APPLICATION OF A NEURAL NETWORK IN MODELING THE SLUDGE BLANKET DEPTH OF A SECONDARY CLARIFIER

By

QIN ZHAO

Bachelor of Science

Shanxi Agricultural University

Taigu, Shanxi

People's Republic of China

1983

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillments of
the requirement for
the Degree of
MASTER OF SCIENCE
May, 1993

OKLAHOMA STATE UNIVERSITY

APPLICATION OF A NEURAL NETWORK IN

MODELING THE SLUDGE BLANKET DEPTH

OF A SECONDARY CLARIFIER

Thesis Approved:

Thesis Adviser

Thomas C. Collins

Dean of the Graduate College

AKNOWLEDGMENTS

I wish to express sincere appreciation to my adviser, Dr. Charles T. Haan, for his kind guidance, patience, advice, and throughout my graduate program. Many thanks go to Dr. Daniel Storm and Dr. Marvin L. Stone for serving on my graduate committee. Their suggestions and support were very helpful throughout the study.

I also wish to thank my former adviser, Dr. Willard H. Downs, for his advice and assistance for my reasearch.

I would like to acknowledge the Department of Agricultural Engineering and Stover and Associates, Inc. for their financial support in the form of reasearch assistantships through my graduate study at Oklahoma State University.

A note of thanks goes to Dr. Enos L. Stover and Chris K. Campana in Stover and Associates, Inc. for their advice in developing the model.

I would like to thank my friends Jian Yue and Dingding Chen in my department for their friendship and help.

A special thanks go to my parents, Chunming Zhao and Cuizhen Zhang, who have encouraged and supported me in every aspect of my life's endeavors and helped me stay focused.

To my family, I extend the deepest gratitude. My wife, Shuiyu Zhang, has constantly encouraged me in all of my

efforts. Her love and sustenance have provided a stable foundation for every enterprise I have ever attempted. My daughter, Danfeng Zhao, furnished my initial inspiration to venture to complete a master degree program, and her faith in me was always evident. All of my family deserves a most sincere thank you.

TABLE OF CONTENTS

Chapter		Page

LIST OF TABLES

LIST OF FIGURES

Figure	Page

CHAPTER I

INTRODUCTION

General Statement of the Problem

This thesis describes a neural network used to predict the sludge blanket depth of a secondary clarifier of a waste water treatment plant. The objective of the thesis was to develop a neural network on a personal computer, train the neural network to learn how the sludge depth changes from experience, and then use the trained neural network to predict the sludge blanket depth.

A neural network is a new computing technique in artificial intelligence. Neural networks can learn, memorize, and generalize from experience and events. What the neural network does is mimic the human brain logically. First the neural network is taught example cases. After the neural network has learned, it can simulate the thinking of the human brain based on the examples learned. This project uses the neural network technique to predict the clarifier sludge depth in a waste water treatment process.

In an activated sludge plant, the purpose of the secondary clarifier is to provide a quiescent area in which solids will separate from treated water. Therefore, within the secondary clarifier there are normally three separate areas: an area of clear water relatively free of solids; an area where discrete straggler floc particles are settling down to form a blanket; and an area which contains the solids that have separated from the mixed liquid to form the sludge blanket. The sludge blanket depth is measured to the top of the relatively thin, but quite homogenous, upper surface of the accumulated sludge blanket. The sludge blanket depth is the distance from the surface of the clarifier to the top of the sludge blanket (Hobson, 1986).

In the waste water treatment process, the clarifier sludge blanket depth changes frequently. This depth is a function of the activated sludge process. During a daily cycle, the depth of the sludge blanket in a final clarifier will change mostly due to the effects of changing waste water flow rates. Changes that occur over a longer period of time (days or weeks) are usually caused by operator process adjustments (or the lack of them) (Hobson, 1986). Factors affecting the depth include raw waste water strength, nutrients in the system, dissolved oxygen in the aeration basin, detention time, pH, temperature, influent flow rate, and operator process adjustments.

Sometimes the changing sludge blanket depth can cause problems in the treatment process. The solids concentrations in the final effluent will increase if the sludge blanket is too high in the clarifier. Unmanaged sludge blanket depth changes can also be disastrous to the clarifier. For example, the sludge blanket could rise in a clarifier and flow over the effluent weirs. For plant operation, the sludge blanket depth should stay within the design range (Hobson, 1986).

Therefore, monitoring and predicting the depth of sludge blanket is very important. This allows the operator to anticipate other problems such as denitrafication in the clarifier. It is also important for the operator to make decisions about return sludge flow adjustment.

There are several methods for monitoring the depth of the sludge blanket in the secondary clarifier. Some use permanently installed instruments such as the airlift pump tube and the ultrasonic blanket detector. Others, such as the electronic detector, the sight glass and the caretaker, are more often used manually by the operator to sample or measure the sludge blanket (Hobson, 1986). However, they are all physical methods. They do not predict the sludge blanket depth. Currently it is difficult to find a mathematical equation to predict the sludge depth. Since the neural network can learn from experience, we can collect many cases of the sludge blanket changes and train the network with these cases. After the network is trained, it may serve as a model to predict the sludge blanket depth.

Why a Neural Network?

The reason a neural network was chosen in this research was because conventional techniques can not predict the depth of the sludge blanket. Sludge blanket depth is a complex phenomenon. Neural networks offer improved performance over conventional technologies in the area of complex mapping and modeling complex phenomena.

Neural networks are an emerging computational technology. They have the abilities of adaptive learning, self-organization, and generalization. Adaptive learning is one of the most attractive features of neural networks. By adaptive learning, neural network can learn to discriminate patterns based on examples (Maren et al., 1990). During the training of the network, we do not have to work out an a priori model and specify probability distribution functions. Neural networks use their adaptive learning capabilities to self-organize the information they receive during training. After the neural network is trained, it can predict output based on inputs that the network has never seen before.

Sludge blanket depth depends upon multiple interacting parameters. There are no current mathematical methods to predict sludge blanket depth. When large amount of measured data are available, neural network can be tried. Such is the case for the database of the Gerber Baby Food Waste Water Treatment Plant which is the data source for this study.

Successful neural network applications generally have the above characteristics. Therefore the project is suitable to apply neural network technology (Bailey and Thompson, 1990a).

Objectives of the Study

The objective of this research is to design and implement a multiple layer back propagation neural network for modeling the sludge blanket depth for the Gerber Baby Food Waste Water Treatment Plant.

Procedure

The first step was to prepare the training data and testing data. The training and testing data were collected from the database of Gerber Baby Food Waster Water Treatment Plant. The proper input variables need to be selected based on the parameters affecting the sludge blanket depth. The output variable is the sludge blanket depth. The data were transformed, scaled and divided into two sets - training data and testing data.

To design a neural network, it is necessary to select the neural network paradigm and determine the number of processing units in each layer. Neural network software must be selected. Finally the design must be implemented on the selected software.

The last step is to implement the neural network. The goal is to create a functioning neural network that provides the most accurate and consistent model possible. Iterative building, training, and testing is used to refine the neural network (Bailey and Thompson, 1990a). First the training data are used to train the neural network. The training process is monitored for problems. Then the network is tested using the test data. Finally, the test results are plotted.

Neural Network Simulation Tool

ExploreNet 3000 was used as the development environment. It is a fully automated commercial development package. This software is designed with user-adjustable parameters for easy tailoring to the requirements of the application. It contains the fundamental structure and processing equations for more than twenty different neural network paradigms. It allows the user to control the number of the hidden layers, the size of each layer, and the values of the transfer function constants. Five back-propagation neural networks are available within ExploreNet 3000.

CHAPTER II

BACK PROPAGATION NEURAL NETWORKS

Among the possible architectures for neural networks are back error propagation, counter propagation, adaptive resonance theory, Madline Rule III, and Nestor's Reduced Coulomb Energy. Back error propagation is the most popular and successful neural network learning paradigm. Back propagation can solve problems dealing with general mapping approximations, pattern classification, data modeling, process control, and signal processing, which can be viewed as a pattern recognition problem. Successful back propagation applications in these areas have been reported. NeuralWare's Applications Development Services and Support (ADSS) group has developed a successful bankruptcy prediction application using the back propagation network (Coleman et al., 1990). Beale and Demuth (1992) used the back propagation method to obtain a mathematical model for a real system to be controlled.

The Architecture of Back Propagation Neural Networks

Back-propagation networks are usually layered. The layers, or slabs are referred to as the input slab, the hidden slab(s), and the output slab. There is also a training slab. Figure 1 shows the basic form of a neural network. The bottom layer is the input layer and is the only layer in the network that receives external input. The layers above are the hidden layers. Hidden layer can be one layer or two layers and usually not more than three layers. The top layer is the output layer, which gives the output of the network. Every layer of the network consists of one or more processing units called neurons. Each layer is fully connected to the next higher level. The highest level of the hidden layers is connected to the output layer. The training slab is connected to the output slab in a one-to-one manner (HNC, Inc., 1991c).

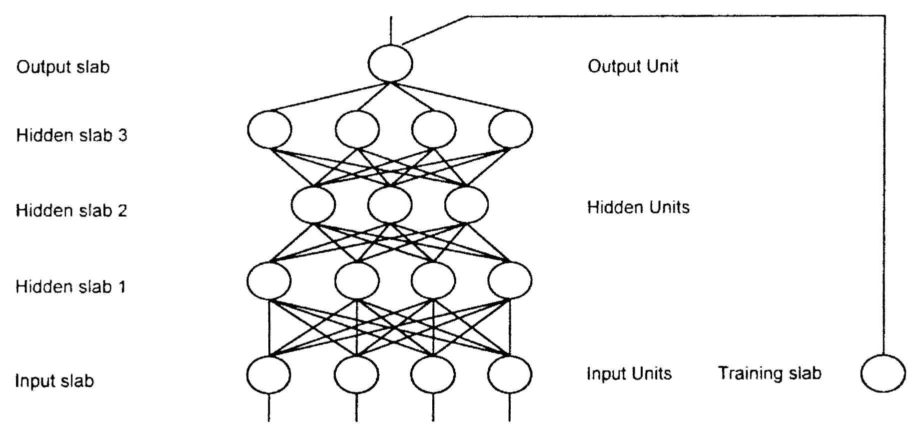


Figure 1. A Five Layered Back-propagation Network

How Does the Back Propagation Network Solve Problems?

There are two modes of back propagation network operation - training mode and producing mode. In the training mode, the network requires two sets of data: an example set of input vectors and a corresponding set of desired output vectors. The goal of back propagation network training is for the network to learn to reproduce a mapping or functional relationship from these two kinds of vectors. After training, the network is used in production mode. If trained correctly, the network should be able to produce correct output responses for any input vectors, which may not have been presented to the network during training.

When the network is given an input vector and corresponding target output vector in the training mode, the input slab receives the input vector and the training slab gets the corresponding output vector. The training slab is connected to the output slab directly without any mathematical computations. The input slab fans out the input data without making calculations. The data flows along the connections toward the hidden slabs and the output slab. Each processing unit of the hidden slab transforms the incoming data by executing the equations associated with each processing unit. It then outputs the transformed data to the next layer. Each processing unit of the output slab does a similar transformation on the data it receives from the last hidden layer. The output of the output layer is then compared to the target output vector in the training slab. The errors between the two vectors are calculated. The error is then used to calculate new weights for all processing units in the hidden slabs and the output slab. The new weight for a processing unit results in a new computing equation associated with the processing unit. The determination of the new weights is based on a simple concept: the weights are corrected so that the error is lessened and as a result future responses of the network are more likely to be correct. This process is repeated until it appears that the network has learned as well as it can.

After the network has learned, it can be used in the producing mode. In the producing mode, the weight associated with each processing unit will not change. The only input to the network is the input vector. The output is the neural network response to such input. If the neural network is trained properly, the response of the trained neural network should agree with the desired output. Even if the inputs to the network have never been seen in the training process, a well trained network can still give an appropriate output response. This is called generalization.

Neuron

A neuron is the basic information processing unit in the neural network. Figure 2 shows a basic back-propagation neuron (Dayhoff, 1990). Every neuron has inputs, weights, an activation function, bias input, and outputs. The neuron is shown in the center, the inputs at the left, and the outputs at the right. There are n inputs, which are the outputs from other neurons or processing units in the previous layer, and m outputs. The outputs are fanned out to become inputs to the next layer of neurons.

There is a bias input to every neuron in the back propagation neural network. Usually the bias input is set to one. The bias can be viewed as a threshold of a neuron. It determines the activation level of a neuron.

There are weights associated with each input to the neuron. The input to the neuron is thus a weighted sum of all inputs on the left and bias input. The output of a neuron to a hidden or output slab is calculated by applying an activation function to the weighted sum of the inputs to the neuron.

Weight

Weights are numbers associated with each interconnection between neurons in the different layers. Before training, they are initialized to random small numbers and are adjusted during learning. After learning is completed, the weights are fixed. These final values of weights are then used during "recall" sessions. Figure 2 illustrates the weights along the incoming connections to the neuron.

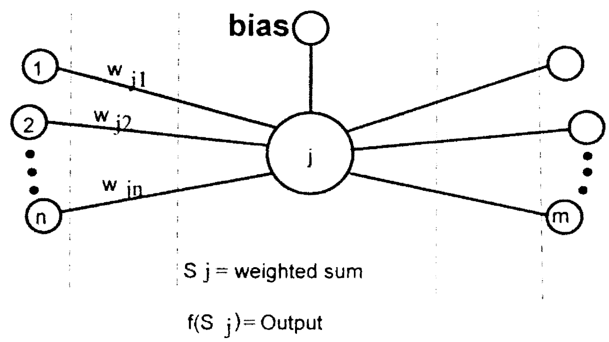


Figure 2. A Basic Back-propagation Neuron

Learning and Testing

Learning and testing are two main steps used in the implementation of a neural network. A network is said to adapt, or learn, if over time the response of the network becomes better. The back-propagation neural network employs supervised learning. During learning the network is presented with the desired output for every input. The actual output of the network is then compared with the desired output to produce a measurement of "error." The weights are adjusted to decrease the error between the network's output and the desired output. A training data set is used for training and is presented to the network many times. After training is stopped, the performance of the network is tested using a testing data set which was never seen by the network during the training.

Learning Algorithm

The back-propagation learning algorithm involves a forward propagating step followed by a backward propagating step. Both the forward and backward propagation steps are done for each input vector and the corresponding output vector. Once the forward propagation and backward propagation are completed on one set of input/output vectors, this iteration of the network is complete and the next iteration of the network is ready to begin.

Forward Propagation

The forward propagation step begins when the input vector is presented to the input slab. The input slab fans out the inputs to each neuron in the first hidden slab. The neurons in the first hidden slab calculate their outputs by applying a sigmoid transfer function to the summation of all the inputs to the neurons. The neurons of the first hidden slab then fan out the calculated outputs to the neurons in the next slab. This process continues on each successive slab. Every neuron sums its inputs and executes a transfer function to calculate its output. Finally the neurons of the output slab calculate the actual output of the network. The forward propagation step stops when the neurons of output slab output the neural network's result (Dayhoff, 1990).

Backward Propagation

After the forward propagation step, the network calculates the error by comparing the actual output of network with the desired output vector. Then the network changes the weights associated with each neuron in the output slab. The changing of the weights usually is based on the learning rule of the specific network. Back propagation neuron networks use the generalized delta rule (Appendix A). This process continues backward, starting with the output slab and moving to the first hidden slab. This process is called the back propagation step. The back propagation step stops when all the weights in the network have been changed. In this back propagation step the network corrects its weights in such a way as to decrease the network calculated error (Dayhoff, 1990).

The Error Value

The generalized delta rule calculates an error for every neuron in the network. The error value associated with each neuron reflects the amount of error associated with the weight of that neuron. This parameter is used during the weight correction while learning is taking place. A large error value indicates that a larger correction should be made to the neuron's weights. The sign of the error reflects the direction in which the weights should be changed (Dayhoff, 1990).

Processing Equations

The Multiple Layer Back Propagation network of the ExploreNet software was used in this study. The processing equations discussed here are from the software manual (HNC, Inc., 1991c).

Forward Propagation Equations

The forward propagation step is initiated when an input pattern is presented to the network and continues until the output slab neuron calculates the network output. For the input slab, the neurons fan out the input vector without any information processing. For the hidden and output slab, the neurons sum the inputs and calculate the output by applying the sigmoid transfer function to the summation.

Figure 3 illustrates a neuron of a hidden slab and the output slab in the forward propagation step. For the hidden slab, the calculated output is sent to upper level neurons as inputs. For the output slab, the neurons do not fan out the calculated output. The calculated output of the output slab is the actual network output.

In Figure 3, the $\mathbf{I}_{lij}$ (j= 0, 1, ... n) are the inputs from connection j. $\mathbf{N}_{li}$ is the neuron i. l indexes the slab. W_{lij} stands for a weight from connection j to neuron i in the slab l. $\mathbf{Z}_{li}$ is the output of the neuron i, which is the input to upper level neurons or the network output if neuron i is in the output slab.

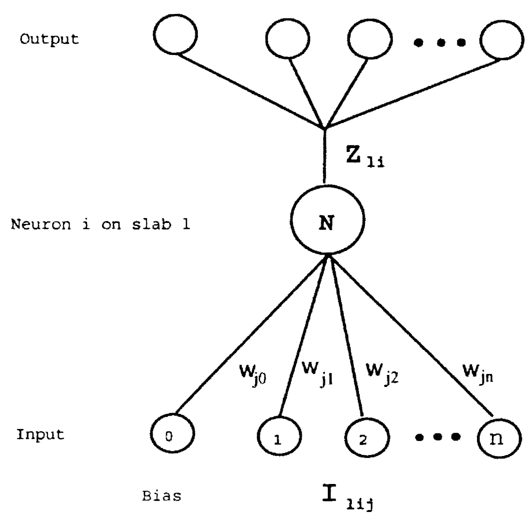


Figure 3. The Forward Propagation Step

Each neuron on a hidden or output slab has one connection from each neuron on the preceding slab (see Figure 1). Associated with each of these connections is an adaptive weight, W_{lij}, where l indexes the slab, i the neuron, and j the connection. In addition, each hidden and output slab neuron receives a constant input value of 1 from an auxiliary slab, called the bias slab. The weight associated with this constant input, W_{li0}, is called the threshold or bias weight. These threshold weights allow the network to approximate a broader class of mappings than would be possible without them. The output of a neuron of a hidden or output slab, z_{li}, is calculated by applying an activation function to the weighted sum of the inputs to that neuron. The equation for this process is given by

$$\mathbf{I}_{li} = \mathbf{W}_{li0} + \sum_{j=1}^{M_{l-1}} \mathbf{W}_{lij} \times \mathbf{Z}_{(l-1)j} \qquad (1)$$

$$\mathbf{Z}_{li} = \mathbf{f}(\mathbf{I}_{li}) \qquad (2)$$

where $\mathbf{M}_{l-1}$ is the size of the preceding slab, $\mathbf{Z}_{(l-1)j}$ is the output of the j^{th} neuron of the preceding slab, and $\mathbf{f}$ is the activation function. The function $\mathbf{f}$ is a sigmoid curve. It is illustrated in Figure 4.

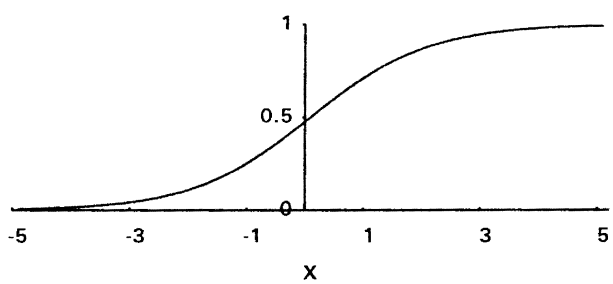


Figure 4. A Sigmoid Transfer Function

The sigmoid curve that was used is given by

$$f(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

This curve is relatively flat at both ends, and has a rapid rise in the middle. When x is less than -3, f(x) is close to 0; when x is greater than 3, f(x) is close to 1.

Since the operand of equation (2) is the weighted sum of neuron k, we have

$$\mathbf{f}(\mathbf{I}_{li}) = \frac{1}{1+e^{-\mathbf{I}_{li}}} \tag{4}$$

After the sigmoid function is computed on **I**, the resulting value becomes the output of neuron k. This value, the output of neuron k, is sent along all output interconnections to upper level neurons.

Backward Propagation Equations

Learning rules for the back propagation network requires that the network's desired output response be known for each input vector. Thus, during training each output neuron must also be supplied with a desired output by the training slab. After forward propagation, the desired output is compared to the actual output of the output slab. The network calculates the error for the output slab based on this comparison. These error values are passed back to neurons on previous slabs which use them to calculate their own error values; hence the name "back propagation." The weights of output and hidden slabs are adjusted using these error values so as to decrease the total mean squared error

(MSE) generated by the network over the training set of data. The MSE is defined as

$$MSE = \sum(obs - pred)^2 /n$$

where obs is the observed value and pred is the value predicted by the network. The learning rules have been derived so as to implement a gradient descent on the error function. The derivation of this result can be found in appendix A.

Figure 5 illustrates the back-propagation step. In this figure information flows from the output slab toward the input slab. Here the error values δ are calculated for neurons of output and hidden slabs and weight changes are computed for all interconnections. The calculations begin at the output layer and progress backward through the network to the input layer.

The calculation of error δ depends on whether the neuron is a member of the output slab or one of the hidden slabs. It is simple to compute for the output layer and somewhat more complicated for the hidden layers. The error for a neuron on the output is

$$\delta_{li} = \mathbf{f}'(\mathbf{I}_{li})(\mathbf{T}_i - \mathbf{Z}_{li}) \tag{5}$$

where

δ_{li} = the error for neuron i.

$\mathbf{T}_i$ = the target value for neuron i.

$\mathbf{Z}_{1i}$ = the output value for neuron i.

$\mathbf{f'}(\mathbf{I}_{1i})$= the derivative of the sigmoid function **f**.

$\mathbf{I}_{1i}$ = weighted sum of inputs to i

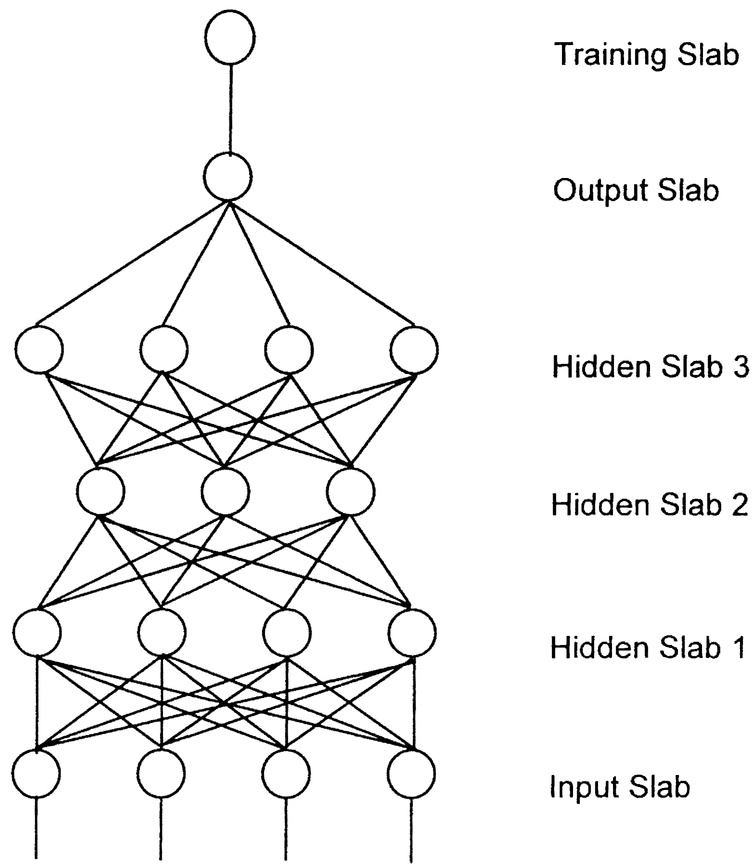


Figure 5. Backward Propagation in Training

The quantity ($\mathbf{T}_i$ - $\mathbf{Z}_{1i}$) reflects the amount of error. The **f'** part of the term "scales" the error to force a stronger correction when the sum is near the rapid rise in the sigmoid curve.

For the hidden layer, the calculation of δ is changed to a form that determines the contribution of each hidden slab neuron to the error seen at the slab above it. This calculation is given by

$$\delta_{li} = f'(I_{li}) \sum_{k=1}^{M_{l+1}} \delta_{(l+1)k} W_{(l+1)ki} \tag{6}$$

where M_{l+1} is the number of neurons on the subsequent slab. The sum over k represents the contribution of the i^{th} hidden element to the errors seen at the subsequent neurons.

The adjustment of the connection weights is done using the δ values of the processing unit. The equations for ΔW_{lij} and W^{new} are the same for output and hidden slabs. Each interconnection weight is adjusted by taking into account the δ value of the unit that receives input from that interconnection. The connection weight adjustment is

$$\Delta W_{lij} = \alpha \delta Z_{(l-1)j} \tag{7}$$

$$W_{lij}^{new} = W_{lij}^{old} + \Delta W_{lij} \tag{8}$$

This is called the generalized Δ rule. The variable α in the weight adjustment equation is the learning rate. Its value is commonly between 0.25 and 0.75 and chosen by the neural network user. The value of α reflects the rate of learning of the network. Values that are very large can lead to instability in the network, and unsatisfactory learning. Values that are too small can lead to excessively slow learning.

Convergence

When a network is trained successfully, it produces more accurate answers more often as the training session progresses. It is important to have a quantitative measure of learning. The MSE is usually calculated to reflect the degree to which learning has taken place in the network. This measure reflects how close the network is to getting the correct answers. As the network learns, its MSE decreases. Generally, a transformed MSE value below 0.1 indicates that a network has learned its training set (Dayhoff, 1990).

CHAPTER III

FACTORS AFFECTING THE SLUDGE BLANKET DEPTH

A Brief Overview of Activated Sludge Systems

The activated sludge process is a biological technique to remove organic matter in waste water. Figure 6 illustrates a typical biological waste water treatment plant. The raw water is the waste water to be treated. The plant mainly consists of two sections. The first is the biological reaction tank called the aeration basin, and the second is called the secondary clarifier. After treatment clear water flows out of the plant from the effluent pipe.

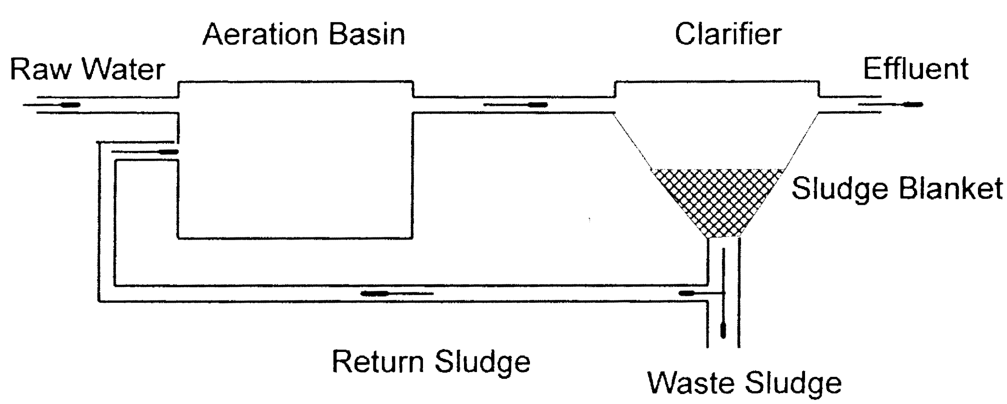


Figure 6. A Typical Activated Sludge Process

The aeration basin is the place where the biological treatment of the waste water takes place. Inside the aeration basin there are many microorganisms, for which the organics in the waste water serve as food. Thus the microorganisms reduce the organics in the waste water by converting the waste constituents to more microorganisms. The purpose of the aeration basin is to provide contact between the organics and the microorganisms in an oxygen-rich environment conducive to the growth of the microorganisms (Junkins et al., 1988).

The purpose of secondary clarifier is to separate the biomass generated by biological treatment. The mixed liquid in the aeration basin forms large flocs which flow into the secondary clarifier. These flocs settle to form a sludge blanket in the clarifier. The clear water is on the top of the clarifier and will flow out from the effluent weir. The sludge in the bottom of the clarifier leaves the clarifier from the bottom opening.

Some of the settled activated sludge leaves the clarifier and goes into the recycle line. The recycled sludge from the clarifier to the aeration basin is called return activated sludge. The purpose of the return sludge is to maintain a sufficient concentration of activated sludge in the aeration basin so that the required degree of treatment can be obtained in the time interval desired.

The excess activated sludge is wasted from the return sludge line. The purpose of wasting sludge is to maintain a constant level of mixed liquor suspended solids and a constant retention time in the aeration basin. The wasted sludge is discharged to the sludge handling facilities.

The basic mechanism of the activated sludge process can be represented by the following biological reaction:

$$\text{Organic Material} + \text{Microorganisms} \xrightarrow{O_2} \text{Microorganisms} + CO_2 + H_2O + \text{Energy}$$

To activate the above biological reaction, oxygen is added to the system in the aeration basin. The microorganisms in the water utilize complex organic material as a food source to produce more microorganisms plus carbon dioxide gas, water and energy. The microorganisms are eventually settled out from the secondary clarifier as waste biomass. The carbon dioxide gas is dispersed into the atmosphere. The water produced in the biological reaction leaves the system as part of the final effluent. The energy produced is utilized by the microorganisms to maintain their life systems (i.e., reproduction, digestion, and movement) (Junkins et al., 1988).

Sludge Blanket Depth

The sludge blanket is located at the bottom of the secondary clarifier as illustrated in Figure 6. The sludge blanket depth is the distance from the water surface of the secondary clarifier to the top of the sludge blanket. It is located at the top of the relatively thin, but quite homogenous, upper surface of the accumulated sludge blanket.

The sludge blanket depth in secondary clarifiers is checked every day. It is important to maintain a sludge blanket approximately three feet deep in clarifiers. Too high a sludge blanket indicates poor settling and/or solids not being withdrawn (recycled and wasted) from the bottom of the clarifier fast enough so that solids are accumulating quicker than they are being removed. Too low a sludge blanket means poor settling due to dispersed growth or solids are being pulled from the bottom of the clarifier at a too high rate. It is desirable to have a concentrated sludge blanket on the bottom of the clarifier in order to reduce the volume of liquid that must be recycled and/or wasted (Junkins et al., 1988).

Factors Affecting the Sludge Blanket Depth

Activities of bacteria in the aeration basin and sludge characteristics of the system determine the sludge blanket depth. Factors affecting the depth include raw waste water strength, nutrients in the system, dissolved oxygen in the aeration basin, detention time, pH, temperature, influent flow rate, and operator process adjustments.

Raw Waste Water Strength

The organic material in the waste water serves as a food source for the microorganisms in an activated sludge system, and the energy required by bacteria is derived from its oxidation as shown by the biological reaction. The chemical oxygen demand (COD) in the biological reaction is usually used as a measure of the amount of organics present in the waste water. COD is defined as the total quantity of oxygen required to oxidize all organic matter to carbon dioxide and water by the action of a strong oxidizing agent under acid conditions. COD reflects the waste water strength.

Any significant changes in the waste water characteristics affect the growth of the microorganisms in the treatment system. If COD loading increases significantly, then there may be too much food present for the microorganisms in the system. This excess food will result in a rapid growth of bacteria, which will produce a young biomass. The young biomass, sometimes called sludge, can cause poor settling in the secondary clarifier. If the organic loading decreases, there will be not enough food for the microorganisms in the system. This will reduce the growth rate and the system's biomass population could diminish. This will form rapid settling flocs and result in an increased suspended solids concentration in the final effluent. Therefore a proper balance between food and microorganism must be maintained (Junkins et al., 1988).

Food and Microorganism Ratio

In order to generate a good settling sludge, a correct microorganism population must be maintained to properly handle the organic materials coming into the system. The food to microorganism (F/M) ratio is defined as the ratio of pounds of influent COD to pounds of MLVSS under aeration basin. MLVSS is the mixed liquid volatile suspended solids, and is used to approximate the bacteria's concentration in the aeration basin (Junkins et al., 1988).

Nutrients

Microorganisms need nutrients such as nitrogen and phosphorus to sustain their life system. In industrial waste water, ammonia and phosphoric acid are usually added to provide sufficient nitrogen and phosphorus. The bacteria require nitrogen to produce other bacteria and phosphorus to generate the enzymes they need to break down organics in the waste water.

A rule of thumb is one pound of phosphoric acid with one pound of ammonia for 100 pounds of COD removed from the system. Insufficient nitrogen can result in filamentous bacterium that settles poorly. In addition, the lack of nitrogen inhibits the production of new cells, while the existing cells continue to remove organic matter. As a result, the microorganisms excrete excess by-products resulting in a fluffy floc which also settles poorly (Junkins et al., 1988).

Dissolved Oxygen

The biological reaction equation shows that the microorganisms need oxygen to survive. Oxygen is added to the aeration basin using mechanical methods. To ensure sufficient oxygen is being added to the system, a dissolved oxygen (D.O.) concentration of 1 to 2 mg/l should be maintained in the aeration basin. In the summer months, the bacteria are more active, and thus need more oxygen. In addition, as the temperature of the waste water increases, the oxygen saturation value decreases. These two phenomena require more oxygen be supplied to maintain the same D.O. concentration. In the winter, the amount of oxygen provided to the system should be decreased (Junkins et al., 1988).

DO Uptake Rate

Dissolved oxygen uptake rate is used to measure the rate at which D.O. is used in a sample of mixed liquid collected from the aeration basin. D.O. uptake rate reflects the activity of the microorganisms in the aeration basin. A low oxygen uptake rate in the aeration basin is an indication of impending problems. For example, lower than normal influent COD loading, improper pH, low D.O., or the presence of toxic material will cause low D.O. uptake rates. A high oxygen uptake rate indicates higher organic loading to the plant than usual. Dissolved oxygen uptake rate is measured every day in a treatment plant (Junkins et al., 1988).

Temperature

Temperature affects the settling characteristics in the secondary clarifier. In the winter, the colder waste water becomes more dense. This results in poorer settling sludge. In the summer, the sludge settles more easily. Also the temperature greatly affects the activity of microorganisms. When temperature increases by 20°F, the activity of the microorganism increases by a factor of two. In the plant operation, increasing MLVSS in winter and decreasing MLVSS in summer are used in order to compensate seasonal temperature change. Usually, the bacteria can thrive in a temperature from 80°F to 90°F (Junkins et al., 1988).

pH

A proper pH value is needed to maintain a healthy and active system. The biological activity is highly related to the pH value. Bacteria can survive in a pH range from 5.0 to 10.0, but they thrive between pH value of 6.5 and 8.5. When pH is below 6.5, fungi become the predominant organism in the water. This will produce poor settling solids. If the pH is too high, phosphorus will precipitate and become unavailable for bacteria. Under extreme high or low pH condition, the plant biological population will be killed (Junkins et al., 1988).

30 Minute Settling Test

The 30 minute settling test is one of the best process monitoring tools. It is a measure of sludge settle ability in the clarifier. The 30 minute settling test simulates the condition of the secondary clarifier. A strict procedure is used to accomplish the test. The sample used in the test represent the mixed liquid flowing out of the aeration basin to the secondary clarifier. The sample is collected either just before it leaves the aeration basin or at some point prior to its entry into the secondary clarifier. This test is a key indicator of sludge condition because it simulates the condition in the secondary clarifier. Therefore, the thirty minute settling test can be used to predict sludge blanket depth (Junkins et al., 1988).

CHAPTER IV

NEURAL NETWORK DESIGN

The Concept Phase Design

The concept phase develops the approach to building the neural network application. It determines which type of application to consider. Then according to the type and requirements of the application, the proper neural network paradigm is chosen. Finally the neural network size, output type, and training method are decided on the basis of the selected network paradigm (Bailey and Thompson, 1990a).

There are about two-dozen neural network paradigms. To select a proper paradigm, the type of application represented by the project must be determined. The current project is to predict the sludge blanket depth based on a number of input variables. It is a data modeling or functional mapping application. For a data modeling application, there are basically two kinds of suitable neural networks. They are counter propagation networks and back propagation neural networks. Since the back propagation network is better than counter propagation network in data modeling, the back propagation network was selected for the network paradigm. Within the back propagation network family, multiple layer back propagation networks are very popular and have more processing power. Thus, a multiple layer back propagation network was selected.

The output of a back propagation neural network can be pattern, real number, and classification output. The output requirements of the application determine the neural network output type. The output of this application is sludge blanket depth, which is a real number. Thus, the output of network must be a real number.

The training method is limited by the selected neural network paradigm. Back propagation neural networks require supervised training. Supervised training requires pairs of data consisting of an input vector and the correct result. The training data for this study is the database of the waste water treatment plant of the Gerber Baby Food factory.

In the training data, the desired output is the sludge blanket depth. It is fed into the training slab to "teach" the network. The input vector consists of eleven parameters. The input vector elements are waste water loading rate, mixed liquid suspended solids in the aeration basin, food-microorganism ratio, ammonia nitrogen, nitrate nitrogen, dissolved oxygen, D.O. uptake rate, temperature in the aeration basin, recycle rate, recycle suspended solids, and 30 minute settling in the clarifier.

The HNC Neurosoft™ Multilayer Back propagation Network (MBPN) software package was used in this study. The HNC version of the MBPN is derived from the work of Professor

David Rumelhart and the Parallel Distributed Processing Group (HNC Incorporated, 1991c).

HNC-MBPN supports up to three hidden layers. It also offers a choice of five parameterized activation functions and can implement four versions of the learning rule: normal update, batching, smoothing, or batching and smoothing. The network may be configured to calculate performance statistics, such as Mean Squared Error. The error value for each neuron is stored as a local memory variable for easy retrieval during network debugging. All network states, weights, and local memory values use 32-bit, floating-point representations, making the input and training data very easy to scale.

The Design Phase Design

The design phase specifies initial values and conditions for the selected neural paradigm at the node, network, and training levels. The design phase comprises several steps to determine the type of nodes or processing elements, size and connectivity of the network layers, and learning algorithm and parameters (Bailey and Thompson, 1990a).

The Neuron Level Design

The neuron level design is to determine what neurons or processing units to use. First the type of inputs to neurons should be determined. The inputs to each neuron in the network was selected as real numbers. The inputs to the first layer neurons need to be scaled so that they all fall within the active range of the transfer function of the neurons. The active range of a transfer is shown in the figure 7. Next a procedure for combining inputs needs to be selected. A commonly used combination of inputs is a weighted sum of the inputs, which is supported by the HNC-MBPN software.

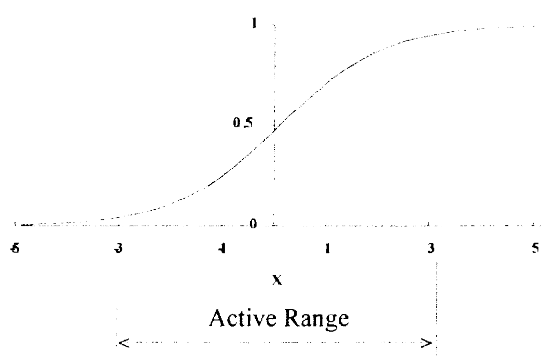


Figure 7. The Active Range of a Transfer Function

The final step in the neuron level design is to choose a transfer function based the type of inputs and outputs and learning algorithm to be used. Back propagation networks require that the transfer function be differentiable at all points. Also the neurons manipulate continuous values such as linear or sigmoidal transfer functions. The sigmoidal function was chosen as the transfer function. HNC-MBPN provides four kinds of sigmoid transfer functions. They are logistic, hyperbolic tangent, threshold linear function, and inverse tangent. Within these four transfer functions, the logistic is the most widely used. Therefore initially the logistic transfer function was chosen. The parameters of the transfer function need to be determined by experiments.

The Network Level Design

The network level design is to determine how to put individual neurons together to form a network. Decisions at the network level design include the number of slabs, the size of each slab, and how to connect each slab (Bailey and Thompson, 1990a).

HNC-MBPN requires one input slab, one output slab, and a variable number of hidden slabs. The maximum number of hidden slabs cannot be over three. Hidden slabs are used to abstract and pull features from inputs. Increasing the number of hidden slabs augments the processing power of the network but significantly complicates training, increases the training time, and requires more training examples. A rule of thumb is to start with one hidden layer and monitor the training results. If it is hard to train the network or the MSE increases, add more hidden slabs.

After selecting the number of slabs for the network, the second step is to determine the number of neurons in each slab. Choosing the proper number of neurons in each slab of the network is critical. Different configurations have a great deal of impact on later network performance.

The input slab is used to present all the input variables to the network. The number of nodes in the input slab should be enough to present input patterns to the network. Since all the input variables are real numbers, a single node in the input slab can be configured to represent one input variable. Thus the number of nodes in the input slab is equal to the number of input variables. The sequence of input variables presented to the input slab can be any order.

For the output slab, the number of neurons required is determined by neural network paradigm and the type of expected output from the network. The output variable is to predict the sludge blanket depth which is a continuous number. A single neuron can hold a real number in this network. For the back propagation network, the number of output slab neurons is equal to the training slab neurons. Therefore the output slab has one neuron which outputs a real number representing the predicted sludge blanket depth.

For the hidden slabs, the number of nodes in each slab is determined through experimentation. Finding the right number of hidden neurons is the most challenging aspect of configuring a back-propagation network. With too few neurons in the hidden slabs, the network can not perform the complex mapping. On the other hand, with too many neurons, the network can easily find a set of weights to memorizes all the input patterns. This is called table lookup. In this case, the network just memorizes all the input patterns which have been presented to the network. The network actually does not extract the salient features of input patterns. When new input patterns are presented to the network, the network will give an incorrect response. Therefore, the network loses the capability of generalization.

As a general rule, to eliminate table lookup, the number of neurons in each hidden slab is decreased. Decreasing the neurons in the hidden slab also improves generalization capabilities and performance to new input patterns. To improve the network processing power and performance accuracy, increase the number of neurons on each hidden slab. The number of neurons in the first hidden slab should approximately equal the number of nodes in the input slab. If there is more than one hidden slab, reduce the number of neurons in the second and third slabs sequentially. The number of neurons in the highest hidden slab can not be equal to the number of output slab neurons. Otherwise the output slab is redundant. Based on these considerations and experimentation, eleven neurons were chosen for the first hidden slab which is equal to the number of nodes in the input slab. Eight neurons were used in the second hidden slab. Six neurons were used in the third hidden slab which is more than the number of output slab neurons.

The connectivity of the network describes how neurons are linked together. The HNC implementation of the Multilayer Back Propagation Network is a feed-forward neural network. Usually feed-forward networks connect nodes in one slab to those in the succeeding slab with no feed-back and no intralayer connections. This is called a fully connecting adjacent slab network. Generally, this kind of connecting configuration is best because it provides the most flexibility when the training algorithm is searching for suitable weights. The training algorithm can nullify unnecessary links by setting their weights to zero. Also the HNC software provides special connections between the input vector and the output vector. If these connections are enabled, each output neuron receives an input from each node in the input slab.

In this work the fully connected network without the input-output connection was used. For the input slab, every node is connected to each neuron in the first hidden slab. Every neuron in the hidden slab is connected to each neuron in the succeeding hidden slab. The neurons in the third hidden slab connect to the neuron in the output slab. There are no feed-back and intralayer connections in the designed network. Figure 8 illustrates the design of the network.

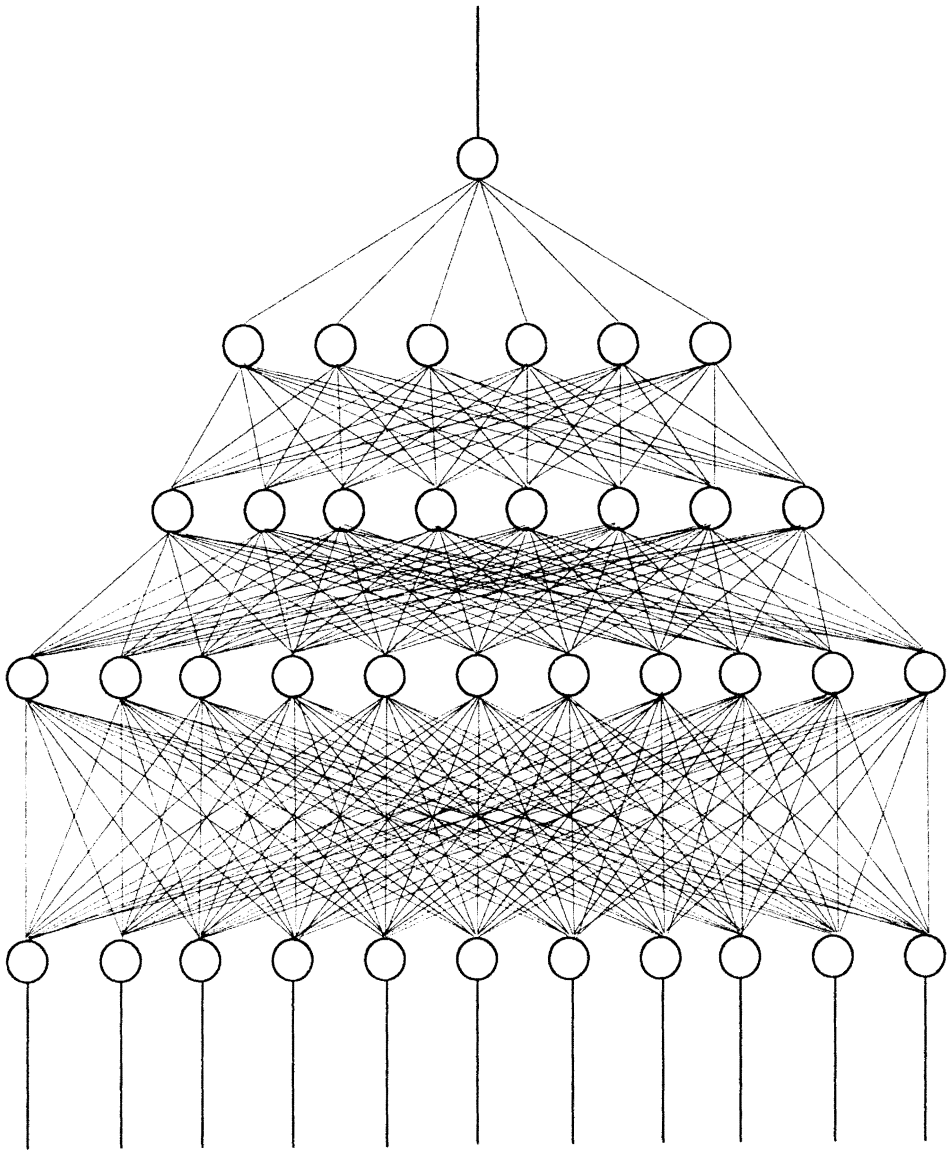


Figure 8. The Design of the Network

CHAPTER V

IMPLEMENTING THE DESIGNED NEURAL NETWORK

The implementation phase builds on the decisions regarding the network characteristics made in the design phase. The goal of implementation is to create a functioning neural network that provides the most accurate, consistent, and robust model possible. Iterative building, training, and testing is used to refine the neural network (Bailey and Thompson, 1990b). The implementing process includes gathering the training and testing data sets, preparing the data sets for training and testing the network, designing the application using the ExploreNet software, building the individual module for the application, training the network, and testing the training process.

Gathering the Training and Testing Data Sets

To implement a neural network, two sets of data are needed. The first one is the training data set. The second is the testing data set. The training data is used to train the network. After the network is trained, its performance is assessed by the testing data. Thus the first step to implement a neural network is gathering the training and testing data sets.

The quality of training and testing data sets plays an important role in implementing a neural network. To train a network properly, the training data should be representitive of the problem. It should include routine as well as unusual observations and all of the boundary condition cases likely to be encountered (Bailey and Thompson, 1990b). Usually collecting as much training data as possible can solve the problem. However, if the training data's quality is poor, the network can not learn the salient features of a function. Thus, rather than just focusing on data volume, one must concentrate on the quality and representativeness of the data set. To improve the quality of a data set, statistical methods are used to select the data set from all possible records. The testing data set is used to test the performance of a network. The testing data is collected at the same time as the training data collection. The data type to collect is determined by the problem. The current study requires continuous data.

The data sets used to implement the network were collected from a database of Stover and Associates. It is a Dbase III Plus database (Dbase III Plus, 1986). Three years of data were available. Every day there is one record. Each record consists of ninety-eight fields. Only twelve fields in this database were used in the network implementation. These fields are the input and output fields of the network to be implemented. The sludge blanket field is used as an output field for the network. The input fields for the network are loading rate, mixed liquor suspended solids in the aeration basin, oxygen uptake rate in the aeration basin, food-microorganism (F/M) ration, pH value in aeration basin, dissolved oxygen in aeration, recycle suspended solid concentration, recycle flow rate, ammonia nitrogen in clarifier, nitrate nitrogen in clarifier, and thirty minutes settling value in clarifier. The output field will connect to the training slab during training and serve as a teacher. During testing the output values do not participate in the calculations of the network and are used to assess the performance of the network. The input fields are input to the input slab during the training and testing of the network. The testing data set was collected from the database at the same time as the training set was collected since both data sets consist of the same fields.

Usually there are no records for the weekend. During the weekday sometimes there are missing data for some fields. When extracting records from the database, records with zero values were not included in the input and output fields from the database. The data was exported from the database into text file format. The operation to export data from the database is done by a Dbase III Plus program (Appendix B).

After the data is exported from the database, it can not feed into the network directly. The data set presented to the network must be meaningful to the neural network. If the training data has too many errors, it will complicate the network's learning process. Therefore, some data preparation is necessary.

To evaluate the quality of records exported from the database, multiple regression was applied. It was found that the R square value for a linear regression relating sludge blanket depth to the input variables based on all of the data was less than 0.1. There is almost no correlation between sludge blanket depth and the selected input variables for the whole data set. However, applying multiple regression to small subsets of records, the R square values vary from 0.1 to 0.9. This means that for some time periods the input and output are correlated and for some they are not. The procedure that was used to screen out data that apparently did not correlate well with the remaining data was based on multiple regression. An R square of 0.7 was selected as a dividing point. Multiple regression was applied to sequences of records. If the R square was less than 0.7, this sequence of records was removed from the data set. After this process, the R square of the whole remaining data set was greater than 0.5. The selected data set had 392 records.

The data set had too many incomplete records to enable observations lagged by one day or more to be used as input variable.

The selected data set was split into a training data set and a testing data set. Sixty records were picked as the testing data set. This ensured that the characteristics of testing data set are the same as the training data set. The sludge blanket depth in the testing data set was not fed into network during the training session.

The sludge blanket depth in the testing set was used to assess the performance of the network by comparing it with network output during the testing session. If the network successfully predicts the value of the sludge blanket depth in the testing data set, the network is trained successfully. The training data set is listed in table 5 in Appendix C. The testing data set is in table 6 in Appendix C.

Data Set Preparation

After the training and testing set was collected and bad data were removed, the data sets were prepared for training and testing the network. The purpose of preparing the data sets for the network was to convert the data into a proper form and make the training process easier. The data were standardized using the Z_score transformation by substracting the mean and dividing by the standard deviation. The transformed sludge blanket depth was scaled to the range -1 to +1.

The reason for using this transformation is as follows. The neural network is a computing technology. It has some special characteristics. First, the neural network responds to absolute magnitudes. It does not place any physical importance on input variables. If a variable has a large magnitude, it will have much influence on the network's output. In other words, the bigger the magnitude of a variable, the larger the variable's influence on the network even though the larger variable may in fact be less significant. Second neural networks not only pay attention to the magnitude of input variables, but to their variability as well. For instance, if there are two input variables with the same magnitudes but one has values fluctuating between the top and bottom of the data range and the other having values staying at one end of the data range, the two variables will have different influence on the network (Crooks, 1992).

For this problem it can be sees that the input variables have different magnitudes. This mainly is because the different input variables have different measuring units. It is hard to say that the bigger the variable, the more significant it is. For example, one can not say that the F/M ratio is less significant than the pH value in determining the sludge blanket depth. The input variables also have a lot of fluctuation. For example, the pH value ranges from 7.0 to 8.0 most of time and seldom is below 6.0. The dissolved oxygen uptake rate is usually below 5.0. It is hard to say that pH is more significant than dissolved oxygen uptake rate. Since the neural network pays attention to the magnitudes and variability of input variables, the data must be scaled to make them comparable in magnitude and variability. One solution is to scale all input variables and sludge blanket depth using the Z_score transformation.

The Z_score is defined as a deviation from the mean in terms of a number of standard deviations. The Z_score is computed by subtracting the mean and dividing the difference by the standard deviation.

$$Z\_score = \frac{\text{Observed} - \text{Mean}}{\text{Standard deviation}} \tag{9}$$

Since there are 11 input variables and one output variable, 12 means and standard deviations are calculated. The mean and standard deviation calculations are based on data in table 5 and table 6 in Appendix C. After the Z_score transformation, the transformed data mainly varied from -3 to +3. This data range is within the activated range of the transfer function of neurons assuming that the sigmoid transfer function is used and the upper and lower bound is from -1 to +1.

The transformed sludge blanket depth ranged from -3 to +3. Since the bias input of the output neuron is one, the output of the network was scaled to range between -1 and +1. This scaling was done by dividing by 3.

All the above Z_score transformations and scaling on the input and output variables are completed within the ExploreNet software. The means and standard deviations for all the variables are calculated outside the software to save network training time.

Design of the Network

After preparation of the training data set, the next step is to design the network using the ExploreNet software. ExploreNet is a workbench for building, training, testing, and applying neural networks. It provides retrieving, formating, transforming, displaying, and saving facilities. ExploreNet applications usually are built into modules. Each module in an application represents a software device to perform some specific task (HNC, Inc., 1991a).

Nine module types are available. The modules used in this application were the file, form, data, network, and graph modules. File modules are used to connect the ExploreNet application to a disk data file. The form module is used to set up interactive customized screen forms. It was used to display the MSE, desired output, and neural network output during training and testing. Data modules provide the capability to manipulate data using mathematical and logical computations. They were used to transform and scale the input data before the data are fed into the network. The network module was used to define, create, and run the network within an ExploreNet application. It was used to implement the multiple layer back propagation paradigm. Graph modules were used to display two dimensional graphs on the screen showing both desired and network outputs during the training and testing of the network.

To identify data to be manipulated by a module, items must be created in ExploreNet. Items are used to store values. A module may have three kinds of items. Input items are items whose value may be supplied by a connection from one module to another module. The values of output items from one module may be connected to supply values to input items for another module. Both input and output items are created, described, and named by the user. The control items are build into the modules and can not be created by the user. The user controls the behavior of a module by setting the values of control items.

Modules are linked by connections. The connections between modules are a mapping of all items from one module to items of another module. The correspondence between source output items and destination input items constitute the mapping. Connections are used to transfer the data between the source module and destination module and to determine which of the connected modules must be executed first. The connections among modules are created by the user.

The design of this application is shown in Figure 9. The name of a module is labeled in this diagram. The input module is a file module. Its purpose is to read from an input file. A set of input data corresponds to one record in the training data file. The input module passes the set of input data to the Z_score module. The Z_score module is implemented by the data module. The Z_score module performs the Z_score transformation. The Z_score transformed record is passed to the scale module. The scale module scales the transformed sludge blanket depth. The scale module is a data module. The transformed records from the scale module are fed into the neural network. Input variables are mapped into the input slab of the network and sludge blanket depth is mapped into the training slab. At the same time, the transformed sludge blanket depth from the scale module is send to the desired output module that is a graph module where it is plotted on screen for monitoring the training process. The network module implements the multiple layer back propagation. The behavior and mode of operation of network modules are controlled by setting predefined control items. The MSE, desired output, and network output are directed to the MSE module. The MSE module is a form module. It displays the actual value of MSE, desired output, and network output in a form for monitoring the training process. The neural network prediction of sludge blanket depth is output to a scale back module. The scale back module converts the network's output into proper units. Finally the network's output is plotted by an output module that is a graph module.

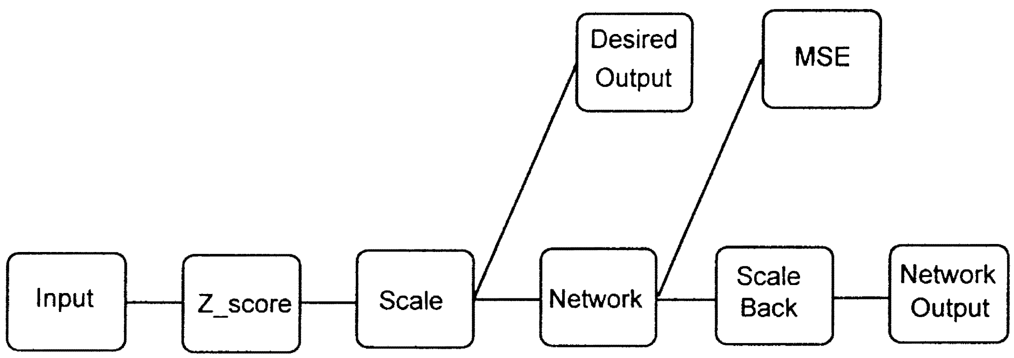


Figure 9. Design of the Application

A module is created by choosing the type of modules needed and specifying a variety of options and parameter values to tailor the module to a specific need. The process of specifying how a module is to behave is called building the module. The ExploreNet has build windows for each module. Each module has a different build window. The build window is used to define how the module will work.

ExploreNet also provides the ability to create a customized item type. Before building an individual module, three customized items were created. The first one was called input[12] item. It is a vector with twelve elements. Each element is defined as a real number. Elements indexed from zero to ten are used to represent independent variables. The twelfth element is used to stand for the dependent variable of sludge blanket depth. The second item was called output[12] item. Output[12] has a similar definition to input[12]. The third item was called predict[1]. It is defined as vector with one real number element. Input[12] is used as an input item to modules and output[12] as an output item. Predict[1] is used to store an output value.

Building the Input Module

The input module is built from a file module. The file module is used to access records in the training data file or cases to be solved after training. By default the file module is set to work with ASCII data files. The files must be delimited data files with commas or spaces between every field and a carriage return or line feed character (or both) after every data record. The input data file used was an ASCII file delimited by blanks between fields and a line feed between records.

The first step in building the input module is to specify the module type as an input module inside the type menu option of its build window. The define menu is used to define the input file name and file search path. Finally under the define menu, the input item is created by naming it and specifying the item type. One item was created in this module called input[] which has an item type of input[12]. Therefore, the input item is a vector with twelve elements. Elements indexed from 0 to 10 are for independent variable. The twelfth element of input item is the value of sludge blanket depth.

Building the Z_score Module

The Z_score module was built from the data module. The Z_score module was used to compute the Z_score transformation of input data. The input vector was passed from the input module to the Z_score module. Every element of the input vector is transformed into its Z_score value. The mean and standard deviation of each variable was calculated using spreadsheet software.

The Z_score module is built within a data module using input items, output items, and transformations. The Z_score module has one input item and one output item. They are input[12] and tr_input[12]. Twelve transformations are created from

$$\mathbf{tr\_input}[\,i\,] = \frac{\mathbf{Input}[\,i\,] - \mathrm{mean}_i}{\mathrm{std}_i} \tag{10}$$

where $mean_i$ and std_i are the mean and standard deviation shown in table 1.

TABLE 1

INPUT MEANS AND STANDARD DEVIATIONS

Index	Variable Name	$Mean_i$	Std_i
0	Loading Rate to Aeration, lbs/day	8894	7609
1	MLVSS in Aeration, mg/l	3773	873.9
2	D.O. Uptake Rate, mg/l	21.85	5.745
3	F/M ratio in Aeration Basin	0.211	0.158
4	pH in Aeration Basin, s.u.	7.334	0.301
5	Dissolved Oxygen, mg/l	3.696	1.636
6	Recycle Suspend Solids, mg/l	6482	1474
7	Recycle Rate, MGD	0.507	0.164
8	NH3-N, mg/l	1.104	3.005
9	NO3-N, mg/l	35.25	15.23
10	30 min. Settling, ml/30 minute	543	240.6
11	Sludge Blanket Depth, ft	8.754	1.614

Building the Scale module

The scale module was also built from the data module. It is used to scale the Z_scale transformed sludge blanket depths. The scale module has two items and two transformations. These two items are input[12] data type.

They are similar to the two items in the Z_scale module. The transformations of the scale module are

$$\mathbf{tr\_\ input[\,0\,..\,10\,]} = \mathbf{Input[\,0\,..\,10\,]}$$

$$\mathbf{tr\_\ input[\,11\,]} = \mathbf{Input[\,11\,]} \div 3 \qquad (11,\ 12)$$

Building the Network module

The network module is the reason for this project. Building the network module is to define, create, and implement the ideas in the network design phase. It is the most important part of this project.

The network module in ExploreNet provides predefined neural networks and interactive control of network parameters and operations. In this project MBPN paradigm is the neural network to be implemented. ExploreNet associates each neural network paradigm with a different network module. All items are predetermined by the neural network paradigm (i.e., type). These items consist of the network's constants, states, weights, and local memory. Building a MBPN network module is to define these constants, states, and local memory. During the implementing cycle of this application, iterative building, training, and testing is used to refine the neural network parameters. Many experiments were done to finalize all these parameters. The experimental results of these parameters are shown in Figure 10. and 11.

MBPN Load Time

Network Size Parameters

Input Slab Size	11	Number of Hidden Slab	3
Output Slab Size	1		

	Hidden 0	Hidden 1	Hidden 2
Hidden Slab Sizes	11	8	6

Miscellaneous Parameters

Connect Input to Outputs	NO	Random Seed	0.0
Initial Weight Maximum	0.1	Learning Method	Normal

Activation Function Parameters

Activation Function Type	Logistic

Lower Bound	-1.0	Upper Bound	+1.0	Steepness	1.0

Activaton Function Lookup Table Parameters

Number of Entries (Zero for no table)	0	Low Limit	0.0	Upper Limit	0.0

Figure 10. MBPN Load Time Constants

MBPN Run Time

Learning Control Parameters

Learning Enable	ON	Learning Method	Normal
Error Tolerance	0.0	Batch Size	0

Learning Rate Parameters

Hidden Slab 0 Alpha	0.1	Hidden Slab 0 Beta	0.0
Hidden Slab 1 Alpha	0.1	Hidden Slab 1 Beta	0.0
Hidden Slab 2 Alpha	0.1	Hidden Slab 2 Beta	0.0
Output Slab Alpha	0.1	Output Slab Beta	0.0

Miscellaneous Parameters

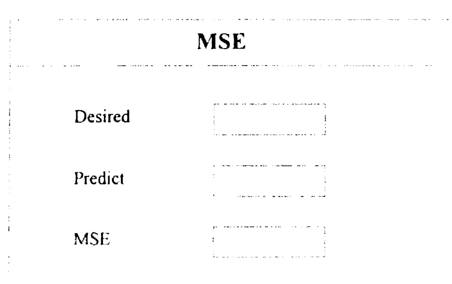
Statistics Enable	on	Table Lookup	off
Linear Activatoin Function on Output Slab	no		

Figure 11. MBPN Run Time Constants

The network parameters can be classified into load time parameters and run time parameters. In ExploreNet, they are called load time and run time constants. The load time constants contain information that is used in creating the network, such as slab sizes, connection patterns, and an initial random seed for initializing weights. These parameters may not be changed after the network has been loaded or created. The run time constants contain parameters, such as learning rates, which may be changed after the network has been created to alter the way in which it operates (HNC, Inc., 1991a).

Building the MSE Module

MSE module is used to monitor the training session. The MSE module displays the desired output, predicted output, and MSE variable of the network for every record presented to the network. During the training session, the MSE module displays the following window dynamically.



The MSE module is built from the form module provided by ExploreNet. During the design of the MSE module, three modules are created. They are called desired item, predict item, and MSE item. All these items are real numbers. Desired item gets the value from the training slab that is the target output. Predict item is mapped from the variable in the output slab and is the network output. MSE item is mapped from the statistics slab and shows the mean square error of the network.

Building the Scale_back Module

The scale back module is used to scale the network prediction back to their original units. The scale back module is built from the data module in the ExploreNet software. This module processes only one input item, one output item, and one transformation. The input item to the scale back module is mapped from the network prediction and is called mbpnout. The output item has a data type of real that is called Out. The output item is used to hold the transformed value of the scale back module, which is a real data type. The transformation that is applied is

$$\text{Out} = \text{mbpnout} \times 3 \tag{13}$$

Building the Screen Display Modules

The screen display modules are used to display graphically the desired output and network's output on customized screen windows. There are two separated display modules in the application. They are called the desired output module and network output module. They are all built from the graph module in ExploreNet. Each module has one item that will be plotted on the customized windows. The item in the desired output module is mapped from the training slab in the network. The item in the network output module is mapped from the output slab in the network. These two items have the data type of time, which means that they are going to plot against the time sequence.

Building the Connections and Mappings Between Modules

After building the individual modules, the connections and the mappings in each connection are created. The connections between modules control the data flow within the application. The mappings inside each connection determine how data flow from one module to another. The connections between modules are shown in the Figure 9. The mappings for each connection are shown in the following equations. In these equations, the left side of an equation is the item name of the destination module; and the right side is the item name of source module.

```
Z_score.input[0:11] = Input.input[0:11]                          (14)
Scale.input[0:11] = Z_score.tr_input[0:11]                       (15)
Network.input_slab.state[0:10] = Scale.tr_input[0:10]            (16)
Network.training_slab.state[0] = Scale.tr_input[11]              (17)
Desired_output.sdp.y = Scale.tr_input[11]                        (18)
MSE.desired = Network.training_slab.state[0]                     (19)
MSE.mse = Network.statistics_slab.state[0]                       (20)
MSE.predict = Network.output_slab.state[0]                       (21)
Scale_back.mbpnout = Network.output_slab.state[0]                (22)
Net_output.sdp.y = Scale_back.out                                (23)
```

In the above equations, equation (14) is the mapping from the input module to the Z_score module. After the input vectors are transformed into Z_score values, they are directed to the scale module through mapping equation (15). Then data are mapped into the network module through equations (16) and (17). Equation (16) does the mapping of the elements indexed from one to ten of tr_input[0:10] into the eleven neurons of the input slab of the network. Equation (17) maps the sludge blanket depth into the neuron of the training slab in the network module. Equation (20) transfers the network's output in the output slab into the scale back module. After the scale back process, the scale back module passes the scaled result to the network output module to plot the result on the screen display window. Equation (18) does the mapping between the scale module and desired output module for displaying the output graphically. Equations (19), (20), and (21) complete the mapping between the network module and MSE module for monitoring the actual value of desired output, predicted output, and MSE during the training process.

Training and Testing the Network

After the networks are constructed for all the modules, connections, and mappings, the training process takes place. During the training session, only the training data set is applied to the network. Usually the training session takes several thousand cycles for this application. When training the network, the training process is monitored through the display of the MSE module and the graphic display of the desired output and network output. Sometimes at the beginning of the training, the network did not show any learning. If this is the case, the procedure is stopped and the network parameters are modified. The training process is restarted. When the mean square error of the training process reaches a minimum, the training process is completed. If the MSE is small, the testing process can be applied. If the MSE is large, the training process has failed and the network must be rebuilt by redefining transformations and using other possible methods that could train the network.

After the training process, the testing process takes place. The testing process is the final step in determining if the training process was successful. If the network has learned the functional relationship contained in the training data set, the testing process serves as a method to access the performance of the network.

During the testing process, only the testing data set is plugged into the network. During testing the process is monitored through the graphic display of desired output and network output. If the plots of desired output and network output basically matched each other, it indicates the network has been trained. Otherwise the network is not trained with the capability of generalization to data outside the training data. This indicates the network did not learn the functional relationship inside the training data. Instead it remembered all the training cases presented in the training set. If this was the case, the network had to be rebuilt and another training process undertaken.

The design of the network parameters, data transform equations, and the mapping equations presented in this chapter are the final results after the training and testing process. After the testing process showed successful training, the network's output and the desired output were placed in a disk file for analysis.

CHAPTER VI

RESULTS AND DISCUSSION

The training processes took about 300,000 training cycles using 332 training records. The final training mean square error was 0.15 feet.

Figure 12 shows the trend plot of the actual sludge blanket depth used in the training process. The actual sludge blanket depth was used as a target output during the training processes. The predictions of the sludge blanket depth for the training data are also shown in the Figure 12. The depth units of the trend plot are feet which are the actual measurement units of the sludge blanket depth.

Figure 13 is a plot of the network prediction against the target output for the training processes. Figure 14 is the plot of the errors of the network output for the training processes. In Figure 14, the mean of the error is 0.028 feet, the standard deviation is 0.39 feet, and the MSE is 0.15 feet.

Figures 12 and 13 show that the trend of the prediction is in agreement with the target trend. This indicates that the network has learned the functional relationship between the input variables and the target output for the training data. The network output is smaller than the target output.

The fact that the predictions are not in exact agreement with the observed data indicates that the network is not over trained. An over trained network memorizes all of the input and output patterns in the training data set. The network prediction and the target output would be exactly the same if the model is over fitted to the training records.

To evaluate the training process, linear regression was applied to the predicted and the observed sludge blanket depths. The regression results for predicted versus observed sludge blanket depth of the training process are shown in Table 2.

TABLE 2

REGRESSION OUTPUT FOR TRAINING DATA

Constant (α)	0.3019
Std Err of Coefficient α	0.0959
Std Err of Y Est	0.269 ft
R Squared	0.9648
No. of Observations	328
Degree of Freedom	326
X Coefficient (β)	0.9656
Std Err of Coeficient β	0.0109

In Figure 13 there four points plotted at 3.9 feet on the predicated scale. This plotting results from constraints in the configuration of the neural network. These four points were omitted in the above regression analysis.

The regression output shows that the overall correlation coefficient for the network prediction and the observed sludge blanket depth is 0.98. The slope of the regression line is 0.97 feet and the intercept is 0.30 feet. To evaluate the regression slope and the intercept, a test of hypotheses concerning the intercept α and the slope β are shown as follows (Haan, 1977)

To test $H_0 : \alpha = 0$ versus $H_a : \alpha \neq 0$, compute

$t = (a - 0.0)/Sa = (0.3019 - 0.0)/0.0959 = 3.148$

$t_{.975,330} = 1.96$

Since $|t| > t_{.975,330}$, reject $H_0 : \alpha = 0$.

To test $H_0 : \beta = 1$ versus $H_a : \beta \neq 1$, compute

$t = (b - 0.0)/Sb = (0.9656 - 1)/0.0109 = -3.156$

$t_{.975,330} = 1.96$

Since $|t| > t_{.975,330}$, reject $H_0 : \beta = 1$.

Even though statistically the intercept is significantly different from zero and the slope is significantly different from one, the actual differences are so small as to have little physical significance.

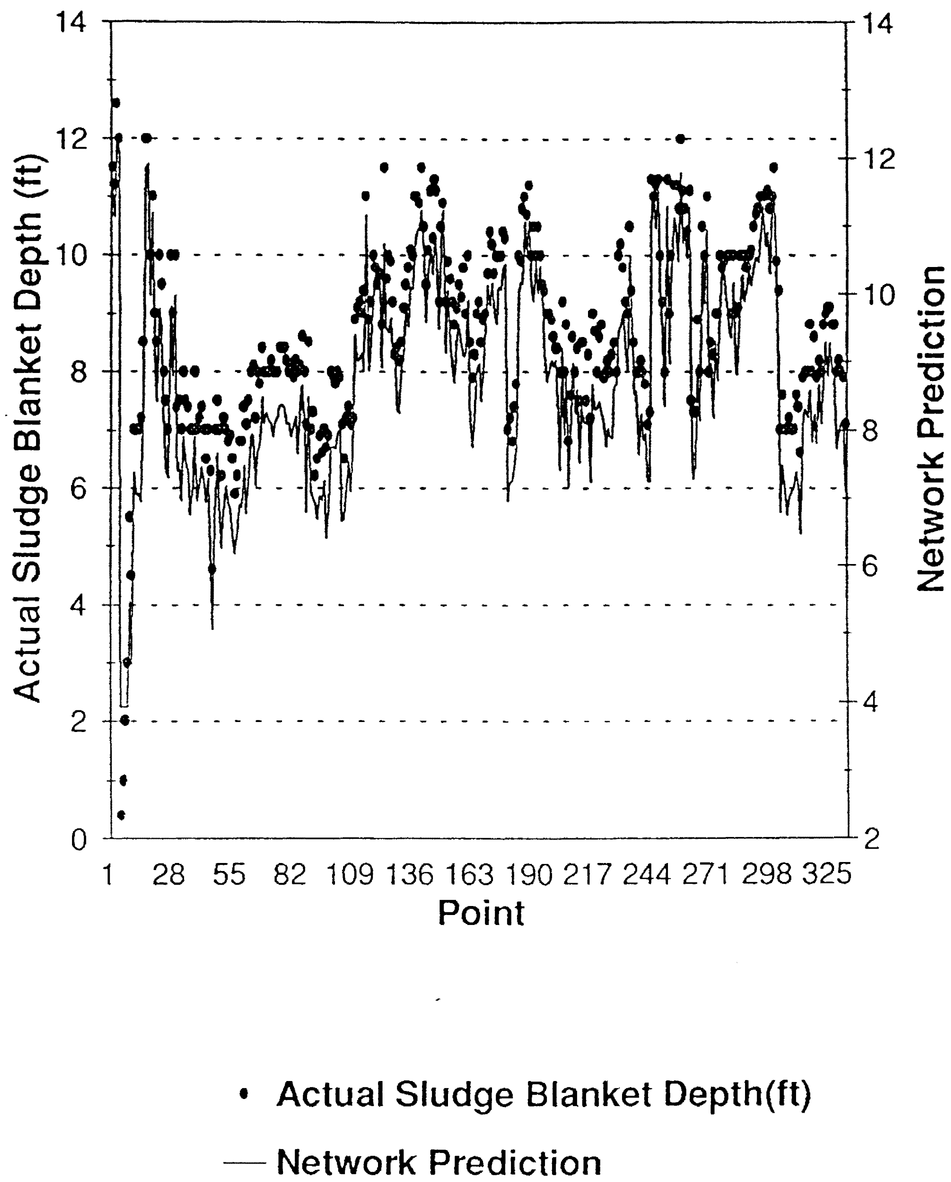


Figure 12. Network Prediction and Target Output of Training Data

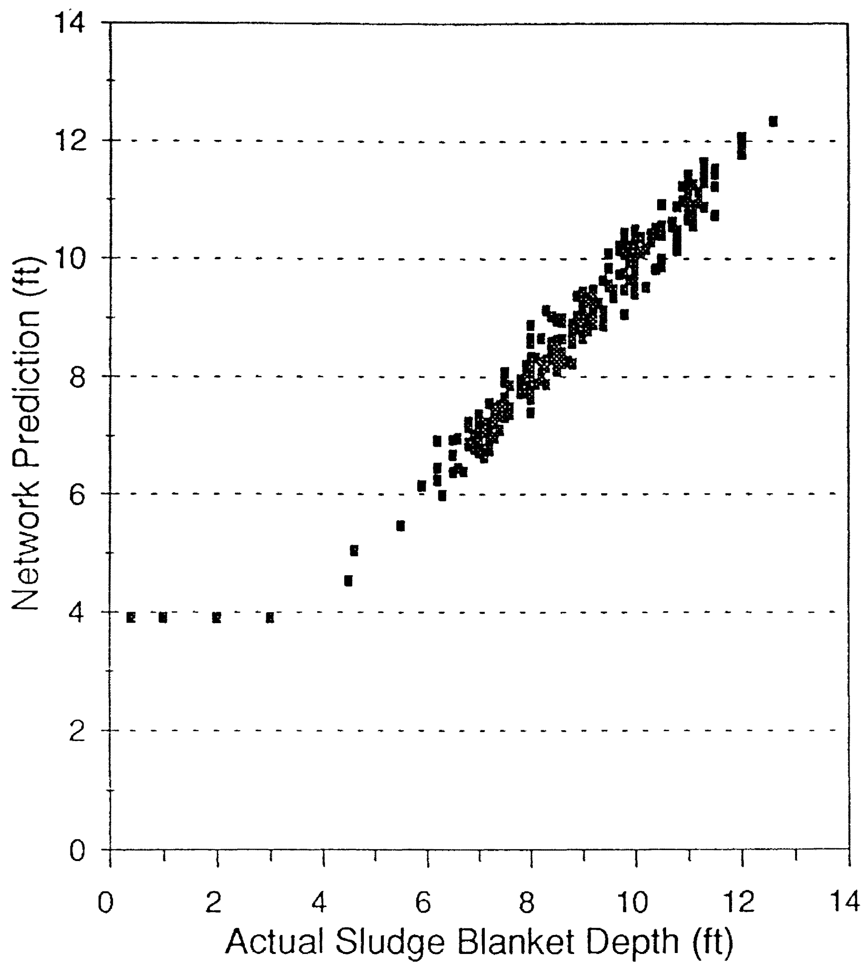


Figure 13. Testing Results of Training Data Set

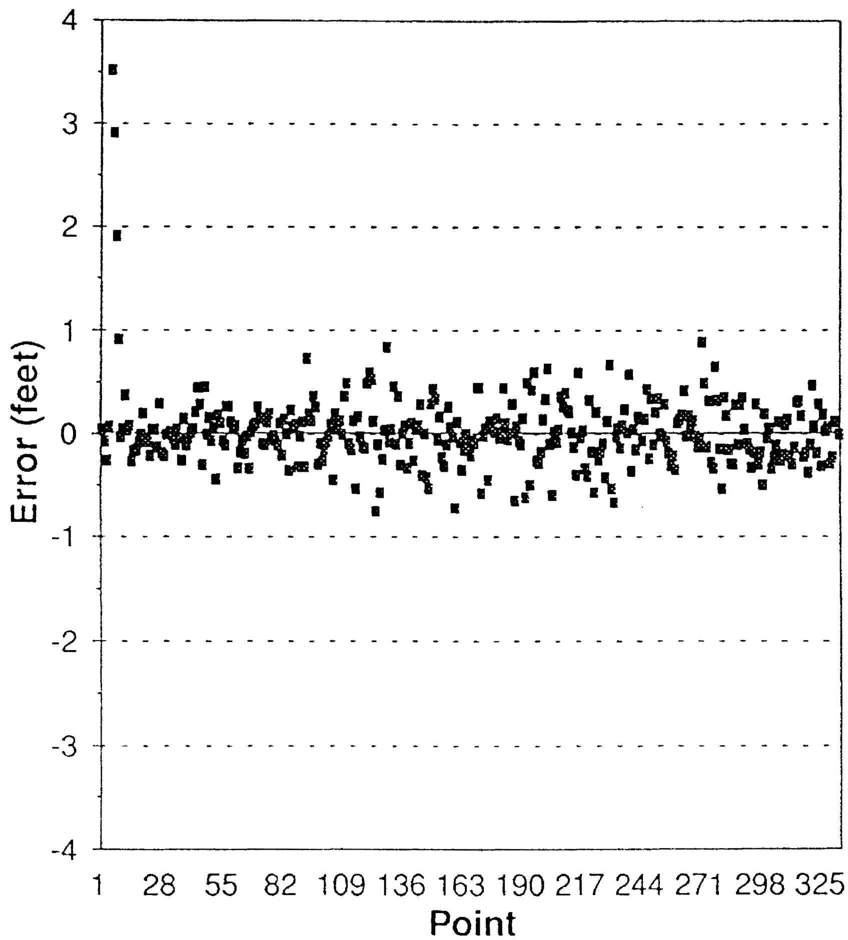


Figure 14. Errors of the Network Output for Training Data

Data normalization is a part of the data scaling process for the sludge blanket depth variable. The data transformation equations used for sludge blanket depth ensure that the sludge blanket depth, which actually varied from 3.9 to 13.6, will be transformed to the interval -1 to +1. The network will generate ±1 for a sludge blanket depth out of this range. In Figure 12 there are four data points below the low boundary set up by the transformation equation. The trained networks gives 3.9 feet as the network prediction when the target output is below 3.9. Figure 12 shows that the lowest value the network can predict is 3.9 feet and introduce some bias in the mean.

The back propagation network has a long training time. The smaller the MSE, the longer the time needed to reduce the MSE. Because of the data normalization procedure with the training data set, continued training did not improve the mean square error. However, from the above plots, it can be seen that the network already learned the function between input and target output for the training data. The errors associated with the normalized output are quite small. Therefore, the training processe was stopped at this point.

After the training process was completed, the testing data set was used to test the trained network. The trend plot of the ranked target output is plotted in Figure 15. This plot is used to assess the performance of the trained network. Figure 16 compares the trend of predictions against the observed sludge blanket depths. Figure 17 is a plot of the regression analysis of the predictions against target output. Table 3 gives the results of the regression analysis. Figure 18 shows the error distribution of the network's prediction for the test data.

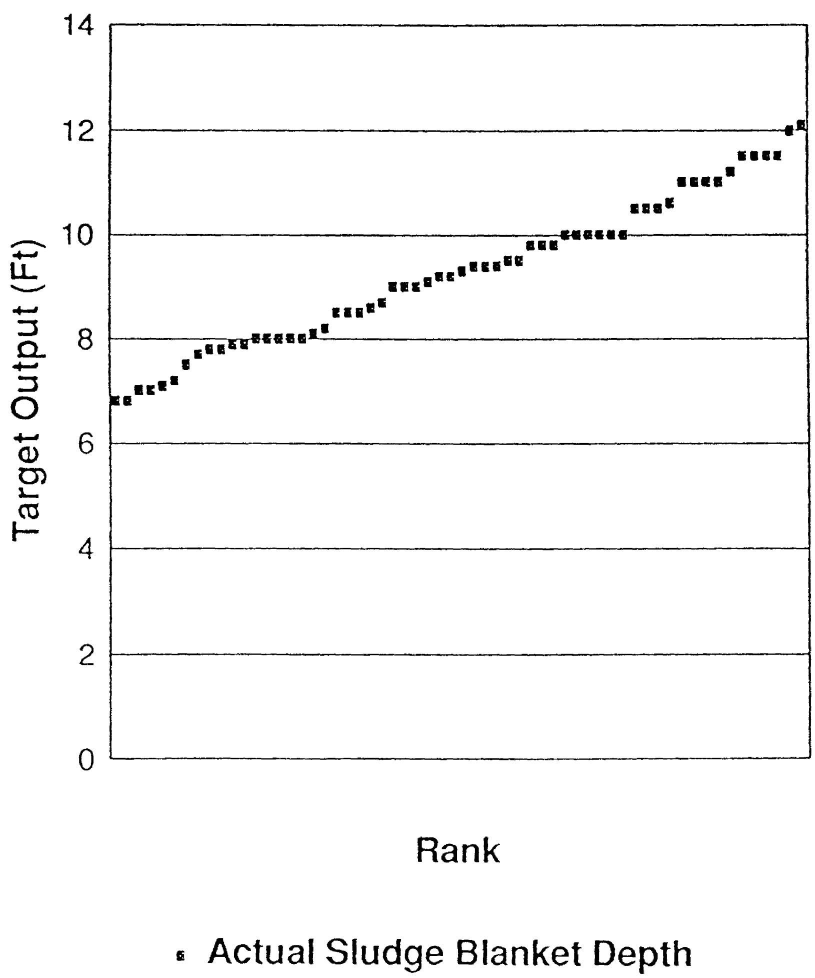


Figure 15. Trend Plot of the Ranked Target Output in Testing

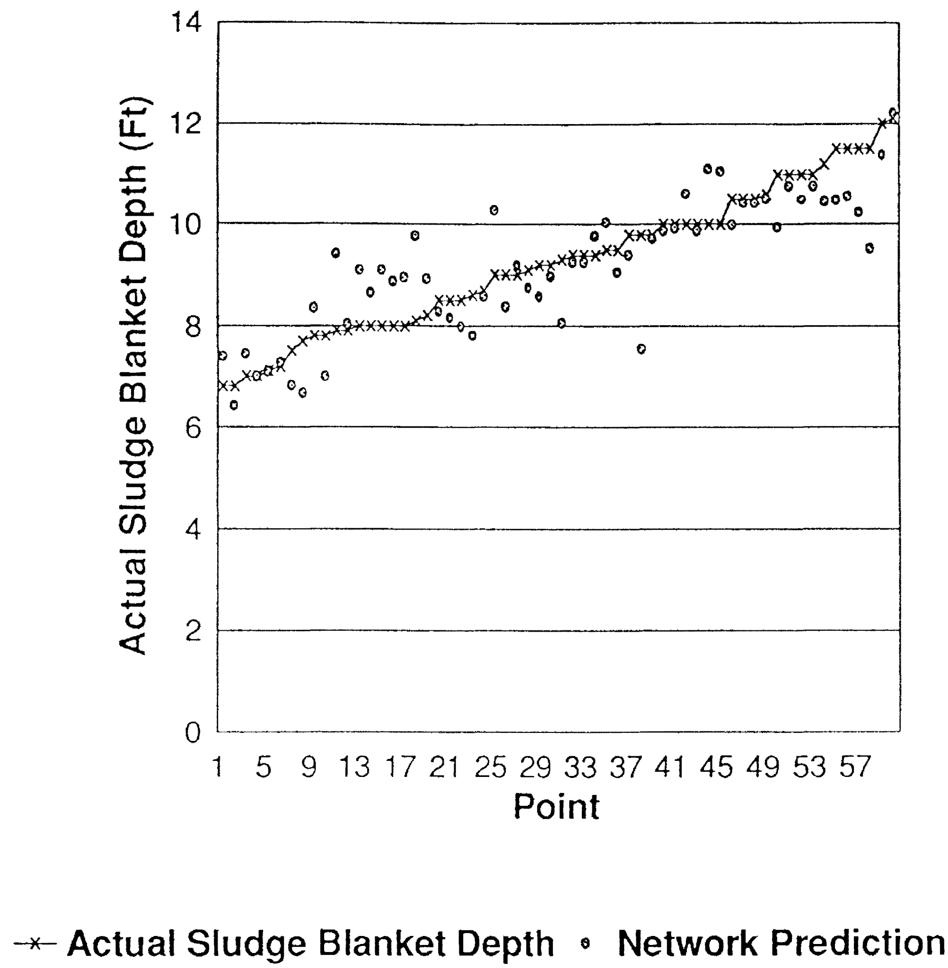


Figure 16. Network Prediction and the Ranked Measured Output

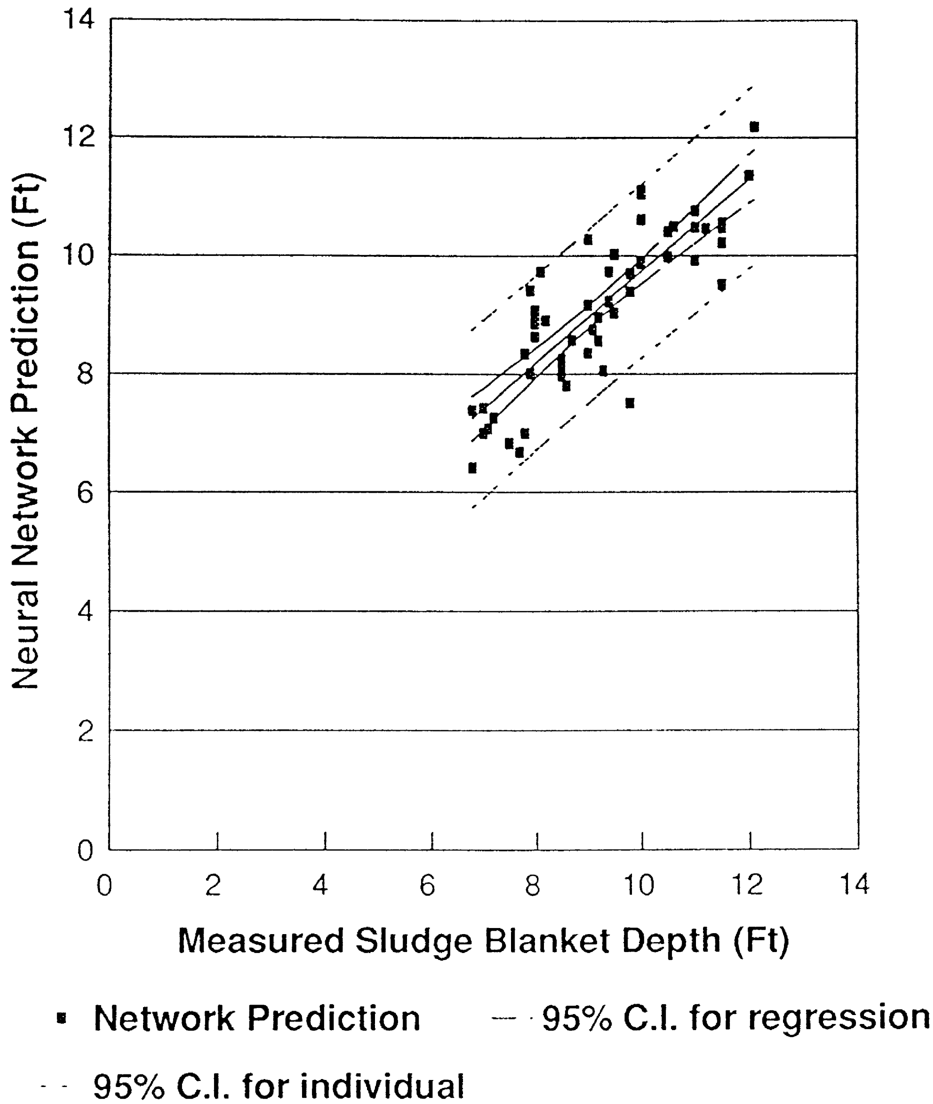


Figure 17. Regression Analysis of the Network Prediction

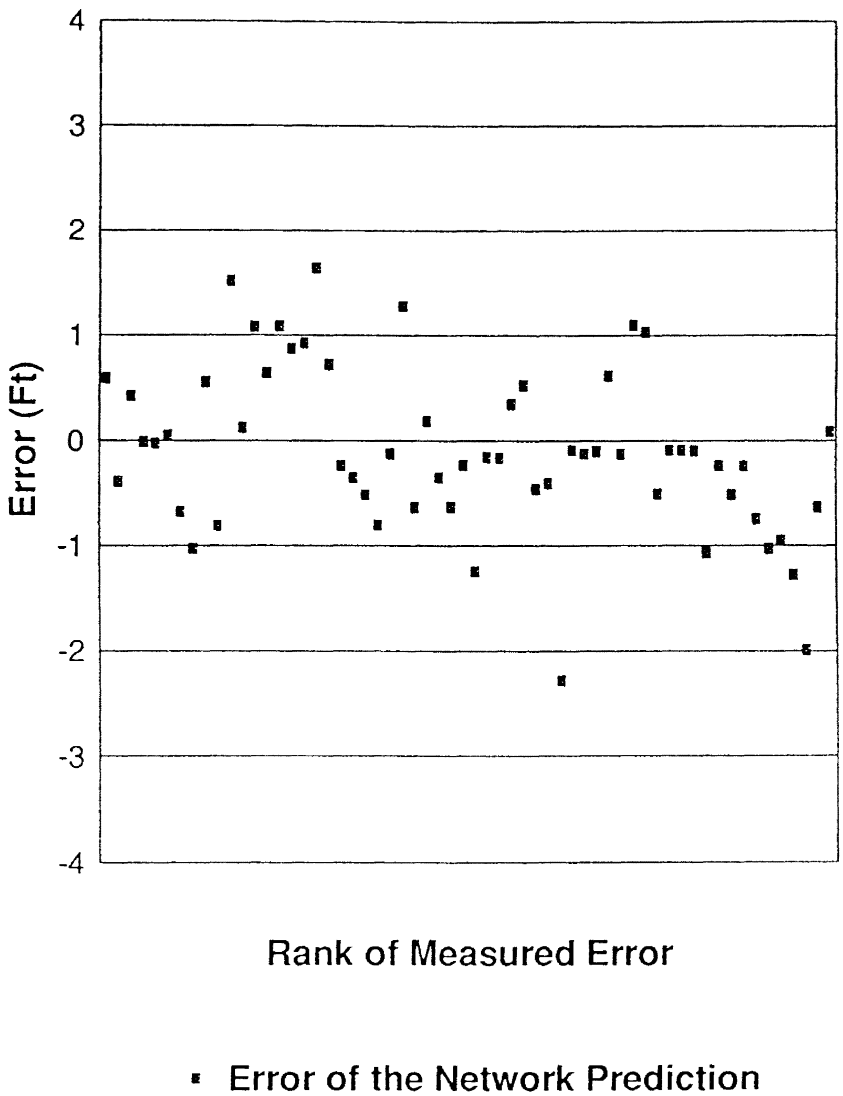


Figure 18. Errors of the Network Output in Testing Process

TABLE 3

REGRESSION OUTPUT FOR TESTING DATA

Constant (α)	1.9565
Std Err of Y Est	0.73 ft
R Squared	0.70
No of Observations	60
Degrees of Freedom	58
X coefficient (β)	0.778
Std Err of Coefficient β	0.067

All of the plots associated with the testing processes are based on the sorted sludge blanket depth. Figure 16 compares the trend plot of network output with the trend of the corresponding sludge blanket depth. From this plot it can be seen that the trends of network prediction agree with the observed sludge blanket depth. This indicates that the network has been trained. The functional relationship embedded in the training data set has been learned to a certain extent by the network.

The network prediction process shows some errors. This can be seen from the trend plot of the testing process. Figure 18 gives the distribution of errors of the predictions for the testing data set. The mean of the prediction errors is -0.1 feet, the standard deviation is 0.79 feet, and the MSE is 0.62 feet.

From the Figure 14 and Figure 18, it can be seen that the means and the sample standard deviations are quite close for the training and testing data. For the testing data the mean of the errors is smaller than and the standard deviation is the same as that for the training data set. This again indicates that the trained network has be trained. The trained network is not over trained or the model is not over fitted. If the model is over fitted, the testing results should be worse than the training results. The errors in the testing results are due to the training because the error distributions are same for both training and testing.

The errors in the training process can be caused from two sources. First, there are a lot of errors in the training data set. Every variable in the training data set has errors. The sludge blanket depth itself has a lot of measurement errors. This can be seen from Figures 12 and 15. These plots of observed sludge blanket depth contain a lot of noise. The measured sludge blanket depth only shows a rough estimate of the settled sludge position in the clarifier. The measurement of the sludge blanket depth can be affected by many factors such as the location of the measurement. The sludge blanket depth in the training data is a teacher to the network. If the teacher contains a lot of errors, the trained network is trained to produce these errors.

Second, because all of the values of sludge blanket depth are not completely transformed between -1 to +1, it is difficult to reduce the networks' training MSE by increasing training time. Therefore, the trained network still contains some errors. The MSE of the trained network is 0.62 feet.

After training and testing the network, another new set of testing data became available (Table 7, Appendix C). This set of new testing data contained 125 records. These new data were independent from previous training and testing data set. There was no data screening on this new set data.

Figures 19 and 20 show the testing results with this new set data. Figure 19 shows the trend plots for both measured and predicted sludge blanket depths. Figure 19 indicates that the trend of prediction is in agreement with the measured trend. When the measured sludge blanket depth had a big jump, the predicted sludge blanket depth also gave a similar jump. Figure 20 is a plot of the network prediction against the measured sludge blanket depth for the new testing data. The regression line for this plot is close to 1.

In these plots, there are several outlying points. There was no data screening associated with this set of new data. Table 4 shows the regression results.

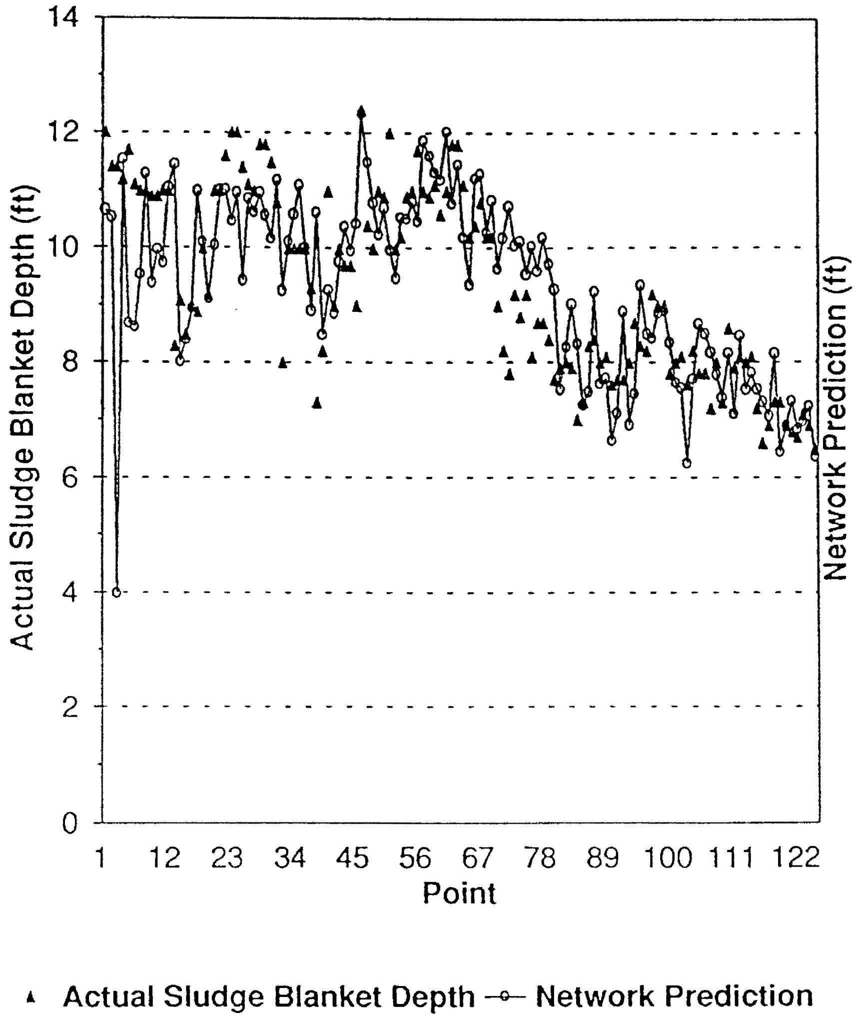


Figure 19. Network Prediction and New Measured Sludge Blanket Depth

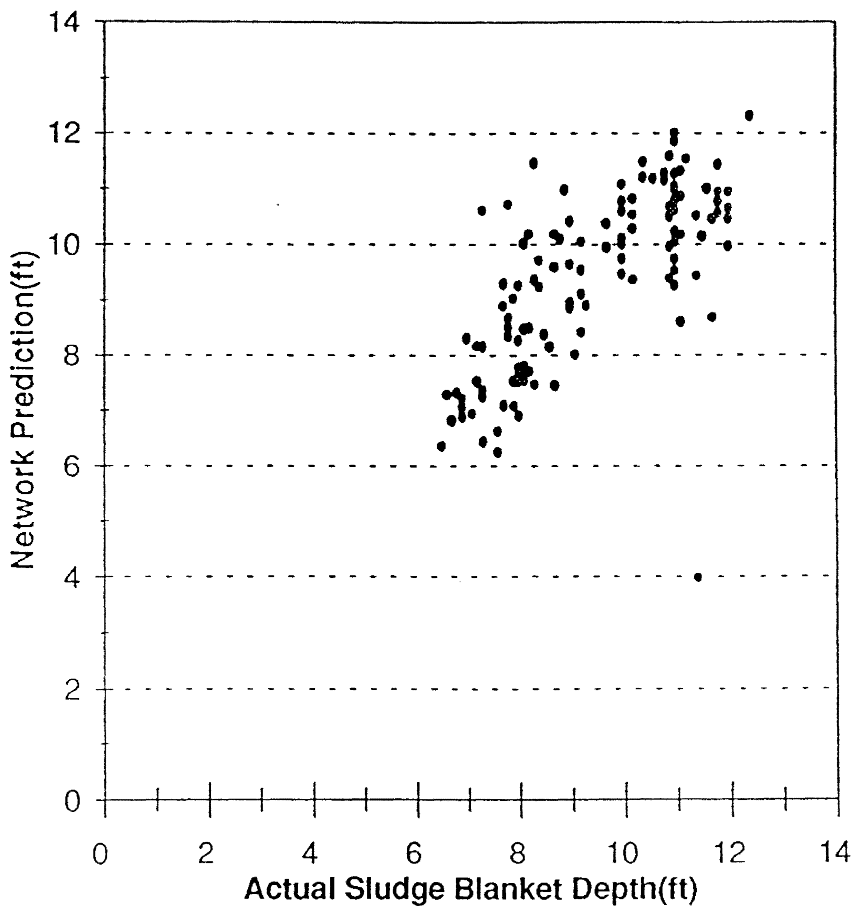


Figure 20. Testing Results of the New Testing Data Set

TABLE 4

REGRESSION ANALYSIS OF NEW TESTING DATA SET

Number-of Points Deleted	R^2	α	β	Std of Error	MSE
0 points	0.30	3.0*	0.676**	5.18	1.53
4 points	0.67	2.19*	0.77**	1.28	0.86
8 points	0.72	1.79*	0.8	0.84	0.67

* significantly differ from zero at 95% level
** significantly differ from one at 95% level

In Table 4, the first colume shows the number of points removed from the testing results when the regression analysis was done. Row 1 shows the regression results with no points taken out from the testing results. Row 2 shows regression results when 4 points were taken out from the 125 data points. All these four points had a prediction error greater than 3 feet. Row 3 shows the results with 8 points, whose absolute prediction error was greater than 2 feet, were removed from the analysis.

Even though the prediction error is statistically large, this model can still be used as an analytical tool. For example, during the daily operation of a waste water treatment plant, what the operators needs to know is what range the sludge blanket is in. If the sludge blanket is too high or too low, the effluent suspended solids will increase. This model can provide an estimate of sludge blanket depth.

The model also shows the possibility of using neural computing techniques in modeling waste water treatment parameters. If the parameters used as the target output in training have fewer errors, the network should model this parameter very well. For example, the 30-minute settling in the secondary clarifier usually has fewer measurement errors. If the neural network is applied to such a parameter, a better model should be obtained.

In conclusion, the results of this study indicate the potential to use the neural network technique to model the sludge blanket depth of the secondary clarifier in the waste water treatment. This can be attributed to the network's abilities of adaptive learning, self-organization, and generalization.

CHAPTER VII

SUMMARY AND CONCLUSIONS

Modeling the sludge blanket depth in the secondary clarifier of an industrial waste water treatment plant was studied using a neural network system. A five layer back propagation neural network system was used. The network had three hidden layers, one input layer, and one output layer. The network was trained using the supervised training algorithm. During the training process, the network was found to be adaptable to the environment. The network's output was trying to mimic the target output. After the network was trained, it was tested using a set of testing data which was not used in the training process.

The modeling was implemented using the ExploreNet software. MBPN (multiple layer back propagation network) was selected as the network paradigm. The application was built into modules. Each module completed a specific task. The network module is the heart of the application. Other modules are used to read data from a disk, write data to a disk file, and display the predictions graphically.

The training of the network took about 300,000 training iterations. The training data set had 332 training records. It was selected from a large collection of measured data.

The mean square error of the network output after the training process was 0.15 feet.

The test result of the trained network showed that the neural network technique is able to model the sludge blanket depth in the secondary clarifier of a waste water treatment plant. The trend in the predictions of sludge blanket depth is the same as the observed sludge blanket depth in the secondary clarifier. The problem with this model is that the predictions of sludge blanket depth have an error with mean equal to -0.1 feet, a standard deviation of error equal to 0.79 feet, and a MSE of 0.62 feet. The model can not predict exact sludge blanket depth partially due to errors in the measured sludge blanket depth. This model can only give an estimate of the depth of the sludge blanket in the clarifier. Since there is no mathematical solution to this problem available, this model still has applicability in an operational setting.

The training by supervised learning requires the target output should be accurate. In order to obtain accurate network prediction, this research suggests that the noise in the signal of sludge blanket depth should be reduced. To improve the quality of the training data set, an expert in the plant operation should be involved in selecting a good training data set.

This research shows that the neural network is a powerful tool in modeling environmental problems. With some modification, the network can be used to model other parameters in the waste water treatment process.

BIBLIOGRAPHY

Bailey, David L. and Donna M. Thompson, (1990a). How to Develop Neural-Network. AI Expert, June 1990.

Bailey, David L. and Donna M. Thompson, (1990b). Developing Neural-Network Applications, AI Expert, September 1990.

Beale, Mark and Howard Demuth, (1992). Preparing Controllers for Nonlinear Systems, AI Expert, July 1992.

Coleman, Kevin G., Timothy J. Graettinger, William F. Lawrence, (1990). Neural Networks for Bankruptcy Prediction: The Power to Solve Financial Problems, AI Review pp48.

Crooks, Ted, (1992). Care and Feeding of Neural Networks, AI Expert, July 1992.

Dbase III Plus, (1986). Programming With dBase III Plus 1. Ashton Tate, 20101 Hamilton Avenue, Torrance, CA.

Dayhoff, Judith E., (1990). Neural Network Architectures. Van Nostrand Reinhold, New York, NY. pp.58-73.

Haan, Charles T., (1977). Statistical Methods in Hydrology. The Iowa State University Press, Ames. pp.180-194.

HNC Incorporated, (1991a). HNC ExploreNet Operating Manual 2.0. HNC, Incorporated, San Diego, CA.

HNC Incorporated, (1991b). HNC ExploreNet Installation and Fast Track Manual 2.0. HNC, Incorporated, San Diego, CA.

HNC Incorporated, (1991c). HNC Neurosoftware Reference Manual. HNC, Incorporated, San Diego, California.

Hobson, Tim, (1986). Foundamentals Process Control, Part IV, Operations Forum. Salina Aera Vocational-Technical School, Salina, KS, pp.21-24.

Junkins, Randy, Kevin Deeny, Thomas Eckhoff, (1988). The Activated Sludge Process: Fundamentals of Operation. Junkins Engineering, Inc., Morgantown, PA.

Maren, Alianna J., Craig T. Harston, Robert M. Pap, (1990). Handbook of Neural Computing Applications. Academic Press.

APPENDICES

APPENDIX A

GENERALIZED DELTA RULE

Generalized Delta rule says that the weight change between the neuron i and j depends on three factors: δ_j, Z_j, and α (Rumelhard & McClelland, 1986). δ_j is the error value of the target unit j. Z_j is the output value for the originating unit i. α is the learning rate whose value is decided by the network user.

Since the error value, denoted by the variable δ_j, is simple to compute for the output slab and somewhat more complicated for the hidden slabs, only the derivation for the output slab is shown here.

Let P be a set of vector-pairs, (x1, t1), (x2, t2), ..., (xp,tp). The error that is minimized by the generalized delta rule is the sum of the squares of the errors produced by the pattern set P for all output neurons.

$$Ep = 1/2 \sum_j (tpj - Opj)^2 \qquad (23)$$

where tpj is the target output of neuron j. Opj is the output value of neuron j.

$$\begin{aligned} \partial Ep/\partial Wij &= 1/2 \bullet 2 \bullet (tpj-Opj)(\partial Opj/\partial Wij) \\ &= -(tpj-Opj)(\partial Opj/\partial Wij) \qquad (24) \end{aligned}$$

For a set of linear units,

$$(\partial Opj/\partial Wij) = \partial(\sum_i Wij \bullet I_{pj})/\partial Wij = - Ipj \quad (25)$$

Where Ipj is the input to neuron j.

Substituting (25) in equation (24),

$$\partial Ep/\partial Wij = - (tpj-Opj) \bullet Ipj = - \delta_j\ Ipj \qquad (26)$$

As far as the magnitude of the weight change is concerned, ΔWij is proportional to the negative gradient.

Thus the weight change ΔWij is calculated by multiplying a step constant α and the negative gradient.

$$\Delta Wij = \alpha \, \delta_j \, Ipj \tag{27}$$

The new weight is,

$$Wij^{new} = Wij^{old} + \Delta Wij \tag{28}$$

APPENDIX B

A DBASE III PROGRAM TO EXPORT DATA

```
*Main.prg --- This is a dBase III Plus program to extract
*training and testing data from data base.
*Following is a list of the field names in the database from
*which the data will be exported.

*EA_LRC: loading rate to the east aeration basin, lbs/day
*EA_MLVSS: MLVSS in east aeration basin, mg/l.
*EA_DOUT: dissolved oxygen uptake rate, mg/l.
*EA_FMC: F/M ratio in terms of COD in east aeration basin.
*EA_pH: pH in east aeration basin, s.u.
*EA_DO: D.O. in east aeration basin, mg/l.
*RSS: recycle suspended solids, mg/l
*RR: recycle rate, MGD.
*EC_NH3: NH3-N in east clarifier, mg/l.
*EC_NO3: NO3-N in east clarifier, mg/l.
*EC_SETT: 30 min. settling in east clarifier, ml/30 min.
*EC_SDP: sludge blanket depth in east clarifier, feet.

Set path to c:\dbase\dbl_sys

CLOSE ALL

select 1
use infobase
index on date_in to c:\dbase\dbl_sys\infobase

go top

SET FILT TO (ea_lrc>0 .and. ea_mlvss>0 .and. ea_dout>0 ;
.and. ea_fmc>0 .and. ea_ph>0 .and. ea_do>0 .and. rss>0 ;
.and. rr>0 .and. ec_nh3>0 .and. ec_no3>0 .and. ec_sett>0;
.and. ec_sdp>0)

COPY TO c:\data.doc FIELDS ea_lrc,ea_mlvss,ea_dout,ea_fmc,
ea_ph,ea_do,rss,rr,ec_nh3,ec_no3,ec_sett,ec_sdp DELIMITED
WITH BLANK

SET FILT TO

CLOSE ALL

SET PATH TO

QUIT
```

APPENDIX C

TRAINING AND TESTING DATA SETS

TABLE 5

TRAINING DATA AND NETWORK PREDICTION

Column Name	Defination	Unit
No.	Record Number	
Input1	Loading Rate to Aeration Basin	lbs/day
Input2	Mixed Liquid Suspended Solids	mg/l
Input3	Dissolved Oxygen Uptake Rate	mg/l
Input4	Food and Microorganisms Ratio in Aeration Basin	
Input5	pH in Aeration Basin	s.u.
Input6	Dissolved Oxygen	mg/l
Input7	Recycle Suspended Solids	mg/l
Input8	Recycle Ratio	MGD
Input9	Ammonia Nitrogen in Clarifier	mg/l
Input10	Nitrate Nitrogen in Clarifier	mg/l
Input11	30 Minute Settling in Clarifier	ml/30 minute
Output	Measured Sludge Blanket Depth	feet
Predict	Neural Network Predicted Sludge Blanket Depth	feet

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
1	6453	3470	21.5	0.16	7.76	4.3	5990	0.577	1.8	1.7	600	11.5	11.6
2	15303	3245	17.4	0.4	7.63	4.2	4780	0.64	11.4	17.3	720	11.2	11.6
3	24691	3570	22.4	0.59	7.4	3.4	4705	0.608	3.2	19.9	770	12.6	12.6
4	16792	3730	26.7	0.39	7.24	1	5030	0.616	2.1	2.6	920	12	11.6
5	36415	3440	32.1	0.91	7.43	1.4	3740	0.589	6.3	4.3	940	0.4	5.1
6	15607	3185	15.8	0.42	7.69	4.6	3455	0.529	32.05	18.8	920	1	3.9
7	16913	3170	16.1	0.46	7.57	5	3270	0.425	19.9	25.5	910	2	3.9
8	27444	3470	18.9	0.68	7.14	3.6	3655	0.451	11.1	48.2	850	3	3.9
9	18896	3325	19.7	0.49	6.99	3.5	4160	0.434	1.5	94	900	5.5	6.1
10	18297	3790	23.7	0.41	6.7	2.6	4725	0.604	3.8	81.5	880	4.5	5.0
11	7971	3650	15	0.19	7.02	4.3	4750	0.619	2.2	36.1	830	7	7.8
12	19649	3440	31.8	0.49	7.08	1.3	4550	0.606	8.1	38	720	7	9.2
13	12874	3635	31.1	0.3	7.14	1.9	4515	0.607	6.9	33.5	720	7	8.3
14	15132	3670	28.1	0.35	6.65	2.3	4560	0.604	6.1	58.5	675	7.2	7.4
15	6170	3435	14.6	0.15	7.5	5	4650	0.59	2.1	38	300	8.5	9.9
16	15347	3175	23.7	0.41	7.08	2.6	3980	0.595	0.9	24.7	290	12	11.8
17	12333	4760	36.1	0.22	7.33	1.2	5905	0.574	0.81	9.46	450	12	11.4
18	4911	3895	16.2	0.11	6.17	4.2	6845	0.594	6.3	28.4	650	10	9.4
19	6755	3720	16.2	0.16	7	4.7	5445	0.61	0.39	24.7	600	11	11.0
20	5648	3825	22.1	0.13	7.12	4.3	5605	0.578	0.63	18.6	600	9	8.8
21	7890	3815	22.5	0.18	6.86	3	5775	0.396	4	18.6	570	8.5	8.3
22	9151	4175	17.6	0.19	7.01	5	7370	0.577	0.49	17.8	480	10	9.9
23	9642	4160	20.1	0.2	6.99	4.7	5750	0.587	0.27	34.4	470	9.5	9.3
24	10653	4490	25.4	0.2	6.55	2.9	7545	0.587	0.39	29.8	470	8	7.9
25	16921	4485	25.4	0.32	6.69	3.2	8425	0.594	2.13	40	440	7.5	7.5
26	10146	4820	20.9	0.18	7.21	3.8	7245	0.582	1.4	46	540	7	7.7
27	7514	4165	11.2	0.15	7.97	6.7	7035	0.576	0.14	25.9	330	10	10.8
28	14953	3880	15	0.33	6.97	5.3	6445	0.579	0.6	19.5	260	9	9.8
29	22768	3875	21.6	0.5	7.08	3.6	5056	0.603	0.58	16.2	310	10	11.6
30	7776	5280	23.2	0.13	7.04	4	7720	0.577	0.78	6.66	340	7.4	7.5
31	6168	4275	17.1	0.12	7.31	5.6	7120	0.585	0.35	4.92	470	7.5	8.1
32	22343	4830	24.9	0.4	6.86	4.9	8230	0.595	11.45	10.8	530	7	7.9
33	19299	4355	17.6	0.38	6.85	4.7	5855	0.582	0.45	21.5	320	8	8.6
34	19591	4225	28.2	0.4	6.4	1.7	6440	0.582	0.78	27.1	370	7.5	8.3
35	16494	4875	28.7	0.29	7.27	2.4	7005	0.586	0.49	21.5	390	7.4	10.0
36	10253	4820	24.3	0.18	7.32	4	8495	0.589	0.65	11.3	600	7	6.9
37	7899	4010	13	0.17	7.08	6.2	6400	0.596	0.49	17.8	480	7	7.1
38	8242	4098	11.7	0.17	7.23	6.9	6185	0.593	0.45	14.2	540	8	8.4
39	6919	3910	16.4	0.15	7.16	5.9	5865	0.59	0.33	9.46	620	7	7.7
40	5694	4150	16.6	0.12	6.45	6.6	6685	0.595	0.63	18.6	530	7.2	7.3
41	6075	4375	13.7	0.12	7.05	7.7	6390	0.593	0.41	25.9	540	7.4	7.8
42	12649	4020	14.4	0.27	6.89	6	6410	0.584	0.43	22.5	610	7	7.2
43	6044	4585	20.2	0.11	7.16	5.3	8615	0.595	0.37	13.2	680	6.5	7.1

TABLE 5 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
44	7976	4200	17.6	0.16	6.69	5.2	6980	0.601	0.54	12.7	745	7	7.5
45	18735	4315	22.6	0.37	6.89	4.4	6880	0.594	0.58	22.5	775	6.3	6.9
46	8376	5005	27.6	0.14	7.25	3.6	8135	0.565	0.73	27.1	870	4.6	5.9
47	13278	4175	20.1	0.27	5.94	5	5910	0.606	0.56	31.2	590	7	7.6
48	17062	4585	29.3	0.32	6.87	3.8	6165	0.601	0.37	29.8	575	7.5	8.2
49	6813	4650	10.1	0.13	7.07	8	8655	0.607	0.2	22.5	650	7	8.3
50	17100	4295	20.7	0.34	5.93	4.8	5760	0.595	4.2	29.8	680	6.2	6.5
51	5900	5250	18.6	0.1	6.94	5.2	7130	0.601	0.69	40	770	7.2	7.2
52	6008	4880	15.9	0.11	7	5.5	7020	0.617	0.24	32.7	860	7	7.3
53	5759	4095	14.8	0.12	6.47	6.5	7220	0.595	0.24	10.8	900	6.8	7.3
54	10825	4340	20.5	0.21	7.12	6.9	8025	0.607	0.24	14.2	850	6.9	7.5
55	11241	5275	17.6	0.18	6.97	6.1	9155	0.613	0.22	25.9	830	6.5	7.0
56	3662	5660	15.6	0.06	6.87	6.5	10045	0.621	0.27	23.6	870	5.9	6.7
57	1259	5695	19.1	0.02	6.7	5	8970	0.6	0.2	16.2	880	6.2	7.0
58	6173	4805	10.8	0.11	7.43	7.9	6345	0.588	0.35	17	920	6.8	7.1
59	4744	4425	10.3	0.09	7.1	6.3	7560	0.588	0.39	18.6	800	6.8	6.8
60	5179	4595	11.4	0.1	7.02	5.1	7310	0.587	0.27	20.5	880	7.4	7.4
61	7214	4185	20.5	0.15	7.1	4.7	8290	0.606	0.39	19.5	770	7.1	7.2
62	4312	4090	22	0.09	7.2	5.9	7460	0.601	0.43	10	700	7.5	7.6
63	4599	4140	29	0.1	6.92	6.1	6530	0.594	0.58	29.8	700	8	7.9
64	6313	3820	16.7	0.14	7.12	6	7080	0.611	0.43	40	665	8.1	8.0
65	4079	4590	13.8	0.08	7.25	9.7	7030	0.601	16.4	44	340	7.2	8.0
66	6548	4375	14.9	0.13	7.07	8.8	15205	0.578	21.8	16.2	315	8	5.6
67	9222	4380	16.4	0.18	6.88	6.9	6775	0.595	15	17	320	7.8	7.8
68	9201	4435	25.3	0.18	6.79	4.2	7610	0.572	2.76	13.2	300	8.4	8.0
69	7226	4560	10.3	0.14	7	8.8	7105	0.594	0.45	19.5	330	8	7.8
70	10077	4815	18.2	0.18	6.81	5.6	8090	0.587	0.37	15.4	300	8	8.2
71	4051	4555	19.6	0.08	6.7	5.5	8150	0.607	0.3	20.5	290	8	7.8
72	3908	4145	17.9	0.08	7.04	6.3	8538	0.601	0.33	27.1	290	8.2	8.3
73	5178	4310	14.2	0.1	6.77	6.2	7650	0.593	0.43	30	250	8	7.9
74	5698	4010	14.3	0.12	6.92	6.2	7438	0.593	0.39	38	280	8	8.5
75	14561	4015	19.7	0.31	7.07	3.7	7385	0.601	0.39	38	280	8	8.7
76	3562	3990	22.2	0.08	6.82	4.6	5436	0.593	0.37	6.25	300	8.4	8.4
77	5149	3980	27.5	0.11	6.39	4.8	6000	0.593	0.96	17.8	255	8.4	8.3
78	12701	3665	22.1	0.3	7.17	4.6	5650	0.611	0.53	40	210	8.4	10.3
79	12790	4000	26.2	0.27	6.96	3.2	6225	0.601	0.45	38	230	8.2	8.4
80	8378	4370	24.5	0.16	7.19	2.9	6735	0.587	0.66	20.5	350	8	8.5
81	10123	4375	23	0.2	7	2.5	7420	0.599	0.6	13.7	355	8.1	8.0
82	6261	4290	22.9	0.12	6.93	2.9	8630	0.601	0.9	10	360	7.9	7.8
83	4332	4570	11.4	0.08	6.53	7	6255	0.601	0.49	36.1	400	8.2	8.5
84	8381	3915	13.5	0.18	6.88	3.7	6685	0.594	0.32	24.7	430	8	7.7
85	9166	4455	17.1	0.18	7.08	4.5	6810	0.592	0.37	31.2	380	8.1	8.3
86	20663	4455	33.5	0.4	7.04	2.1	10575	0.583	0.45	31.2	390	8.6	8.6
87	7486	5025	29.1	0.13	7.34	4.8	10800	0.581	0.62	11.3	500	8	7.5
88	5105	3590	27.2	0.12	6.98	1.9	7538	0.611	1.01	43	720	7.1	7.3
89	8568	3315	31.6	0.22	7.05	1.7	5275	0.642	0.3	23.6	730	8.5	8.4
90	6734	3190	16.8	0.18	6.65	2.1	6180	0.63	0.28	34.4	680	7	7.5
91	7174	3050	25.9	0.2	7.07	1.8	5110	0.601	0.41	40	620	7.3	7.5
92	5613	3475	26.1	0.14	7.26	1.4	7250	0.73	0.51	40	700	6.2	7.3
93	6616	3365	22.3	0.17	7.21	1.4	7325	0.729	0.35	48.6	750	6.5	7.2
94	5405	3330	21.2	0.14	6.93	2.2	5575	0.642	0.35	48.6	790	6.9	7.6
95	7410	4050	11.6	0.16	7.18	5.6	6025	0.617	0.34	73.5	790	6.6	7.5
96	7152	4075	15.8	0.15	7.15	3.3	7700	0.621	0.35	43	840	7	7.7
97	10044	3888	23.8	0.22	7.32	2.2	6625	0.623	0.35	43	830	6.7	7.0
98	13958	3925	43	0.3	7.26	1.3	6325	0.606	3.43	41	770	6.9	7.8
99	8822	4425	32.4	0.17	7.17	2.8	7025	0.62	0.57	21.2	710	8	7.9
100	3696	3730	23.2	0.08	7.23	2.1	7575	0.533	0.51	41.4	550	7.9	8.0
101	10185	3205	25.8	0.27	7.25	1.5	4725	0.61	0.43	38	570	7.8	9.7
102	6639	3240	26.6	0.18	7.38	1.5	5975	0.764	0.52	36.1	640	8	8.8
103	14535	3475	28.7	0.36	7.41	2.1	6025	0.601	0.46	36.1	680	7.9	9.6
104	4365	3170	20.5	0.12	7.26	2.2	4980	0.594	0.38	11.7	790	7.1	7.2
105	3731	3095	21	0.1	7.04	2.2	5210	0.639	0.33	14.2	670	6.5	7.2

TABLE 5 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
106	7712	2975	19.9	0.22	6.98	2.3	4490	0.673	0.31	34.3	770	7.2	7.8
107	7924	2915	18.6	0.23	7.04	2.2	4775	0.61	0.46	34.4	770	7.4	8.0
108	7292	2860	16.8	0.22	7.06	2.8	6950	0.801	1.57	47.3	500	7.1	7.6
109	11432	3325	26.1	0.29	7.26	3.1	9450	0.06	0.91	43	700	7.2	7.1
110	5801	3400	27.2	0.15	7.14	2.6	6550	0.224	0.94	50	500	8.9	8.8
111	7286	3025	26.9	0.21	7.3	4.1	8350	0.175	1.31	50	480	9.1	8.2
112	9152	2850	23.4	0.28	7.46	2.6	8350	0.236	1.72	33.3	470	9.2	8.7
113	10221	2800	23.8	0.31	7.46	3.1	5638	0.337	0.54	38	490	9	9.0
114	6417	2838	26.1	0.19	7.38	2.8	6275	0.224	0.94	31.2	570	9.4	8.7
115	8402	2295	20.2	0.31	7.24	4	4055	0.245	0.4	25.9	600	11	11.2
116	6998	2650	25.4	0.23	7.22	2.9	7288	0.239	0.58	40	510	8.9	8.4
117	26659	3080	31.6	0.74	7.12	3.2	5375	0.273	0.52	41.4	370	9.2	12.6
118	6055	3115	24.1	0.17	6.99	2.6	6875	0.273	0.63	56.5	480	10	9.4
119	3728	2835	19.3	0.11	7.33	3.9	6425	0.227	0.36	32.25	290	9.8	10.9
120	1648	2455	19	0.06	7.27	5	5160	0.281	0.31	25.9	220	9.5	11.1
121	6175	2085	21.3	0.25	7.29	3.6	4515	0.325	0.34	23.6	180	9.7	11.6
122	6030	2140	22.6	0.24	7.27	3.3	5580	0.365	0.4	22.5	160	8.8	11.1
123	3796	2215	19.3	0.15	7.25	3.5	5250	0.267	0.52	19.5	170	11.5	11.7
124	5458	2475	15.5	0.19	7.32	5.1	5800	0.399	0.4	54	310	9.6	10.2
125	4417	2545	13.7	0.15	7.27	4.8	4913	0.393	0.38	47.2	370	10	10.4
126	5271	2470	13.7	0.18	7.23	4.7	4025	0.381	0.36	62.4	400	9.9	10.0
127	7469	2395	14.8	0.27	7.29	4.5	5785	0.354	0.33	47.2	500	9.2	9.4
128	4915	2445	12.5	0.17	7.37	4.7	4188	0.37	0.34	47.3	630	8.3	9.6
129	6046	2200	13.8	0.24	7.3	4.8	4950	0.391	0.34	17	620	8.4	10.1
130	4732	2330	9.5	0.17	7.26	6.3	4930	0.383	0.52	51.8	470	8.2	9.6
131	4111	2510	15.5	0.14	7.19	4	6455	0.09	0.36	47.2	320	8.5	9.0
132	7094	2460	19.3	0.25	7.11	2.7	4495	0.203	0.39	37.6	270	9.1	9.6
133	3426	2200	16.7	0.13	7.08	5.3	2500	0.303	0.49	59.6	150	9.5	10.8
134	4936	2438	22.5	0.17	7.16	3	4038	0.274	0.53	68.8	140	9.8	9.5
135	4705	2300	26.4	0.18	7.15	10.4	5000	0.279	0.88	81	160	10.1	10.7
136	5648	2250	23.5	0.21	7.17	2	4300	0.3	4.3	77	130	10	10.0
137	6858	2330	23	0.25	7.33	4.1	4950	0.35	0.55	23.6	260	11	11.7
138	10448	2530	26.6	0.35	7.28	2.4	6350	0.112	0.49	31.2	270	11	10.6
139	13181	2585	26.9	0.44	7.32	2.5	6480	0.16	0.74	21.5	260	10.9	11.0
140	6808	2110	15.5	0.28	7.41	5.3	5255	0.35	0.64	14.2	270	11.5	11.6
141	8490	2275	23.5	0.32	7.36	2.7	4755	0.36	0.35	12.7	330	10.5	11.3
142	10268	2715	28.6	0.32	7.38	1.8	5580	0.107	0.57	13.2	350	9.5	10.3
143	10539	2550	31.8	0.35	7.42	1.5	5245	0.163	0.6	21.5	290	10.1	9.9
144	8468	2360	27.9	0.31	7.48	1.9	5455	0.228	0.69	37.2	240	11.1	10.8
145	7658	2470	28.3	0.27	7.45	1.8	6730	0.274	0.69	40	200	10.3	10.2
146	7905	2670	28.8	0.25	7.36	2.4	5100	0.024	0.64	62.4	290	11.3	10.4
147	8737	2600	24.9	0.29	7.41	4.4	4305	0.121	0.74	59.6	360	11.1	10.3
148	17165	2830	29.3	0.52	7.45	4.2	6025	0.305	0.53	72.2	370	9.2	9.2
149	10051	3010	26.5	0.29	7.34	3.6	5945	0.115	0.74	62.4	300	10.5	10.3
150	6935	1249	28.9	0.48	7.41	1.7	5500	0.294	0.96	68.8	300	10.9	10.6
151	9746	2970	20.2	0.28	7.46	2	5030	0.23	0.53	36.1	410	9.2	9.7
152	12306	2345	19.4	0.45	7.43	2.5	4715	0.284	0.59	38	330	9.9	11.0
153	8521	2785	20.4	0.26	7.43	2.3	5270	0.42	0.71	38	350	9.6	9.8
154	6316	3055	21.8	0.18	7.4	2.5	5855	0.262	0	34.4	445	9.2	9.1
155	5914	2620	14.8	0.19	7.43	3.1	4970	0.373	0.82	29.8	630	8.8	9.0
156	8201	2305	20.5	0.3	7.43	3.1	6935	0.349	0.45	31.2	560	9.1	9.2
157	6759	2150	18.8	0.27	7.43	2.6	7235	0.391	0.57	36.1	620	9.5	9.6
158	8592	2560	21	0.29	7.43	2.4	5640	0.42	0.55	32.7	650	9.3	9.2
159	5239	3300	13.7	0.14	7.49	4.9	5775	0.339	0.64	40	600	9.8	9.4
160	9286	2955	15.9	0.27	7.53	3.2	7025	0.325	0.37	41.4	450	9	9.3
161	16305	3175	28	0.44	7.46	2	6530	0.239	0.47	36.1	590	10	9.7
162	8486	2085	12.5	0.35	7.57	6.9	9145	0.374	4	27.1	830	8.5	8.4
163	11524	3020	25.5	0.33	7.31	4.8	6465	0.12	3.8	22.5	870	7.9	7.2
164	7938	2970	26.2	0.23	7.25	3.5	6860	0.287	3.7	27.1	830	8.3	8.0
165	6834	3240	22	0.18	7.14	3.4	6960	0.216	2.08	40	800	9	8.5
166	8143	3010	22	0.23	7.07	3.2	7390	0.154	0.57	40	720	9.2	8.6
167	6880	2950	19.8	0.2	7.3	3.6	5990	0.269	0.23	40	850	8.5	8.4

TABLE 5 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
168	4971	2625	21.1	0.16	7.3	3.2	6460	0.267	0.25	32.7	750	8.9	8.6
169	4716	2765	13.4	0.15	7.46	6.7	6416	0.231	0.29	41.4	700	9	9.6
170	9287	2640	19.9	0.3	7.12	5.6	5730	0.314	0.34	41.4	480	9.7	10.1
171	4738	2685	18	0.15	7.35	5.9	5380	0.352	0.87	40	410	10.4	10.5
172	10500	2520	12.8	0.36	7.37	8.2	6655	0.397	0.61	50	430	10.2	11.2
173	8918	3150	26.4	0.24	7.99	5.2	6000	0.204	15	48	460	9.7	9.9
174	9274	2985	18.6	0.27	7.38	6.9	7660	0.336	6.35	73.5	360	10	9.5
175	14167	2890	21.3	0.42	7.33	5	8050	0.335	0.4	41.4	390	10	9.7
176	8878	2880	14.9	0.26	7.54	6.9	7425	0.273	0.48	34.4	570	10	10.8
177	6441	3200	14.6	0.17	7.54	7	6170	0.354	0.48	34.4	470	10.4	11.3
178	9837	3040	19	0.28	7.13	6.5	6645	0.396	0.36	28.4	390	10.3	10.5
179	7811	2610	30.4	0.26	7.28	1.6	4305	0.29	0.38	45	960	7	7.6
180	6110	2665	24.4	0.2	7.3	2.2	4845	0.442	1.75	43	940	7.2	7.7
181	10130	2805	26	0.31	7.41	2.7	6115	0.526	0.48	40	950	6.8	7.9
182	5996	3170	30.9	0.16	7.3	3.8	5675	0.306	0.4	30.8	920	7.4	7.9
183	3270	3000	21.2	0.09	7.42	5.5	3350	0.575	0.5	41.4	870	7.8	8.1
184	5001	2980	30.4	0.14	7.4	4.3	2980	0.588	0.59	38	580	10	11.5
185	10033	2530	24.4	0.34	7.4	3.9	4460	0.231	0.48	43	410	9.9	10.0
186	6439	2610	29.8	0.21	7.36	2.9	4915	0.355	0.52	41.4	500	10.8	10.0
187	3969	2595	22.8	0.13	7.47	4	4080	0.518	0.48	41.4	470	11	11.3
188	6175	2835	23.7	0.19	7.48	2.7	4455	0.347	0.38	59.6	450	10.7	10.1
189	6399	2590	27	0.21	7.51	3.6	5245	0.501	0.42	28.4	350	11.2	11.7
190	13093	2805	20.8	0.4	7.42	2.7	5090	0.312	0.4	45	390	10	10.3
191	5467	3075	23.7	0.15	7.4	3.3	4890	0.474	0.54	30.8	460	10.5	9.9
192	7296	2975	28.5	0.21	7.48	2.6	4985	0.503	0.39	36.1	440	10	10.2
193	6021	3080	24	0.17	7.49	3.2	5290	0.361	0.59	41.4	400	10.5	9.9
194	3955	3080	27.8	0.11	7.49	2.6	5990	0.52	0.44	38	430	10	10.2
195	8779	2895	22.8	0.26	7.54	2.7	6155	0.392	0.42	54.22	480	9.5	9.8
196	9314	2748	15.7	0.29	7.48	4.2	5280	0.526	0.5	43	530	9.4	10.3
197	8239	3025	19.9	0.23	7.5	3.1	5950	0.404	0.33	40	580	9	8.7
198	5711	3125	19	0.16	7.81	3.4	5635	0.492	0.44	32.7	570	9	9.2
199	2593	3025	20.5	0.07	7.48	2.6	5800	0.487	0.48	40	570	8.9	9.0
200	7829	3240	18.3	0.21	7.52	3.2	5880	0.48	0.48	43	740	8.6	8.9
201	8147	3330	21.8	0.21	7.55	2.1	6110	0.471	0.63	38	790	8.4	9.2
202	8516	3710	22.3	0.2	7.49	2	5880	0.5	0.87	37.2	790	8.4	8.8
203	5218	3755	22.3	0.12	7.49	1.2	6305	0.621	0.96	36.1	840	8	7.8
204	5378	3430	11.6	0.13	7.59	6	4770	0.621	0.26	56.5	890	9.2	9.2
205	6090	3585	11.1	0.15	7.59	6	5015	0.706	0.26	52	910	8	8.5
206	8212	3585	11.7	0.2	7.5	4.8	5350	0.706	0.26	40	950	8.8	8.7
207	6439	4010	18.1	0.14	7.48	5.1	6130	0.503	0.31	49.4	910	6.8	7.5
208	13944	3705	18.9	0.32	7.56	2.8	6200	0.5	0.43	40	910	7.6	8.4
209	9968	3320	19.7	0.26	7.58	2.8	7330	0.514	0.53	40	860	8.6	9.0
210	6483	3495	14.6	0.16	7.62	3.5	5630	0.456	0.47	29.8	910	8	8.6
211	7856	3620	18.6	0.19	7.6	3.3	5405	0.595	0.33	32.7	870	8.4	8.8
212	3766	3560	24.2	0.09	7.57	3.6	5470	0.593	0.45	12.2	940	7.5	7.3
213	10235	3510	16	0.25	7.58	3.9	5510	0.524	0.26	41.4	930	8.5	8.7
214	5752	3665	17.4	0.13	7.62	2.8	6400	0.411	0.45	36.1	910	8.5	7.5
215	12800	3500	20.3	0.31	7.67	4.2	4705	0.414	0.49	29.8	830	7.5	8.3
216	11173	3600	15.7	0.27	7.71	3.5	5020	0.429	0.37	32.7	920	8.3	8.6
217	8667	3496	25.3	0.21	8.34	4.4	4936	0.444	0.35	38	900	7.2	7.6
218	13258	3620	22	0.31	7.6	3.5	5605	0.439	0.53	28.4	900	9	8.5
219	7190	3980	19.1	0.15	7.6	4.5	4772	0.359	0.69	40	900	8.7	8.7
220	12675	3650	21.2	0.3	7.48	3.9	4870	0.473	0.51	38	880	8	8.6
221	10014	3800	16.2	0.23	7.49	3.9	5300	0.456	0.2	16.2	920	8.6	8.4
222	8440	4020	17.1	0.18	7.42	4.9	5745	0.52	0.24	36.1	870	8.8	8.3
223	12462	4095	23.5	0.26	7.46	3.3	6155	0.342	0.43	28.2	900	7.9	8.1
224	14283	3910	29.2	0.31	7.44	3.4	5500	0.319	0.35	22.52	900	8	8.4
225	4762	4020	20.3	0.1	7.41	4.6	5580	0.451	0.8	38	850	8.2	8.1
226	9598	3415	24.3	0.24	7.44	13.4	5200	0.462	0.72	43	920	8	8.0
227	11296	3705	21.1	0.26	7.53	2.9	6280	0.398	0.53	41.1	890	8.3	8.1
228	10134	3905	21	0.22	7.52	3.7	5800	0.455	0.64	38	910	8.5	8.6
229	7226	4125	25.5	0.15	7.54	3.1	5775	0.29	0.85	32.7	880	8	7.7

TABLE 5 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
230	7062	3620	21	0.17	7.53	2	5130	0.608	0.61	35.6	840	10	9.6
231	11593	3455	24.3	0.29	7.62	2.6	5988	0.457	0.4	45	800	10.2	9.5
232	8771	3660	21.4	0.21	7.62	1.8	7000	0.516	0.34	40	750	9.8	9.6
233	12325	3710	23.5	0.28	7.63	1.4	5955	0.321	0.81	12.2	850	9.2	9.8
234	13019	3720	28.3	0.3	7.68	1.5	5385	0.473	0.5	11.7	750	9	9.9
235	9075	3795	26.9	0.2	7.87	2.6	5715	0.597	0.36	8.21	430	10.5	11.0
236	5135	3670	22.8	0.12	7.35	2.2	5245	0.583	0.29	48.7	540	9.4	9.5
237	7431	3555	25.6	0.18	7.38	1.7	6125	0.552	0.34	67.8	550	8.5	8.7
238	7156	3700	22.9	0.17	7.46	2.3	6670	0.571	0.52	68.8	600	8	8.7
239	6907	3860	18.7	0.15	7.42	3.1	6570	0.488	0.28	58.2	750	8	7.9
240	4373	4296	19.7	0.09	7.48	3.8	6200	0.573	0.43	26.2	820	8.2	7.6
241	7217	4320	16.1	0.14	7.53	3.9	6575	0.506	0.74	12.5	900	8	7.2
242	10493	4475	23.2	0.2	7.53	2.2	6710	0.492	0.41	9.8	900	7.8	8.1
243	4834	4355	25.5	0.1	7.43	2	3915	0.516	0.33	39.2	890	7.1	7.7
244	5949	3915	17.5	0.13	7.52	4.2	7295	0.443	0.31	76	860	7.3	7.6
245	6073	3915	17.7	0.13	7.59	4.7	4895	0.574	0.21	17.5	210	11.3	11.8
246	13684	3500	20.9	0.33	7.66	3.4	6540	0.49	0.13	27.9	210	11	11.7
247	11483	4255	27.8	0.23	7.54	2.7	7913	0.342	0.41	15.5	220	11.2	11.1
248	14720	4065	24.7	0.31	7.46	2.4	7988	0.511	0.21	21.3	210	11.3	11.5
249	3366	3970	15.3	0.07	7.69	5	7010	0.56	0.19	18.1	250	10	10.5
250	3546	4075	15.2	0.07	7.59	5.5	7415	0.571	0.18	21.2	270	9.2	10.2
251	2178	4535	18.6	0.04	7.67	4.7	8425	0.634	0.15	25.2	260	8	9.0
252	7872	3988	11.9	0.17	7.47	6.1	8050	0.496	0.19	28	290	11.3	11.5
253	8837	4275	15.5	0.18	7.92	4.6	8825	0.24	0.16	31.7	290	9	9.3
254	16402	4550	22.9	0.31	7.46	2.7	8575	0.265	0.22	34.8	250	10	10.0
255	9472	3800	19.4	0.21	8.01	3.2	7150	0.603	0.15	50.4	240	11.2	11.7
256	8140	4300	18.6	0.16	7.71	3.8	7125	0.649	0.21	40	230	11.2	11.7
257	7995	4838	16.5	0.14	7.57	4.8	6000	0.389	0.12	47.2	220	10.8	11.4
258	12997	4475	20.8	0.25	7.64	3.3	6225	0.39	0.16	34.6	200	12	11.9
259	5365	4170	14.6	0.11	7.64	5.7	6250	0.697	0.29	13.4	250	11.1	11.3
260	10581	4615	17	0.2	7.84	4.9	6580	0.703	0.41	13.3	270	10.8	11.0
261	9732	4720	31.6	0.18	7.34	1.7	7220	0.563	0.13	17.2	280	10.4	10.2
262	8924	4455	12.2	0.17	7.36	5.8	6885	0.468	0.5	29.3	290	11.1	11.4
263	4547	3950	18.3	0.1	7.56	4.6	6150	0.703	0.25	31.4	460	7.5	8.7
264	19571	4875	19.2	0.34	7.6	4.5	6750	0.707	0.25	33	460	7.3	9.7
265	10596	4625	19.2	0.2	7.6	4.5	6750	0.707	0.39	45.4	460	7.3	8.1
266	12325	4750	25.1	0.22	7.63	2.7	6200	0.57	0.26	36.2	590	8.9	9.5
267	20023	4625	26.8	0.37	7.66	1.8	6225	0.613	0.25	25	440	8	11.7
268	12948	4775	31.9	0.23	7.49	3.5	7625	0.194	0.31	49.8	230	10.5	10.4
269	4881	4975	27.1	0.08	7.49	1.9	6350	0.39	0.22	47.8	220	10	9.9
270	6162	4145	29.9	0.13	7.52	1.8	7550	0.648	0.23	32.7	220	11	10.8
271	6520	4375	24.7	0.13	7.62	2.5	6600	0.666	0.22	47.2	390	8	8.8
272	5721	4350	20.2	0.11	7.65	3.1	6125	0.558	0.59	54.5	380	8.5	9.4
273	11992	4175	29.7	0.25	7.64	1.8	6100	0.489	0.64	61.5	340	8.3	8.5
274	6752	3950	15.7	0.15	7.61	5.3	7200	0.636	0.59	34.8	320	9	10.6
275	6992	4125	17.3	0.15	7.52	4.6	6800	0.675	0.17	35.7	300	9	10.2
276	8443	3975	19.3	0.18	7.58	3.6	7125	0.602	0.14	32.4	300	10	10.7
277	5354	4250	19.6	0.11	7.55	3.7	5990	0.677	0.2	44	300	9.8	11.3
278	7490	3975	21.5	0.16	7.56	3.1	6600	0.744	0.19	31.5	300	9.9	10.9
279	3612	4550	17.8	0.07	7.63	3.9	8350	0.501	0.19	52.4	310	10	10.7
280	7685	4475	26.2	0.15	7.6	2.2	7350	0.759	0.2	49.2	350	10	9.3
281	11439	4500	33	0.22	7.61	1.6	6700	0.561	0.24	47.6	320	9	9.4
282	9667	4275	29.1	0.19	7.66	4.1	6700	0.535	0.19	48.7	290	10	11.0
283	17281	4700	22.4	0.31	7.65	2.6	4840	0.464	0.21	50.8	290	9	11.7
284	8632	4840	21.1	0.15	7.66	3.2	6715	0.663	0.19	42.2	310	9.1	9.4
285	7645	4505	24	0.15	7.64	2	7610	0.576	0.18	41.4	310	10	10.0
286	7100	4455	26.3	0.14	7.6	2.1	8310	0.457	0.16	48.6	300	10	10.4
287	4515	4235	21.1	0.09	7.64	4.2	8185	0.662	0.15	53.8	280	10	10.5
288	8379	4305	21.3	0.17	7.51	3.3	9325	0.606	0.13	49.6	260	9.8	9.8
289	7710	4880	28.7	0.14	7.59	2.2	8675	0.694	0.11	46	250	10	10.0
290	7721	4490	27.3	0.15	7.57	2.4	6720	0.68	0.15	25.5	290	10.1	10.5
291	5620	4640	26.7	0.1	7.56	3	7620	0.638	0.18	29.9	300	10.5	10.7

TABLE 5 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
292	4322	4635	27.2	0.08	7.55	3.2	8470	0.727	0.16	30.4	270	10.7	10.7
293	8217	4635	20.6	0.15	7.5	3.1	7845	0.711	0.15	28.7	270	10.8	10.9
294	4417	4705	25.1	0.08	7.53	2.1	7740	0.642	0.11	47.2	280	11	10.7
295	6354	4385	29.6	0.12	7.57	2.1	8345	0.753	0.11	53.6	250	11	11.2
296	9861	4690	29.4	0.18	7.57	3.1	8060	0.642	0.1	47	270	11	10.6
297	13177	4740	37.5	0.24	7.56	2.6	8435	0.639	0.13	52.6	270	11.1	10.9
298	6845	4605	23.2	0.13	7.49	2.4	6260	0.569	0.21	55.8	300	10.8	10.3
299	6280	4945	28.5	0.11	7.59	3.4	7280	0.547	0.27	57	300	11	11.1
300	9378	4710	35.2	0.17	7.56	4	6270	0.769	0.2	57.6	290	11.5	11.4
301	5892	4805	21.2	0.11	7.57	2.4	7380	0.78	0.15	60	500	9.9	10.0
302	6729	4370	30.2	0.13	7.54	1.5	6495	0.625	0.17	46.2	610	9.4	9.2
303	5610	4655	32	0.1	7.55	3.3	6450	0.633	0.2	49.6	660	7	7.5
304	5683	4945	28.4	0.1	7.49	4.1	9685	0.507	0.2	49.6	700	7.6	7.9
305	8563	4975	15.6	0.15	7.5	3	8595	0.74	0.15	38.8	820	7	7.5
306	6142	5795	21.4	0.09	7.51	1.9	8850	0.548	0.11	27.6	850	7	7.0
307	3577	5590	22.5	0.05	7.55	3	8985	0.579	0.15	37.62	850	7.2	7.4
308	5834	5645	26.6	0.09	7.48	1.9	9270	0.514	0.13	36.8	810	7	7.4
309	16006	5515	30.3	0.25	7.48	3.3	8355	0.629	0.19	30.1	770	7	8.0
310	4627	4780	16.3	0.08	7.58	3.3	7850	0.681	0.14	49	860	7.6	7.8
311	9406	5080	21.2	0.16	7.56	3.6	8585	0.697	0.12	50	800	7.4	7.7
312	7208	5110	28.4	0.12	7.52	1.8	8090	0.597	0.11	43	780	6.6	7.0
313	6702	5192	19.8	0.11	7.54	3.3	8360	0.648	0.16	56.2	680	7.9	8.5
314	4776	5285	23.3	0.08	7.57	2.6	7430	0.653	0.19	52.8	640	8	8.5
315	6440	4645	30.5	0.12	7.4	2.4	7260	0.718	0.26	57.4	390	8	8.8
316	3705	5005	26.7	0.06	7.45	3.8	7345	0.689	0.16	61.2	370	8.8	9.0
317	2667	4060	27.8	0.06	7.52	3.1	8580	0.824	0.13	56.6	320	8	7.7
318	11147	5385	29.3	0.18	7.36	2.5	9755	0.675	0.89	58.4	300	8.6	8.7
319	5600	5715	32.2	0.08	7.48	2.3	10590	0.568	0.23	56.6	320	7.9	7.8
320	9293	5230	29.8	0.15	7.44	2.5	8535	0.643	0.2	54.4	320	8.2	9.1
321	8554	4930	15.1	0.15	7.43	5.3	6920	0.613	0.24	50	330	8	8.9
322	5738	4855	18.1	0.1	7.49	4.3	8700	0.622	0.18	44.4	360	8.8	8.4
323	4927	4780	18.4	0.09	7.53	4.9	7585	0.537	0.16	32.4	410	9	9.7
324	3849	4472	20.8	0.07	7.49	3.5	7185	0.808	0.16	28.2	410	9.1	9.0
325	8638	4300	19.6	0.17	7.57	3.2	6235	0.628	0.12	33.4	630	9.1	9.5
326	5781	4185	18.8	0.12	7.56	3.1	7380	0.792	0.1	21	710	8.8	7.9
327	2967	4660	21.4	0.05	7.53	2.5	6990	0.801	0.1	21.1	800	8.8	8.1
328	6921	4680	12.8	0.13	7.52	4.3	6795	0.685	0.19	27	890	8	7.4
329	6667	4730	19.7	0.12	7.54	2.6	7695	0.62	0.14	24.4	820	8.2	7.6
330	5936	4770	18.3	0.11	7.54	3	6465	0.67	0.15	24	860	8	7.5
331	3375	4570	17.1	0.06	7.52	2.9	4570	0.648	0.17	27.1	880	7.9	7.7
332	7223	3895	9.4	0.16	7.44	5.7	4485	0.793	0.14	43.6	960	7.1	7.9

TABLE 6

TESTING DATA AND NETWORK PREDICTION

Column Name	Defination	Unit
No.	Record Number	
Input1	Loading Rate to Aeration Basin	lbs/day
Input2	Mixed Liquid Suspended Solids	mg/l
Input3	Dissolved Oxygen Uptake Rate	mg/l
Input4	Food and Microorganisms Ratio in Aeration Basin	
Input5	pH in Aeration Basin	s.u.
Input6	Dissolved Oxygen	mg/l
Input7	Recycle Suspended Solids	mg/l
Input8	Recycle Ratio	MGD
Input9	Ammonia Nitrogen in Clarifier	mg/l
Input10	Nitrate Nitrogen in Clarifier	mg/l
Input11	30 Minute Settling in Clarifier	ml/30 minute
Output	Measured Sludge Blanket Depth	feet
Predict	Neural Network Predicted Sludge Blanket Depth	feet

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
1	31730	3520	31.7	0.77	7.11	0.7	4630	0.59	0.73	31.2	310	12.1	12.2
2	8197	3815	19.2	0.18	7.05	4	6355	0.6	2.7	21.5	500	10.5	10.0
3	13589	4325	28.5	0.27	7.36	3.2	8915	0.59	20.9	40	430	8	8.9
4	12483	4790	29.6	0.22	6.84	3.5	8008	0.589	6.5	34	400	7	7.0
5	5278	4885	19.1	0.09	7	4.4	7240	0.607	0.45	24.7	730	7	7.4
6	7750	5035	11.7	0.13	6.91	5.3	7660	0.612	0.35	27.1	860	6.8	7.4
7	15064	4150	16.9	0.31	7.05	4.8	7300	0.6	0.45	31.2	330	8	9.1
8	14867	4615	27.6	0.28	7.09	2.7	9165	0.595	0.53	31.2	320	8	8.9
9	6774	3535	21.7	0.16	7.12	2.3	5675	0.801	0.41	40	470	9.8	7.5
10	8196	3790	23.5	0.19	7.08	1.7	7962	0.602	0.49	41.4	470	9.3	8.1
11	6117	3205	20.2	0.16	7.21	2	7688	0.53	0.46	4.14	630	7.7	6.7
12	7982	3162	23.8	0.22	7.14	2.2	4912	0.271	2.42	43	750	7.9	8.0
13	4967	1885	19.1	0.23	6.97	4.9	3254	0.232	0.61	20.5	590	9.8	9.4
14	6321	2145	18.1	0.25	7.11	5.6	2605	0.357	4.67	27.1	560	11	10.5
15	4632	2290	18	0.17	7.24	4.2	5900	0.42	0.33	34.4	200	10	11.1
16	4172	2445	14.2	0.15	7.16	4.7	6695	0.337	0.48	56.5	270	9.4	9.2
17	6161	2530	16.6	0.21	7.18	4.7	5938	0.402	0.36	48.6	590	7.9	9.4
18	3821	2660	13.9	0.12	7.23	5	7020	0.252	0.52	43.5	590	8	8.6
19	7464	2750	22.5	0.23	7.24	1.8	4375	0.29	1.19	43.5	170	11	9.9
20	6324	2000	14.2	0.27	7.31	4.4	4075	0.395	1.42	50	165	12	11.4
21	12222	2540	32	0.41	7.41	4	5425	0.49	0.47	15.4	240	10	11.0
22	6957	2736	22.4	0.22	7.48	3.1	5240	0.159	0.71	43	215	11.5	10.5
23	5948	2635	25.1	0.19	7.43	2.2	5050	0.357	0.6	59.6	300	9	10.3
24	8964	2430	25.2	0.32	7.44	2.2	7005	0.341	0.57	25.9	390	9.8	9.7
25	10292	2890	24.4	0.31	7.43	2.5	5565	0.191	0.53	40	430	9.5	10.0
26	15274	2900	21.8	0.45	7.47	3.1	5745	0.42	0.37	31.2	500	9.2	9.0
27	5337	3405	26.6	0.13	7.12	2.9	6630	0.389	1.65	32.7	810	8.5	8.1
28	5211	3500	22.1	0.13	7.03	4	8285	0.287	0.49	34.4	870	8.5	8.3
29	15921	2655	15.9	0.51	7.38	5.6	6830	0.233	0.22	34.4	860	9	8.4
30	10615	3045	17.7	0.3	7.35	5.5	7605	0.238	0.34	41.4	800	9.2	8.6
31	12483	2760	21.2	0.39	7.42	2.2	5235	0.239	1.05	28.4	870	8.7	8.6
32	6830	3205	28.2	0.18	7.44	2	4890	0.47	0.45	34.4	670	9.4	9.7
33	6902	2905	22.2	0.2	7.57	2.6	4585	0.445	0.5	45	530	9.4	9.2
34	5807	2910	20.7	0.17	7.46	2.9	5310	0.409	0.44	41.4	530	9.5	9.0
35	3970	2725	18.5	0.12	7.43	3.7	5680	0.491	0.59	29.8	600	9	9.2
36	3510	2735	20.5	0.11	7.5	3.2	5150	0.717	0.61	28.4	550	9.1	8.7
37	5413	3655	18.2	0.13	7.42	4.9	6000	0.634	0.31	43	910	8.6	7.8
38	5562	3810	18.6	0.13	7.61	3.2	6275	0.33	0.37	34.4	910	7.8	7.0
39	7138	3915	16.2	0.16	7.63	3.6	5725	0.287	0.37	34.4	890	8.5	8.0
40	11641	3555	23.1	0.28	7.59	3.1	5550	0.4	0.6	32.7	850	8.2	8.9
41	6833	4340	24.3	0.13	7.53	1.8	6140	0.438	0.35	20.4	900	7.8	8.4
42	6526	4440	29.4	0.13	7.41	1.6	8100	0.376	0.34	27.5	950	7.2	7.3
43	9477	5412	29.4	0.15	7.46	2.7	9225	0.274	0.6	50.6	890	6.8	6.4
44	12948	4775	31.9	0.23	7.49	3.5	7625	0.194	0.31	49.8	230	10.5	10.4

TABLE 6 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
45	4881	4975	27.1	0.08	7.49	1.9	6350	0.39	0.22	47.8	220	10	9.9
46	6162	4145	29.9	0.13	7.52	1.8	7550	0.648	0.23	32.7	220	11	10.8
47	12948	4775	31.9	0.23	7.49	3.5	7625	0.194	0.31	49.8	230	10.5	10.4
48	4881	4975	27.1	0.08	7.49	1.9	6350	0.39	0.22	47.8	220	10	9.9
49	6162	4145	29.9	0.13	7.52	1.8	7550	0.648	0.23	32.7	220	11	10.8
50	8865	4250	22.4	0.18	7.6	2.9	7750	0.524	0.13	43.4	300	11.5	10.2
51	7044	4145	28.4	0.15	7.59	2.7	7875	0.638	0.14	40	320	11.5	10.6
52	15882	4150	45.1	0.33	7.58	1.6	6525	0.591	0.16	31.3	320	11.5	9.5
53	6009	4255	23.5	0.12	7.59	3.8	7000	0.692	0.29	34.2	330	10	10.6
54	5365	4825	29.6	0.1	7.6	2.4	6775	0.622	0.19	44	330	10	9.9
55	7898	5125	32.1	0.13	7.55	4.2	7765	0.641	0.21	58.4	320	11.2	10.5
56	6183	4050	16.3	0.13	7.49	3.5	7470	0.752	0.23	63	400	10.6	10.5
57	7893	5300	16.4	0.13	7.64	3.1	8760	0.482	0.14	43.2	880	7.1	7.1
58	6718	5090	20.7	0.11	7.55	1.9	8125	0.474	0.07	49.2	870	7.5	6.8
59	5521	5340	33.4	0.09	7.44	2.3	8980	0.567	0.19	44.4	320	8.1	9.7
60	5767	5140	27.5	0.1	7.44	2.4	7345	0.766	0.24	39.2	310	8	9.1

TABLE 7

NEW DATA AND NETWORK PREDICTION

Column Name	Defination	Unit
No.	Record Number	
Input1	Loading Rate to Aeration Basin	lbs/day
Input2	Mixed Liquid Suspended Solids	mg/l
Input3	Dissolved Oxygen Uptake Rate	mg/l
Input4	Food and Microorganisms Ratio in Aeration Basin	
Input5	pH in Aeration Basin	s.u.
Input6	Dissolved Oxygen	mg/l
Input7	Recycle Suspended Solids	mg/l
Input8	Recycle Ratio	MGD
Input9	Ammonia Nitrogen in Clarifier	mg/l
Input10	Nitrate Nitrogen in Clarifier	mg/l
Input11	30 Minute Settling in Clarifier	ml/30 minute
Output	Measured Sludge Blanket Depth	feet
Predict	Neural Network Predicted Sludge Blanket Depth	feet

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
1	6188	4185	20.5	0.13	7.48	2.8	6850	0.673	0.21	35	210	12	10.7
2	6076	4640	27.9	0.11	7.46	2.4	6160	0.688	0.16	33	220	11.4	10.5
3	7737	4475	46.1	0.15	8.38	2.7	6740	0.675	0.22	33.54	250	11.4	4.0
4	10939	4655	25.9	0.2	7.66	2.9	5630	0.58	4	23.9	250	11.2	11.5
5	19603	4505	42.3	0.37	7.61	1.7	6595	0.687	0.26	23.9	250	11.7	8.7
6	10030	4730	41.8	0.18	7.91	1.3	6172	0.752	0.23	20.3	300	11.1	8.6
7	9014	5045	32	0.15	7.58	1.3	7525	0.685	0.96	22.7	300	11	9.5
8	7944	5310	25.3	0.13	7.5	3.9	8145	0.662	0.34	22.9	280	11	11.3
9	6826	5105	20.9	0.11	7.43	1.7	9025	0.652	0.38	29.6	300	10.9	9.4
10	5165	5100	21.8	0.09	7.45	2.5	7415	0.7	0.17	33.9	300	10.9	10.0
11	8423	4850	21.9	0.15	7.51	2.1	6925	0.698	0.16	31.9	300	11	9.7
12	4771	5140	24.7	0.08	7.53	3.9	7610	0.66	0.25	30.4	330	11	11.1
13	8149	4940	28.2	0.14	7.46	3.5	4940	0.736	0.18	54	325	8.3	11.5
14	5424	5020	30.4	0.09	7.4	4.6	8365	0.622	0.15	47.6	350	9.1	8.0
15	8404	5145	35.1	0.14	7.4	3.8	7680	0.702	0.16	55.6	300	8.5	8.4
16	9144	4680	25.3	0.17	7.25	3.7	7050	0.669	0.25	59	280	9	8.9
17	14034	5535	30.1	0.22	7.69	4.4	6760	0.697	0.17	60	260	8.9	11.0
18	5977	5245	36.9	0.1	7.44	1.9	9405	0.551	0.17	49	260	10	10.1
19	4359	4880	25	0.08	7.37	5.3	9405	0.707	0.19	35	260	9.2	9.1
20	7016	5030	27.7	0.12	7.47	1.6	7755	0.612	0.15	58	300	11	10.0
21	5626	4450	18.7	0.11	7.43	4.3	6533	0.709	0.26	44	230	11	11.0
22	12658	4675	31.8	0.23	7.47	1.9	6185	0.707	0.16	48.6	230	11.6	11.0
23	10959	4850	43.2	0.19	7.52	2.6	9500	0.626	0.19	40.4	260	12	10.5
24	9223	5650	39.9	0.14	7.53	2.9	5850	0.595	0.49	37	290	12	11.0
25	9873	5512	34.7	0.15	7.53	3.4	8225	0.758	0.36	24.6	280	11.4	9.4
26	11935	5288	35.3	0.19	7.45	1.5	7555	0.704	0.31	32.6	200	11.1	10.9
27	8654	5045	41.7	0.15	7.49	3	8080	0.645	3.65	50	250	11	10.6
28	4183	5015	31.8	0.07	7.51	2.1	6950	0.729	0.33	27.2	270	11.8	11.0
29	9258	4830	27	0.16	7.51	1.9	7730	0.773	0.24	34.8	260	11.8	10.6
30	14904	4815	27.5	0.27	7.57	2.7	8925	0.687	0.24	34.8	370	11.5	10.2
31	9357	5380	30.1	0.15	7.52	2	9020	0.744	0.24	18.6	470	10.8	11.2
32	9600	5150	26.9	0.16	8.41	4.5	9265	0.648	0.19	18.2	530	8	9.3
33	5497	5580	19.5	0.08	7.66	2	8085	0.569	0.19	17.5	620	10	10.1
34	6620	5290	20	0.11	7.56	2	8450	0.69	0.18	23.3	630	10	10.6
35	3515	5510	23.5	0.05	7.51	1.5	8810	0.68	0.19	24.6	710	10	11.1
36	5126	5120	21.6	0.09	7.52	2.9	6850	0.702	0.16	28.9	710	10	10.0
37	12299	5140	17.2	0.2	7.49	3.4	8365	0.801	0.14	29.8	630	9.3	8.9
38	9907	6950	29.4	0.12	7.49	3.5	9790	0.7	0.37	31.1	750	7.3	10.6
39	8176	5820	21.7	0.12	7.59	3.1	10165	0.761	0.17	28.9	850	8.2	8.5
40	4120	4875	18	0.07	7.55	2.7	8480	0.532	0.2	25.8	830	11	9.3
41	5196	5080	19.3	0.09	7.49	2.5	9370	0.727	0.1	29.8	830	9	8.9
42	6556	5625	24.7	0.1	7.53	2	9575	0.726	0.13	30.4	820	10	9.8

TABLE 7 (Continued)

No.		Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
43	6514	5465	24.3	0.1	7.56	2.5	9290	0.687	0.17	27.1	790	9.7	10.4
44	5070	6315	23.3	0.07	7.58	2.2	13138	0.673	0.14	29.4	770	9.7	10.0
45	5524	6670	24.7	0.07	7.58	3.5	11745	0.754	0.15	22.6	790	9	10.4
46	5828	2325	13	0.21	7.79	5.4	5400	0.796	0.1	11	180	12.4	12.3
47	8863	3465	17.1	0.22	7.67	4.1	9775	0.763	0.15	9.9	270	10.4	11.5
48	8687	4880	27.2	0.15	7.6	3.5	9485	0.749	0.13	10.1	380	10	10.8
49	5746	5255	26.1	0.09	7.49	3.2	9475	0.712	0.35	17.9	410	11	10.2
50	9462	5195	37.1	0.16	7.51	2.1	10750	0.721	0.3	15.9	380	10.9	10.7
51	11890	5670	36.1	0.18	7.51	3.5	9275	0.701	0.35	17.6	430	12	10.0
52	12638	6000	30.9	0.18	7.45	4	8700	0.688	0.37	20.1	370	10	9.5
53	7360	5245	17.5	0.12	7.38	4.4	8325	0.693	0.35	25.1	370	10.2	10.5
54	5036	5385	22.6	0.08	7.37	3.2	8590	0.699	0.21	24.5	305	10.9	10.5
55	6136	5080	30.4	0.1	7.49	1.8	7640	0.736	0.21	23.5	300	11	10.8
56	7757	5005	38.2	0.13	7.41	3.4	9300	0.746	0.26	20.6	280	11.7	10.5
57	5540	5220	35.5	0.09	7.48	4.9	6390	0.716	0.27	30.1	280	11	11.9
58	4975	5034	35.5	0.08	7.39	4.2	8650	0.725	0.25	28.4	270	10.9	11.6
59	12716	4625	36	0.24	7.41	2.6	9275	0.689	0.29	31.2	240	11.1	11.3
60	5302	5085	38.2	0.09	7.48	2.5	8335	0.693	0.27	29.3	240	10.6	11.2
61	4825	4610	26	0.09	7.49	4	7925	0.731	0.27	29.2	240	11	12.0
62	7561	4595	27.3	0.14	7.43	2.3	7930	0.806	0.21	23.9	250	11.8	10.8
63	6436	4585	27.3	0.12	7.44	2.3	7930	0.806	18.2	20.6	250	11.8	11.5
64	8271	4575	34.5	0.15	7.42	3.3	8440	0.327	0.26	27.4	275	11.1	10.2
65	5531	4680	15.2	0.1	7.38	5.1	9200	0.73	0.18	30.6	350	10.2	9.4
66	6212	4415	19.1	0.12	7.43	3.7	7750	0.769	0.14	24.9	330	10.4	11.2
67	9214	3930	15.9	0.2	7.48	4.6	7705	0.784	0.22	19.6	340	10.8	11.3
68	10302	4740	21.4	0.19	7.45	2.3	8400	0.256	0.18	17.1	400	10.2	10.3
69	4386	5260	26.5	0.07	7.41	3	8545	0.71	0.22	19.4	420	10.2	10.8
70	6754	4795	11.5	0.12	7.37	6	8165	0.727	0.18	21.9	520	9	9.6
71	9791	4800	20	0.17	7.47	2.7	9345	0.752	0.14	22	430	8.2	10.2
72	6833	4695	25.3	0.12	7.45	2.1	9010	0.698	0.1	19.5	450	7.8	10.7
73	4327	4705	22.1	0.08	7.45	3.3	8315	0.751	0.11	16.5	570	9.2	10.0
74	10923	4415	22	0.21	7.38	1.4	8855	0.699	0.1	20.2	450	8.8	10.1
75	7300	4920	16.9	0.13	7.48	4.9	9156	0.723	0.08	19.7	530	9.2	9.5
76	6521	5190	23.8	0.11	7.41	2.8	9850	0.658	0.08	19.7	480	8.1	10.0
77	6454	5875	29.1	0.09	7.44	2.3	8705	0.748	0.13	16.1	450	8.7	9.6
78	8427	4690	31.5	0.15	7.5	1.7	8470	0.724	0.18	18	440	8.7	10.2
79	11199	4980	33	0.19	7.42	3.2	7205	0.759	0.16	17.1	570	8.4	9.7
80	5418	5430	28.4	0.09	7.33	3	8235	0.777	0.23	27.7	465	7.7	9.3
81	5959	5455	18.9	0.09	7.42	3.1	8575	0.71	0.17	31.6	620	7.9	7.5
82	10311	5145	27.2	0.17	7.89	2.7	8645	0.721	0.17	31	530	8	8.3
83	6102	5255	24.6	0.1	7.57	2.3	7500	0.757	0.2	25.3	560	7.9	9.0
84	6760	5260	12.7	0.11	7.49	7.2	8970	0.674	0.24	25.2	610	7	8.3
85	6190	5310	22.5	0.1	7.25	3.5	8760	0.634	6.71	27.6	670	7.3	7.3
86	4763	4760	17.2	0.09	7.4	5.3	9035	0.745	4.61	30.2	720	8.3	7.5
87	7738	4950	25.9	0.13	7.5	2.9	8470	0.734	0.16	25.8	700	8.4	9.2
88	6024	5355	23.8	0.1	7.49	2.9	8945	0.688	8.9	19.4	630	8	7.6
89	3301	4970	21.1	0.06	7.52	4.1	7920	0.726	0.2	17.4	660	8.1	7.7
90	3672	4760	15.5	0.07	7.42	4.9	7530	0.72	0.18	20.2	700	7.6	6.6
91	6004	4965	18.5	0.1	7.43	3.9	8565	0.724	0.15	17.8	570	7.7	7.1
92	7915	4855	29.8	0.14	7.38	3.7	7640	0.673	0.11	19.4	520	7.7	8.9
93	2740	5080	23.3	0.05	7.34	4.4	7275	0.655	0.33	33.2	480	8	6.9
94	3604	4860	20.3	0.06	7.39	3.5	8155	0.719	0.14	27.3	490	8.7	7.5
95	6518	4660	28.7	0.12	7.3	2.5	7815	0.728	0.12	35.1	390	8.3	9.4
96	4610	5415	19.8	0.07	7.42	2.6	7760	0.676	1.38	36	400	8.2	8.5
97	3879	5215	19.3	0.06	7.44	4.8	7435	0.735	0.17	26.4	390	9.2	8.4
98	4258	4635	17.9	0.08	7.46	4.4	7465	0.769	0.17	22.3	415	9	8.9
99	7102	4740	21.3	0.13	7.5	2.6	9715	0.726	0.15	26	470	9	8.9
100	6811	5300	11.7	0.11	7.45	6.1	8100	0.688	0.16	28	530	7.8	8.4
101	6096	5100	16.2	0.1	7.33	4.7	4505	0.635	0.12	30	510	8	7.6
102	4267	4505	16.8	0.08	7.38	3.6	7095	0.808	0.15	27.4	470	8.1	7.5
103	5272	4700	11.6	0.1	7.33	7.7	7925	0.751	0.43	26.6	720	7.6	8.5
104	10977	4695	25	0.2	7.31	2.3	7425	0.698	0.56	30	560	8.2	8.2

TABLE 7 (Continued)

No.	Input1	Input2	Input3	Input4	Input5	Input6	Input7	Input8	Input9	Input10	Input11	Output	Predict
105	4038	5130	26.8	0.07	7.39	2.6	8515	0.822	0.38	71.2	580	7.8	8.2
106	3155	4660	7.2	0.06	7.46	8	7655	0.696	0.36	84	820	7.8	8.8
107	5414	4750	10.1	0.1	7.42	7	8260	0.651	0.3	83.5	820	7.2	8.8
108	5230	4605	11.7	0.1	7.4	6.3	7730	0.68	0.31	72	760	8	8.8
109	9313	5080	21	0.16	7.33	2.7	8185	0.726	0.32	65	730	7.3	7.8
110	5294	5075	19.5	0.09	7.46	4.5	8235	0.709	0.55	71.75	790	8.6	6.3
111	6025	4700	17.1	0.11	7.28	5.5	8430	0.695	0.29	77	900	7.9	7.7
112	4499	4955	18.2	0.08	7.36	5.9	8030	0.697	8.74	99.5	830	8.1	8.7
113	4139	5055	13.3	0.07	7.39	6.8	7780	0.733	0.31	75.25	860	8	7.4
114	4065	5025	16.3	0.07	7.33	5.5	8650	0.719	0.28	58.5	850	8.1	8.2
115	4112	4920	15.4	0.07	7.52	6.1	7620	0.726	0.44	43	880	7.2	7.1
116	10457	4490	14.8	0.2	7.26	5.9	10250	0.715	0.36	31.4	910	6.6	8.5
117	7148	4970	24.2	0.12	7.27	3.8	8445	0.644	4.12	59.2	890	6.9	7.5
118	4161	5785	18.9	0.06	7.35	4.5	9825	0.677	0.34	68	865	7.3	7.8
119	6712	6150	21.9	0.09	7.37	2.4	8590	0.68	0.29	50.25	900	7.3	7.5
120	4289	5665	9.8	0.06	7.52	7.2	7910	0.684	0.21	21.1	945	6.9	7.3
121	5747	5595	14.5	0.09	7.46	4.2	9240	0.697	0.27	26.7	930	6.8	7.1
122	4131	5910	30.3	0.06	7.45	1.7	8635	0.627	0.38	49	910	6.7	8.2
123	6066	5955	20.6	0.09	7.4	3.5	9930	0.684	3.75	45	910	7.1	6.4
124	4397	5450	14.3	0.07	7.52	5.4	8795	0.645	0.22	43	920	6.9	6.9
125	12380	5060	21.8	0.21	7.31	3.3	9525	0.696	0.29	19.2	950	6.5	7.3

VITA

Qin Zhao

Candidate for the Degree of

Master of Science

Thesis: APPLICATION OF A NEURAL NETWORK IN MODELING THE SLUDGE BLANKET DEPTH OF A SECONDARY CLARIFIER

Major Field: Agricultural Engineering

Biographical:

Personal Data: Born in Taiyuan, Shanxi, P.R. China, June 15, 1962, the son of Chunming Zhao and Cuizhen Zhang.

Education: Graduated from Taiyuan 20th High School, Taiyuan, Shanxi, P.R. China, in July 1979; received Bachelor of Science Degree in Agricultural Engineering from Shanxi Agricultural University at Taigu, Shanxi, P.R. China in July, 1983; completed requirements for the Master of Science degree at Oklahoma State University in May, 1993.

Professional Experience: Programming and trouble shooting for the industrial group in Stover and Associates, Inc., Stillwater, Oklahoma, June 1992 to May 1993; Research Assistant, Department of Agricultural Engineering, Oklahoma State University, January 1990 to June 1992; Visiting Scholar, Department of Agricultural Engineering, Oklahoma State University, January 1989 to January 1990; Instructor, Department of Agricultural Engineering, Shanxi Agricultural University, Shanxi, P.R. China, September 1983 to December 1989.