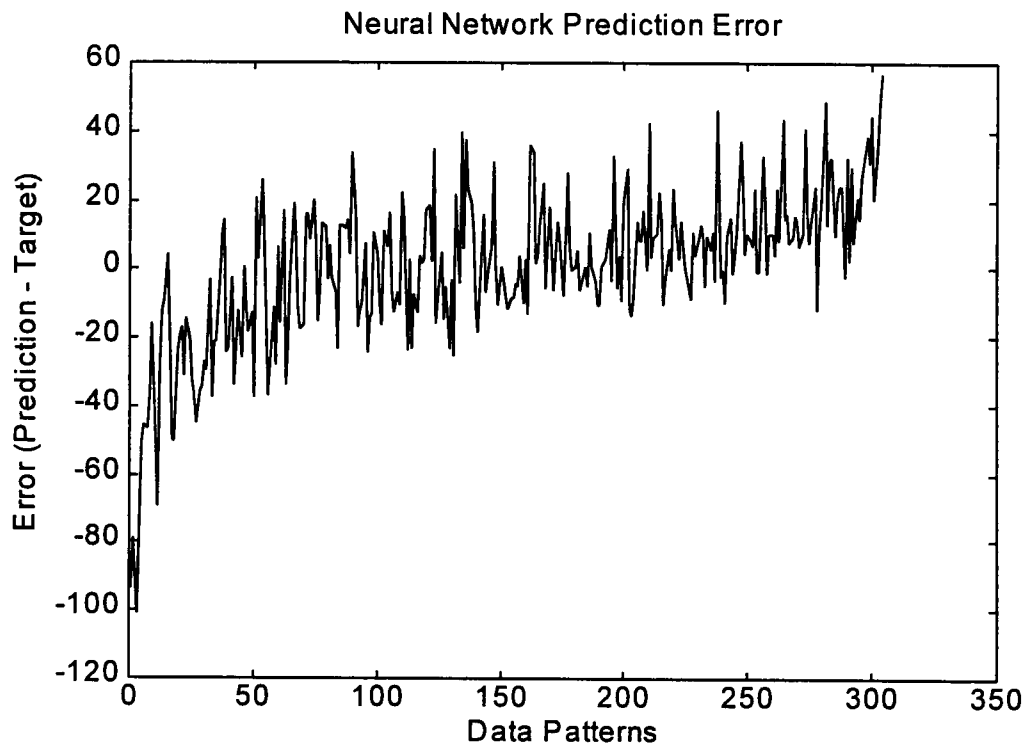


Figure 5.33 Session 3-1.1f: Test Set Prediction Error

Table 5.9: Neural Network Training Results for Classification of Cylinders Using One-in-N Codes

NUMBER OF INPUT VARIABLES: 18, coded in 1-in-N format
 TRAINING SET FILE NAME: N18var.m

TARGET RMSE: 0.1
 TARGET SSE: 3.04

TRAINING SESSION	INPUT SCALING	NUMBER OF HIDDEN LAYERS	HIDDEN LAYER 1 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 1	HIDDEN LAYER 2 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 2	HIDDEN LAYER 3 TRANSFER FUNCTION	NUMBER OF HIDDEN NEURONS IN HIDDEN LAYER 3	OUTPUT LAYER TRANSFER FUNCTION	MAXIMUM TRAINING EPOCHS	ACTUAL TRAINING EPOCHS	NETWORK TRAINING SSE	NETWORK TRAINING RMSE	TEST SET SSE	TEST SET RMSE	TRAINING SET CORREL. COEFF. (R)	TRAINING SET R^2	TEST SET CORREL. COEFF. (R)	TEST SET R^2	REMARKS
4-1.1a	MCUV	1	tansig	10	NA	NA	NA	NA	sigmoid	25	25	3.35	0.10	38.92	0.36	0.9241	0.8540	0.1098	0.0120	overtraining started @ 3rd epoch
4-1.1b	MCUV	1	tansig	2	NA	NA	NA	NA	sigmoid	500	500	11.21	0.19	31.65	0.32	0.7100	0.5041	0.0711	0.0051	overtraining started @ 3rd epoch
4-1.1c	linear	1	sigmoid	1	NA	NA	NA	NA	sigmoid	50	50	14.01	0.21	30.08	0.31	0.6154	0.3787	0.1432	0.0205	overtraining started @ 20th epoch
4-1.2c	linear	1	tansig	1	NA	NA	NA	NA	sigmoid	10	10	18.54	0.25	25.46	0.29	0.4407	0.1942	0.1464	0.0214	overtraining started @ 8th epoch
4-2.1a	MCUV	2	sigmoid	1	tansig	1	NA	NA	sigmoid	50	50	16.39	0.23	25.05	0.29	0.5226	0.2731	0.1534	0.0235	overtraining started @ 6th epoch
4-2.1b	MCUV	2	tansig	1	sigmoid	1	NA	NA	sigmoid	15	15	14.42	0.22	27.44	0.30	0.6003	0.3604	0.1769	0.0313	overtraining started @ 4th epoch
4-3.1a	linear	3	sigmoid	2	tansig	1	tansig	1	sigmoid	10	10	18.82	0.25	26.03	0.29	0.4072	0.1658	0.0533	0.0028	overtraining started @ 8th epoch
4-3.1c	linear	3	tansig	1	sigmoid	1	tansig	1	sigmoid	12	12	15.76	0.23	28.99	0.31	0.5493	0.3017	0.1055	0.0111	overtraining started @ 2nd epoch

to overtrain after only a few epochs. This occurred even with the simplest network architecture – one hidden layer and one hidden neuron. Overall, the network performance did not improve as expected with the use of One-of-N codes. In fact, the highest test set R^2 value achieved – 0.0313 (Session 4-2.1b), was much lower than that achieved in the previous section. This showed that network training did not improve with the use of One-of-N codes. Figures 5.34 – 5.38 show the training results of Session 4-2.1b.

Overall, the correct classification rate was 80.6%, although most of the correct classifications were on cylinders greater than 250 mils. Of the 31 less-than-250-mils cylinders in the test set, 23 were incorrectly identified as greater than 250 mils (74.2%). Therefore, it was concluded that even though a different coding scheme was used, the inherent information that lied in the input data set was not sufficient enough to classify the cylinders according to their minimum wall thickness.

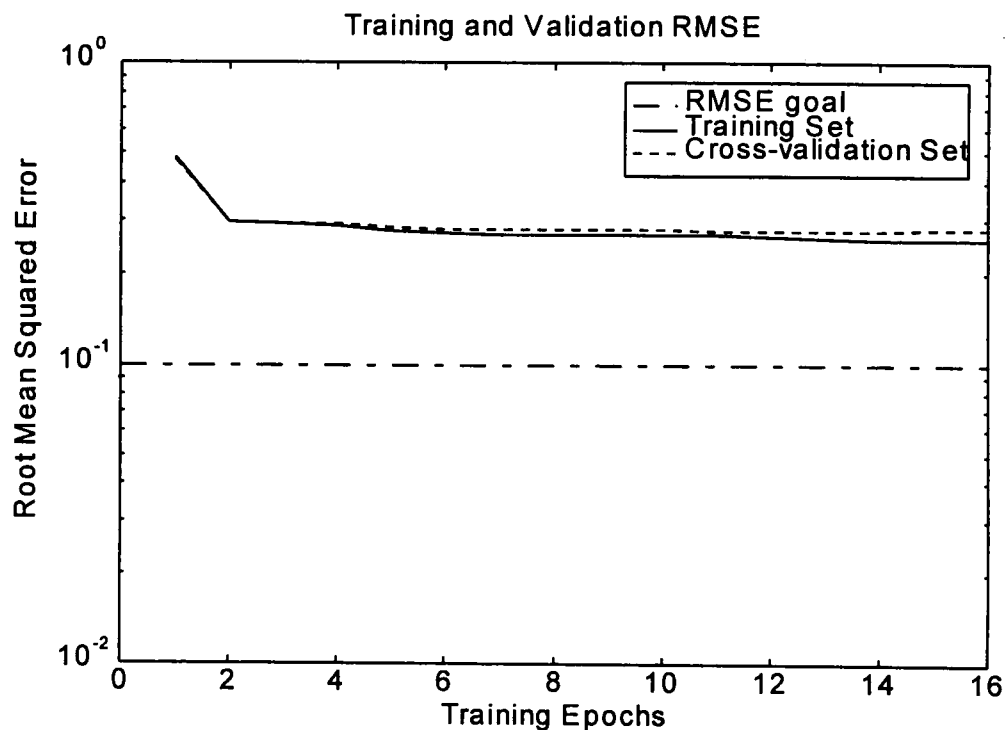


Figure 5.34 Session 4-3.1b: RMSE

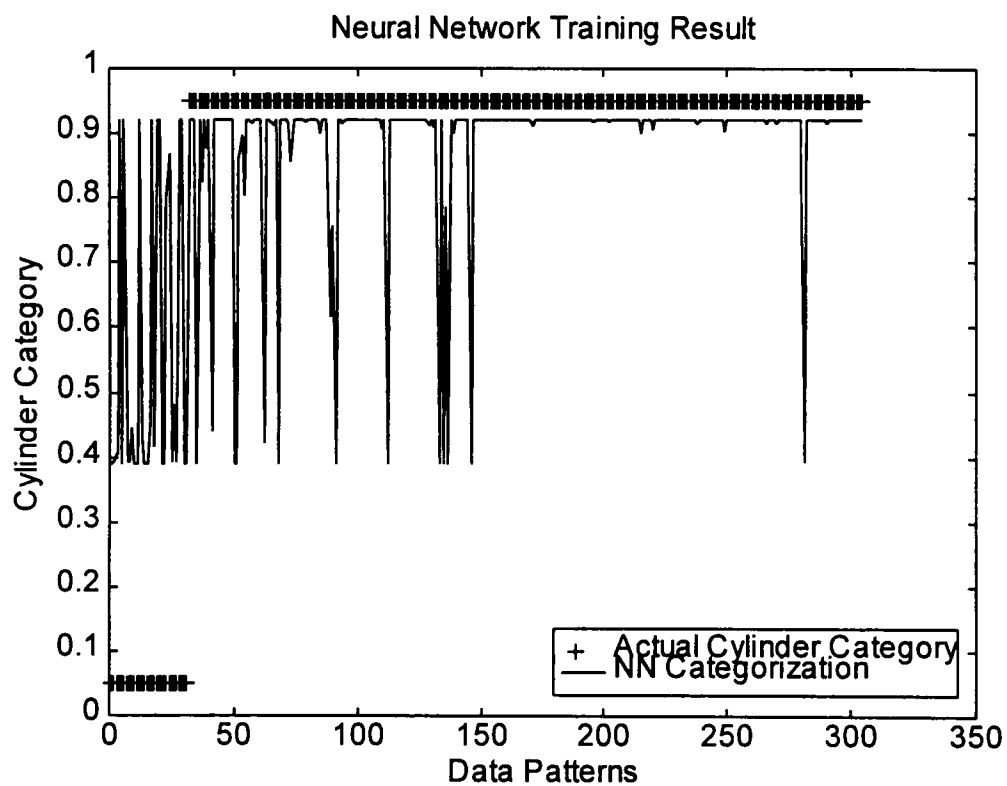


Figure 5.35 Session 4-2.1b: Prediction Results for Training Set

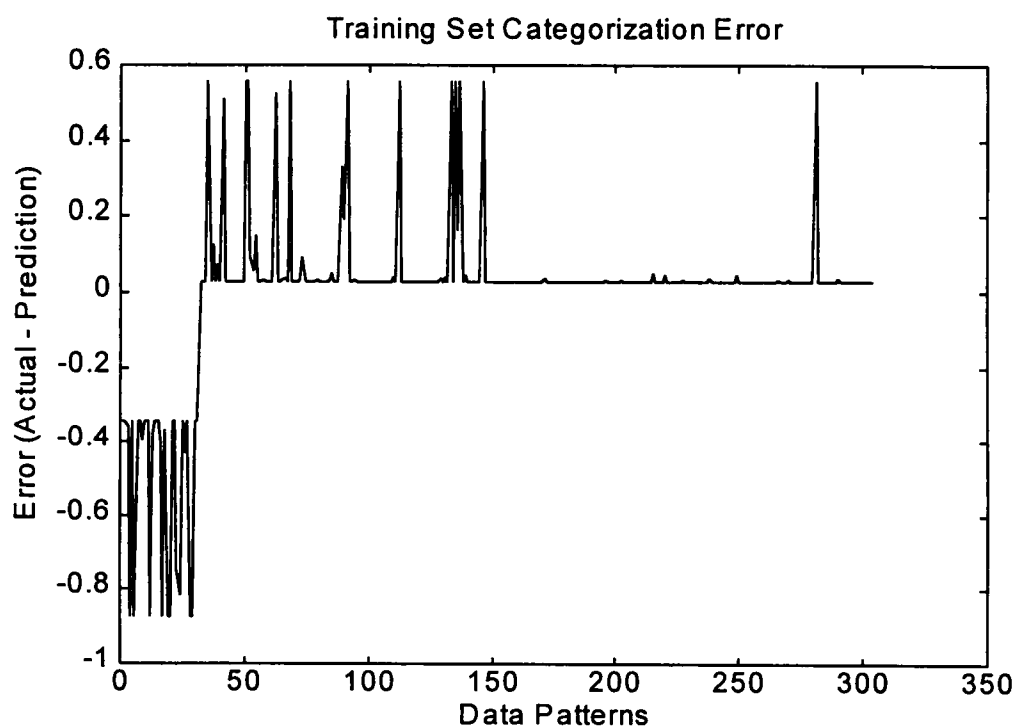


Figure 5.36 Session 4-3.1b: Training Set Prediction Error

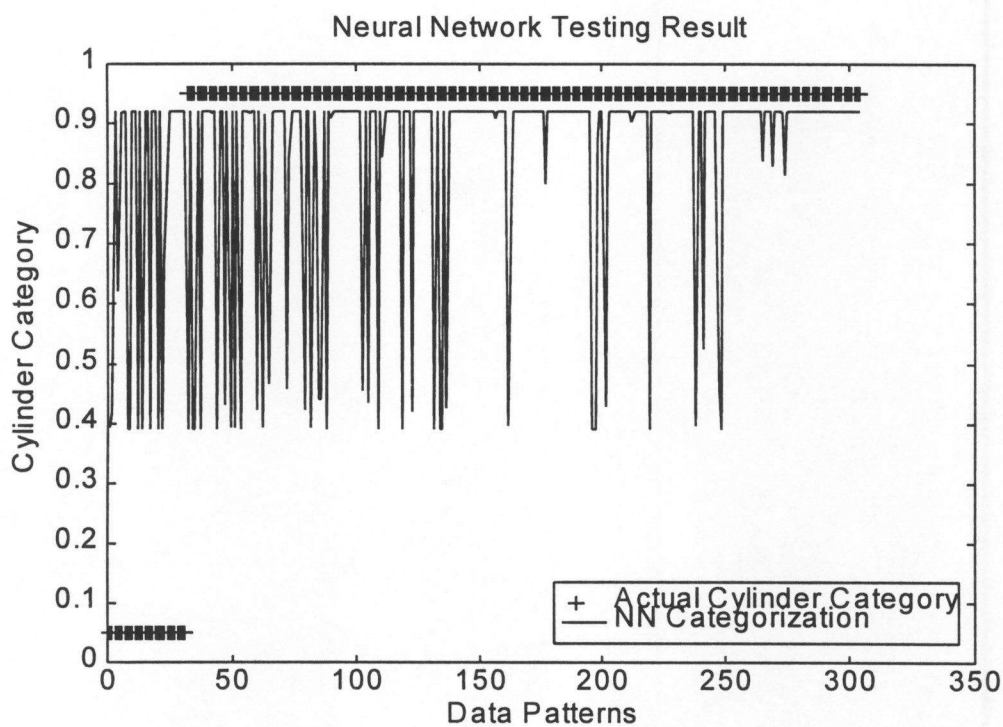


Figure 5.37 Session 4-3.1b: Prediction Results for Test Set

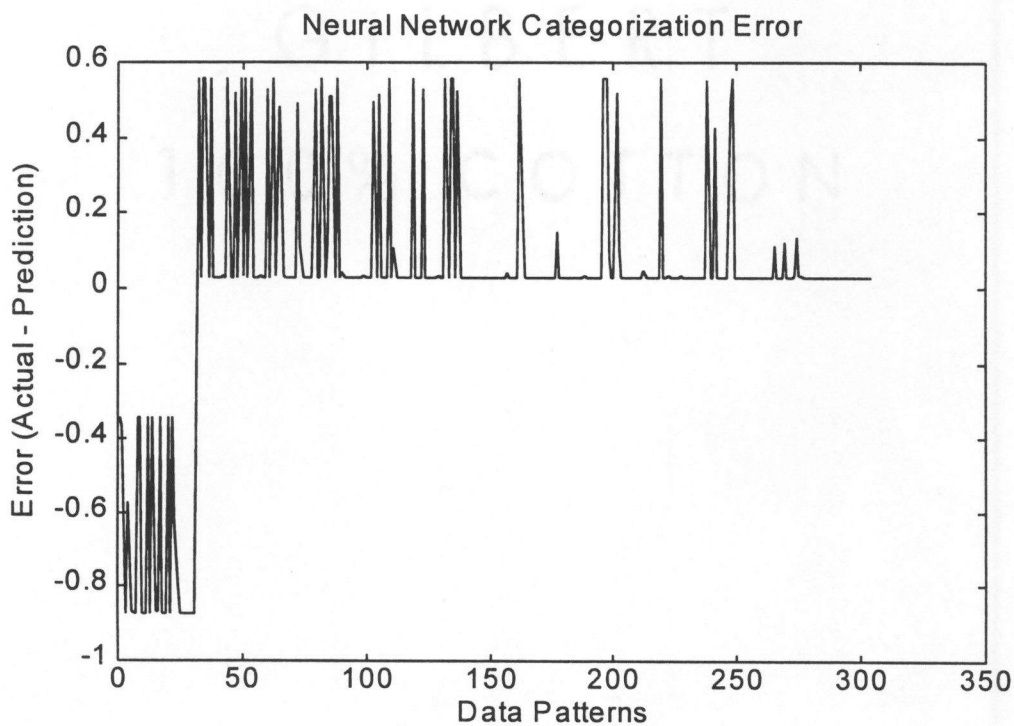


Figure 5.38 Session 4-3.1b: Test Set Prediction Error

5.4 Genetic Algorithm/Neural Network (GA/NN) Training

Since multiple linear regression and neural network training were unable to produce satisfactory results, a hybrid neural network and genetic algorithm program was used to train the data set. The software package *NeuralWorks Predict* (from here on referred to as “*Predict*”) was used to perform the so-called supportive hybrid system analysis, which involved the selection of variables by genetic algorithm for neural network training.

Predict incorporated five phases into the neural network model building process: train/test set selection, data analysis, creation of the transformed data set, input variable selection by genetic algorithm, and neural network training. Six different training sessions were carried out in which three were for the prediction of minimum wall thickness while the other three were for the categorization of the cylinders.

Tables 5.10 and 5.11 show the training parameters selected for these training sessions. The *Problem Type* parameter refers to the type of desired target output in which three selections -- prediction, ranking, and classification, are available. The prediction problem type can take on any continuous or discrete values. The ranking problem type is similar to the prediction type except that the rankings of outputs, rather than the actual values, are given. The classification problem type is suited for outputs that are exclusive categories. The prediction problem type was chosen for the prediction of minimum wall thickness while the classification problem type was used for cylinder categorization.

Table 5.10 Training Parameters Used in *Predict's* GA/NN Training (I)

	TRAINING SESSION 5-1	TRAINING SESSION 5-2	TRAINING SESSION 5-3
GENERAL TRAINING LEVEL			
Problem Type	prediction	prediction	prediction
Noise Level	noisy	very noisy	very noisy
Data Transformation Level	scale data only	comprehensive	comprehensive
Variable Selection Level	exhaustive	exhaustive	exhaustive
Network Search Level	exhaustive	exhaustive	exhaustive
VARIABLE SELECTION			
Variable Selection Model	Neural Network	Neural Network	Neural Network
Maximum Generations	30	30	30
Patience	5	5	7
Tolerance per Generation	0.001	0.001	0.001
Parent 1 Selection	rank	rank	rank
Parent 2 Selection	uniform	uniform	uniform
GENETIC ALGORITHM REPRODUCTION			
Cross-over Probability	0.7	0.7	0.7
Mutation Factor	1	1	1
Elitist Fraction	0.05	0.05	0.05
NEURAL NETWORK			
Network Architecture	10-3-1	8-0-1	15-1-1
Network Type	adaptive gradient	Kalman filtering	adaptive gradient
Number of Input Variables	10	8	15
Hidden Neurons	3	0	1
Active Fields (Variables)	10	8	12
Total Transforms	50	88	88
Active Transforms	10	8	15
Output Transfer Function	linear	sigmoid	linear
Evaluation Function	root mean power error	root mean power error	root mean power error
Internal Correlation	0.1551	0.1804	0.1868
Direct Connections	yes	yes	yes
Cascaded Connections	yes	yes	yes

Table 5.11 Training Parameters Used in Predict's GA/NN Training (II)

	TRAINING SESSION 6-1	TRAINING SESSION 6-2	TRAINING SESSION 6-3
GENERAL TRAINING LEVEL			
Problem Type	classification	classification	classification
Noise Level	noisy	very noisy	very noisy
Data Transformation Level	comprehensive	comprehensive	comprehensive
Variable Selection Level	exhaustive	exhaustive	exhaustive
Network Search Level	exhaustive	exhaustive	exhaustive
VARIABLE SELECTION			
Variable Selection Model	Neural Network	Neural Network	Neural Network
Maximum Generations	30	50	30
Patience	5	5	5
Tolerance per Generation	0.001	0.001	0.001
Parent 1 Selection	rank	rank	rank
Parent 2 Selection	uniform	uniform	uniform
GENETIC ALGORITHM REPRODUCTION			
Cross-over Probability	0.7	0.7	0.7
Mutation Factor	1	1	1
Elitist Fraction	0.05	0.05	0.05
NEURAL NETWORK			
Network Architecture	5-3-1	3-5-1	5-5-1
Network Type	adaptive gradient	adaptive gradient	adaptive gradient
Number of Input Variables	5	3	5
Hidden Neurons	3	5	5
Active Fields (Variables)	5	4	3
Total Transforms	31	31	32
Active Transforms	5	5	3
Output Transfer Function	SoftMax	sigmoid	sigmoid
Evaluation Function	average classification rate	classification rate	correlation
Internal Correlation	0.7938	0.7826	0.3521
Direct Connections	yes	yes	yes
Cascaded Connections	yes	yes	yes

Worst case scenarios (noisy and very noisy data) were assumed for the *Noise Level* parameter since the results from the previous neural network training were not satisfactory. Except for Training Session 5-1 which scaled the data only, comprehensive data transformation was applied to input data for all training sessions. Data transformation refers to the transformation of original variables into new variables. For example, new variables can be created from original variables by finding the square root or log values of the original variables.

A genetic algorithm is used by the input variable selection process to search for good subsets of the full input variable set created during the data analysis and transformation process. Each input variable subset is assigned a fitness value. The fitness value is based on how successful a model can be built using only the variables in the subset. Variables that carry no significant information or are not related to the output variable have smaller chances of being selected. The objective function used to determine the fitness is the root mean power error function, which is shown in Equation 5.1:

$$e = \sqrt[\mu]{\frac{1}{T} \sum_{t=1}^T \frac{1}{M} \sum_{m=1}^M (d_m^{(t)} - y_m^{(t)})^\mu} \quad (5.1)$$

For Training Sessions 5-1, 5-2, 5-3, 6-1, 6-2, and 6-3, the exhaustive mode was used for the *Variable Selection* and *Network Search* parameters. This meant that several subsets of input variables were selected to train a few neural networks with the best subset and network retained. Default values were used for the genetic algorithm search process.

5.4.1 Prediction of Minimum Wall Thickness

As shown in Table 5.10, three training sessions for the prediction of minimum wall thickness were run. The architectures for the three networks were 10-3-1, 8-0-1, and 15-1-1. The three values represented the number of input variables, hidden neurons, and output variables respectively. Table 5.12 shows the input variables selected by the genetic algorithm and the data transformation, if any, that the variables underwent. Table 5.13 shows the errors of the training and test set for Sessions 5-1, 5-2, and 5-3. 638 out of 912 sample cylinders were used in the training process while the remaining 274 were used as test patterns. The value R refers to the linear correlation between the target output and the actual model output. The average absolute error, maximum absolute error, and the root-mean-squared error (RMSE) are the errors between the target output and the actual model output. The R^2 and RMSE values obtained from the GA/NN model show a slight improvement from the values obtained from the network training using *MATLAB*.

Table 5.13 lists the errors of the training and test set for Sessions 5-1, 5-2, and 5-3. The value R is the correlation between the target output and the actual model output. The R^2 and error values obtained from the GA/NN model did not show any improvement over the results obtained from the neural network training using *MATLAB*.

The training results for Sessions 5-1, 2, and 3 are shown in Figures 5.39 – 5.44. The plots show that the model was producing mostly random prediction values much like the results obtained from Session 3 from neural network training. It is obvious from the plots that none of the three GA/NN models are capable of predicting the cylinders' minimum wall thickness.

Table 5.12 GA Selected Input Variables and Their Data Transformation (I)

Training Session 5-1		Training Session 5-2		Training Session 5-3	
Input Variables	Data Transformation	Input Variables	Data Transformation	Input Variables	Data Transformation
Site	Identity function	Site	$\log[x/(1-x)]$	Yard	Square function
Yard	Identity function	Row	Logical	Row	Logical
				Age	
Row	Logical	Plant	Hyperbolic tangent function	When Measured	Identity function
Drawing	Identity function	Modifier	Identity function		Exp of Exp
					Hyperbolic tangent function
Skirt	Logical	Drawing	$\ln[x/(1-x)]$	Plant	$\log[x/(1-x)]$
Assay	Identity function	Assay	Identity function		$\log[x/(1-x)]$
Tare	Identity function	Tare	$\log$	Drawing	$\log[x/(1-x)]$
A05	Logical	B02	Logical	PO	square function
				ORD_yymmdd	Identity function
C06	Logical				Inverse function (1/x)
C13	Logical				Identity function
				Assay	Identity function
				Tare	Square root function
					Fuzzy right transformation
				Net	
				B02	Logical
				C17	Logical

Table 5.13 GA/NN Model Linear Correlation and Error Values (I)

	Data Set	Input Patterns	R	R^2	Average Absolute Error	Maximum Absolute Error	Root Mean Squared Error
Training Session 5-1	All	912	0.59	0.35	14.13	106.27	19.65
	Train	638	0.58	0.33	14.40	106.27	20.18
	Test	274	0.62	0.39	13.49	73.02	18.34
Training Session 5-2	All	912	0.59	0.34	14.20	103.15	19.99
	Train	638	0.58	0.34	14.19	103.15	20.06
	Test	274	0.60	0.36	14.23	101.02	19.81
Training Session 5-3	All	912	0.57	0.33	14.42	110.55	20.28
	Train	638	0.56	0.32	14.44	110.55	20.35
	Test	274	0.59	0.35	14.38	94.69	20.13

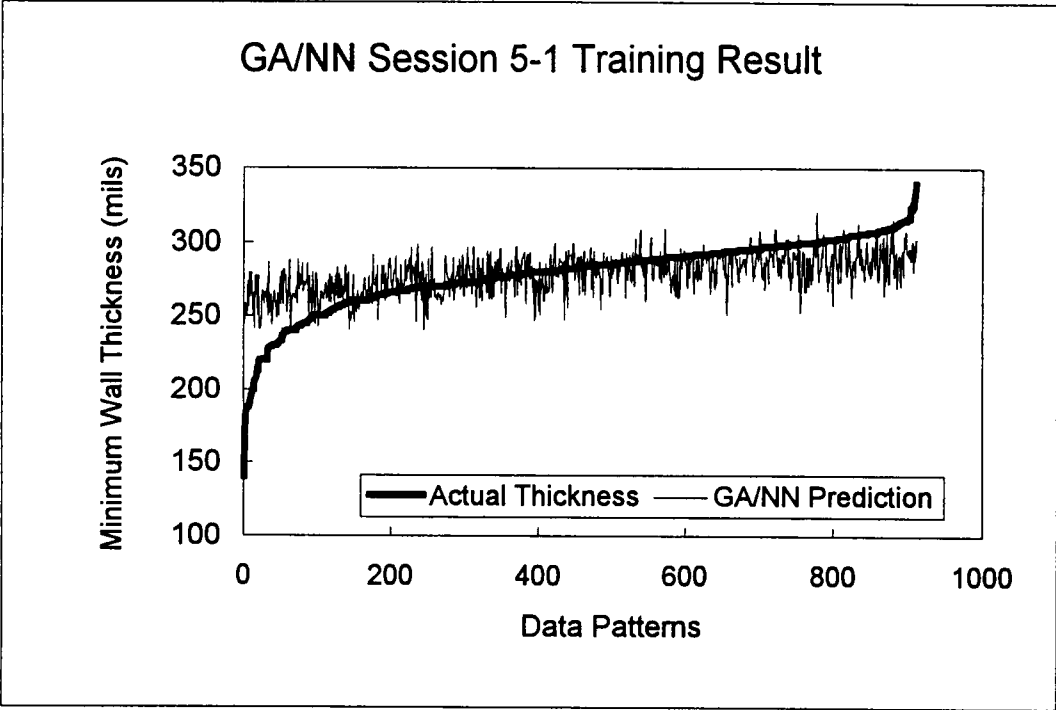


Figure 5.39 Session 5-1: Prediction Results for Training Set

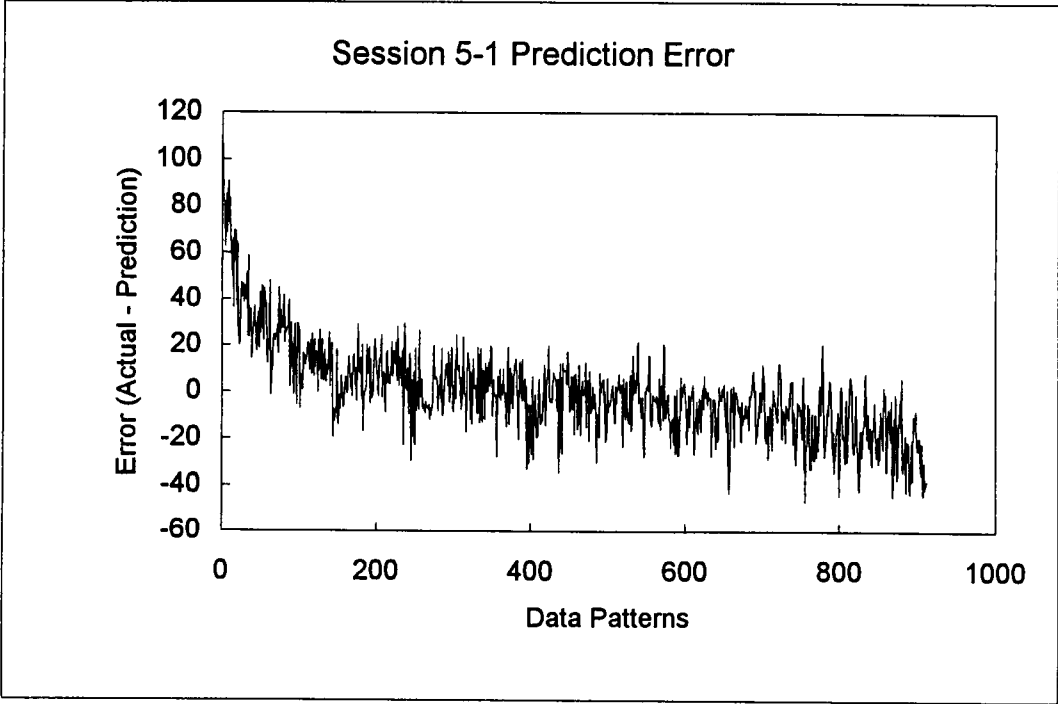


Figure 5.40 Session 5-1: Training Set Prediction Error

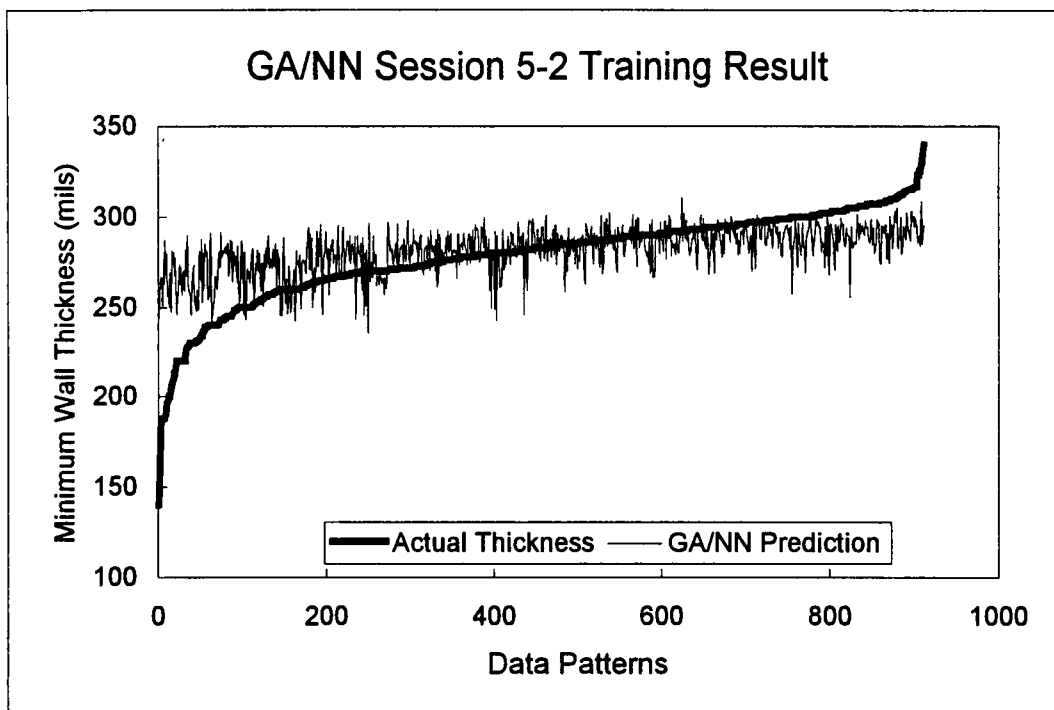


Figure 5.41 Session 5-2: Prediction Results for Training Set

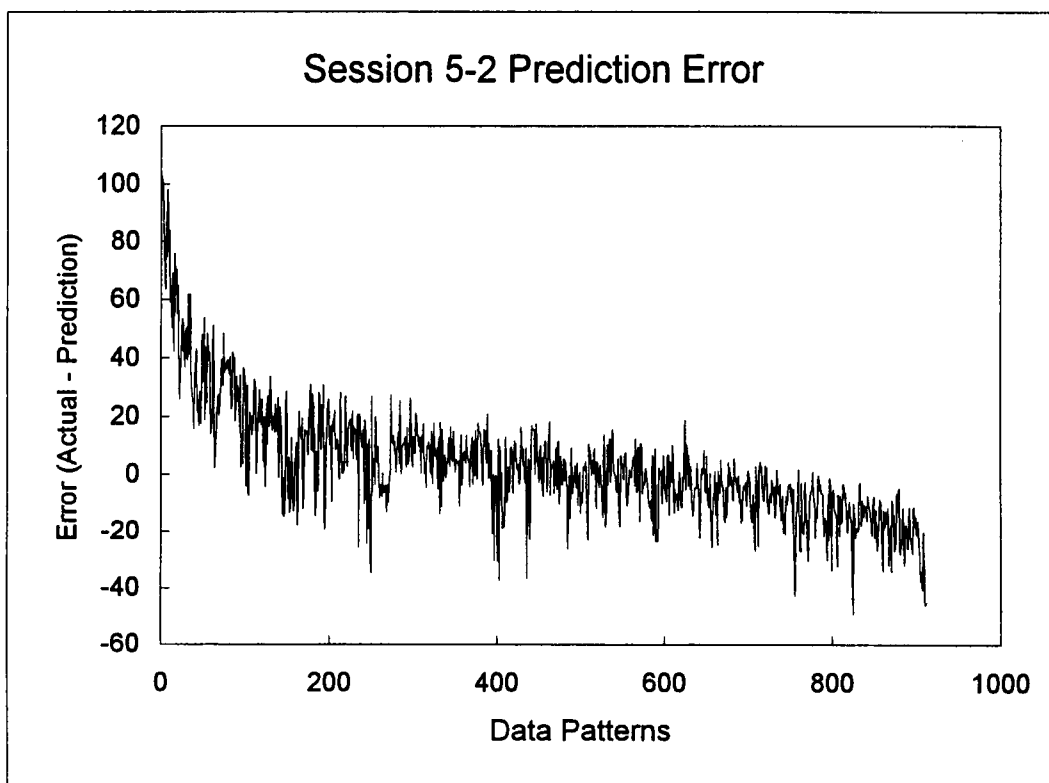


Figure 5.42 Session 5-2: Training Set Prediction Error

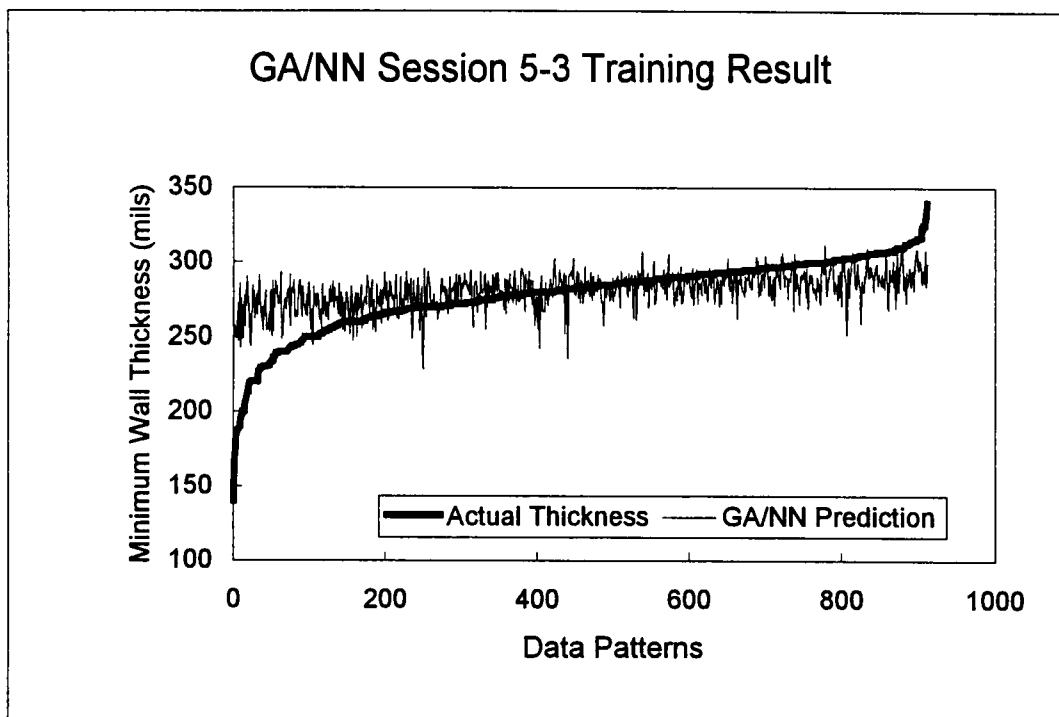


Figure 5.43 Session 5-3: Prediction Results for Training Set

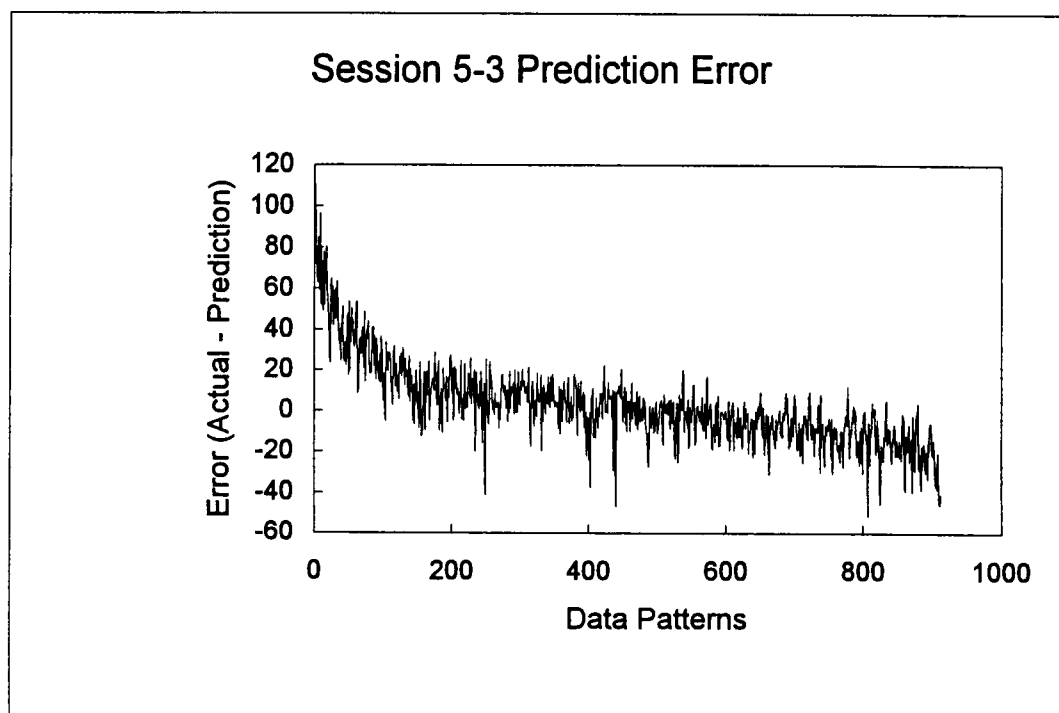


Figure 5.44 Session 5-3: Training Set Prediction Error

5.4.2 Classification of Cylinders

Three training sessions for the classification of cylinders were run and the training parameters were shown earlier in Table 5.11. The network architectures of the three models were 5-3-1, 3-5-1, and 5-5-1 respectively. Input variables selected by the genetic algorithm and the data transformation, if any, that these variables underwent are shown in Table 5.14. Table 5.15 lists the errors of the training and test set for Sessions 6-1, 6-2, and 6-3. The value R is the correlation between the target output and the actual model output. The R^2 and error values obtained from the GA/NN model did not show any improvement over the results obtained from the neural network training using *MATLAB*.

Figures 5.44 – 5.56 show the training and testing results for Sessions 6-1, 6-2, and 6-3. Similar to the results from previous training sessions, the GA/NN model could not correctly categorize cylinders into the two desired categories. Based on the results from both the neural network training and GA/NN models, it was concluded that even with the help of a genetic algorithm, there was no sufficient inherent information in the input data set to either predict the minimum wall thickness or classify the UF₆ cylinders. The number of permutations of *Predict*'s training parameters is almost endless. The GA/NN models used here assumed the worst case scenarios and recorded results similar to those of *MATLAB*'s neural network training. Therefore, it was not encouraging to spend more time and resources searching for either a better neural network or GA/NN model.

Table 5.14 GA Selected Input Variables and Their Data Transformation (II)

Training Session 6-1		Training Session 6-2		Training Session 6-3	
Input Variables	Data transformation	Input Variables	Data transformation	Input Variables	Data transformation
Yard	Fuzzy right transformation	Cost	Log of log	Yard	Identity function
ORD_yymmdd	log	B07	logical	Cost	1/(Square power function)
Tare	Fuzzy right transformation	Skirt	logical	Row	logical
B07	logical			Skirt	logical
Assay	1/(Fourth root function)				

Table 5.15 GA/NN Model Linear Correlation and Error Values (II)

	Data Set	Input Patterns	R	R^2	Maximum Absolute Error	Sum Squared Error	Root Mean Square Error
Training Session 6-1	Training	304	0.17	0.03	0.9	95.58	0.56
	Test	304	0.00	0.00	0.9	25.11	0.29
Training Session 6-2	Training	304	0.28	0.08	0.9	45.36	0.39
	Test	304	0.12	0.02	0.9	53.46	0.42
Training Session 6-3	Training	304	0.36	0.13	0.9	47.79	0.40
	Test	304	0.22	0.05	0.9	50.22	0.41

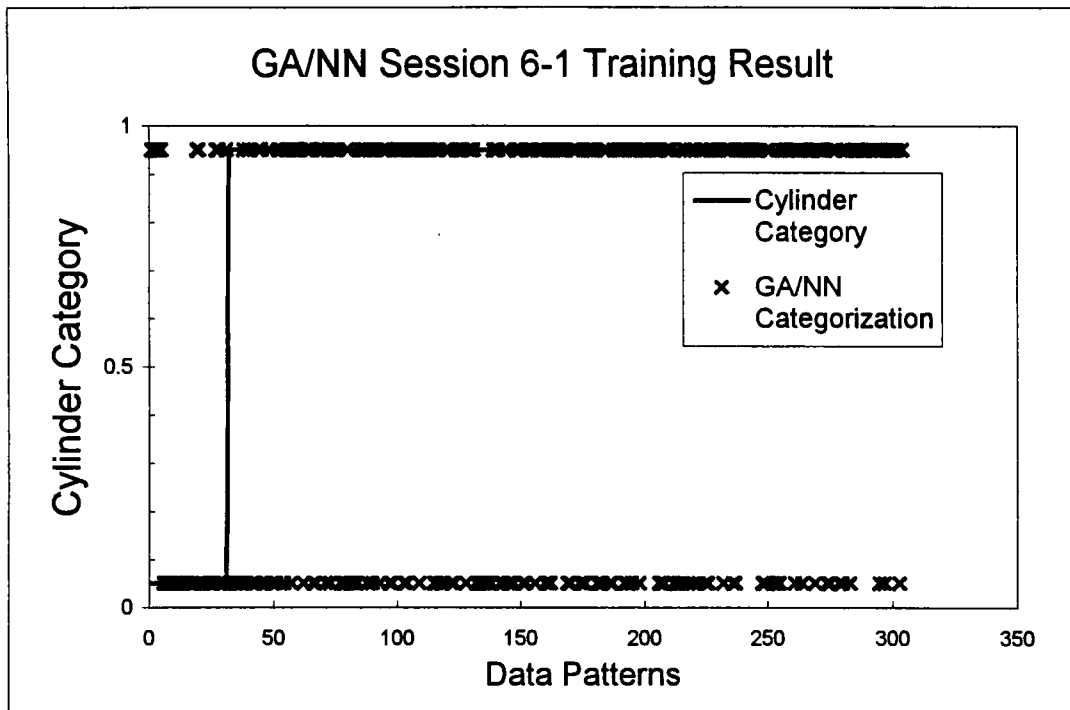


Figure 5.45 Session 6-1: Prediction Results for Training Set

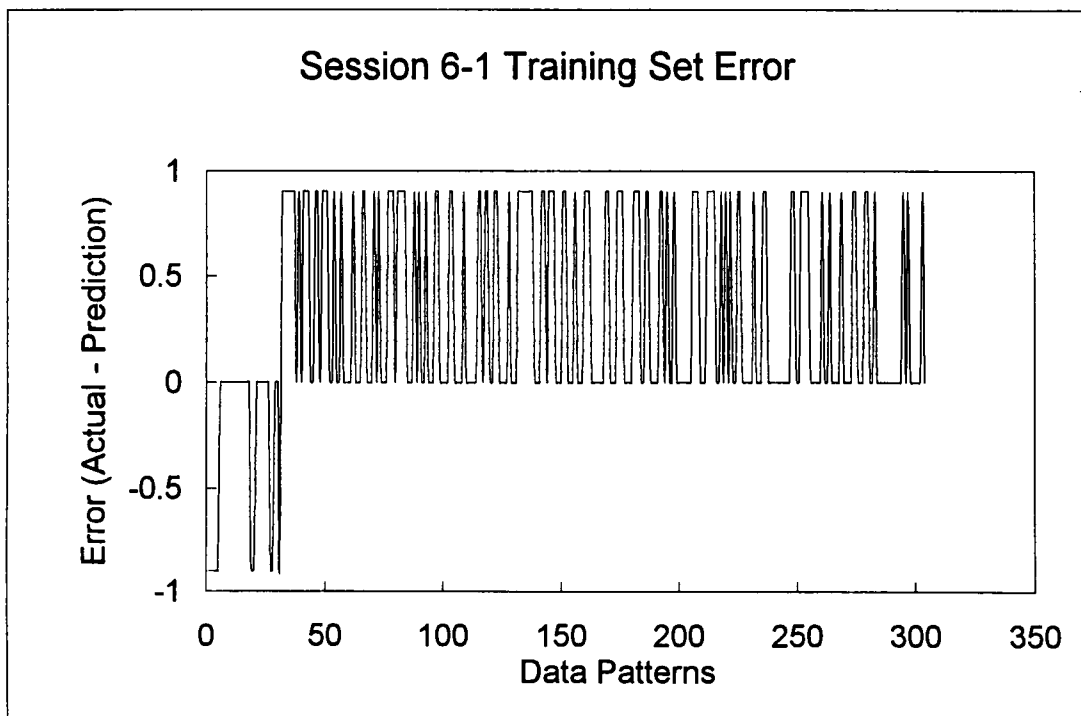


Figure 5.46 Session 6-1: Training Set Prediction Error

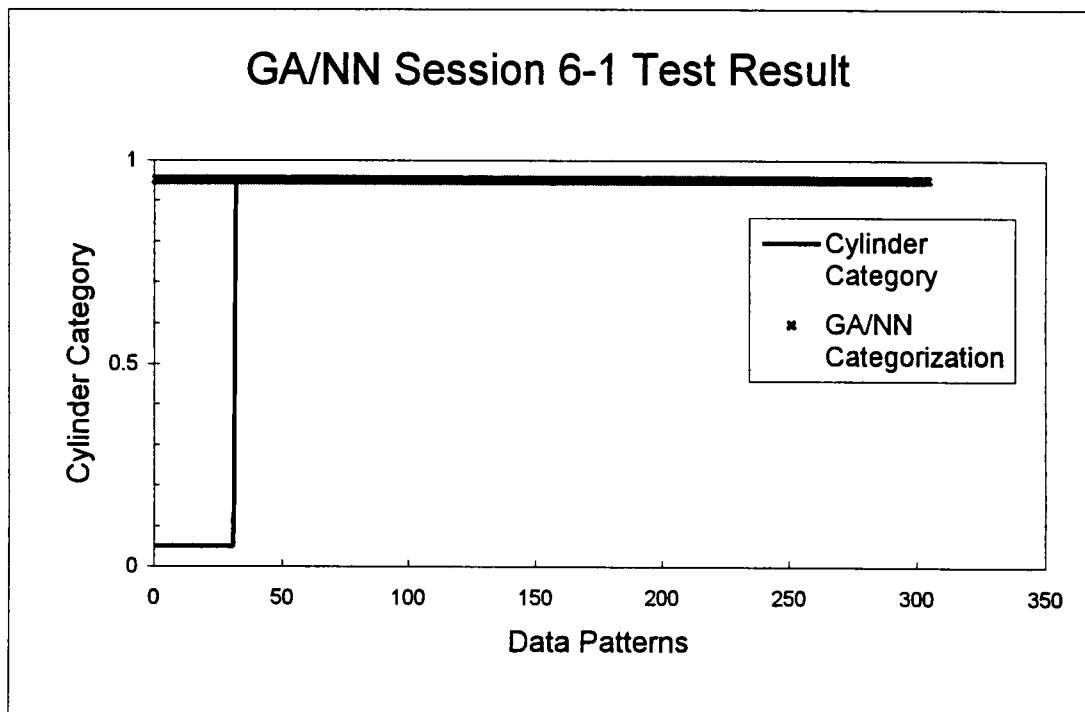


Figure 5.47 Session 6-1: Prediction Results for Test Set

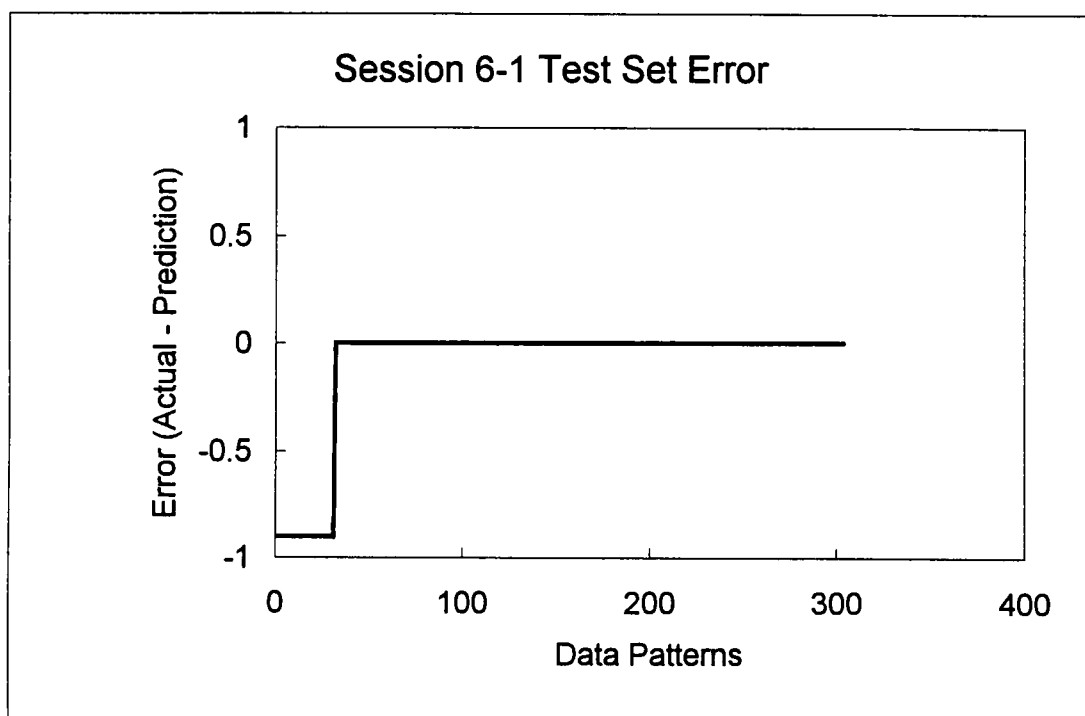


Figure 5.48 Session 6-1: Test Set Prediction Error

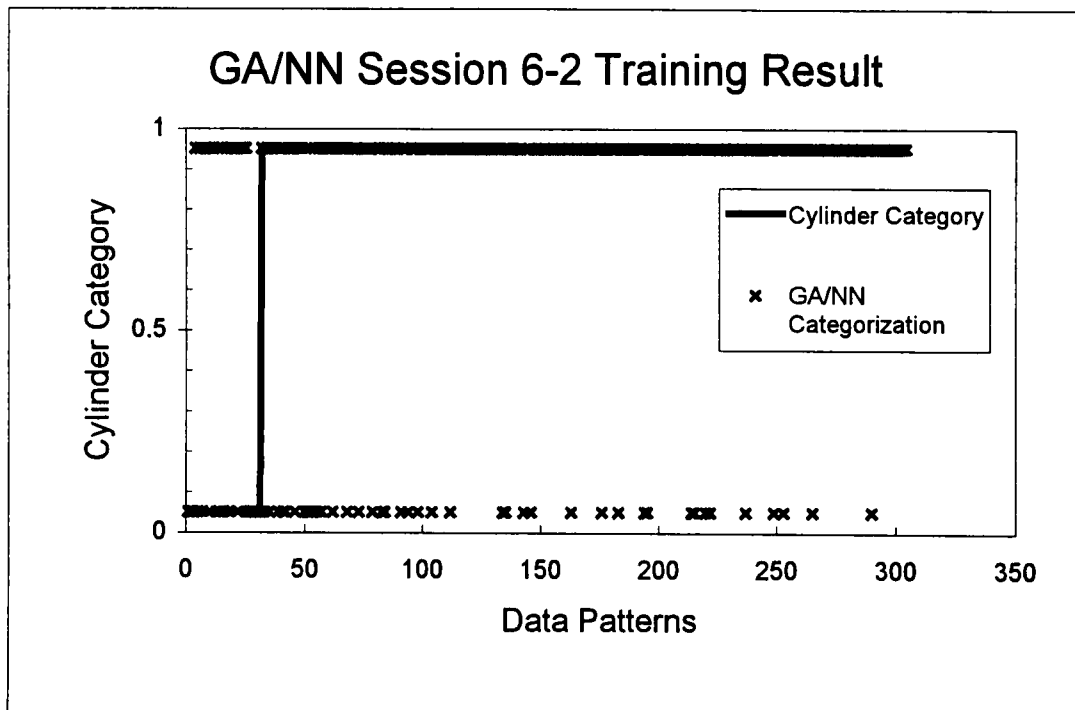


Figure 5.49 Session 6-2: Prediction Results for Training Set

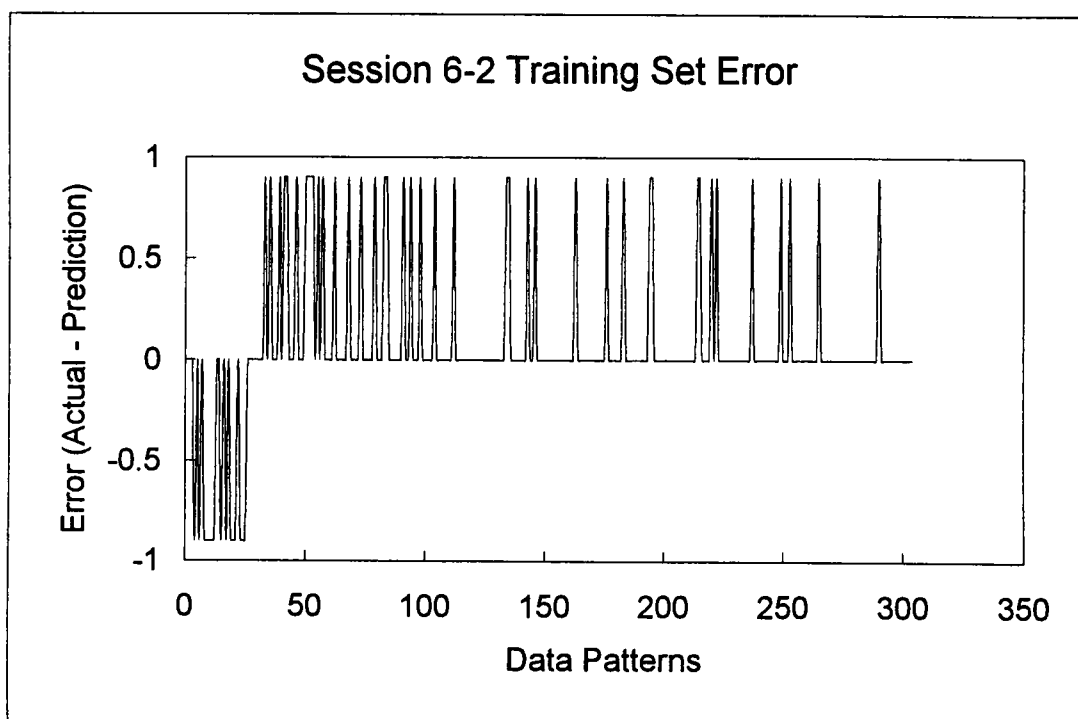


Figure 5.50 Session 6-2: Training Set Prediction Error

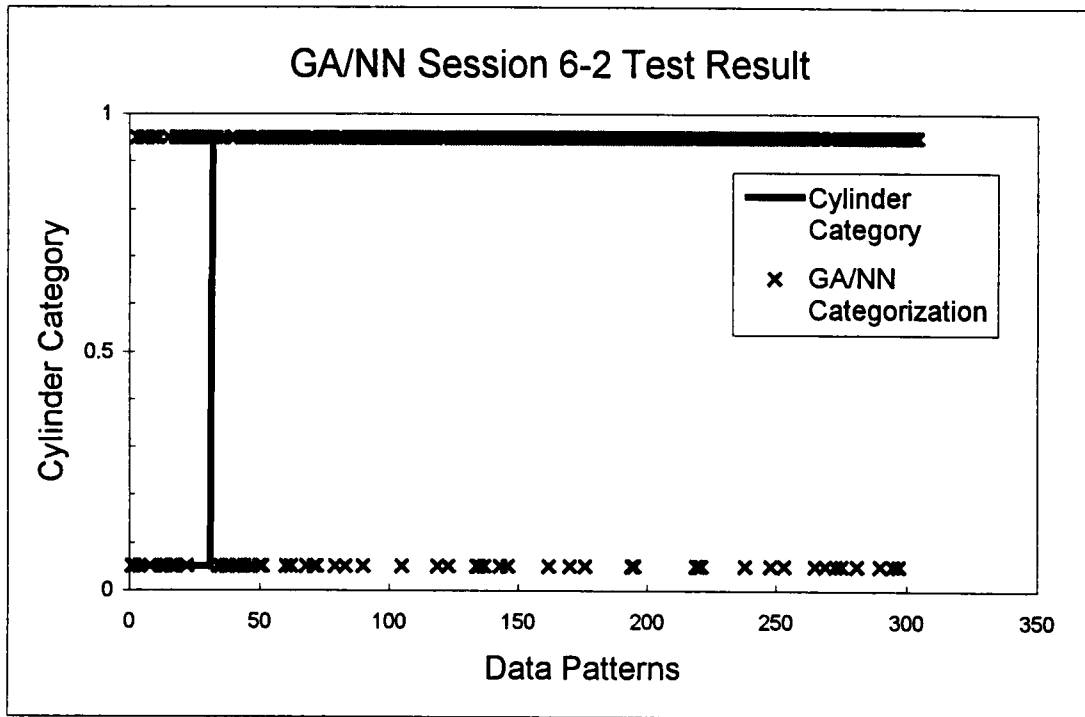


Figure 5.51 Session 6-2: Prediction Results for Test Set

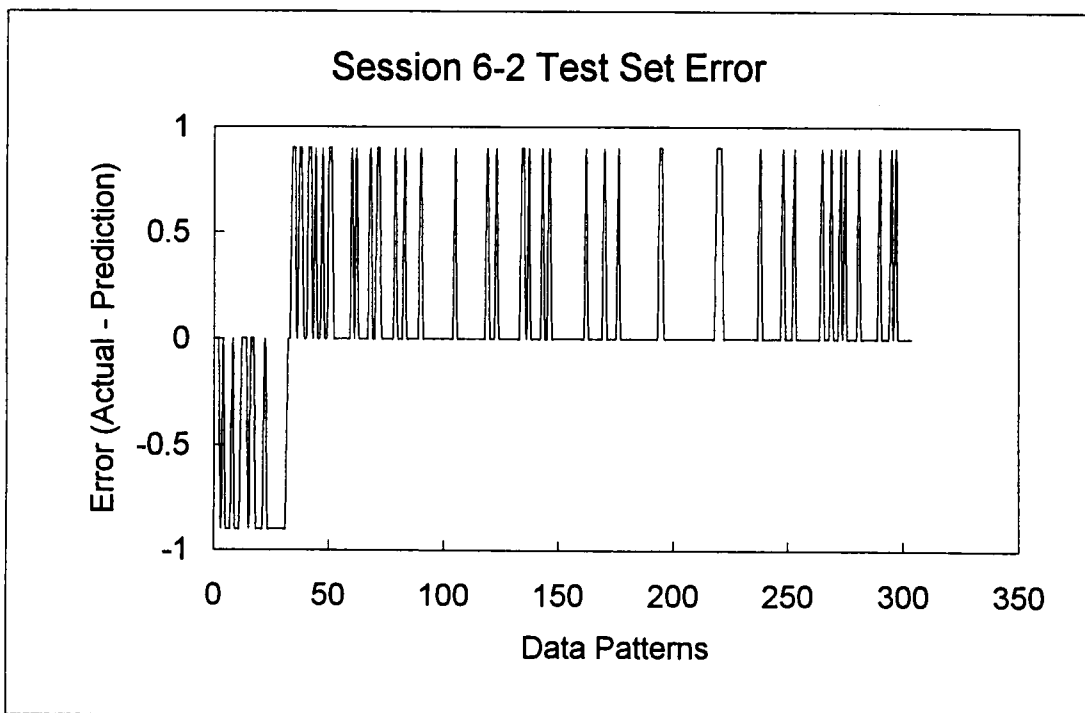


Figure 5.52 Session 6-2: Test Set Prediction Error

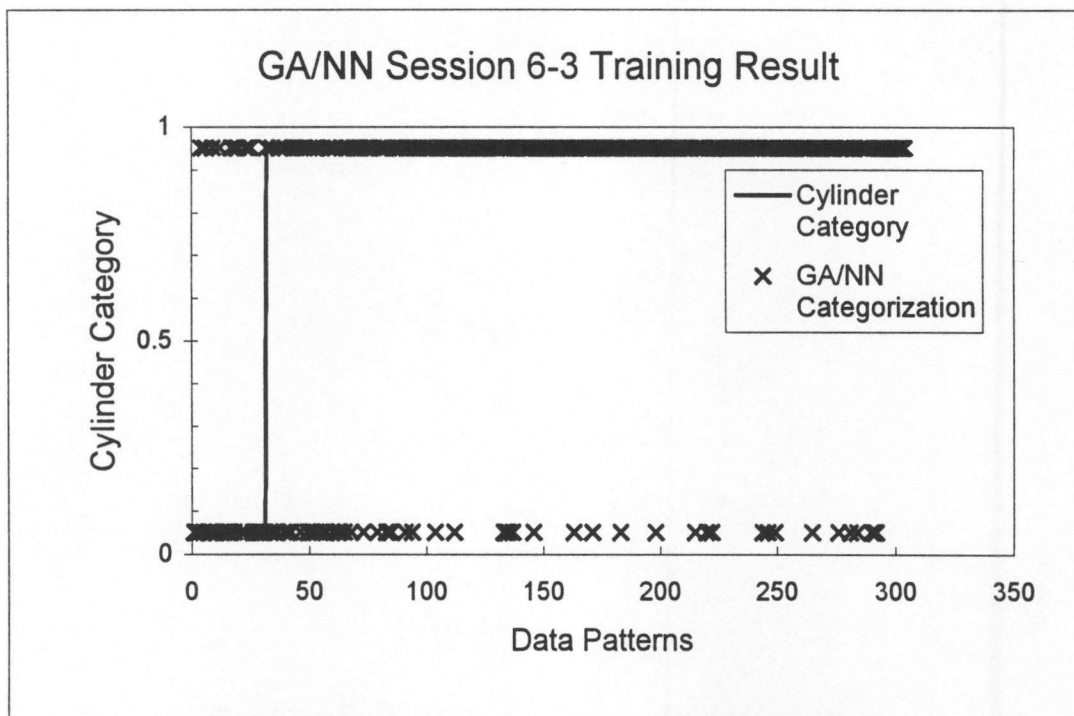


Figure 5.53 Session 6-3: Prediction Results for Training Set

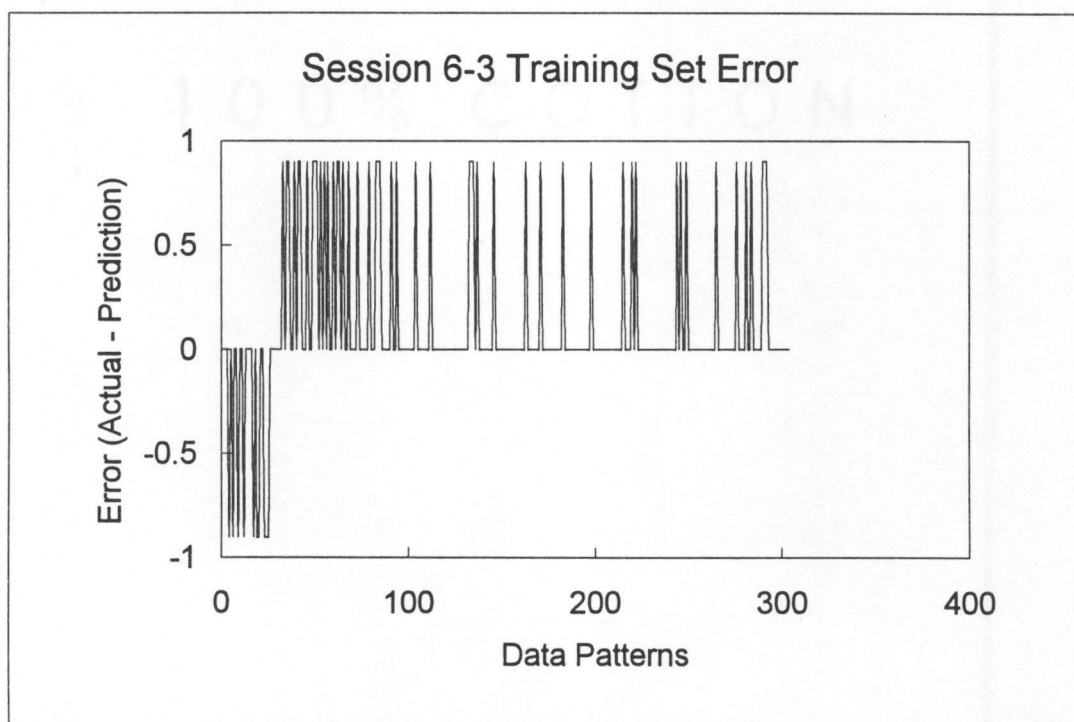


Figure 5.54 Session 6-3: Training Set Prediction Error

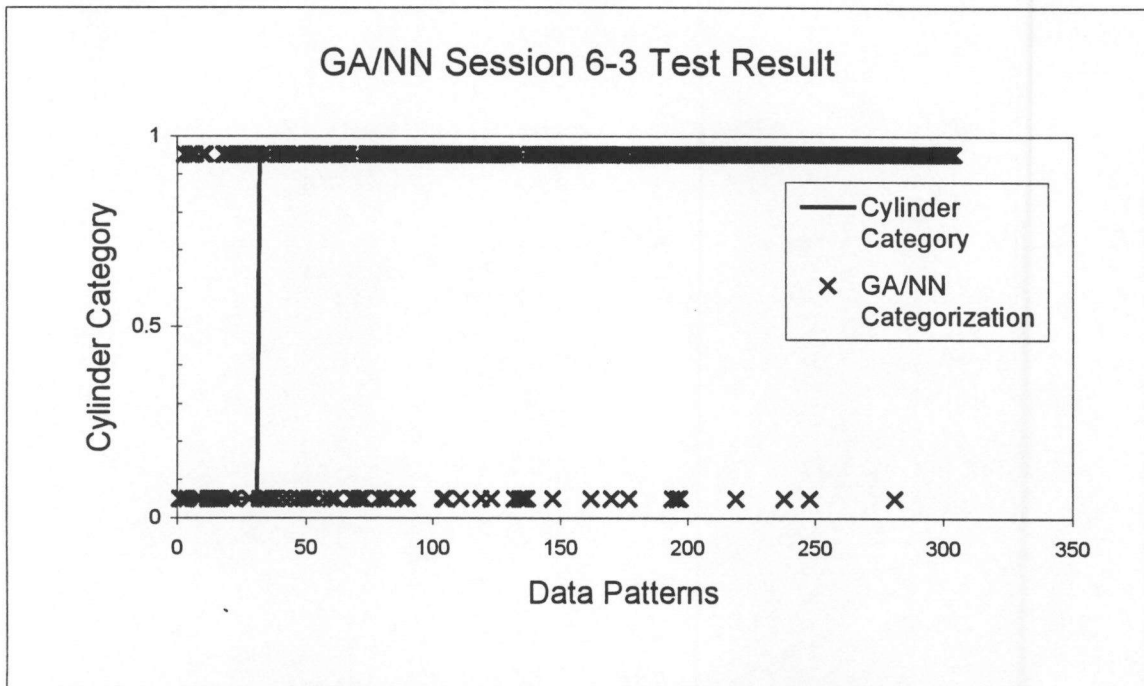


Figure 5.55 Session 6-3: Prediction Results for Test Set

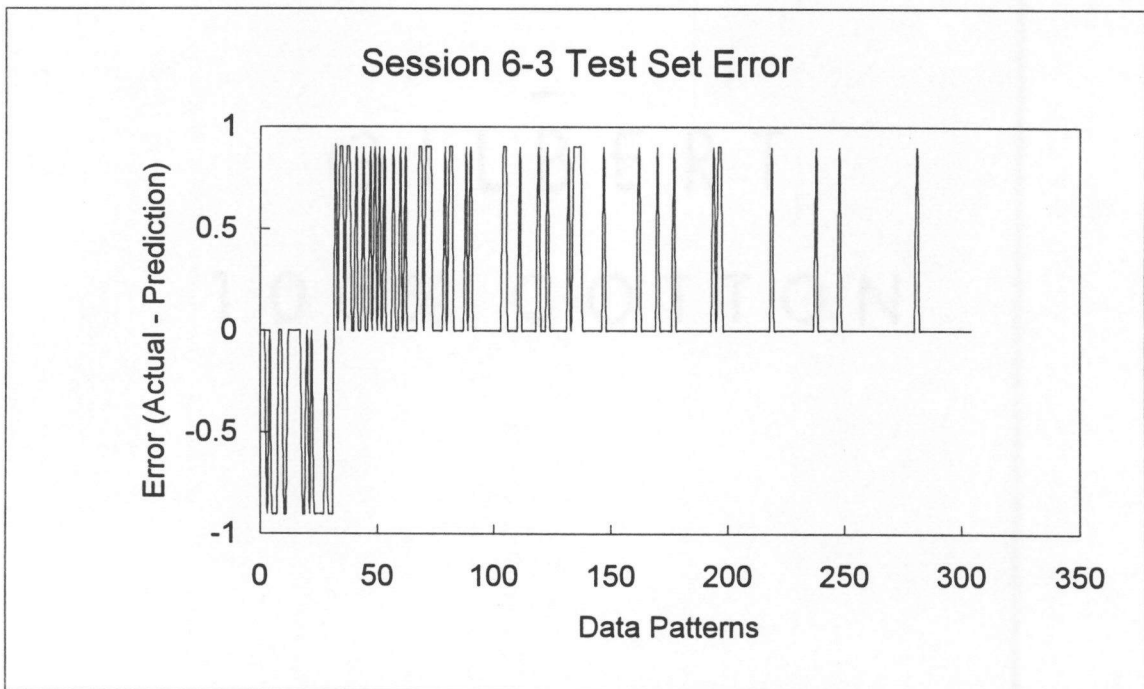


Figure 5.56 Session 6-3: Test Set Prediction Error

5.5 Summary of Results

The results obtained from the MLR analysis, neural network training, and GA/NN model analysis are summarized in Tables 5.16 and 5.17. For the prediction of minimum wall thickness, R^2 values obtained from the GA/NN model analysis were fairly close to that of the MLR analysis that used the whole data set. The R^2 value of the MLR analysis using the training set's coefficients yielded a lower value (0.292). On the other hand, results from the neural network training yielded the lowest R^2 values. This may be because the networks were trained to achieve the minimum root mean squared error, and not to achieve the highest R^2 values. Therefore, comparing the performance of a neural network to the MLR analysis based on R^2 values might put the neural network at a disadvantage.

One possible reason the GA/NN model yielded R^2 values better than those of the neural network training was a better way of selecting the input variables compared to the method used for the neural network training. By selecting different combinations of variables for network training, its data set might have more useful information compared to the 17 variables chosen for the network training based solely on correlation coefficients.

For the classification of cylinders, the MLR analysis using the whole data set produced the highest R^2 value ($R^2 = 0.2115$). However, when only the training set was used to obtain the regression coefficients, the R^2 value from MLR analysis was very close to the values produced by neural network training and the GA/NN model. This showed that the data set did not have sufficient information to categorize the cylinders.

Table 5.16 Summary of Results for Prediction of Minimum Wall Thickness

	Data Coding	R	R^2
MLR (whole data set)	Original	0.6292	0.3959
	One-of-N	0.6372	0.4060
MLR (training set)	Original	0.5401	0.2916
Neural Network Training (Best Results)		Test Set R	Test Set R^2
Training Session 1-1.1c	Original	0.4603	0.2119
Training Session 1-1.1h	Original	0.4884	0.2385
Training Session 3-1.1d	One-of-N	0.5001	0.2501
Training Session 3-1.1f	One-of-N	0.4719	0.2227
GA/NN Model		Test Set R	Test Set R^2
Training Session 5-1	Original	0.6225	0.3875
Training Session 5-2	Original	0.6021	0.3625
Training Session 5-3	Original	0.5882	0.3460

Table 5.17 Summary of Results for the Classification of Cylinders

	Data Coding	R	R^2
MLR (whole data set)	Original	0.4511	0.2035
	One-of-N	0.4599	0.2115
MLR (training set)	Original	0.2021	0.0409
Neural Network Training (Best Results)		Test Set R	Test Set R^2
Training Session 2-1.2a	Original	0.2119	0.0449
Training Session 2-3.1a	Original	0.1806	0.0326
Training Session 4-2.1b	One-of-N	0.1769	0.0313
GA/NN Model		R	R^2
Training Session 6-1	Original	0.0000	0.0000
Training Session 6-2	Original	0.1240	0.0154
Training Session 6-3	Original	0.2194	0.0482

5.6 Data Sorting and Comparison Using *Visual Basic*

It was concluded earlier that neither the use of a neural network nor a GA/NN model was able to effectively predict the minimum wall thickness and/or categorize the UF₆ cylinders. Since both linear models (multiple linear regression) and non-linear models (neural networks and GA/NN models) were used to analyze the data set and negative results were produced, it was suspected that the data set did not have sufficient inherent information to achieve the desired goal. In order to confirm this hypothesis, a sorting program (written in the *Visual Basic* language) provided by Dr. R. Bruce Robinson was used to group cylinders that had similar values for most, if not all, of their variables.

Before using the sorting program, cylinders that had minimum wall thickness less than 250 mils were first identified. Every one of these cylinders was then compared to all the cylinders in the data set. The program would identify cylinders that closely resembled the target cylinder by comparing the values of the cylinders' input variables. Any cylinder that had an error value less than 3 when compared to the target cylinder was selected. In this case, the error was defined as:

$$Error = \sum_i \left[\frac{Abs.(Value_{TC} - Value_{CC})}{Max.(Value_{TC}, Value_{CC}) * } \right] \quad (5.2)$$

where $Value_{TC}$ = value of the target cylinder's input variable

$Value_{CC}$ = the comparison cylinder's input value

$i = 61$ variables in database (excluding *Cylinder I.D.*, *minimum wall thickness*, and *Cylinder Category*)

* -- if the maximum value of the target cylinder and the cylinder being compared

to was = 0, the denominator was set as 1 and the error for that variable would be computed as 0, as expected.

Table 5.18 shows an example of a cylinder (cylinder I.D. 12888) with thickness less than 250 mils and four other cylinders selected that closely resembled the target cylinder. The cylinder in column *Comparison 0* essentially is the target cylinder; the target cylinder is compared to itself to make sure that the *Visual Basic* program works correctly. Another example (cylinder I.D. 11636) is shown in Table 5.19. Note that from these two examples most of the discrete variables have identical input values. In addition, the difference in input values for the continuous variables is too small to provide any meaningful information.

The examples above are only two of the numerous cases whereby a cylinder with thickness less than 250 mils has input values that are hardly different from those of a cylinder with thickness greater than 250 mils. This finding is consistent with the results obtained from *MATLAB*'s neural network training and the GA/NN model and explains why neither a linear nor a non-linear model could perform the tasks of predicting minimum wall thickness and/or categorizing the cylinders. Table 5.20 lists the number of cylinders with thickness greater than 250 mils that has error value equal or less than 3 when compared to cylinders with thickness less than 250 mils. The full sorting results by the *Visual Basic* program is listed in Appendix 5.1.

Table 5.18 Comparison Between Cylinders With Similar Input Variable Values (I)

Variable Type	Variable	Target Cylinder	Comparison 0	Comparison 1	Comparison 2	Comparison 3	Comparison 4
	Cylinder ID	12888	12888	14047	15535	14243	16600
	Min Wall Thickness	198	198	231	252	254	264
	Category	0.05	0.05	0.05	0.95	0.95	0.95
Continuous	Age When Measured	37	37	36	36	36	36
	Cost	318	318	318	320	318	320
	Assay	0.002497	0.002497	0.002487	0.002495	0.002496	0.002517
	Tare	2431	2431	2453	2455	2445	2433
	Net	27967	27967	27945	27968	27953	27956
	Drawing	13	13	13	11	13	11
Discrete	Plant	2	2	2	2	2	2
	Yard	5	5	5	5	5	5
	Yardname	4	4	4	4	4	4
	Row	2	2	2	2	2	2
	Modifier	0	0	0	0	0	0
	Model	2	2	2	2	2	2
	Size	14	14	14	14	14	14
	Material	1	1	1	1	1	1
	Skirt	0	0	0	0	0	0
	ASME_Stamp	1	1	1	1	1	1
	PO	11	11	11	12	11	12
	ORD_yymmdd	4	4	4	5	4	5
	Vendor	3	3	3	3	3	3
	Del_Plt	2	2	2	2	2	2
	A02	0	0	0	0	0	0
	A04	0	0	0	0	0	0
	A05	0	0	0	0	0	0
	A06	0	0	0	0	0	0
	A07	1	1	1	1	1	1
	A08	1	1	1	1	1	1
	A09	1	1	1	1	1	1
	A10	1	1	1	1	1	1
	A11	0	0	0	0	0	0
	A12	1	1	1	1	1	1
	B01	0	0	0	0	0	0
	B02	1	1	1	1	1	1
	B03	0	0	0	0	0	0
	B04	0	0	0	0	0	0
	B05	0	0	0	0	0	0
	B06	0	0	0	0	0	0
	B07	1	1	1	1	1	1
	B09	0	0	0	0	0	0
	B10	0	0	0	0	0	0
	C01	0	0	0	0	0	1
	C02	0	0	0	0	0	0
	C03	0	0	0	0	0	0
	C04	0	0	0	0	0	0
	C06	0	0	0	0	0	0
	C07	0	0	0	0	0	0
	C09	1	1	1	1	1	1
	C10	0	0	0	0	0	0
	C11	0	0	0	0	0	0
	C13	0	0	0	0	0	0
	C14	0	0	0	0	0	0
	C15	0	0	0	0	0	0
	C16	0	0	0	0	0	0
	C17	0	0	0	0	0	0
	C18	0	0	0	0	0	0
	C19	0	0	0	0	1	0
	D02	0	0	0	0	0	0
	D03	0	0	0	0	0	0
	D08	0	0	0	0	0	0
	D09	0	0	1	1	1	0
	D10	1	1	1	1	1	1
	X00	0	0	0	0	0	0
	Error	Target	0.00	1.04	1.48	2.03	1.48

Table 5.19 Comparison Between Cylinders With Similar Input Variable Values (II)

Variable Type	Variable	Target Cylinder	Comparison 0	Comparison 1	Comparison 2	Comparison 3	Comparison 4
	Cylinder ID	11636	11636	14666	11532	12097	11953
	Min Wall Thickness	200	200	197	249	282	294
	Category	0.05	0.05	0.05	0.05	0.95	0.95
Continuous	Age When Measured	36	36	34	37	37	37
	Cost	308	308	320	308	308	308
	Assay	0.001962	0.001962	0.001988	0.00196	0.00197	0.001979
	Tare	2435	2435	2474	2404	2393	2437
	Net	28067	28067	28070	28058	28053	28111
	Drawing	12	12	11	12	12	12
Discrete	Plant	2	2	2	2	2	2
	Yard	5	5	5	5	5	5
	Yardname	4	4	4	4	4	4
	Row	2	2	2	2	2	2
	Modifier	0	0	0	0	0	0
	Model	2	2	2	2	2	2
	Size	14	14	14	14	14	14
	Material	1	1	1	1	1	1
	Skirt	1	1	0	1	1	1
	ASME_Stamp	1	1	1	1	1	1
	PO	10	10	12	10	10	10
	ORD_yymmdd	3	3	5	3	3	3
	Vendor	3	3	3	3	3	3
	Del_Plt	2	2	2	2	2	2
	A02	0	0	0	0	0	0
	A04	0	0	0	0	0	0
	A05	0	0	0	0	0	0
	A06	0	0	0	0	0	0
	A07	1	1	1	1	1	1
	A08	1	1	0	1	1	1
	A09	0	0	0	0	0	0
	A10	1	1	1	1	1	1
	A11	1	1	1	1	0	1
	A12	1	1	1	1	1	1
	B01	0	0	0	0	0	0
	B02	1	1	1	0	1	0
	B03	0	0	0	0	0	0
	B04	0	0	0	0	0	0
	B05	0	0	0	0	0	0
	B06	0	0	0	0	0	0
	B07	0	0	0	0	0	1
	B09	0	0	0	0	0	0
	B10	0	0	0	0	0	0
	C01	0	0	0	0	0	0
	C02	0	0	0	0	0	0
	C03	0	0	0	0	0	0
	C04	0	0	0	0	0	0
	C06	0	0	0	0	0	0
	C07	0	0	0	0	0	0
	C09	0	0	0	0	0	0
	C10	0	0	0	0	0	0
	C11	0	0	0	0	0	0
	C13	0	0	0	0	0	0
	C14	1	1	1	1	1	1
	C15	0	0	0	0	0	0
	C16	0	0	0	0	0	0
	C17	0	0	0	0	0	0
	C18	0	0	0	0	0	0
	C19	0	0	0	0	0	0
	D02	0	0	0	0	0	0
	D03	0	0	0	0	0	0
	D08	0	0	0	0	0	0
	D09	0	0	0	0	0	0
	D10	1	1	1	1	1	1
	X00	0	0	0	0	0	0
	Error	Target	0.00	2.77	1.04	1.05	2.04

Table 5.20 Number of Cylinders (>250 mils) With Error < 3

Cylinder I.D. (< 250 mils)	Number of Cylinders (>250 mils) with Error ≤ 3.0	Cylinder I.D. (< 250 mils)	Number of Cylinders (>250 mils) with Error ≤ 3.0
7340	1	11670	0
10499	2	16121	1
16036	1	9068	10
10375	2	8200	122
14232	0	13927	0
16027	0	12317	0
8444	4	6687	121
8539	11	16807	1
101879	30	6701	15
113778	2	18746	14
14666	1	115251	55
12888	3	8521	1
11636	2	5463	1
101799	32	9131	3
7486	0	100970	34
7933	15	101256	32
112169	3	15908	9
17299	32	18124	0
11628	2	100810	32
113290	9	101803	32
11570	1	7314	2
10232	1	12500	3
12027	0	100751	46
18762	10	8542	35
100552	38	115344	33
100802	38	5119	4
18432	30	16008	7
100736	42	16753	1
101413	32	8357	7
101854	27	115219	61
8723	4	11881	9
113561	27	8965	121
11334	0	102097	8
11954	3	13329	8
8120	110	14806	7
14219	0	18012	6
11833	0	18651	4
13193	1	18592	8
11346	0	115308	7
100938	38	12389	2
5280	0	8061	70
9014	2	17825	4
100430	46	5120	4
100861	42	11111	2
10333	1	114573	27
14506	0	11532	2
14047	4		

6 Conclusions and Recommendations

6.1 Summary and Conclusion

A study was conducted on approximately one thousand UF₆ cylinders that were stored at three Gaseous Diffusion Plants at Oak Ridge, Tennessee; Paducah, Kentucky; and Portsmouth, Ohio. It involved the use of neural networks and GA/NN models to predict these cylinders' minimum wall thickness and to categorize them based on whether they have minimum wall thickness greater or less than 250 mils.

The analyses conducted for this study are summarized below:

- The data set of 61 variables from the UCLIM database was encoded in two formats suitable for neural network training and GA/NN modeling: a) linear encoding, and b) one-of-N data encoding.
- Multiple linear regression analysis was performed on the 61 encoded variables from the UCLIM database.
- Correlation coefficients between the variables and the minimum wall thickness were computed and 17 variables with the highest correlation values were retained as input variables for neural network training. Repeating the same procedure for the variables and the categorized cylinders, 18 variables that best correlated with the categorized cylinders were used as neural network input variables for networks trained to identify cylinders that were less than 250 mils.
- Four groups of networks were trained. The first network group was trained on

the 17 linearly coded variables to predict minimum wall thickness. The second network group was trained to predict the wall thickness with the same 17, this time one-of-N encoded, variables mentioned above. The other two groups of networks were trained to categorize the cylinders into whether they have minimum wall thickness greater or less than 250 mils using the 18 selected variables. Again, one group was trained with the original linearly coded data while the other group with the one-of-N codes.

- A GA/NN model was used to predict the minimum wall thickness and to categorize the cylinders' conditions.
- A sorting program written in the *Visual Basic* language was used to compare cylinders with minimum wall thickness less than 250 mils to the rest of the samples. The number of cylinders with wall thickness greater than 250 mils that closely resembled the target cylinder was recorded.

For the prediction of minimum wall thickness, the highest R^2 value was obtained from the MLR analysis using the whole one-of-N data set ($R^2 = 0.4060$). The R^2 value of the MLR analysis using the training set's regression coefficients yielded a lower value (0.292). The best neural network R^2 value was 0.2501 while the best GA/NN model R^2 value was 0.3875.

One possible reason the GA/NN model yielded R^2 values better than those of the neural network training was a better way of selecting the input variables compared to the method used for the neural network training. The reason the networks yielded lower R^2 values may be because the networks were trained to achieve the minimum root mean squared error, and not to achieve the highest R^2 values. Therefore, comparing the

performance of a neural network to the MLR analysis based on R^2 values might put the neural network at a disadvantage.

For the classification of cylinders, again, the highest R^2 value was obtained from the MLR analysis using the whole one-of-N data set ($R^2 = 0.2115$). When only the training set was used to obtain the regression coefficients, the R^2 value from MLR analysis was very close to the values produced by neural network training and the GA/NN model. The best values managed by the MLR analysis, neural network training, and GA/NN model were 0.041, 0.0449, and 0.0482 respectively.

Based on the results gathered from the above mentioned analyses, it was concluded that the 61 variables from the UCLIM database were unable to provide enough information to either deterministically predict the cylinders' minimum wall thickness or to categorize the cylinders based on their minimum wall thickness. This conclusion was supported by the consistent results obtained from the multiple linear regression analysis, neural network training, and the GA/NN modeling. The conclusion was further strengthened by the findings of many cylinders that were almost identical but were classified into two different categories (greater or less than 250 mils). Statistical approaches for estimating the probabilities of cylinder wall thickness may produce more meaningful results but were beyond the scope of this project.

6.2 Recommendations

It is clear that the existing information in the UCLIM database was not sufficient to predict the UF₆ cylinders' wall thickness and/or to categorize them according to their wall thickness. However, it should be noted that the UCLIM database was initially

developed as a management tool to record and maintain a history of the UF₆ cylinders only. It was not expected to serve as a predictive tool for the categorization of the cylinders.

Gathering more data or information may not even help solve the problem.

Therefore, it is recommended that the UCLIM database not be used to deterministically predict cylinder minimum wall thickness and/or categorize the cylinders according to their minimum wall thickness. This recommendation only applies when one intends to predict an absolute wall thickness value. It may not apply if it is desired to analyze the cylinders using statistical distribution.

REFERENCES

REFERENCES

- Bäck, T. (1996) *Evolution algorithms in theory and practice*, Oxford University Press Inc., New York, NY.
- Belew, R. K., McInerney, J., and Schraudolph, N. N. (1992) "Evolving networks: Using genetic algorithms with connectionist learning," *Proceedings of the Workshop on Artificial Life*, Santa Fe, NM, 511-547.
- Blum, A. (1992) *Neural Networks in C++*, John Wiley & Sons, Inc., New York, NY.
- Brill, F. Z., Brown, D. E., and Martin, W. N. (1992) "Fast genetic selection of features for neural network classifiers," *IEEE Transactions on Neural Networks*, 3 (2), 324-328.
- Buch, A. M., Mazumdar, H. S., and Pandey, P. C. (1993) "Application of artificial neural networks in hydrological modeling: a case study of runoff simulation of a himalayan glacier basin," *Proceedings of 1993 International Joint Conference on Neural Networks (IJCNN)*, 971-974.
- Chalmers, D. J. (1990) "The evolution of learning: an experiment in genetic connectionism," *Proceedings of the 1990 Connectionist Summer School*, Morgan Kaufmann, San Mateo, CA, 82-90.
- Chang, E. J., and Lippmann, R. P. (1991) "Using genetic algorithms to improve pattern classification performance," *Advances in Neural Information Processing 3*, Lippmann, R. P., Moody, J. E., and Touretsky, D. S., Eds., San Mateo, CA, 797-803.
- Chester, D. L. (1990) "Why two hidden layers are better than one," *Applications Track of the Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 1, Washington D. C., 265-268.
- Chow, M. Y., and Teeter, J. (1994) "An analysis of weight decay as a methodology of reducing three-layer feedforward artificial neural networks for classification problems," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 1, Orlando, FL, 600-605.
- Demuth, H., and Beale, M. (1994) *Neural Network Toolbox*, The MathWorks Inc., Natick, MA.

- Deng, C., and Haykin, S. (1990) "A multi-layer neural network classifier for radar clutter," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 1, San Diego, CA, 241-246.
- Dowla, F. U., and Rogers, L. L. (1995) *Solving Problems in Environmental Engineering and Geosciences with Artificial Neural Networks*, MIT Press, Cambridge, MA.
- Engelbrecht, A. P., and Cloete, I. (1996) "A sensitivity analysis algorithm for pruning feedforward neural networks," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 2, Washington D.C., 1274-1277.
- Fukuda, T., Ishigami, H., Arai, F., and Shibata, T. (1993) "Structure organization of fuzzy neural networks by genetic algorithm," *Proceedings of the IFSA*, 964-967.
- Fukumi, M., Omatu, S., Nishikawa, Y. (1995) "Designing a neural network by a genetic algorithm with partial fitness," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 4, Perth, Western Australia, 1834-1838.
- Funahashi, K. (1989) "On the approximate realization of continuous mapping by neural networks," *Neural Networks*, 2(3), 183-192.
- Garrett, Jr., J. H., Ranjithan, S., and Eheart, J. W. (1992) "Application of neural network to groundwater remediation," *ASCE Monograph Knowledge Representation*, Allen, R., Ed., Chapter 11, 259-269.
- Goldberg, D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Publishing Company, Inc., Reading, MA.
- Grubert, J. P. (1995) "Application of neural networks in stratified flow stability analysis," *Journal of Hydraulic Engineering*, ASCE, Vol. 121, No. 7, July, 523-532.
- Guo, Z., and Uhrig, R. E. (1992) "Using genetic algorithms to select inputs for neural networks," *International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, Los Alamitos, CA, 223-234.
- Hagan, M. T., Demuth, H. B., and Beale, M. (1996) *Neural Network Design*, PWS Publishing Company, Boston, MA.
- Hagan, M. T., and Menhaj, M. B. (1994) "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Transactions on Neural Network*, Vol. 5, No. 6, 989-993.

- Harp, S. A., Samad, T., and Guha, A. (1989) "Towards the genetic synthesis of neural networks," *Third International Conference on Genetic Algorithms*, Schaffer, J. D., Ed., San Mateo, CA, 360-369.
- Haykin, S. (1994) *Neural Networks*, Macmillan College Publishing Company Inc., Englewood Cliffs, NJ.
- Holland, J. H. (1992) "Genetic algorithms," *Scientific American*, Vol. 267, 66-72.
- Janson, D. J., and Frenzel, J. F. (1993) "Training product unit neural networks with genetic algorithms," *IEEE Expert*, Vol. 8, No. 5, 26-33.
- Jockusch, S., and Ritter, H. (1994) "Self-organizing maps: local competition and evolutionary optimization," *Neural Networks*, Vol. 7, No. 8, 1229-1239.
- Karunanithi, N., Grenney, W. J., Whitley, D., Bovee, K. (1994) "Neural networks for river flow prediction," *Journal of Computing in Civil Engineering*, ASCE, Vol. 8, No. 2, April, 201-220.
- Kelly, J. D., and Davis, L. (1991) "Hybridizing the genetic algorithm and the K nearest neighbors classification algorithm," *Fourth International Conference on Genetic Algorithms*, Blew, R. K., and Booker, L. B., Eds., 377-383.
- Kessler, W., and Kessler, R. W. (1995) "Kohonen network as a classifier and predictor for the qualification of metal-oxide-surfaces," *Neural Network Engineering in Dynamic Control Systems*, Hunt, K. J., Irwin, G. R., and Warwick, K., Eds., Springer, Great Britain, 201-220.
- Kitano, H. (1993) "Neurogenetic learning: an integrated method of designing and training neural networks using genetic algorithms," *Physica D*, Vol. 75, No. 1-3, 225-238.
- Krovvidy, S., and Wee, W. G. (1990) "A knowledge based neural network approach for waste water treatment system," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 1, San Diego, CA, 327-332.
- Lee, K. W., and Lam, H. N. (1995) "Optimising neural network weights using genetic algorithms: a case study," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 3, Perth, Western Australia, 1384-1388.

- Li, J. Y., Chow, T. W. S., and Yu, Y. L. (1995) "The estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 3, Perth, Western Australia, 1229-1233.
- Lyon, B. F., and Stumb, P. C. (1996) *Draft Report for Cylinder Categorization*, Technical Report, Oak Ridge National Laboratory (ORNL), Oak Ridge, February.
- Masters, T. (1993) *Practical Neural Networks Recipes in C++*, Academic Press, San Diego, CA.
- Medsker, L. R. (1995) *Hybrid Intelligent Systems*, Kluwer Academic Publishers, Norwell, MA.
- Monclova-Ortiz, L. A. (1996) *Cylinder Categorization Using Neural Networks*, Technical Report, Oak Ridge National Laboratory (ORNL), Oak Ridge, August.
- Murray, A. L. (1995) *Applications of Neural Networks*, Kluwer Academic Publishers, Norwell, MA.
- NeuralWorks Predict* (1995), NeuralWare Inc., Technical Publications Group, Pittsburgh, PA.
- Onoda, T. (1995) "Neural network information criterion for the optimal number of hidden units," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 1, Perth, Western Australia, 275-280.
- Ramani, N., Patrick, P. H., Hanson, W. G., Anderson, H. (1990) "Fish detection and classification using a neural-network-based active sonar system – preliminary results," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 2, Washington, D.C., 527-530.
- Rao, M., et al. (1994) *Corrosion Monitoring in the UF₆ Cylinder Yards at the Oak Ridge K-25 Site: FY 1994 Report*, Technical Report, Oak Ridge National Laboratory (ORNL), 1BK-02268C, Oak Ridge, October.
- Rogers, L. L., Dowla, F. U., and Johnson, V. M. (1995) "Optimal field-scale groundwater remediation using neural networks and the genetic algorithm," *Environmental Science and Technology*, Vol. 29, No. 5, 1145-1155.

- Romaniuk, S. G. (1994) "Towards minimal network architectures with evolutionary growth networks," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 3, Orlando, FL, 1710-1713.
- Saha, S., and Christensen, J. P. (1994) "Genetic design of sparse feedforward neural networks," *Information Sciences*, Vol. 79, No. 3-4, 191-200.
- Sarle, W. S., *Neural Network FAQ*, <ftp://ftp.sas.com/pub/neural/FAQ.html>; last modified: 01-07-1997.
- Schaffer, J. D., Caruana, R. A., and Eshelman, L. J. (1990) "Using genetic search to exploit the emergent behavior of neural networks," *Emergent Computation*, S., Forrest, Ed., North Holland, Amsterdam, 244-248.
- Schaffer, J. D., Whitley, D., and Eshelman, L. J. (1992) "Combinations of genetic algorithms and neural networks: a survey of the state of the art," *International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, Baltimore, MD, 1-37.
- Schmuller, J. (1990) "Neural networks and environmental applications," *Expert Systems for Environmental Applications*, Hushon, J. M., Ed., Washington, D.C., Chapter 4, 52-68.
- Shyu, H. J., Libert, J. M., Mann, S. D. (1990) "Classifying seismic signals via RCE neural network," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 1, San Diego, CA, 101-105.
- Smith, G., and Longstaff, D. (1995) "A neural network component in a texture classification system," *IEEE Proceedings of the International Conference on Neural Networks (ICNN)*, Vol. 1, Perth, Western Australia, 43-48.
- Stumb, P. C. (1996) *A Fuzzy Goal Programming Approach to the Management of Depleted Uranium Hexafluoride*, dissertation presented to the University of Tennessee, Knoxville, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.
- Swingler, K. (1996) *Applying Neural Networks: A Practical Guide*, Academic Press Inc., San Diego, CA.

- Vrckovnik, G., Chung, T., and Carter, C. R. (1990) "Classifying impulse radar waveforms using principle components analysis and neural networks," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 1, San Diego, CA, 69-74.
- Whitley, D. (1989) "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best," *Proceedings of the Third International Conference on Genetic Algorithms*, 116-121.
- Wilson, S. W. (1990) "Perceptron redux: emergence of structure," *Emergent Computation*, S., Forrest, Ed., North Holland, Amsterdam, 249-256.
- Yoon, B., Holmes, D. J., Langholz, G., and Kandel, A. (1994) "Efficient genetic algorithms for training layered feedforward neural networks," *Information Sciences*, Vol. 76, No. 1-2, 67-85.

APPENDICES

Appendix 4.1 Complete UCLIM Database Variables

UCLIM Database Variable Name	Comment
UCL Container	Container ID. Note: the first two characters can be letters.
UCL Plant	Plant where container is currently stored 1 = K-25 2 = PAD 3 = PORTS
UCL Modifier	Cylinder modifier
UCL Drawing	Cylinder drawing no.
UCL Model	Model type
UCL Size	Cylinder Size
UCL Material	Steel material
UCL Wall Gauge	Wall gauge
UCL Skirt	Presence of skirt
UCL ASME Stamp	Whether cylinder is an ASME pressure vessel
UCL PO	Purchase order
UCL ORD YYMMDD	Date ordered
UCL Vendor	Initials of vendor
UCL Cost	Cost of cylinder
UCL Del Plt	Which plant the cylinders were delivered to
UCL Del Yr	Year delivered
UCL Del Mo	Month delivered
UCL Test yymmdd	*Date tested
UCL Fill yymmdd	*Data filled
UCL Assay	Assay of DU in cylinder (fraction ?)
UCL Tare	Tare weight of cylinder (lbs)
UCL Net	Net weight of cylinder (lbs)
UCL Tran yymmdd 1	Date cylinder was moved to current location
UCL Plant 1	Plant where cylinder is currently located
UCL Yard 1	Yard where cylinder is currently located
UCL Section 1	Section of yard where cylinder is currently located
UCL Row 1	Row where cylinda is currently located (Note: not top vs. Bottom)
UCL Position 1	Position where cylinder is currently located
UCL Tran yymmdd 2	*Date cylinder was moved to previous location
UCL Plant 2	*Plant where cylinder was located
UCL Yard 2	*Yard where cylinder was located
UCL Section 2	*Section of yard where cylinder was located
UCL Row 2	*Row where cylinda was located (Note: not top vs. Bottom)
UCL Position 2	*Position where cylinder was located
UCL Tran yymmdd 3	*Date cylinder was moved to previous location
UCL Plant 3	*Plant where cylinder was located
UCL Yard 3	*Yard where cylinder was located
UCL Section 3	*Section of yard where cylinder was located
UCL Row 3	*Row where cylinda was located (Note: not top vs. Bottom)
UCL Position 3	*Position where cylinder was located
UCL Tran yymmdd 4	*Date cylinder was moved to previous location
UCL Plant 4	*Plant where cylinder was located
UCL Yard 4	*Yard where cylinder was located
UCL Section 4	*Section of yard where cylinder was located
UCL Row 4	*Row where cylinda was located (Note: not top vs. Bottom)
UCL Position 4	*Position where cylinder was located
UCL Tran yymmdd 5	*Date cylinder was moved to previous location
UCL Plant 5	*Plant where cylinder was located
UCL Yard 5	*Yard where cylinder was located
UCL Section 5	*Section of yard where cylinder was located
UCL Row 5	*Row where cylinda was located (Note: not top vs. Bottom)
UCL Position 5	*Position where cylinder was located
UCL Tran yymmdd 6	*Date cylinder was moved to previous location
UCL Plant 6	*Plant where cylinder was located
UCL Yard 6	*Yard where cylinder was located
UCL Section 6	*Section of yard where cylinder was located
UCL Row 6	*Row where cylinda was located (Note: not top vs. Bottom)
UCL Position 6	*Position where cylinder was located
UCL Insp ID	Inspection ID number for last inspection
UCL Insp yymmdd	Date inspected
UCL Badge1	Badge number of inspector
UCL Badge2	Badge number of inspector
UCL Badge3	Badge number of inspector
UCL Mech Note	Note made by inspector
UCL Defect Code 01	Defect code
UCL Defect Code 02	Defect code
UCL Defect Code 03	Defect code
UCL Defect Code 04	Defect code
UCL Defect Code 05	Defect code
UCL Defect Code 06	Defect code
UCL Defect Code 07	Defect code
UCL Defect Code 08	Defect code
UCL Defect Code 09	Defect code
UCL Defect Code 10	Defect code
UCL Defect Code 11	Defect code
UCL Defect Code 12	Defect code
UCL Defect Code 13	Defect code
UCL Defect Code 14	Defect code

*Rarely has a value.

Appendix 4.2 Cylinders Not Used in Neural Network Training

Note: 165 out of 1077 cylinders were left out from the neural network data set

Cylinder I.D.	Reason for Exclusion from Neural Network Training
2	not found in UCLIM database (thick cylinder)
3	not found in UCLIM database (thick cylinder)
15	not found in UCLIM database (thick cylinder)
72	not found in UCLIM database (thick cylinder)
79	not found in UCLIM database (thick cylinder)
87	not found in UCLIM database (thick cylinder)
93	not found in UCLIM database (thick cylinder)
169	not found in UCLIM database (thick cylinder)
170	not found in UCLIM database (thick cylinder)
178	not found in UCLIM database (thick cylinder)
197	not found in UCLIM database (thick cylinder)
202	not found in UCLIM database (thick cylinder)
215	not found in UCLIM database (thick cylinder)
235	not found in UCLIM database (thick cylinder)
237	not found in UCLIM database (thick cylinder)
246	not found in UCLIM database (thick cylinder)
249	not found in UCLIM database (thick cylinder)
267	not found in UCLIM database (thick cylinder)
277	not found in UCLIM database (thick cylinder)
295	not found in UCLIM database (thick cylinder)
309	not found in UCLIM database (thick cylinder)
316	not found in UCLIM database (thick cylinder)
327	not found in UCLIM database (thick cylinder)
329	not found in UCLIM database (thick cylinder)
347	not found in UCLIM database (thick cylinder)
351	not found in UCLIM database (thick cylinder)
371	not found in UCLIM database (thick cylinder)
378	not found in UCLIM database (thick cylinder)
390	not found in UCLIM database (thick cylinder)
406	not found in UCLIM database (thick cylinder)
408	not found in UCLIM database (thick cylinder)
420	not found in UCLIM database (thick cylinder)
429	not found in UCLIM database (thick cylinder)
431	not found in UCLIM database (thick cylinder)
445	not found in UCLIM database (thick cylinder)
476	not found in UCLIM database (thick cylinder)
488	not found in UCLIM database (thick cylinder)
492	not found in UCLIM database (thick cylinder)
502	not found in UCLIM database (thick cylinder)
507	not found in UCLIM database (thick cylinder)
516	not found in UCLIM database (thick cylinder)
564	not found in UCLIM database (thick cylinder)
565	not found in UCLIM database (thick cylinder)
569	not found in UCLIM database (thick cylinder)
571	not found in UCLIM database (thick cylinder)
589	not found in UCLIM database (thick cylinder)
593	not found in UCLIM database (thick cylinder)
619	not found in UCLIM database (thick cylinder)
641	not found in UCLIM database (thick cylinder)
657	not found in UCLIM database (thick cylinder)
662	not found in UCLIM database (thick cylinder)
664	not found in UCLIM database (thick cylinder)
673	not found in UCLIM database (thick cylinder)
693	not found in UCLIM database (thick cylinder)
696	not found in UCLIM database (thick cylinder)
711	not found in UCLIM database (thick cylinder)
725	not found in UCLIM database (thick cylinder)
726	not found in UCLIM database (thick cylinder)
735	not found in UCLIM database (thick cylinder)
740	not found in UCLIM database (thick cylinder)
766	not found in UCLIM database (thick cylinder)
783	not found in UCLIM database (thick cylinder)
804	not found in UCLIM database (thick cylinder)
811	not found in UCLIM database (thick cylinder)
818	not found in UCLIM database (thick cylinder)
827	not found in UCLIM database (thick cylinder)
838	not found in UCLIM database (thick cylinder)
841	not found in UCLIM database (thick cylinder)
848	not found in UCLIM database (thick cylinder)
889	not found in UCLIM database (thick cylinder)
900	not found in UCLIM database (thick cylinder)
901	not found in UCLIM database (thick cylinder)
902	not found in UCLIM database (thick cylinder)
913	not found in UCLIM database (thick cylinder)
919	not found in UCLIM database (thick cylinder)
921	not found in UCLIM database (thick cylinder)
928	not found in UCLIM database (thick cylinder)
931	not found in UCLIM database (thick cylinder)

Appendix 4.2 (contd.)

Cylinder I.D.	Reason for Exclusion from Neural Network Training
937	not found in UCLIM database (thick cylinder)
945	not found in UCLIM database (thick cylinder)
947	not found in UCLIM database (thick cylinder)
949	not found in UCLIM database (thick cylinder)
972	not found in UCLIM database (thick cylinder)
979	not found in UCLIM database (thick cylinder)
981	not found in UCLIM database (thick cylinder)
989	not found in UCLIM database (thick cylinder)
998	not found in UCLIM database (thick cylinder)
1164	not found in UCLIM database (thick cylinder)
1209	not found in UCLIM database (thick cylinder)
1216	not found in UCLIM database (thick cylinder)
1220	not found in UCLIM database (thick cylinder)
1255	not found in UCLIM database (thick cylinder)
1258	not found in UCLIM database (thick cylinder)
1304	not found in UCLIM database (thick cylinder)
1343	not found in UCLIM database (thick cylinder)
1416	not found in UCLIM database (thick cylinder)
1624	not found in UCLIM database (thick cylinder)
1669	not found in UCLIM database (thick cylinder)
1772	not found in UCLIM database (thick cylinder)
1785	not found in UCLIM database (thick cylinder)
1987	not found in UCLIM database (thick cylinder)
2018	not found in UCLIM database (thick cylinder)
2062	not found in UCLIM database (thick cylinder)
2072	not found in UCLIM database (thick cylinder)
2101	not found in UCLIM database (thick cylinder)
2137	not found in UCLIM database (thick cylinder)
2233	not found in UCLIM database (thick cylinder)
2235	not found in UCLIM database (thick cylinder)
2252	not found in UCLIM database (thick cylinder)
2282	not found in UCLIM database (thick cylinder)
2307	not found in UCLIM database (thick cylinder)
2325	not found in UCLIM database (thick cylinder)
2489	not found in UCLIM database (thick cylinder)
3001	not found in UCLIM database (thick cylinder)
3007	not found in UCLIM database (thick cylinder)
3009	not found in UCLIM database (thick cylinder)
3023	not found in UCLIM database (thick cylinder)
3046	not found in UCLIM database (thick cylinder)
3052	not found in UCLIM database (thick cylinder)
3057	not found in UCLIM database (thick cylinder)
3076	not found in UCLIM database (thick cylinder)
3086	not found in UCLIM database (thick cylinder)
3087	not found in UCLIM database (thick cylinder)
3100	not found in UCLIM database (thick cylinder)
3101	not found in UCLIM database (thick cylinder)
3114	not found in UCLIM database (thick cylinder)
3132	not found in UCLIM database (thick cylinder)
3159	not found in UCLIM database (thick cylinder)
3174	not found in UCLIM database (thick cylinder)
3185	not found in UCLIM database (thick cylinder)
3191	not found in UCLIM database (thick cylinder)
3206	not found in UCLIM database (thick cylinder)
3234	not found in UCLIM database (thick cylinder)
3294	not found in UCLIM database (thick cylinder)
3295	not found in UCLIM database (thick cylinder)
3312	not found in UCLIM database (thick cylinder)
3332	not found in UCLIM database (thick cylinder)
5343	not found in UCLIM database (thick cylinder)
5634	not found in UCLIM database (thick cylinder)
6004	possible outlier due to suspiciously low <i>Net Weight</i> value
6234	possible outlier due to suspiciously low <i>Net Weight</i> value
6325	possible outlier due to suspiciously low <i>Net Weight</i> value
6407	possible outlier due to suspiciously low <i>Net Weight</i> value
6435	possible outlier due to suspiciously low <i>Net Weight</i> value
6561	possible outlier due to suspiciously low <i>Net Weight</i> value
6727	possible outlier due to suspiciously low <i>Net Weight</i> value
6824	possible outlier due to suspiciously low <i>Net Weight</i> value
7953	Minimum Wall Thickness = 0 mil
7958	possible outlier due to suspiciously low <i>Net Weight</i> value
8179	possible outlier due to suspiciously low <i>Net Weight</i> value
8571	<i>PO</i> , <i>ORD</i> , <i>yymmdd</i> , <i>Vendor</i> , and <i>Cost</i> unknown
17093	<i>YardName</i> not identified
18594	not found in UCLIM database
100157	<i>PO</i> , <i>ORD</i> , <i>yymmdd</i> , <i>Vendor</i> , and <i>Cost</i> unknown
101244	Minimum Wall Thickness = 0 mil
102019	0 value for all variables
110002	not found in UCLIM database
110126	not found in UCLIM database
110630	<i>Del Yr</i> and <i>Del Mo</i> not identified
111441	<i>Del Yr</i> and <i>Del Mo</i> not identified
114385	<i>Del Yr</i> and <i>Del Mo</i> not identified
115035	<i>Del Yr</i> and <i>Del Mo</i> not identified
115150	not found in UCLIM database
115688	<i>Yard</i> and <i>Row</i> not identified
130334	not found in UCLIM database

Appendix 5.1 Visual Basic Program Sorting Results

Table	Journal	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
-------	---------	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

[illegible]

VITA

Yeeming Leong was born on May 22, 1972, in Kuala Lumpur, Malaysia. He completed his high school and pre-university (Form 6) in Malaysia. In 1992, he enrolled at Inti College (Malaysia) and studied for two semesters before transferring to the University of Arkansas, Fayetteville, where he received a B.S. degree (Magna Cum Laude) in Civil Engineering in December 1995. In the spring of 1996, he furthered his studies at the University of Tennessee, Knoxville, and received a M.S. degree in Environmental Engineering in December 1997.