To the Graduate Council:

I am submitting herewith a thesis written by G. Gray Paschal entitled "Use of Principal Component Analysis for Data Reduction for Training Neural Networks." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Robert E. Uhrig, Major Professor

We have read this thesis and recommend its acceptance.

Accepted for the Council

Associate Vice Chancellor and
Dean of The Graduate School

Use of Principal Component Analysis for Data Reduction for Training Neural Networks

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

G. Gray Paschal

December 1996

DEDICATION

This thesis is dedicated to my parents,

Mr. and Mrs. Glen G. Paschal,

who have provided me

encouragement, support, and love

and to my wife,

Rebbecca Lynn Paschal,

who waits patiently on me.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my esteemed major professor, Dr. Robert E. Uhrig, for his patience and guidance. I am also indebted to my other committee members and faculty members that have knowingly and unknowingly left an indelible impression on my life. Great appreciation to Dr. Kerlin who created the financial opportunity for my return to school.

ABSTRACT

The use of neural networks has been effective in identifying relationships between plant data and a desired output. As the amount of data presented to a neural network increases the difficulty in training the networks increases. Principal component analysis is a technique available to reduce the size of data and thus may help the use of neural networks.

Principal component analysis is a method of systematically reducing a large number of dependent and independent variables to a smaller, conceptually more coherent set of variables. This smaller set of uncorrelated variables still retains most of the information of the original data set. This lends itself well as a preprocessing tool for neural networks where large data sets are involved. The smaller data set can be used to train neural networks quicker due to its reduced size. Uncorrelated variables provide the networks with simpler relationships and reduced noise which further improves performance.

An application of this technique is presented using data from TVA's Kingston Fossil Plant. A comparison is made between the use of raw data and data preprocessed with use of principal component analysis to train a neural network. The principal component analysis proves to be a very useful tool in reducing the number of inputs to a neural network and improving its performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Much has been gained through the use of neural networks to analyze and grasp the relationship of plant data to a desired output. Neural networks can be effective at identifying these relationships. Large amounts of data can be difficult to manipulate and can confuse or slow the training of neural networks. Principal components analysis may help neural networks efficiently deal with large data inputs.

Principal component analysis is a method of breaking data into its root building blocks. It finds relationships between variables and may find a simpler set of data that represents the same relationship. A decorrelator can perform a principal component analysis by finding the correlation between data and breaking it down into fewer components which represent the vast majority of the information in the original data set. This simpler, smaller data set can then be used to train neural networks faster and better; faster because of the reduced data size and simpler relationships; better because noise and spurious data fields have been reduced.

Though principal component analysis has been around since the beginning of the century, little has been done to define the number of principal components to keep in developing a model. Genetic algorithms attempt to find, develop, and improve links (i.e., optimize the relationship) between data and desired output.

Genetic algorithms are used here to indicate which principal components influence the output.

An application of this methodology using data from TVA's Kingston Fossil Plant Unit 5 is presented. The number of inputs was reduced using a principal component analysis tool and a genetic algorithm optimization for a neural network trained to predict net unit heat rate (NUHR). Though this work uses fossil power plant data, certainly other systems where large amounts of data define the system can also be investigated. Nuclear power plants and chemical manufacturing sites have many systems where this technique may be useful.

1.1 Objective

The purpose of this research was to investigate the use of principal component analysis to reduce the number of inputs to a neural network to be trained and quantify any benefits of this methodology. Further, investigation into the use of genetic algorithms was carried out as a selection tool of relevant principal components. An application of this methodology will be shown using data from TVA's Kingston Fossil Plant data in which NUHR will be predicted. A comparison between the use of principal components and the original data as inputs to train a neural network was made.

1.2 Principal Component Analysis

Principal component analysis is a statistical technique that linearly transforms an original set of variables into a substantially smaller set of uncorrelated variables that represent most of the information in the original set. This smaller set of uncorrelated variables is much easier to understand and use in further analyses than a larger set that contain correlated variables. Principal component analysis was originally conceived by Pearson in 1901 and independently developed by Hotelling in 1933. [1][2]

1.3 Genetic Algorithms

Genetic algorithms were developed by John Holland in the early 1970s and are nonlinear method of finding relationships between data. [3] The primary objective of genetic algorithms is the optimization of a system in accordance with a fitness function. By mimicking natural evolution, software packages attempt to find optimal solutions or arrangement of data for specific output. Thousands of generations are performed. Through each generation, reproduction by the means of mating and mutation are performed. Survival of the best characteristics result in the best data to predict the desired output.

1.4 Application to TVA's Kingston Fossil Plant

An application of these techniques will be made to TVA's Kingston Fossil

Plant Boiler Unit 5 data to predict the Net Unit Heat Rate (NUHR). Improvements of thermodynamic efficiency in a 200 MWe power plant like Kingston Unit 5 by one half of a percent could result in savings of a half million dollars per year (about 175 BTU/Kwhr based on an average net heat rate of 9,000 BTU/Kwh and a fuel cost of $2.50/million BTU). A decorrelator will be used to perform the principal component analysis on the raw data from the plant, and a comparison between a neural network trained with the raw data and the decorrelated data will be shown for varying number of components.

CHAPTER 2

PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is a classical multivariate technique. In 1901, Pearson conceived it as a means of fitting planes by orthogonal least squares, and later in 1933 Hotelling used it for the purpose of analyzing covariance and correlation structures.[4] Principal component analysis can be considered as:

1. a method for transforming correlated variables into uncorrelated ones,
2. a method for finding linear combinations with relatively large or relatively small variability,

and

3. a tool for data reduction. [5]

Principal component analysis attempts to linearly transform p dimensional random vector, $\mathbf{X}$, $(x_1, x_2, x_3, \ldots x_p)$ into a p dimensional vector $\mathbf{Y}$ such that;

$$\mathbf{Y} = a_1x_1 + a_2x_2 + \ldots + a_px_p.$$

In principal component analysis, the weights (i.e., $a_1, a_2, \ldots, a_p$) are mathematically

determined to maximize the sum of squared correlations of the principal component with the original variables. The principal components are ordered with respect to the amount of the variation accounted for in the original variables. Thus the first few principal components account for most of the variation of the original data set[1].

The first principal component geometrically represents the line of closest fit to *n* observations in the *p* dimensional variable space. It minimizes the sum of squared distances of the n observations from the line representing the first principal component. The second principal component represents a line perpendicular to the first component and is the line of closest fit to the residuals from the first principal component. The first two principal components define a plane of closest fit to the points in the *p* dimensional variable space. The first three components define a three dimensional space of closest fit, and so on.

The first principal component algebraically represents a linear combination of **X**,

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p = \sum_{i=1}^{p} a_{1i}x_i$$

such that the variance of y_1 is maximized given the constraint that the sum of

[1]Many authors refer to this variation explained as "knowledge," and it will be occasionally referred to here as knowledge

squared weights is equal to one,

$$\sum_{i=1}^{p} a_{1i}^2 = 1$$

Principal component analysis finds the optimal weight vectors and the associated variances for y_1 which is usually denoted by λ_1, as will be shown later known as latent roots or eigenvalues. Similarly, the second principal component involves finding a second weight vector that maximizes the variance of

$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p = \sum_{i=1}^{p} a_{2i}x_i$$

but is constrained to being uncorrelated with the first principal component and the sum of its squared weights is equal to one,

$$\sum_{i=1}^{p} a_{2i}^2 = 1$$

The second principal component is orthogonal to the first component and is therefore uncorrelated with it. It's weight vector when crossed with the first equals zero which as will be seen later all of the weight vectors form an orthogonal matrix.

$$\sum_{i=1}^{p} a_{1i}a_{2i} = 0$$

This process continues until as many components as variables have been calculated. This results in the first principal component accounting the largest sum of squared correlations with the original variables; the second component with the second largest, and so on. With each component accounting for a decreasing amount of the variance of the original variables, the first few principal components can be used to represent the original data. Thus the data set can be reduced without drastically reducing the knowledge retained.

There exists p principal components as p linear transformations of p original variables. They are;

$$y_1 = \sum_{i=1}^{p} a_{1i}x_i$$

$$y_2 = \sum_{i=1}^{p} a_{2i}x_i$$

$$\vdots$$

$$y_p = \sum_{i=1}^{p} a_{pi}x_i$$

Which can easily be expressed in matrix algebra as,

$$\mathbf{Y} = \mathbf{A'X}$$

where **Y** is a p element vector of principal components, **A'** is a p x p matrix of weight vectors or eigenvectors, and **X** is a p element column vector of the original variables.

Since **A'** is an orthogonal matrix, $\mathbf{AA'} = \mathbf{I}$. Thus the original variables, **X** can be expressed as a linear combination of its principal components, **Y**, as

$$\mathbf{X} = \mathbf{AY}.$$

Performing some matrix manipulation gives,

$$\mathbf{XX'} = \mathbf{AYY'A'}$$

Taking the expectation of both sides

$$\mathbf{E(XX')} = \mathbf{AE(YY')A'}$$

yields

$$\mathbf{R} = \mathbf{A}\,\Lambda\,\mathbf{A'}$$

where **R** represents the covariance matrix of **X** and **Λ** represents a diagonal matrix of the variances of the principal components, **Y**.

Further simplifying,

$$\mathbf{RA} = \mathbf{A\Lambda}$$

$$\mathbf{RA} - \mathbf{A\Lambda} = 0$$

which reduces to,

$$(\mathbf{R} - \Lambda)\,\mathbf{A} = 0$$

which can be simplified to

$$(\mathbf{R} - \lambda\mathbf{I})\,\mathbf{A} = 0$$

Which is an eigenvalue problem where λ_i are the eigenvalues of the covariance matrix, **R**.

The above derivation of principal components was based on using eigenvectors and eigenvalues of the covariance matrix of the original variables. In practice, it is more common to use the correlation matrix of the original variables

to define the principal components. Now let

$$\mathbf{Y} = \mathbf{A}'\mathbf{X}^*$$

where $\mathbf{X}^*$ consists of standardized variables of $\mathbf{X}$ such that

$$x^*_j = x_j/\sigma_{jj}^{1/2}$$

where σ_{jj} is the variance of x_j. The covariance matrix for $\mathbf{X}^*$ is the correlation matrix of $\mathbf{X}$, thus the derivation is much the same as above with this change in $\mathbf{X}$. The principal components, $\mathbf{Y}$, and the eigenvectors, $\mathbf{A}$, are based upon the correlation matrix of $\mathbf{X}$.

Principal components found from covariance and correlation matrices do not give equivalent information and cannot be derived directly from each other. Principal components found using correlation matrices are generally used in practice because principal components based on covariance matrices are sensitive to the units of measurement used for the elements of the original variables. If there are large differences between variances of the elements of the original variables, then those variables whose variances are largest will tend to dominate the first few principal components.

There are several criteria for selecting the number of principal components

to keep. In 1960, Kaiser recommended dropping principal components with latent roots (eigenvalues) less than one.[5] His reasoning was that any principal component with a latent root less than one contained less knowledge than a single standardized variable with a variance equal to one. Some have argued that this is too strict and may result in discarding components that were important.

Jolliffe suggested a cutoff of 0.7 for the latent roots. He proposed that if a variable in the original data set was largely independent of all other variables, it would dominate one principal component with its variance close to one, but it would have small coefficients in the other principal components. This variable and its related principal component provides its own independent information from the other variables and should not be discarded. However, deletion will occur if Kaiser's rule is used, and if, due to sampling variation its variance is less than one [4]. Based on simulation studies, Jolliffe found that a value of 0.7 is roughly the correct cutoff for the latent roots. Though in some cases, this has resulted in keeping twice as many components as Kaiser's criterion.

In 1966, Cattell proposed the use of a graph of the latent roots versus their principal components. He suggested keeping the components where the line was "steep" and discarding the components where the line was "not steep".[5] This idea of "steep" and "not steep" is arbitrary and in some principal component analysis there is no definitive change.

Perhaps the simplest method or criteria was to retain the number of

principal components to account for a given percentage of the total variation, e.g. 80%.

CHAPTER 3

GENETIC ALGORITHMS

In the early 1970s, John Holland began the study of genetic algorithms. Over the past two decades, the field of genetic algorithms has developed into a useful and intriguing tool. The primary objective of genetic algorithms is optimization. The process by which the tool obtains this optimization is to closely mimic the natural evolutionary process.

Evolution takes place in the natural world through reproduction. During reproduction, mutations and mixing of the parents' gene pool create an offspring that has similar but different characteristics of the parents. Over many generations, the process of natural selection determines which of these characteristics and offspring survive. In nature, only the strong survive, but "strong" as defined by nature and its surroundings. Evolution is this process in which creatures change to better adapt to its surroundings.

Genetic algorithms try to mimic the evolutionary process. The better the process is at copying the evolution, the more likely it is to produce an optimal solution. Evolution takes place in the chromosomes and during reproduction they are mated and mutated to give different characteristics. Evolution takes place over many generations which in life takes a long time. Genetic algorithms attempt the same kind of changes, but through the use of a computer, thousand of generations can be computed in seconds.

During reproduction, the parents' chromosomes are mixed and mutated to produce the next offspring. The mixing of two chromosomes is process called crossover, and two new chromosomes are produced which have some of the characteristics from each of the parent chromosomes. In a uniform random process, a crossover point on the parents' chromosomes is identified. At this point the parents' chromosomes split and recombine with the other's split chromosome, producing two new chromosomes that have those characteristics of the parent that came with that parent's chromosome. A mutation is a process when a single element of a chromosome is changed and is not a result of the parents' donation. This a completely random process and is an abrupt change in evolution which will influence every subsequent generation if it is selected for reproduction.

As a result of reproduction, these new chromosomes go into a large gene pool for possible reproduction. If the chromosome is of poor quality, it will be discarded. Its good and bad features will be lost forever unless a subsequent mutation recreates an identical gene. The selection of new chromosomes for reproduction continues. This is somewhat a random process, but the better chromosomes are more likely to be selected first. For example, the strongest lion mates with the lioness of the pride unless he is killed off. Generation after generation better characteristics are retained and poorer ones discarded, and over many generations, the quality of the gene pool improves.

Genetic algorithms attempt to mimic this reproduction process and use a

fitness function as its criterion for natural selection. Two steps are required: 1.) encode variables into strings of binary bits, and 2.) define the fitness function. Input data are represented as bit strings and take on the look of binary chromosome strings, e.g. 100110101. These "chromosomes" are then entered into the gene pool for reproduction. Each string is evaluated for its fitness, and ones with a higher fitness value will have a higher probability of contributing offspring to the next generation. The mating pool for the next generation is selected according to this probability, i.e., the ones with the higher probability being selected for reproduction.

The fitness function must be defined in order to guide the search process. The function may combine criteria to reduce training error and criteria to reduce the number of inputs used in training. The fitness function gives the user flexibility to design the genetic algorithm to select "chromosomes" by his definition of "strong".

The selection process is repeated for many generations until there is little change in the average value of the fitness functions for the pool. The binary string with the highest fitness function is then decoded to find the optimal value.

CHAPTER 4

NET UNIT HEAT RATE APPLICATION

4.1 Data

The data used was from TVA's Kingston Fossil Plant Unit 5. Data was collected at one minute intervals over a period of five days. A total of 140 different variables and more than 7200 data sets were recorded for analysis. Some of the variables are calculated from other signals. Net Unit Heat Rate was a calculated variable which was chosen as the important indicator of thermal efficiency. The 140 original signals were reviewed. Editing for incomplete data sets and dead signals reduced the number of variables to 108 time series and just over 7100 data sets.

The net unit heat rate (NUHR) was removed from this data set to be the variable to be predicted. The rest of the data was to undergo a principal component analysis to find new uncorrelated variables. The NUHR would be added to these new variables as a target for the neural network. The first 5100 data sets would then be used for training sets for the neural networks and the last 2000 to be used as test sets.

4.2 Principal Component Analysis

The principal component analysis was accomplished by the use of a decorrelator. Jurik Research offered a Microsoft Excel tool that performed this

function. In personal communications Jurik indicated that the transformation is performed within the Excel environment by matrix manipulation.[6] All variables, except NUHR, and time series were standardized by their variances, and the principal components were based on the correlation matrix of the original variables. From the original data set (107 x 7100), the decorrelator performed the principal component analysis within the Excel environment to provide an identically sized data set of uncorrelated variables. Also provided are the percent of the variance of the total explained by each principal component. This percent variance of the total indicated the amount of knowledge contained by each component.

There are general "rules of thumb" or criteria for selecting how many principal components to retain, i.e., Kaiser (1960), Jolliffe (1972), and Cattell (1966) as discussed earlier. For the purposes of analyzing this technique and the goal of training a usable neural network, a variety of cases with a different number of principal components were examined.

4.3 Genetic Algorithms

The genetic algorithm analysis was accomplished through the use of a software package, NeuralWorks PREDICT, from NeuralWare, Inc. The

PREDICT[2] software operated within the Microsoft excel environment. The PREDICT software used a proprietary fitness function based on an Adaptive Gradient learning rule. All 107 decorrelated variables, principal components, from the Jurik decorrelator were used in the PREDICT software along with the net unit heat rate as the desired output. The results of the analysis were used as another method of principal component selection, and a neural network was trained from these results.

4.4 Neural Network

Back-propagation neural networks were trained using the uncorrelated variables obtained from the principal component analysis to predict the net unit heat rate. Further the raw data from the plant containing all input variables was also used as an input to a network for comparison with this technique and past work, (Black 1994). The Neural Network Toolbox within MATLAB was selected for its convenience and reliability.

A three layer back-propagation network was designed. Figure 4.1 shows the neural network architecture.

---

[2]PREDICT is a commercially available software program that utilizes genetic algorithms to select the choice of variables based on a fitness function using an Adaptive Gradient learning rule. It also implements a cascade correlation neural network that is not used here. Rather MATLAB is used to implement a back-propagation neural network in order to be consistent with the neural networks used for the other cases presented here.

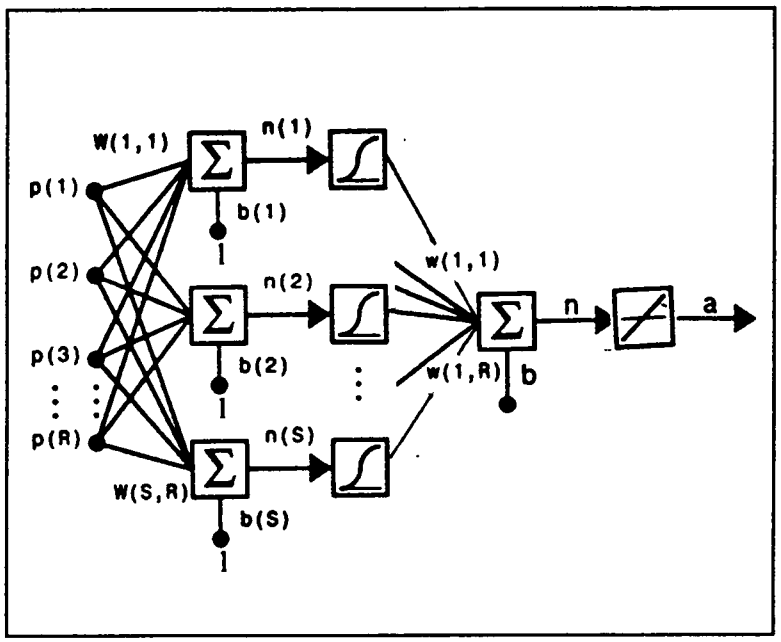

Figure 4.1: Neural network architecture (modified from MATLAB Neural Network Toolbox User's Guide)

A logsig transfer function was used for the output of the hidden layer and a linear transfer function for the output of the output layer. Figures 4.2 and 4.3 show the transfer functions. Both inputs and the output were scaled between 0.1 and 0.9. Due to the size of the data, every tenth point was used for training and the networks were trained to a total output error tolerance of 10.

Networks using 107 original variables and 30, 20, 10, and 5 uncorrelated variables (principal components) as inputs were trained and tested. Neural networks were also trained and tested using principal components selected by the PREDICT genetic algorithm.

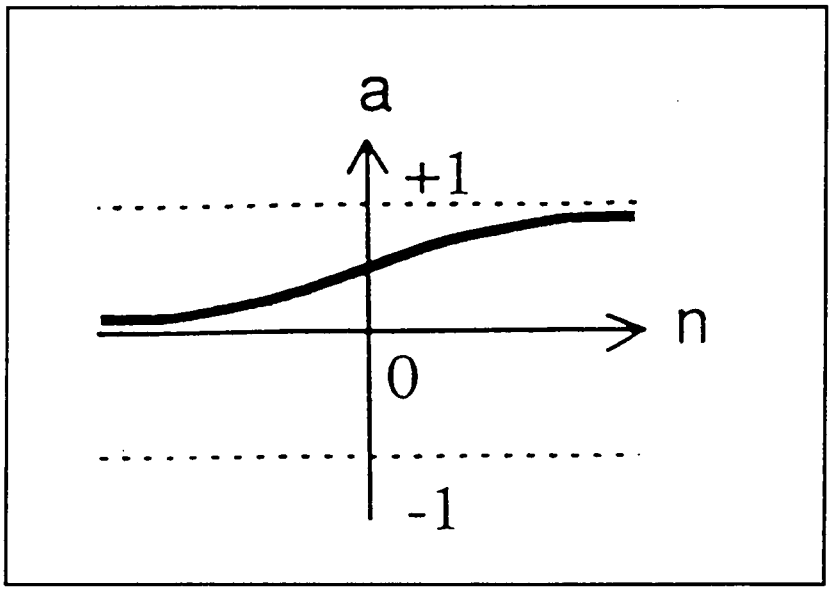

Figure 4.2: Logsigmoidal transfer function.

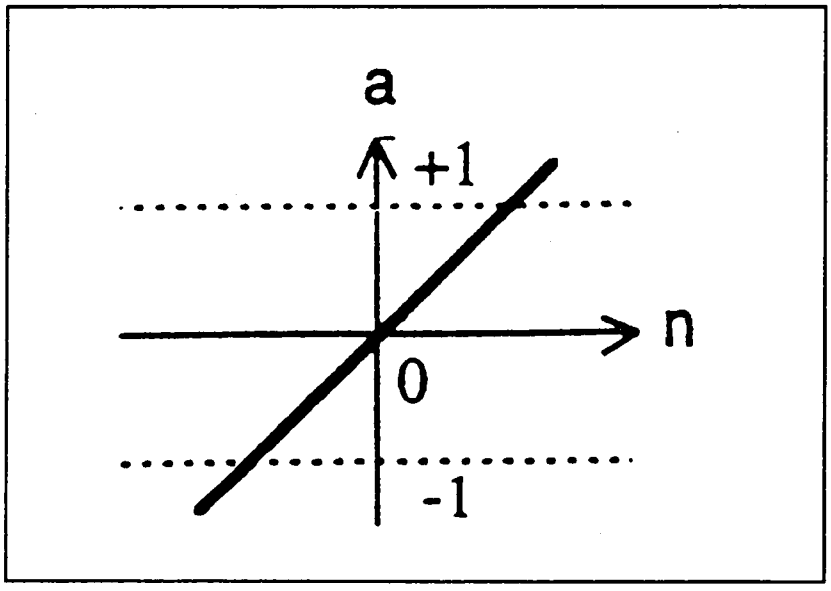

Figure 4.3: Linear transfer function.

CHAPTER 5

RESULTS

5.1 Principal Component Analysis

The principal component analysis for the (107 x 7100) data set from TVA's Kingston steam plant Unit #5 was completed by the Jurik Research decorrelator. The principal components were checked for correlations with each other and none were found. Figure 5.1 shows all principal components with the percent of the original variables' variance explained.

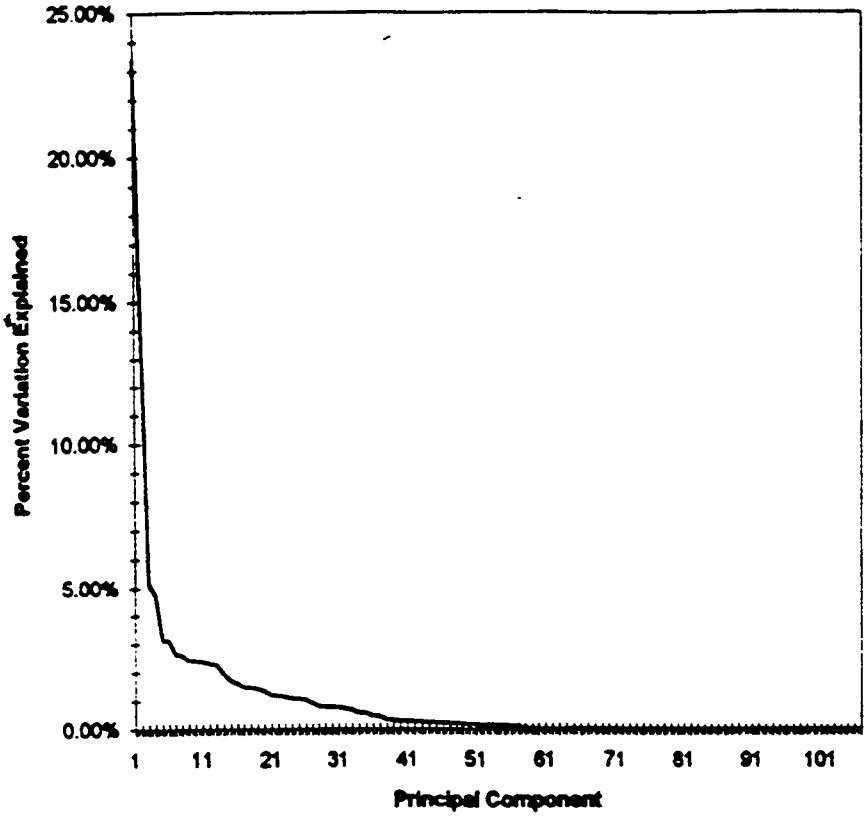

Figure 5.1: Percent Variation explained by each principal component.

This graph is similar to a graph of the eigenvalues, λ_i, for each principal component, y_i. This information along with the eigenvalue for each principal component can be used to select the number of principal components to keep by Cattell's slope method.

Table 5.1 shows the number of principal components to retain to account for a certain level variance explained. By Kaiser's and Jolliffe's criterion, 26 and 33 principal components respectively should be kept.

Table 5.1: The number of principal components to retain to account for the variance explained.

Amount of Variance Explained	Number of Principal Components
50.6%	5 (Cattell's criteria)
63.8%	10 (Cattell's criteria)
75.0%	16
81.8%	20
85.0%	23
88.2%	26 (Kaiser's criteria)
91.8%	30
94.3%	33 (Jolliffe's criteria)

In selecting the number of principal components to keep, it was decided to retain a certain level of variance explained of the original variables. A level of

80% and 90% was selected for those levels. Two sets of data were selected using 20 principal components and 30 principal components representing greater than 80% and greater than 90% of the total variance of the original variables, respectively. These were also very close to the number of components suggested by Kaiser's or Jolliffe's criterion respectively.

To examine Cattell's slope criterion, a couple of data sets must considered. Using Figure 5.1, it can be seen that the slope of the graph changes dramatically at 5 principal components and then again at 10. Cattell's method of choosing the number of components where the graph is "steep" was somewhat arbitrary, so two data sets were chosen that represent or bound the region in which the slope changes the most. These two data sets consisted of 5 and 10 principal components.

Table 5.2 shows the amount of variance of the original data set explained by each of the six sets of principal components selected.

Table 5.2: Amount of variance explained by each principal component set.

Number of Principal Components	Amount of Variance Explained
30	91.8%
20	81.8%
10	63.8%
5	50.6%
11 PREDICT	9.7%

5.2 Genetic Algorithms

As another method of principal component selection, the PREDICT genetic algorithm recommends an optimal neural network with 11 principal components as inputs for the net unit heat rate. The following components were selected by PREDICT:

6 7 11 37 43 45 47 51 52 56 91

These 11 components accounted for approximately 9.66% of the original variables' variance. This is a shocking result and is discussed in terms of the results obtained from the neural network using the inputs indicated above.

5.3 Neural Network

Six neural networks were trained and tested. The original raw data set was used as a baseline and was also used as a comparison to earlier work (Black, 1994). Table 5.3 shows the training times and the number of epochs during training (a pass through all of the training data and target is called an epoch). Figure 5.2 and figure 5.3 show the actual NUHR of the plant and the neural network estimate versus time, respectively. Figure 5.4 shows the residual of the NUHR which is the difference between prediction and actual NUHR.

Table 5.3: Training times.

Model	Training time	No. of Epochs
107 original variables	6 hr. 15 min.	1007
30 principal components	3 hr.	425
20 principal components	2 hr. 15 min.	355
10 principal components	1 hr. 20 min.	155
5 principal components	45 min.	75
11 PREDICT p. components	1 hr. 20 min.	153

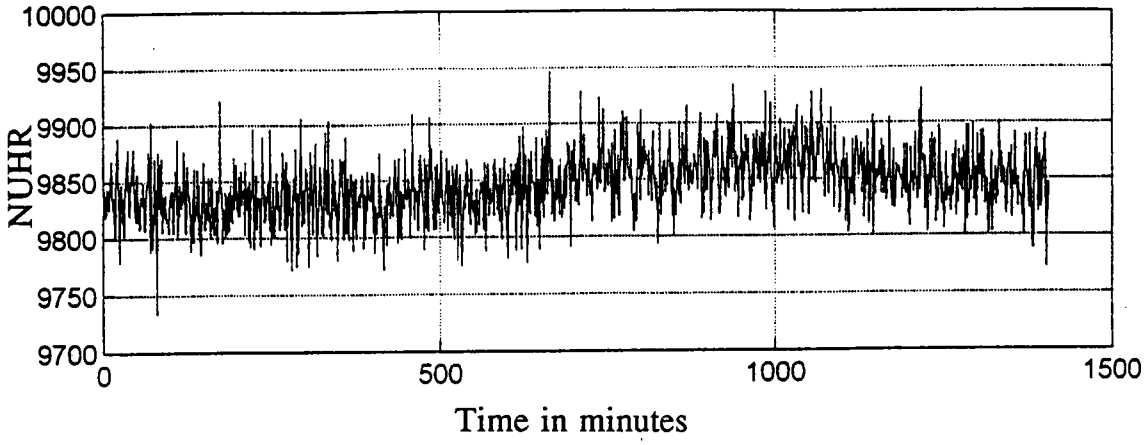

Figure 5.2: Actual Net Unit Heat Rate (BTU/kW$_e$ Hr) for the plant.

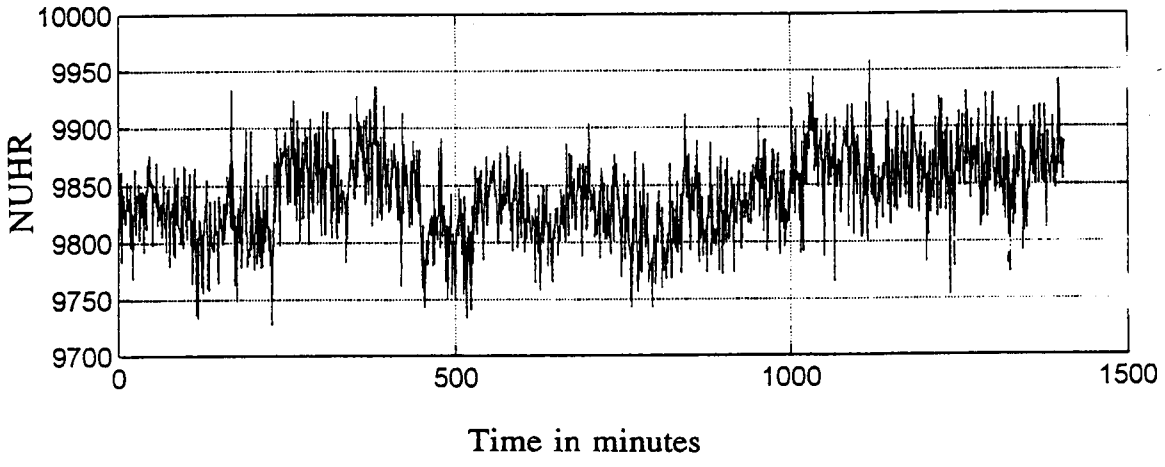

Figure 5.3: NUHR prediction (BTU/kW$_e$ Hr) using original data (107 variables).

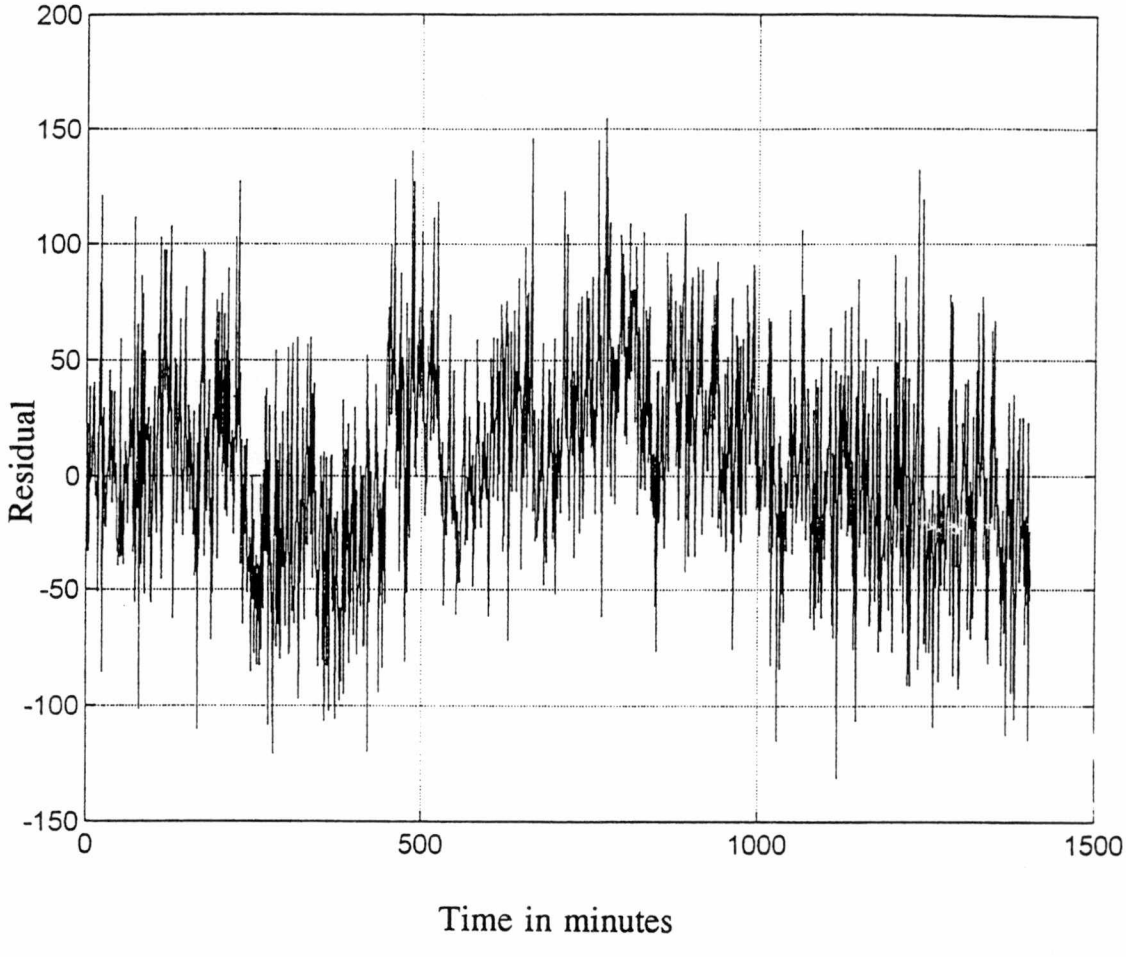

Figure 5.4: Residual NUHR (BTU/kW$_e$ Hr) using original data (107 variables).

The first neural network to be compared to this baseline would be the one using the 30 principal components as inputs to the neural network. These 30 components represented over 90% of the variance of the original 107 variable data set. Figure 5.5 and figure 5.6 show the actual value of NUHR of the plant and the neural network prediction versus time, respectively.

Note that the 30 principal component prediction is not as variant as the prediction from the original data model. Figure 5.7 shows the residual value of the NUHR versus time using 30 principal components. This also shows reduced variability over the original data set model indicating reduced noise in the model.

Figures 5.8, 5.9, and 5.10 show the actual NUHR, the predicted NUHR, and the residual NUHR using 20 principal components as inputs to the neural network. The prediction for the 20 principal component model, shown in figure 5.6, shows less noise than the previous 30 principal component model, and as will be seen as the principal components are reduced, the "noise" is further and further reduced. As the number of principal components was further reduced, the prediction begins to flatten to the average value of the net unit heat rate.

Similarly, figures 5.11, 5.12, and 5.13 show the actual NUHR, the prediction of NUHR, and the residual NUHR, respectively for the use of 10 principal components as input to the neural network.

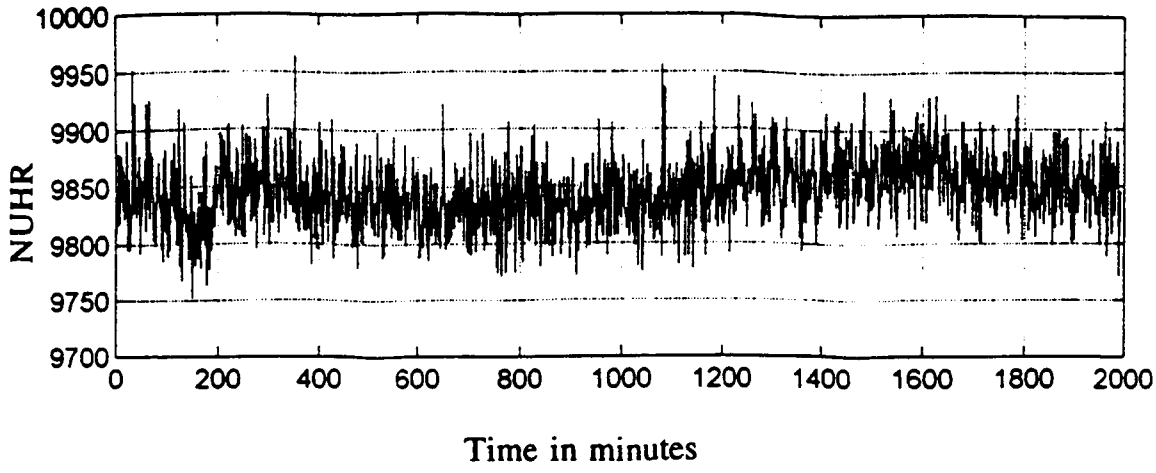

Figure 5.5: Actual NUHR (BTU/kW$_e$ Hr) for the plant.

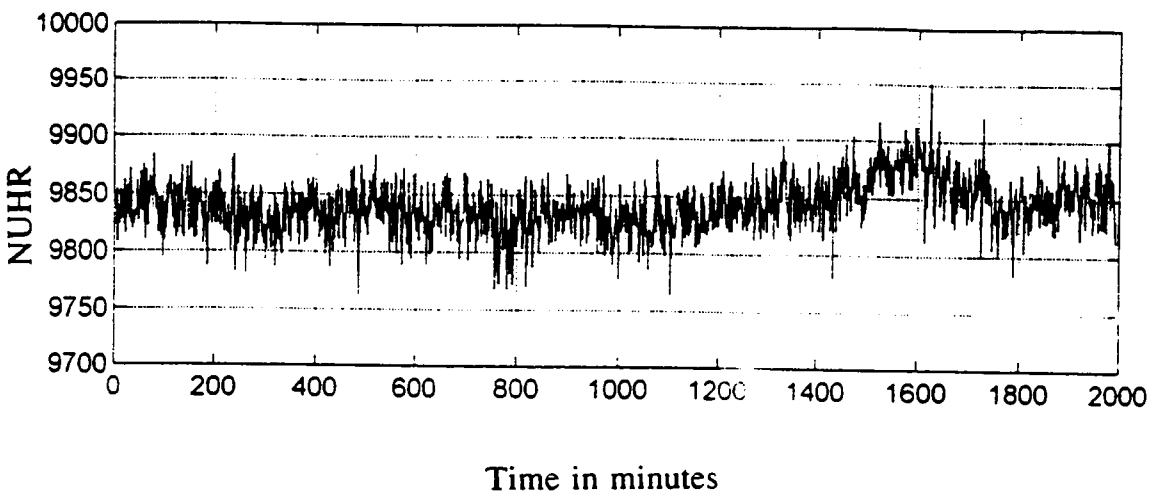

Figure 5.6: NUHR prediction (BTU/kW$_e$ Hr) using 30 principal components.

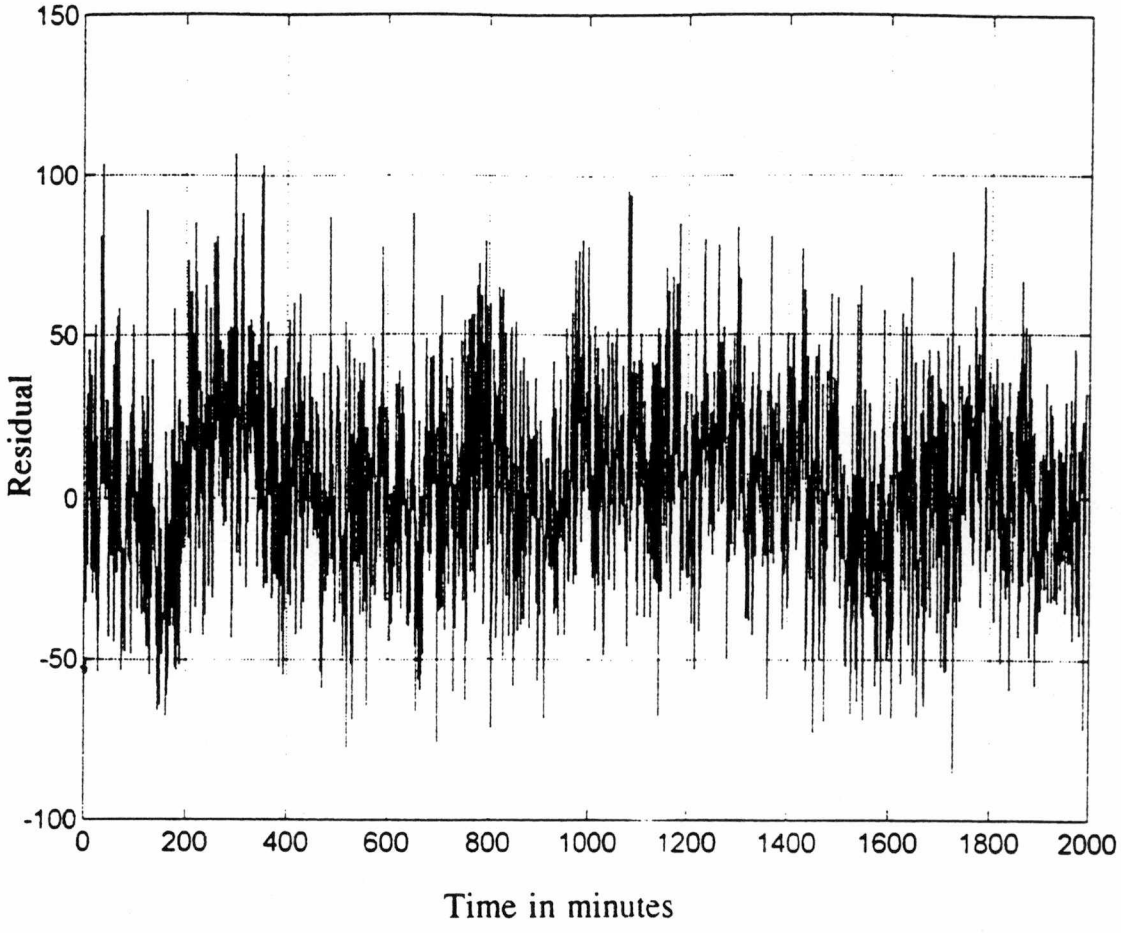

Figure 5.7: Residual NUHR (BTU/kW_e Hr) using 30 principal components.

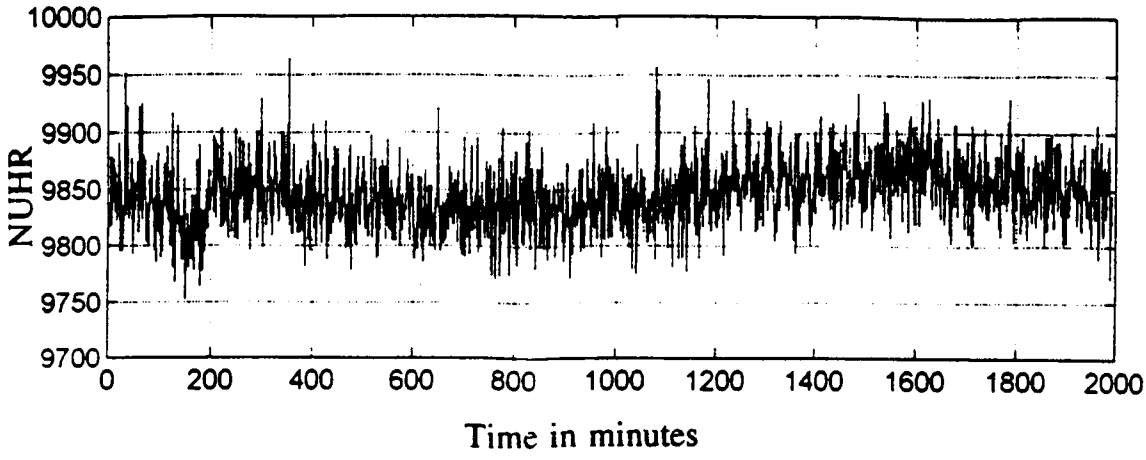

Figure 5.8: Actual NUHR (BTU/kW$_e$ Hr) for the plant.

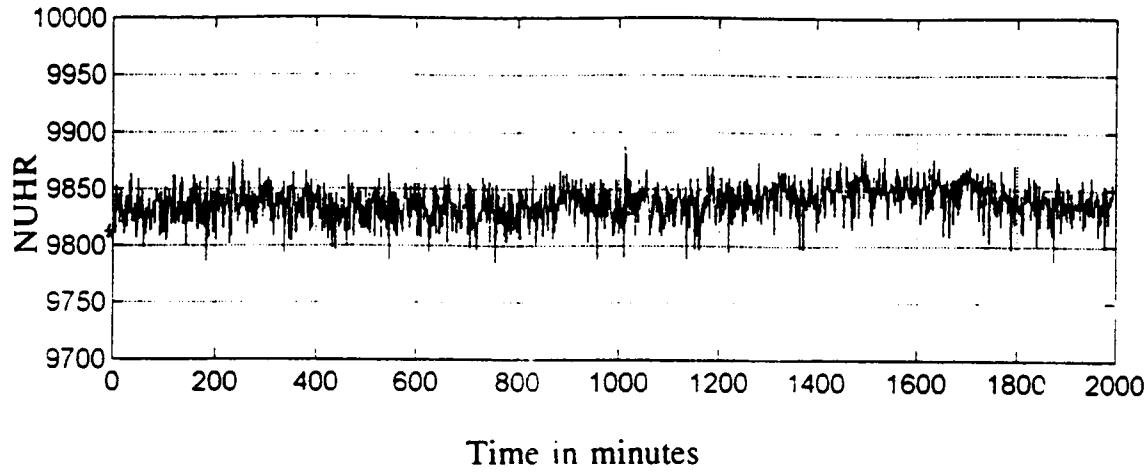

Figure 5.9: NUHR prediction (BTU/kW$_e$ Hr) using 20 principal components.

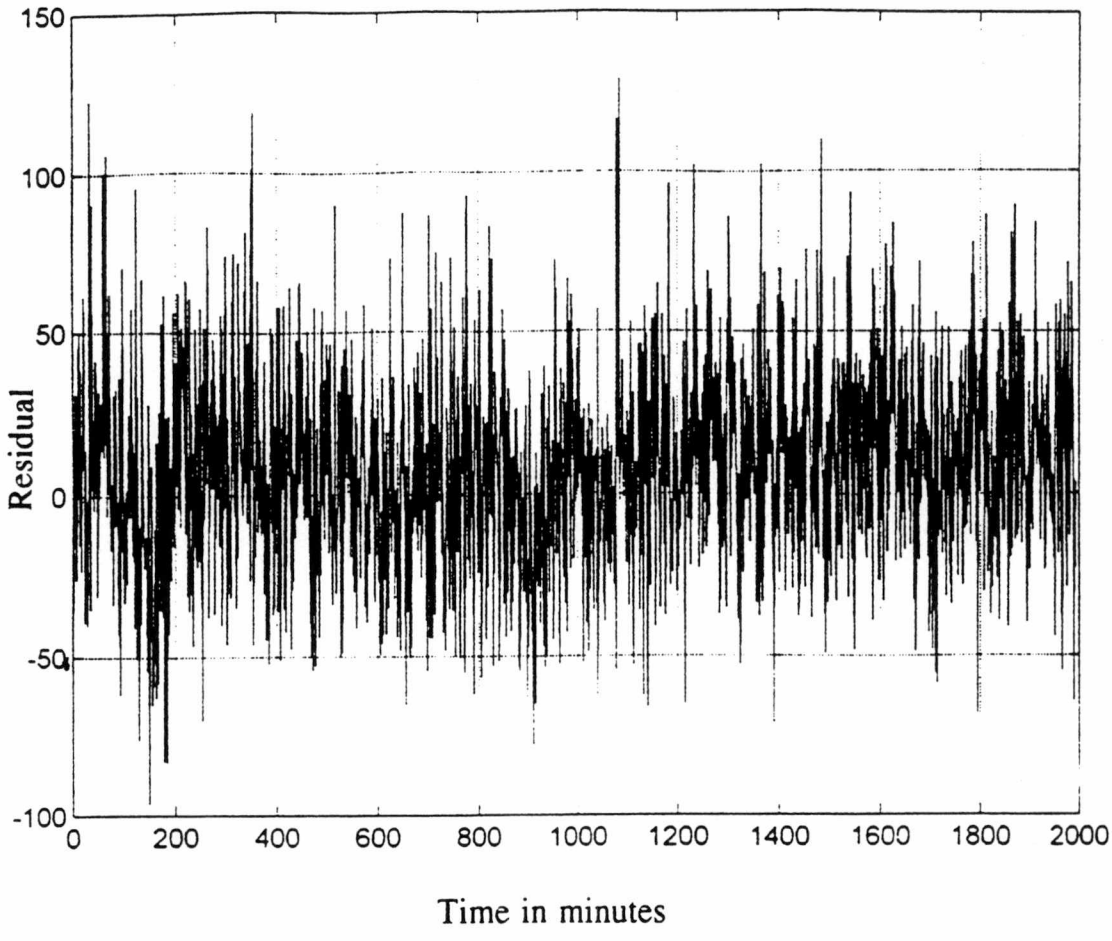

Figure 5.10: Residual NUHR (BTU/kW_e Hr) using 20 principal components.

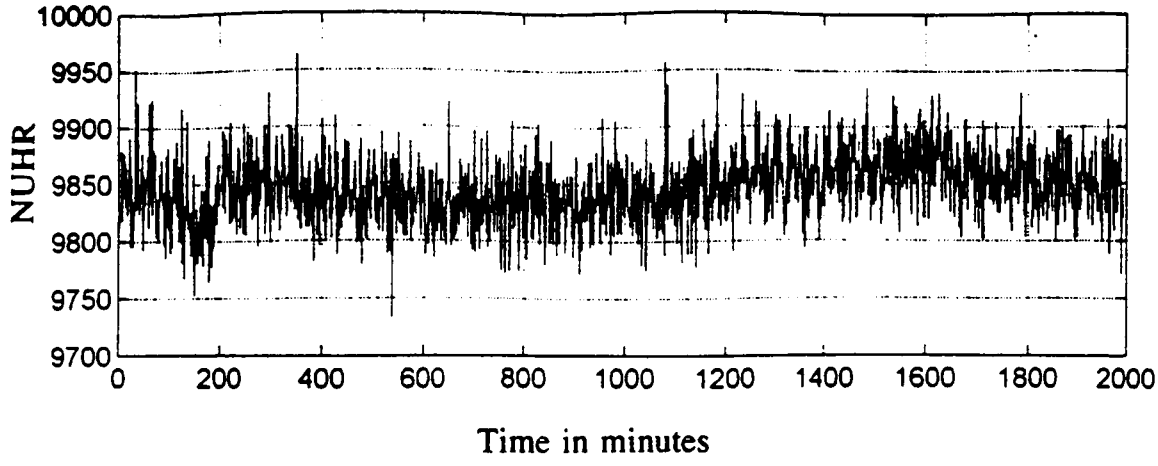

Figure 5.11: Actual NUHR (BTU/kW$_e$ Hr) for the plant.

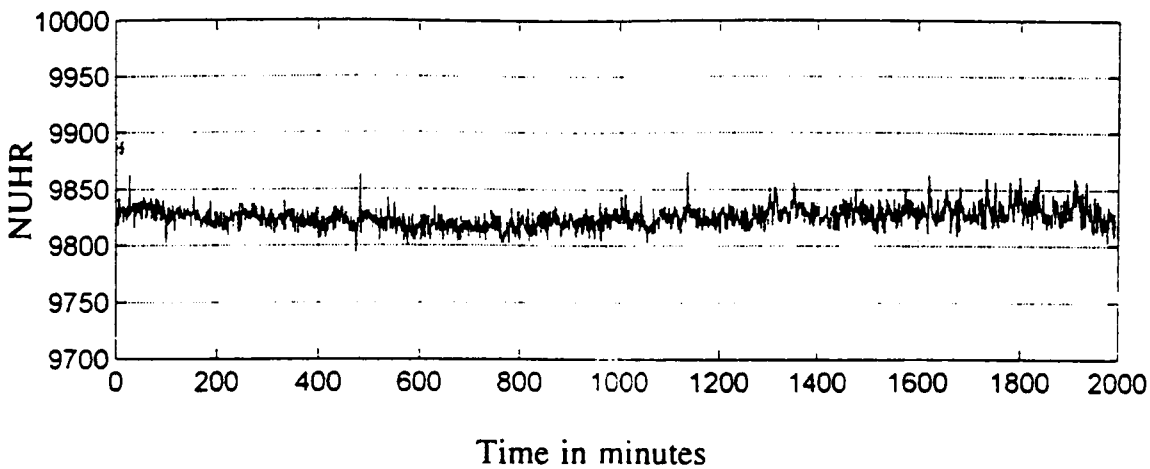

Figure 5.12: NUHR prediction (BTU/kW$_e$ Hr) using 10 principal component

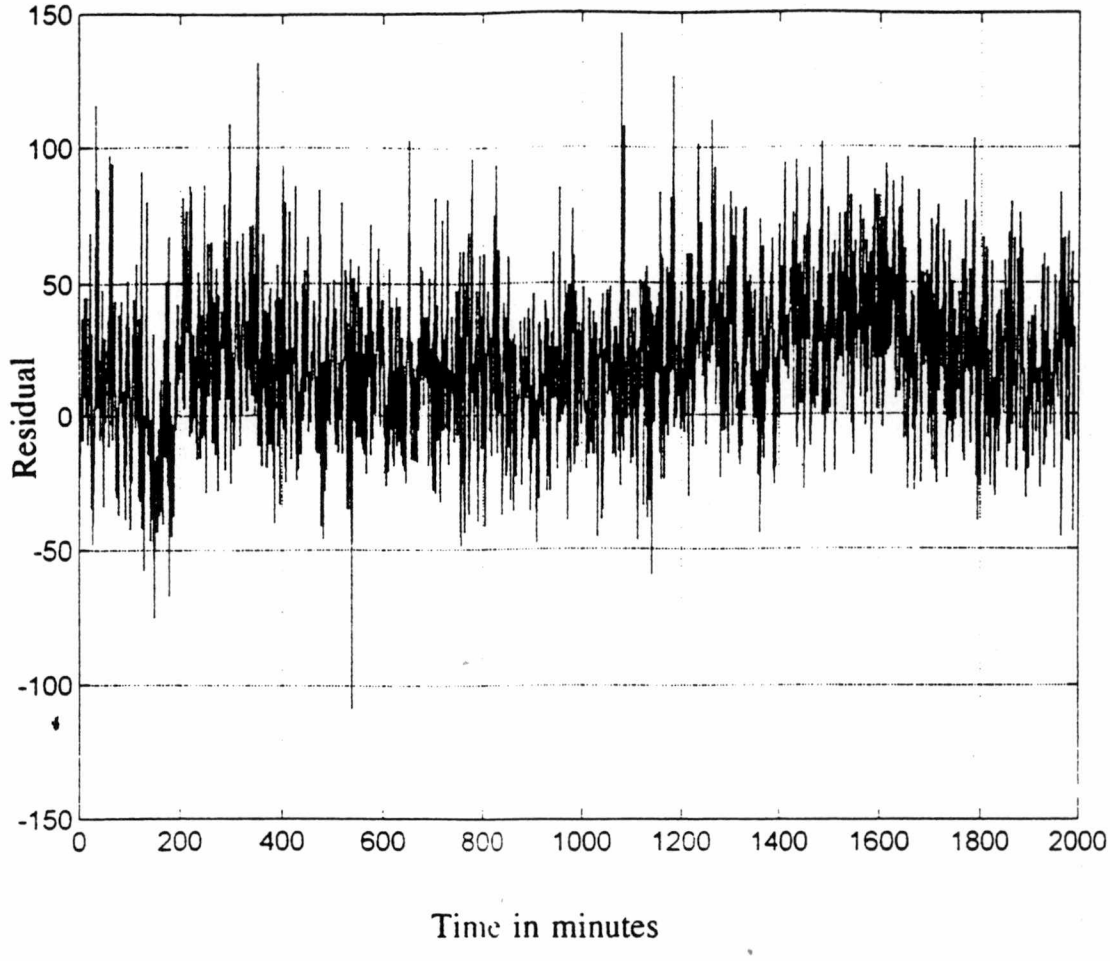

Figure 5.13: Residual NUHR (BTU/kW_e Hr) using 10 principal components.

Finally, Figures 5.14, 5.15, and 5.16 similarly show the same using only 5 principal components input to the neural network for the prediction of NUHR. As seen in figure 5.15, the model's prediction is very flat and close to the actual NUHR average. Because of this, the residual, shown in figure 5.16, is very similar to the variability of the actual NUHR.

Figures 5.17 and 5.18 show the actual and predicted NUHR for the neural network trained using the 11 principal components selected using the genetic algorithm software package, PREDICT. Figure 5.19 show the residual between the two for this model.

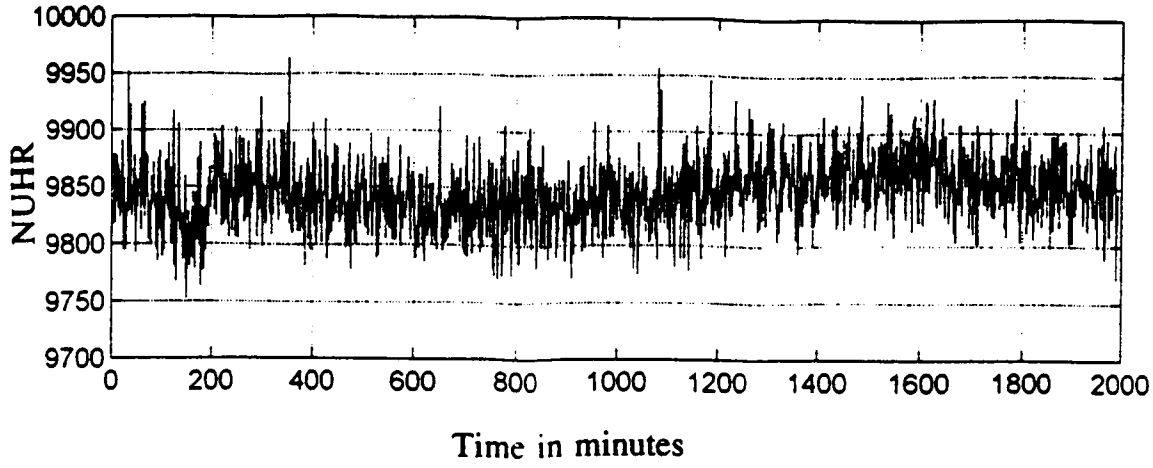

Figure 5.14: Actual NUHR (BTU/kW$_e$ Hr) for the plant.

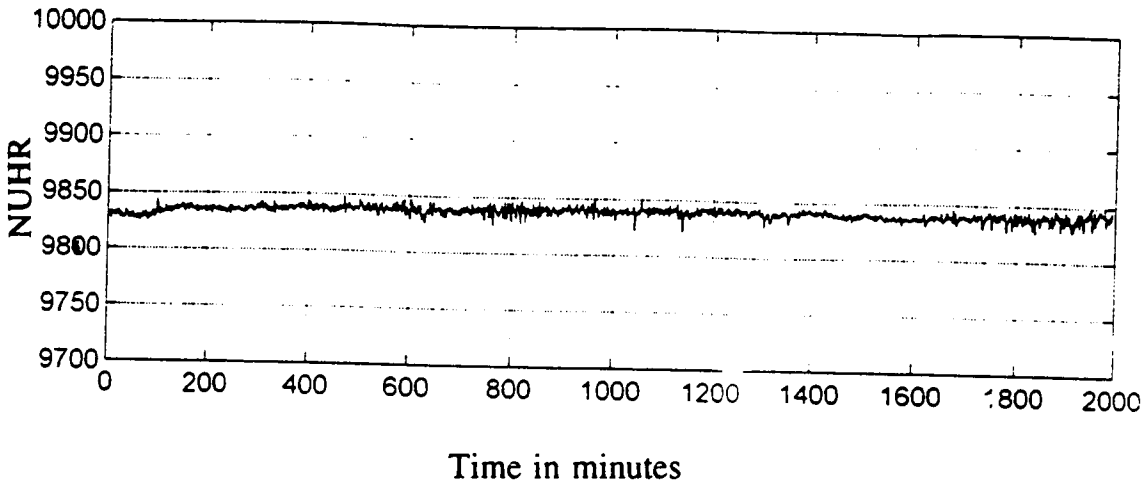

Figure 5.15: NUHR prediction (BTU/kW$_e$ Hr) using 5 principal components.

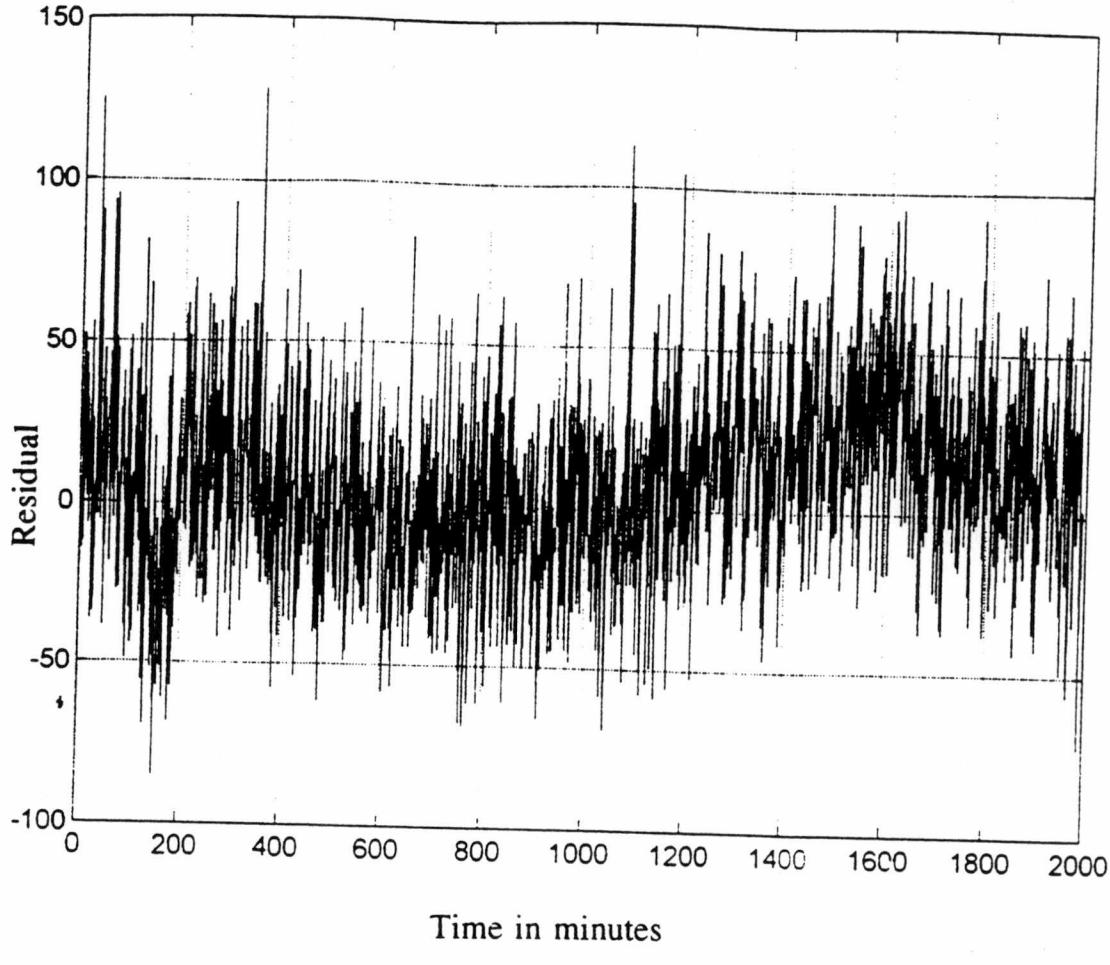

Figure 5.16: Residual NUHR (BTU/kW_e Hr) using 5 principal components.

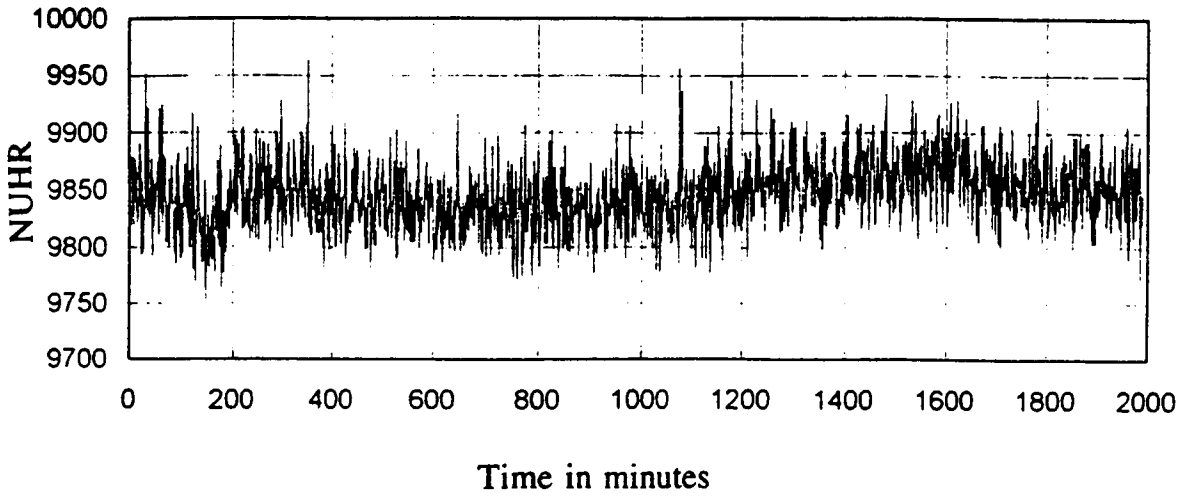

Figure 5.17: Actual NUHR (BTU/kW$_e$ Hr) for the plant

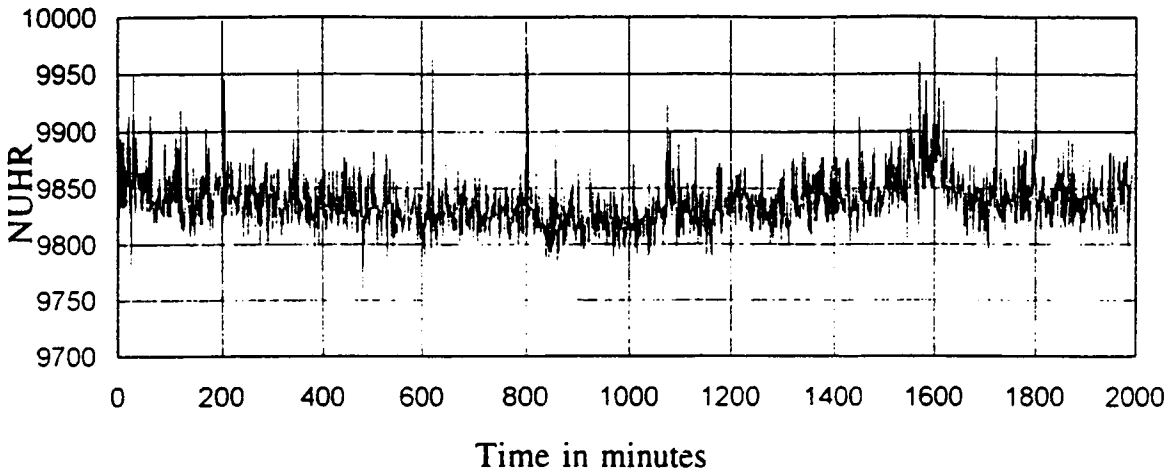

Figure 5.18: NUHR prediction (BTU/kW$_e$ Hr) using 11 principal components selected by PREDICT.

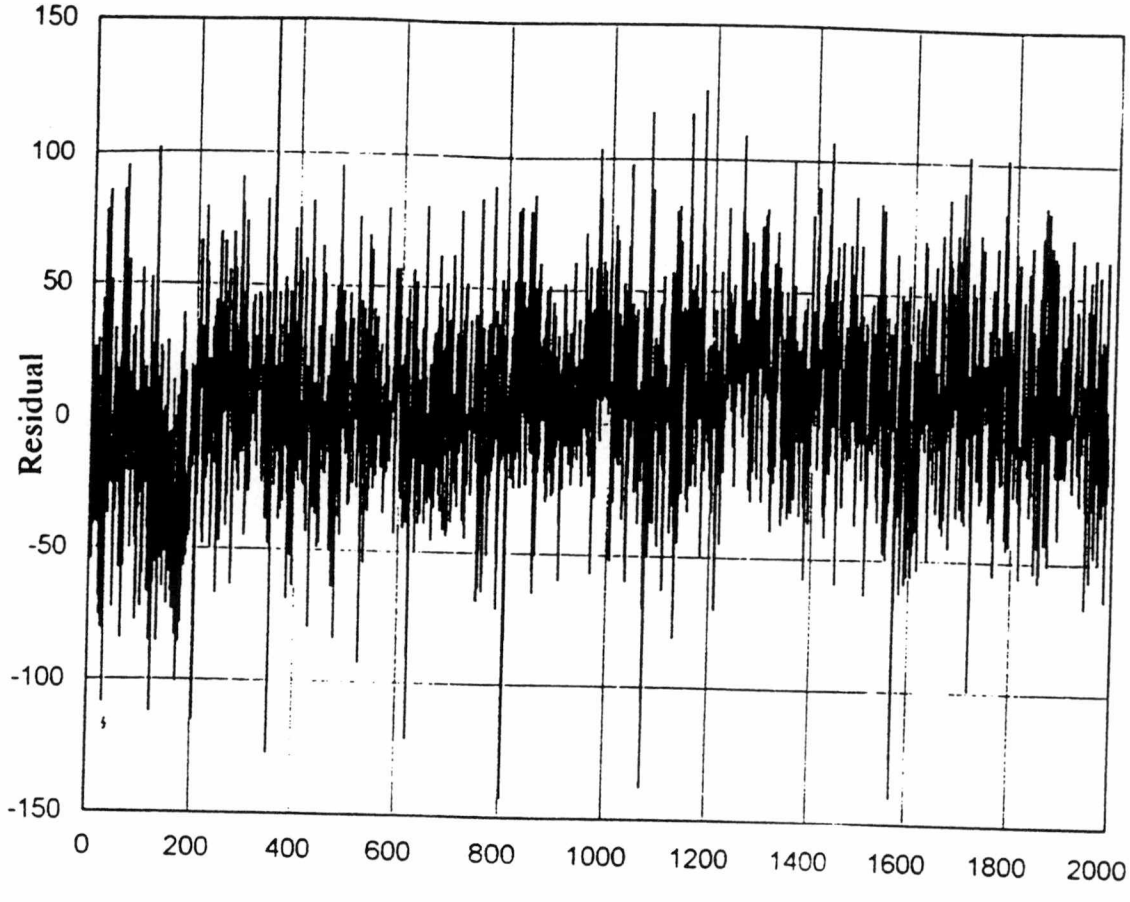

Figure 5.19: NUHR Residual (BTU/kW_e Hr) using the 11 principal components selected by PREDICT.

CHAPTER 6

DISCUSSION OF RESULTS

In analyzing the technique of principal components as a method of data reduction to train neural networks, each model's result must be compared. Table 6.1 shows the statistics of the actual NUHR and the 6 different neural network predictions. As can be from these statistics as well as from the figures that as the number principal components are reduced the predictions begin to flatten out. The standard deviation of the predicted NUHR decreased with the decreasing number of principal components. This shows that reducing the principal components reduces noise, but too great a reduction causes loss of knowledge. This loss of knowledge can be seen in the prediction taking on the look of the average value and its variability disappearing.

The method of forming confidence intervals around each's mean will be employed to check for differences between each mean. Overlapping of the intervals will confirm that those means are similar and not statistically different. A simple method of determining confidence intervals uses the following formula:

$$\text{Confidence Interval} = \text{mean} +/- 2\sigma/(n)^{0.5}$$

where

n is the number of observations

Table 6.1: Statistics of actual and predictions of NUHR.

Model	% Variance accounted	NUHR Mean	NUHR std. dev.
Actual NUHR		9,845.6	28.45
Prediction using 107 original variables	100.0	9,840.8	37.50
Prediction using 30 principal components	91.8	9,841.3	22.58
Prediction using 20 principal components	81.8	9,837.9	14.82
Prediction using 10 principal components	63.8	9,825.1	9.49
Prediction using 5 principal components	50.6	9,837.7	3.80
Prediction using 11 PREDICT p. components	9.7	9,838.1	22.92

Table 6.2 shows these confidence intervals around the means of the predictions. None of the models intervals overlap with the actual NUHR, but in comparing the

2/24 WANG

nponents to using all original variables show that the 30 model compares to the original variables model. The other are different enough to argue their mean and prediction is from the use of original variables.

ce Intervals around NUHR prediction means.

l	Confidence Interval
	9,845.6 +/- 1.27
es	9,840.8 +/- 1.98
nents	9,841.3 +/- 1.01
nents	9,837.9 +/- 0.66
nents	9,825.1 +/- 0.42
nts	9,837.7 +/- 0.17
nponents	9,838.1 +/- 1.02

Perhaps a better comparison between model's predictions would be to

examine their residual values. Table 6.3 shows the statistical analysis of the residuals for each of the five cases.

This shows that perhaps the 30 principal component model performed better than the model using original data. Its mean is closer to zero and its standard deviation is lower. As expected as the number of principal components were reduced the residual mean climbed, except the model with 5 principal components which as discussed approached a flat average value. This could be attributed to the plant level operating rate. Because of the 10 and 5 principal component models' flatness, its standard deviation is very similar to that of the actual data. To further examine the residual values, confidence intervals were set up about each's mean. Table 6.4 shows the residuals and the confidence intervals for each model. Only the 10 principal component model's confidence interval falls outside of the others. This shows the difference of the prediction from the actual NUHR for each are not statistically different from each other, except for the 10 principal component model.

Table 6.3: Statistical analysis of the residuals.

Model	Residual Mean	Residual Std. Dev.
107 original variables	5.44	45.34
30 principal components	4.35	29.72
20 principal components	7.72	30.14
10 principal components	20.46	28.42
5 principal components	7.93	28.87
11 PREDICT principal c.	7.50	33.86

Table 6.4: Confidence Intervals about Residual means.

Model	Confidence Interval
107 original variables	5.44 +/- 2.39
30 principal components	4.35 +/- 1.33
20 principal components	7.72 +/- 1.35
10 principal components	20.46 +/- 1.27
5 principal components	7.93 +/- 1.29
11 PREDICT principal c.	7.50 +/- 1.51

CHAPTER 7

CONCLUSIONS

Results of all of the models to the actual were moderate. The 30 principal component model performed the best. Its three sigma (standard deviation) of +/- 100 on an average predicted value of 9,841 BTU/kWh is fairly good. In comparison, the original data model's three sigma was approximately +/-140. This improvement can be attributed to the reduction of noise in the data. Poor performance and lengthy training time of the original data model has brought about questions about the data itself[3].

The focus remained to compare the use of principal components in a neural network model to a neural network trained with original data, and in that principal component analysis proved its value. The 30 principal component model performed better than the original variables model. The reduction to 30 principal components eliminated some of the noise of the original data. Even the 20 and 5 principal component models showed some advantages over the original variables' model. Their reduced data size shortened training time by 4 hours for the 20 principal component model and by five and a half hours for the 5 principal component model. Their residual values were similar to the original variables' model with

---

[3]Information recently obtained from the EPRI Maintenance and Diagnostics center at TVA's Kingston Plant indicates that the wide fluctuation in NUHR is due to a sensor problem which introduced erratic fluctuations in the calculated values of NUHR.

significantly improved residual standard deviation (33% improvement by each).

Further advantage of the principal component analysis was the preprocessing of the data. This aided in identifying bad data. Because the first couple of principal components contained most of the original variance, they also contained any of the significant errors in the data. Searching the first few columns of a large data field for abrupt changes from one time sequence to the next was much easier than searching the entire data field.

Principal component analysis was an extremely powerful tool in reducing data. However, it did not find links between the decorrelated data and the desired output. Principal component analysis reduced the data and accounted for the original variance of the data, but it had no insight into what variables might be important to a certain desired output. For use in neural networks, it left the user the task of selecting which and how many principal components to keep with very little indication of which components had influence on the desired output.

Genetic algorithms helped bridge the gap between principal components and input to a neural network. The genetic algorithm looked for links between the components and the desired output. The PREDICT software selected 11 principal components which represented less than 10% of the original variables variance and the trained network performed better than the model using 20 principal components.

The use of 11 principal components to predict the net unit heat rate and the

subsequent results were shocking. While the number of principal components was larger than two other sets presented here, the selected components accounted for little of the original variable's variance (9.7% according to the decorrelator), and eight of those eleven components were outside of any documented criteria for selection. Further, the last principal component selected by PREDICT (principal component #91) did not account for practically any of the variance according to the decorrelator. However, the results were nearly as good as the results using 30 principal components which represented more than 90% of the information. The reason may be that while the 30 component model explained greater than 90% of the original variance (knowledge), the net unit heat rate could be predicted using the 9.7% selected by PREDICT.

The good results obtained from the PREDICT selected components model should be tempered by the fact that the actual NUHR had a large amount of variance in comparison to its change over the data set and that the 5 component model which was a fairly flat prediction did a fair job at prediction.

CHAPTER 8

FUTURE STUDY

Principal component analysis and genetic algorithms proved to be very powerful tools and produced some spectacular results. Due to some flaws in the input data, it is unlikely to use any of the models in production, and correcting this error by obtaining new data will produce even better, usable results.

Some work has been done using neural networks to find principal components which would provide an interesting alternative to Jurik's decorrelator because neural networks can provide nonlinear principal components.

REFERENCES

REFERENCES

[1] K. Pearson, "On lines and planes of closest fit to systems of points in space.", Phil. Mag. (6), 2, 1901.

[2] H. Hotelling, "Analysis of a complex of statistical variables into principal components.", J. Educ. Psychol., 24, pp. 417-441, 498-520, 1933.

[3] John Holland, Adaption in Natural and Artificial Systems, University of Michigan Press, 1975.

[4] B. Flury, Common Principal Components and Related Multivariate Models, John Wiley & Sons, 1988.

[5] G. Dunteman, Principal Component Analysis, Sage Publications, 1989.

[6] R. Jurik, Jurik Research & Consulting, Personal Communication, March 1996.

[7] I. T. Jolliffe, Principal Component Analysis, Springer-Verlag, 1986.

[8] K. I. Diamantaras and S. Y. Kung, Principal Component Neural Networks, John Wiley & Sons, 1996.

[9] C. Black, "Artificial Neural Network Sensitivity Analysis and Process Modeling with Application to Kingston Fossil Plant Net Unit Heat Rate Optimization", Masters' Thesis, Univ. of Tennessee, May 1994.

[10] David Lewis, TVA Kingston Fossil Plant, Personal Communication, September 1995.

[11] Paul Hawk, NeuralWare Inc., Personal Communication, August 1996.

APPENDICES

APPENDIX A

LIST OF ORIGINAL VARIABLES

Variable Number	Signal Channel Code	Signal Description
1	A000	Generator Load
2	A001	Station Service
3	A002	Main Steam Pressure @ Boiler Outlet
4	A003	Main Steam Pressure @ Turbine
5	A004	HP Turbine, First Stage Pressure
6	A005	Cold Reheat Pressure @ Turbine
7	A006	Cold Reheat Pressure @ Boiler Inlet
8	A007	Hot Reheat Pressure @ Boiler Outlet
9	A008	Hot Reheat Pressure @ Turbine Inlet
10	A009	#3 Feedwater Heater Inlet Pressure
11	A010	Feedwater Pressure At Economizer
12	A011	Condenser Backpressure
13	A012	Extraction #1 Pressure
14	A013	Extraction #2 Pressure
15	A014	Heater 3 Shell Pressure
16	A015	Heater 2 Shell Pressure
17	A016	Heater 1 Shell Pressure
18	A017	Boiler Drum Pressure
19	A018	Cross Over Pressure
20	A019	Cross Under Pressure
21	A024	RH Furnace Draft Pressure
22	A025	SH Furnace Draft Pressure
23	A040	Main Steam Dp
24	A041	Reheat Dp
25	A043	Waterbox DP, West
26	A044	Waterbox DP, East
27	A045	Outlet DP, West
28	A046	Outlet DP, East
29	A047	SH Furn H.T. Superheater DP
30	A048	SH Furn L.T. Superheater DP
31	A049	RH Furn L.T. Superheater DP
32	A050	RH Furn Reheater DP
33	A060	Main Steam Temperature @ Boiler Outlet
34	A061	Cold Reheat Temperature @ Boiler Inlet
35	A062	Hot Reheat Temperature @ Boiler Outlet
36	A063	Feedwater Temperature @ Economizer
37	A064	Outside Air Temperature (inlet)
38	A065	Air Temperature Entering Air Heater A

Variable Number	Signal Channel Code	Signal Description
39	A066	Air Temperature Entering Air Heater B
40	A067	Gas Temperature Leaving Air Heater A
41	A068	Gas Temperature Leaving Air Heater B
42	A069	Air Temperature Leaving Air Heater A
43	A070	Air Temperature Leaving Air Heater B
44	A071	Crossover Chamber Upper Surf. Metal T
45	A072	First Stage Inner Surface Metal Temp
46	A073	Gas Temperature Entering Air Heater A
47	A074	Gas Temperature Entering Air Heater B
48	A075	Gas Temperature Leaving RH Furnace
49	A076	Gas Temperature Leaving SH Furnace
50	A078	Thrust Brg Temperature, Front Plate Btm
51	A079	Thrust Brg Temperature, Front Plate Top
52	A080	Thrust Brg Temperature, Rear Plate Btm
53	A081	Thrust Brg Temperature, Rear Plate Top
54	A082	#1 Bearing Metal Temperature, Rear
55	A083	#1 Bearing Oil Temperature
56	A084	#2 Bearing Metal Temperature, Rear
57	A085	#2 Bearing Oil Temperature
58	A087	#3 Bearing Oil Temperature
59	A088	#4 Bearing Metal Temperature, Rear
60	A089	#4 Bearing Oil Temperature
61	A090	#5 Bearing Metal Temperature, Rear
62	A091	#5 Bearing Oil Temperature
63	A092	#6 Bearing Metal Temperature, Rear
64	A093	#6 Bearing Oil Temperature
65	A094	#7 Bearing Metal Temperature, Rear
66	A095	#7 Bearing Oil Temperature
67	A096	#8 Bearing Metal Temperature, Rear
68	A097	#8 Bearing Oil Temperature
69	A110	O2 In Flue Gases, RH Furnace
70	A111	O2 In Flue Gases, SH Furnace
71	A113	Feed Rate, Coal Scale A
72	A114	Feed Rate, Coal Scale B
73	A116	Feed Rate, Coal Scale D
74	A117	Feed Rate, Coal Scale E
75	A119	Burner Tilt RF, RH Furnace
76	A125	Burner Tilt LF, SH Furnace
77	A126	Burner Tilt LR, SH Furnace

Variable Number	Signal Channel Code	Signal Description
78	A127	Shaft Speed, Tachometer
79	A128	Vibration, #1 Bearing
80	A129	Vibration, #2 Bearing
81	A130	Vibration, #3 Bearing
82	A131	Vibration, #4 Bearing
83	A132	Vibration, #5 Bearing
84	A133	Vibration, #6 Bearing
85	A134	Vibration, #7 Bearing
86	A135	Vibration, #8 Bearing
87	A160	HP Turbine Inlet Temp - Actual
88	A161	Cold Reheat Temperature @ Turbine 1
89	A162	Cold Reheat Temperature @ Turbine 2
90	A165	North Concentrate Temperature
91	A166	South Concentrate Temperature
92	A167	Heater 3 Feedwater Inlet Temperature
93	A168	Heater 3 Feedwater Outlet Temperature
94	A169	Heater 2 Feedwater Outlet Temperature
95	A170	Heater 1 Feedwater Outlet Temperature
96	A173	CCW East Inlet Temperature
97	A174	CCW West Inlet Temperature
98	A175	CCW West Outlet 1 Temperature
99	A176	CCW West Outlet 2 Temperature
100	A177	CCW West Outlet 3 Temperature
101	A178	CCW West Outlet 4 Temperature
102	A179	CCW East Outlet 1 Temperature
103	A180	CCW East Outlet 2 Temperature
104	A181	CCW East Outlet 3 Temperature
105	A182	CCW East Outlet 4 Temperature
106	A183	Extraction 1 Temperature
107	A184	Extraction 2 Temperature
108	A185	Heater 3 Drain Temperature
109	A186	Heater 2 Drain Temperature
110	A187	Heater 1 Drain Temperature
111	A188	Cross Over 1 Temperature
112	A189	Cross Over 2 Temperature

APPENDIX B

EQUATIONS FOR CALCULATED QUANTITIES OF HEAT RATE

(NET, GROSS, NET UNIT, AND GROSS UNIT)

Heat Rate

(TFB 6/10/93 version)

Heat rate is the energy supplied to the unit (or turbine cycle) divided by the electrical output of the unit. It is the most commonly used measure of overall efficiency.

OLDMS calculates heat into the turbine cycle from steam flows, temperatures and pressures, and divides heat input by load to get turbine cycle heat rate. The boiler efficiency module uses heat input to calculate boiler efficiency, which the heat rate module divides into turbine cycle heat rate to get unit heat rate. Note that by convention in TVA, turbine cycle heat rate includes heat supplied to building heat, sootblowing steam, etc., as well as main and reheat steam.

Inputs:

Temperatures, degrees F
main steam at boiler outlet, Tms
hot reheat steam at boiler outlet, Thrh
cold reheat steam at boiler inlet, Tcrh
Feedwater at boiler inlet, Tfw
sootblowing steam, Tsb
blowdown, Tbd
building heating steam, Tbh
reheat attemperation, Trha
superheat attemperation, Tsha
auxiliary steam from primary superheater (Allen only), Tpsh

Pressures, psia
main steam at boiler outlet, Pms
hot reheat steam at boiler outlet, Phrh
cold reheat steam at boiler inlet, Pcrh
feedwater at boiler inlet, Pfw
sootblowing steam, Psb
blowdown, Pbd
building heating steam, Pbh
reheat attemperation, Prha
superheat attemperation, Psha
auxiliary steam from primary superheater (Allen only), Ppsh

Flows, pounds per hour
main steam, Mms
hot reheat steam, Mhrh
sootblowing steam, Msb
blowdown, Mbd
building heating steam, Mbh
reheat attemperation, Mrha
superheat attemperation, Msha
auxiliary steam from primary superheater (Allen only), Mpsh

Other
boiler efficiency in percent, boiler_eff
gross load in kw, Wg
station service in kw, SS

Intermediate Values:

Enthalpies determined from ASME Steam Tables, Btu/lbm
main steam, Hms
hot reheat steam, Hhrh
cold reheat steam, Hcrh
feedwater, Hfw
sootblowing steam, Hsb
blowdown, Hbd
building heating steam, Hbh
reheat attemperation, Hrha
superheat attemperation, Hsha
auxiliary steam from primary superheater (Allen only), Hpsh

Other
heat added to turbine cycle in Btu/hr, Qt

Outputs:

Gross turbine cycle heat rate in Btu/kwh, GTHR
Gross unit heat rate in Btu/kwh, GUHR
Net turbine cycle heat rate in Btu/kwh, NTHR
Net unit heat rate in Btu/kwh, NUHR

Calculations:

Enthalpies are calculated using Norris Labs' ASME Steam Table routines. Superheated steam and compressed liquid enthalpies are computed using pressure and temperature as inputs. Saturate steam and liquid enthalpies are calculated using pressure.

Qt, heat added to the turbine cycle, is computed from mass and energy balances around the boiler. Most flows are measured or assumed, with the exception of feedwater and cold reheat.

Feedwater flow, Mfw, is calculated from a mass balance around the economizer, drum, and superheater flow streams:

$$Mfw + Msha = Mms + Msb + Mbd + Mbh + Mpsh, \text{ or}$$

$$Mfw = Mms + Msb + Mbd + Mbh + Mpsh - Msha$$

Cold reheat flow is computed from a mass balance around the reheater:

$$Mcrh + Mrha = Mhrh$$

$$Mcrh = Mhrh - Mrha$$

The energy balance to compute heat input to the turbine cycle is:

$$Qt + (Mfw * Hfw) + (Mcrh * Hcrh) + (Msha * Hsha) + (Mrha * Hrha) = (Mms * Hms) + (Mhrh * Hhrh) + (Msb * Hsb) + (Mbd * Hbd) + (Mbh * Hbh) + (Mpsh * Hpsh)$$

$$Qt = (Mms * Hms) + (Mhrh * Hhrh) + (Msb * Hsb) + (Mbd * Hbd) + (Mbh * Hbh) + (Mpsh * Hpsh) - (Mfw * Hfw) - (Mcrh * Hcrh) - (Msha * Hsha) - (Mrha * Hrha)$$

By substituting the equations for Mfw and Mcrh into the above equation for Qt, the following alternative expression is obtained:

$$Qt = Mms * (Hms - Hfw) + Mhrh * (Hhrh - Hcrh) + Msb * (Hsb - Hfw) + Mbd * (Hbd - Hfw) + Mbh * (Hbh - Hfw) + Mpsh * (Hpsh - Hfw) + Msha * (Hfw - Hsha) + Mrha * (Hcrh - Hrha)$$

Divide heat input by generator load to get gross turbine cycle heat rate:

GTHR = Qt / Wg

Divide heat input by net load, Wg - SS, to get net turbine cycle heat rate:

NTHR = Qt / (Wg - SS)

Unit heat rate is turbine cycle heat rate divided by boiler efficiency expressed as a fraction, so gross unit heat rate is

GUHR = GTHR / (boiler_eff / 100)

And net unit heat rate is:

NUHR = NTHR / (boiler_eff / 100)

The expressions for unit heat rate (i.e., turbine cycle heat rate divided by boiler efficiency) are not obvious. The next section derives the equations for unit heat rate. The symbol W will be used for load in kw; substitute Wg for W in the following equations for gross heat rates, and (Wg - SS) for W for net heat rates.

Let Mc = coal rate in lbm per hour
hhv = coal higher heating value in Btu/lbm
Q = sum of boiler losses in Btu/lbm of fired fuel
L = sum of boiler losses in percent

Unit heat rate is defined as follows:

UHR = heat supplied in fuel / electrical output, or

UHR = Mc * hhv / W

The Boiler Efficiency Module defines boiler efficiency as:

Boiler_eff = 100 - L

The Boiler Efficiency Module also gives the relationship between Q and L:

Q = (L / 100) * hhv

An energy balance on the boiler, also described in the Boiler Efficiency Module,

gives:

$$Mc * hhv = Qt + Q * Mc$$

Substituting for Q in the last equation gives:

$$Mc * hhv = Qt + ((L / 100) * hhv) * Mc$$

Dividing both sides of the equation by (Mc * hhv) gives:

$$1.0 = Qt / (Mc * hhv) + (L / 100), \text{ or}$$

$$1.0 - (L / 100) = Qt / (Mc * hhv), \text{ or}$$

$$100 - L = (Qt / (Mc * hhv)) * 100$$

Using the definition of boiler efficiency given above:

$$boiler\_eff = (Qt / (Mc * hhv)) * 100$$

Turbine cycle heat rate is:

$$THR = Qt / W, \text{ or}$$

$$Qt / W = THR$$

Dividing both sides by the reciprocal of boiler efficiency:

$$(Qt / W) / boiler\_eff = THR / boiler\_eff$$

Substituting for boiler efficiency on the LHS:

$$(Qt / W) / ((Qt / (Mc * hhv)) * 100) = THR / boiler\_eff, \text{ or}$$

$$Mc * hhv / W = THR / (boiler\_eff / 100)$$

The LHS is the definition of unit heat rate, so

$$UHR = THR / (boiler\_eff / 100)$$

The relationship is valid for both net and gross heat rates.

APPENDIX C

SAMPLE OF ORIGINAL VARIABLES DATA

Variable Number	TIME (h:mm:ss) 0:00:00	0:01:00	0:02:00	0:03:00	0:04:00	0:05:00
1	194.539	193.852	193.672	194.228	193.754	193.762
2	11.89031	11.8758	11.7664	11.8788	11.906	11.9331
3	1829	1823	1821	1823	1822	1822
4	1834	1828.3	1825.4	1828.8	1827.7	1828.7
5	1301.1	1297.2	1295	1297.9	1296.9	1297.6
6	389.3	388	387.2	388.7	388.2	388.1
7	395	394	394	393	394	393
8	355	354	354	354	354	354
9	352.1	350.8	350	351.4	351.1	351
10	1990.9	1985.8	1994	1985.3	1984.3	1985.8
11	1974	1969	1967	1969	1968	1969
12	1.9	1.9	1.9	1.9	1.9	1.9
13	627.3	625	623.9	626.5	625.7	625.9
14	386.5	385.1	384.3	385.9	385.5	385.4
15	241.7	240.8	240.3	241.2	241.1	240.8
16	386	384.6	383.9	385.4	385	385
17	624.1	621.6	620.6	623.1	622.3	622.5
18	1982	1978	1975	1978	1978	1976
19	36.5	36.2	36.1	36.3	36.4	36.2
20	242.7	241.4	241	242	241.8	241.7
21	-0.6	-0.5	-0.6	-0.5	-0.4	-0.4
22	-0.5	-0.5	-0.5	-0.4	-0.3	-0.5
23	1303	1298	1295	1298	1297	1294
24	1122	1109	1113	1123	1110	1115
25	9.6	9.7	10.2	9.7	9.5	9.3
26	9.3	9.4	9.2	9.2	9	9.5
27	23.6	21.9	22.3	23.2	23.1	23.3
28	18.2	17.3	18	18.5	17.8	18.6
29	0.9	0.9	0.9	0.9	0.8	0.9
30	1.7	1.7	1.8	1.7	1.7	1.7
31	1.8	1.8	1.8	1.8	1.8	1.8
32	1.2	1.2	1.2	1.2	1.2	1.2
33	637	637	638	637	637	638
34	480	480	480	480	471	480
35	92	92	92	92	92	91
36	131	130	130	131	129	130
37	131	131	130	131	131	131
38	336	336	335	336	336	336

39	337	337	337	338	337	337
40	563	564	563	564	564	563
41	589	587	589	588	588	588
42	943	943	943	943	943	943
43	1402	1402	1402	1402	1402	1402
44	642	642	642	641	642	642
45	654	654	653	654	653	653
46	643	643	643	643	643	643
47	186	186	186	186	186	186
48	192	193	192	192	193	192
49	146	146	146	146	146	146
50	147	147	147	147	147	147
51	176	178	175	177	177	176
52	166	166	166	166	166	166
53	164	164	164	163	163	163
54	136	136	136	136	136	136
55	140	140	140	140	140	140
56	146	146	146	146	146	146
57	132	132	132	132	132	132
58	150	150	150	150	150	150
59	136	136	136	136	136	136
60	187	187	187	187	187	187
61	133	133	133	133	133	133
62	171	171	171	171	171	171
63	138	138	138	138	138	138
64	189	189	189	189	189	189
65	136	136	136	136	135	136
66	2.4	2.4	2.4	2.4	2.4	2.4
67	2.6	2.8	3	2.8	2.6	2.9
68	23086	23115	23115	22998	23145	23085
69	25044	25220	25259	25184	25044	25044
70	24418	24418	24418	24253	24253	24220
71	25081	25081	25081	25081	25081	25151
72	14	14	14	14	14	14
73	-9	-9	-9	-9	-9	-9
74	3598	3597	3597	3598	3597	3597
75	1	1	1	1	1	1
76	2.2	2.1	2.1	2.2	2.1	2.1
77	3.1	3.2	3.2	3.2	3.2	3.2
78	1.4	1.4	1.4	1.4	1.4	1.4
79	1.7	1.7	1.7	1.7	1.7	1.7

80	1.4	1.4	1.4	1.4	1.4	1.4
81	1.6	1.6	1.6	1.6	1.6	1.6
82	1002.4	1002.4	1002.5	1002.7	1002.8	1002.9
83	635.8	635.8	635.8	635.8	635.8	635.9
84	649.7	649.8	649.8	649.7	649.7	649.8
85	88	88	88	88	88	88
86	101	101	101	101	101	101
87	329.1	329.2	330	329.2	329.1	329.2
88	381.3	381.2	381.2	381.1	381.2	381.1
89	435	434.9	434.7	434.8	434.8	434.9
90	471.8	471.7	471.6	471.5	471.6	471.5
91	74	74	74	73	73	73
92	74	74	74	74	74	74
93	89	89	89	89	89	89
94	89	89	89	89	89	89
95	88	87	87	87	87	87
96	88	88	88	88	88	88
97	87	87	87	87	87	87
98	84	84	84	84	84	84
99	90	90	90	90	90	89
100	89	89	89	89	89	89
101	768.2	768.3	768.3	768.4	768.4	768.4
102	663.1	663.2	663.2	663.2	663.2	663.2
103	358.3	358.3	358.3	358.3	358.3	358.3
104	199.2	199.1	199.5	199.1	199.5	199.3
105	455.3	455.4	455.3	455.1	455.1	455.1
106	522.5	522.7	522.7	522.6	522.5	522.4
107	513.4	513.6	513.5	513.4	513.3	513.3

APPENDIX D

SAMPLES OF PRINCIPAL COMPONENT DATA

Principal Component	TIME (hr:mm:ss) 0:00:00	0:01:00	0:02:00	0:03:00	0:04:00	0:05:00
1	10.68	9.33	9.04	9.45	8.24	8.84
2	6.24	9.08	8.88	5.37	7.54	4.48
3	21.30	14.87	12.28	13.35	14.37	12.78
4	-12.12	-8.58	-8.29	-8.31	-9.49	-8.12
5	0.22	0.24	0.23	0.22	0.26	0.23
6	-0.73	-2.96	1.83	0.41	-5.18	-1.21
7	-0.73	-0.60	-0.64	-0.69	-0.62	-0.65
8	8.25	11.89	18.05	3.78	-4.53	0.42
9	-14.71	-11.41	-10.47	-13.59	-9.83	-13.52
10	14.96	15.03	14.93	16.08	12.82	18.95
11	4.43	0.52	-1.29	3.11	1.96	5.35
12	2.75	0.84	-2.82	2.97	-10.25	16.93
13	-1.38	10.99	-16.05	-8.42	0.00	8.69
14	-11.89	19.98	17.09	-4.83	-3.36	-11.88
15	6.02	-2.15	-1.69	-11.45	-30.52	-8.78
16	5.59	4.01	11.13	4.13	9.23	7.18
17	8.51	-0.96	-1.19	-7.70	-10.76	-5.88
18	-2.71	-0.79	0.70	13.16	12.93	-6.09
19	1.67	-0.62	-7.05	1.98	1.92	1.09
20	-5.89	-17.66	-18.90	-13.08	-11.13	-12.98
21	-5.02	-14.03	-16.31	-9.79	-7.12	-9.59
22	-9.80	-4.93	-4.92	-8.79	-12.46	-17.94
23	-12.26	-9.78	-12.67	-10.48	8.85	-8.50
24	6.43	6.34	8.94	2.42	10.13	10.05
25	0.75	0.78	0.86	0.74	0.67	0.67
26	16.43	11.14	18.77	2.76	2.39	-13.63
27	-2.37	-17.25	-18.48	-15.58	-14.86	-19.81
28	-39.12	-40.09	-34.66	-26.60	-27.32	-19.02
29	-5.02	-9.79	-9.22	18.12	17.79	24.13
30	-14.16	-15.48	-8.75	-5.66	-9.16	-17.46
31	-0.01	-4.15	-4.75	-4.53	56.94	-6.28
32	7.70	2.37	8.01	-3.33	-1.01	11.95
33	3.49	3.88	5.61	6.15	1.18	7.93
34	-8.04	-8.17	-9.90	11.56	11.23	17.90
35	-10.56	-16.95	-13.10	-8.07	-7.96	-6.31
36	-3.54	-0.46	-2.12	1.01	0.96	5.62
37	-2.85	-4.36	-3.64	-3.06	-2.08	-3.88
38	1.40	-0.95	-1.10	0.21	0.62	0.48

39	2.01	-6.81	-5.30	14.69	-1.31	11.72
40	-2.01	-1.59	-2.25	-1.77	-2.35	-0.32
41	-9.59	17.15	7.04	-14.67	14.39	3.00
42	3.01	3.87	2.52	-3.89	-1.89	4.10
43	-0.75	-1.10	-1.16	-0.77	-0.70	-0.89
44	0.20	-4.97	1.05	1.75	-4.61	-2.19
45	0.11	-0.59	0.07	-1.37	-4.23	-2.26
46	-1.00	-2.20	0.88	-0.14	-0.82	-0.23
47	11.46	7.85	-0.03	1.55	8.00	-6.12
48	-1.00	-1.02	-1.37	0.18	-7.42	-1.25
49	1.15	-2.50	-5.47	-6.49	-0.92	-8.97
50	1.27	1.30	1.18	1.41	1.29	1.67
51	-1.80	-1.16	-0.91	-1.88	-1.71	-2.15
52	-0.93	0.12	-2.02	-1.81	-1.00	-0.80
53	-9.65	1.76	-6.87	-11.91	0.20	-2.88
54	0.31	-0.57	-0.75	0.07	0.42	0.63
55	0.22	0.41	0.42	0.22	-0.04	0.13
56	-6.17	-6.34	-3.35	-16.89	2.03	-7.18
57	2.97	3.74	6.56	-4.04	-2.73	-6.67
58	-2.87	1.41	3.63	-14.07	-1.64	-4.30
59	-2.39	-2.40	-2.18	-5.95	-2.04	-2.43
60	-6.52	-5.63	-8.45	-3.15	2.97	1.65
61	0.56	1.23	3.42	2.03	3.86	0.93
62	-9.92	-12.28	0.21	-13.40	-11.81	-8.44
63	-7.08	0.41	-1.47	-9.26	4.06	-5.07
64	-5.14	0.39	-4.05	2.66	5.53	2.06
65	-10.63	-6.37	-9.31	-11.57	-5.70	-8.61
66	-1.63	2.82	2.90	-2.28	-1.46	0.04
67	1.25	-5.37	-6.82	1.11	-1.76	-2.29
68	-6.74	-4.18	-5.73	-9.27	-10.05	-6.03
69	-8.03	-6.32	-5.81	5.57	7.74	7.91
70	-2.50	-6.88	-4.53	-1.18	-3.75	-4.53
71	-9.23	0.60	5.24	-16.63	-7.79	1.11
72	-5.54	4.35	-2.31	-1.72	8.29	-3.78
73	-0.77	-5.47	1.94	-8.12	-2.88	-14.08
74	30.58	15.95	19.48	13.03	-1.07	2.76
75	12.60	9.65	5.37	4.24	1.57	3.16
76	10.86	11.83	5.24	15.84	12.45	17.39
77	-10.89	-18.81	-8.17	-6.75	-1.72	-0.06
78	-6.20	-5.58	-6.22	-10.53	-10.28	-4.43
79	-7.50	-16.36	-4.24	-11.09	-11.52	-5.53

80	2.40	0.76	0.60	2.43	9.02	1.61
81	-16.88	-18.91	-17.95	-18.88	-30.10	-19.88
82	-4.43	13.95	-4.90	-4.19	11.13	-2.30
83	-7.22	4.67	-4.38	-3.73	6.42	-2.62
84	22.33	15.66	18.71	18.72	25.38	30.13
85	6.64	-10.78	7.31	8.36	-11.34	8.43
86	1.91	-5.23	3.41	6.56	-9.58	-7.32
87	8.60	14.75	12.51	12.17	20.28	6.39
88	11.48	16.55	11.46	11.94	14.18	6.79
89	-5.81	-5.64	-4.96	-5.41	-4.74	-2.72
90	11.02	0.26	9.60	4.91	-5.71	1.73
91	-7.00	-13.64	-0.78	-17.58	6.40	7.11
92	-12.55	-9.61	-10.77	-11.43	-8.57	-5.60
93	1.39	4.82	5.01	2.05	8.34	3.51
94	-3.39	-12.14	-20.99	-3.01	-5.71	2.26
95	-8.88	-7.51	0.86	-5.38	-10.42	1.71
96	-6.86	-8.08	1.53	-11.45	-7.78	-6.07
97	14.19	8.55	3.18	20.25	0.41	15.81
98	13.22	7.24	8.72	2.09	1.52	-10.77
99	13.12	-16.14	-8.06	-10.44	-13.63	-13.15
100	4.70	-10.02	-8.99	13.96	-5.40	-4.61
101	6.45	9.69	1.48	8.41	24.78	1.82
102	7.31	8.38	12.05	-9.99	-22.36	4.63
103	-1.12	-3.24	-6.81	-8.34	-11.94	-5.03
104	0.14	18.11	21.27	13.68	16.81	14.80
105	9.65	9.83	-1.88	1.01	0.81	10.02
106	-11.38	-11.90	-0.93	-10.72	10.31	-12.16
107	-14.71	-11.24	8.61	-14.65	16.44	-13.14

VITA

Glen Gray Paschal was born in Murfreesboro, Tennessee on February 20, 1966. He graduated from Hixson High School in Chattanooga, Tennessee in June of 1984 and entered The University of Tennessee, Knoxville the following Fall in pursuit of a degree in mechanical engineering. During the Summers of 1985 and 1986, he worked as an engineering aide for the Tennessee Department of Transportation. In the Spring of 1985, he was selected as a Kodak Eastman Scholar and worked for Eastman Kodak during the summer of 1987 as a summer technical employee. He obtained his Bachelor of Science degree in mechanical engineering in June 1988 and went to work for Tennessee Eastman Chemical Company in Kingsport, Tennessee until the Fall of 1994. In the Fall of 1994, he took an educational leave of absence to return to the University of Tennessee and received an INPO fellowship. His research work was in the field of Expert Systems and Neural Networks in the Department of Nuclear Engineering. He expects to receive the Master of Science Degree in December 1996.