WATER DEMAND FORECASTING AND COMPARISON OF NEURAL NETWORK MODELS

By

SIVA RAMA KRISHNA KAVUTURU

Bachelor of Engineering

Osmania University

Hyderabad, India

1990

Submitted to the faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the degree of
MASTER OF SCIENCE
December 1993

OKLAHOMA STATE UNIVERSITY

WATER DEMAND FORECASTING AND

COMPARISON OF NEURAL

NETWORK MODELS

Thesis Approved:

Huizhu Lu

Thesis Adviser

Ramesh Sharda

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGEMENTS

It is a great pleasure to acknowledge my debt to all of my committee members. I wish to express my sincere appreciation and gratitude to my graduate advisor, Dr. Huizhu Lu, for the advice and guidance throughout the study period at Oklahoma State University and in the preparation of this thesis. I would like to thank Professor Ramesh Sharda for his valuable suggestions, guidance and encouragement throughout this study. His constructive criticism helped me in gaining confidence during the thesis preparation. I would like to thank Dr. Paul Benjamin for serving on my graduate committee. His suggestions helped me in preparing this thesis.

I would like to express my grateful appreciation to Professor Ramesh Sharda for his financial support through research assistantship.

This work would not have been possible without the generous support and assistance of many institutions, colleagues and friends. The help of the members of City of Stillwater, OSU Water Pumping Station is gratefully acknowledged. Special thanks to Chandu, Lonnie Hamm and Vish Daita for their help in this research.

Special thanks to my parents, Prasad Rao K., and Bharathy K., brother, Srinivas, and sister for their patience, encouragement and many sacrifices throughout my education.

TABLE OF CONTENTS

LIST OF FIGURES

Figure | Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Water is very important to mankind, plants, and animals. Where water is available, civilizations flourish, settlements are created, and stability is attained. The most important benefit from improving the quality and the quantity of water supplies is the improvement in public health. The effect of water supply on the community health depends on the level of use of the supply, and service provided. Improving water supply is an essential step in industrial and agricultural development and productivity.

For effective water system planning, the needs must be assessed realistically considering the potential variability of water use. Understanding and managing water use can result in substantial savings in capital investments eventually. One way to realize these savings is from precise estimates of future water demand.

In water demand forecasting there are many natural factors that cannot be expressed through the statistical terms. These factors may include the day light hours, temperature, season of the year, etc. Neural networks are able to solve such problems efficiently. Neural networks also have the capability of expressing the natural phenomena.

The next two sections describe the forecasting problem and the neural networks and their advantages and disadvantages.

Forecasting

Forecasting is a major part of decision making for all organizations. Forecasting methods can be classified into two categories: causal and time series methods. The time series methods differ from the causal methods in that they do not assume the cause and effect relationship between input and output variables. Whatever forecasting method is used, the main objective for all forecasters is to get a precise forecast.

In time-series forecasting using regression the problem of prediction can be generalized in the following manner:

$\mathbf{Y}(t+1) = \mathbf{F}(X(t), X(t-1), X(t-2), \ldots, E(t))$

where

$\mathbf{Y}(t+1)$ is the forecasted value of $X(t+1)$ using the past values of X at times t, t-1, t-2, . . . and error term $E(t)$. In this model the main aim is to decrease the value of the error term $E(t)$ when forecasting $\mathbf{Y}(t+1)$.

Neural Networks

According to Robert Hetch-Neilsen (1989), Artificial Neural Network (ANN) is ". . . a computing system made up of a number of simple, highly interconnected processing elements, which processes information by its dynamic state response to external inputs."

Artificial neural networks ("neural nets") go by many names such as parallel distributed processing models, connectionist models, neuromorphic systems. In a serial computer, everything happens in a deterministic sequence of operations. In contrast, a neural network is neither sequential nor, even necessarily, deterministic. It is composed of many simple processing elements that usually process a weighted sum of all their inputs. Instead of executing a series of instructions, a neural network responds, in parallel, to the inputs given to it. The final result consists of the complete state of the network after it has reached a steady-state condition, which correlates patterns between the sets of input data and corresponding output or target values. The final network can be used to predict outcomes from the new input data. The goal of the neural networks is to achieve good forecasting performance by densely interconnecting many processing elements. In this aspect the neural networks imitate the human nervous system. Neural networks have been used in the fields of speech recognition and image processing where high parallel computation is completely exploited, and the present technology could not outperform human nervous system.

Neural networks are described in terms of input, output, and hidden neurons, a transfer function, learning or training rules etc. These learning rules specify how to setup and modify weights to improve the performance.

Processing elements used in neural nets are nonlinear, analog and slower than the digital circuits. A processing element sums all its weighted inputs and passes out nonlinearly transformed result. Examples of nonlinear transfer functions include sigmoid, gaussian, step, and threshold logic.

Adaptation of learning is a major advantage of neural networks. This attribute plays an important role in speech recognition where training data is limited and new speakers, new phrases, and new dialects are encountered continuously. Neural nets are robust in that they compensate even minor variabilities in the characteristics of processing elements. Artificial neural networks make no assumptions about the distribution of data in contrast to traditional statistical classifiers. Thus neural networks prove to be better when the data to be modeled may have been generated by a nonlinear source [Lippmann, 1987].

As the ANNs are fault tolerant, the loss of a few processing elements often will not degrade the performance of the whole network. ANNs are found to be effective in the areas where the conventional approaches are not adequate or are difficult to implement but precision is not a critical measure. Some examples include speech and pattern recognition. Neural networks are not known to provide a high degree of accuracy. However, Lapedes and Farber have shown that a multilayer network trained with backpropagation yielded higher numerical accuracy than conventional methods for prediction.

Statement of the Problem

The problem of prediction can be generalized in the following manner:

$\mathbf{Y}(t+1) = \mathbf{F}(X(t), E(t))$

Where,

$\mathbf{Y}(t+1)$ is the forecasted value of $\mathbf{X}$ at time t+1, by observing values of X at times 1, . . ., t.

$\mathbf{F}$ is the vector valued function containing an error term $E(t)$.

$X(t)$ is the vector of inputs for observation at time t.

The neural network models that are considered in this thesis, backpropagation, Jordan network and Experimental network, differ in defining the vector valued function, **F**, which minimizes the error term **E**(t) in predicting **Y**(t+1).

Smolensky (1986) specifies a dynamic feed-forward network in the form of:

$u_i(t+1) = F[\Sigma_k W_{ki} G(u_k(t))]$

Where

$u_i(t)$ is the activation of the unit i at time t,

F is a nonlinear sigmoid transfer function,

G is nonlinear threshold function and

W_{ki} is the connection strength or weight from unit k to unit i.

This relationship can be used to forecast the future values. The main aim of the forecasting problem is to find the function F that minimizes the error in $u_i(t+1)$.
The objective of this study is to build three adaptive systems that learn to estimate the future values and compare them with respect to their precision of estimation and the time consumed to build that model.

In this thesis the $u_i(t)$ is a univariate time series. The water pumpage is used as the estimate of the water demand. As the data is a real world one there is a possibility to encounter the worst case performance, even though the performance depends on many other factors that include number of hidden neurons, the learning parameters, transfer function and the history that we are providing to forecast the future value. There are other ways to build the adaptive system using a simulated data that may not capture and explain all the features of the adaptive system built.

Objectives of the Study

1. Develop and implement the neural network models for water demand forecasting using traditional forecasting methods, backpropagation algorithm, and recurrent backpropagation algorithm.
2. Identify and analyze the problems that are affecting the performance of the neural network models
3. Compare and contrast the models with each other.
4. Analyze the effect of input window on the performance at different lead times (i.e., 1, 4, 12, and 52 weeks ahead).

CHAPTER II

LITERATURE REVIEW

One of the main problems affecting the management of water distribution system is the forecasting of consumer demands ahead of time. It is particularly important that short-term forecasts of water demand be accurately estimated in order that minimum-cost pumping schedules may be computed. With the increasing federal role in all aspects of water resource planning, the need for a planning model to enable calculations of projected municipal water requirements has become evident.

The following two sections describe the papers which dealt with water demand forecasting and comparisons of various methods in neural networks.

Water Demand Forecasting

Results from a forecast simulation (Miaou, S. P., 1986) indicate that the adaptive models do provide slightly better forecast performance for smaller lead times. But the improvement tends to decline as the lead-time increases, and the adaptive model eventually performs worse than the constant parameter models.

In extreme arid environment, requirement approach is more appropriate in forecasting water use. However, variations in the socioeconomic characteristics that were predictors of residential water use for major urban areas (of Israel) were not associated with variations in residential water consumption. Only technical water-using appliance specifications were found to be correlates (Darr et al., 1975).

For the purpose of on-line control of a water supply distribution system, it is necessary to obtain one-step ahead prediction, on a short-term basis of hourly or daily demands. In their study, Chen et al. (1988), found that the water demand could be expressed using the difference equation known as the ARIMA model. The time series of hourly demand was determined from two methods: one was formed by considering each following hour of the same day, another was formed by considering the same hour for each the following day. Time series analysis proved suitable for the modeling and short-term predicting of water demands. However, it is felt that higher precision could be obtained by combining the two methods for predicting hourly demands.

In order to provide more efficient estimates and eliminate potential ordinary least squares bias in single equation models of demand for goods sold through block rate pricing systems because of possible price endogeneity, Chicone et al., (1986) employed an estimation technique incorporating the error structure using the three-stage-least-squares.

Burke, Thomas R. (1970), developed an economic model of municipality water requirements that incorporates variables reflecting the various factors affecting water demand (demographic, social, and industrial).

Pooled time-series models are more demanding in their structure and data requirements, but often provide better estimates of the impact of price variables than simple time-series analysis. Combining time-series data with cross-sectional attributes can be a very effective method of analysis for forecasting and measuring price elasticity. It isolates the differential response among cross-sectional characteristics that are obscured in pure time-series analysis. At EBMUD, significant elasticities, in the range of -0.1 to -0.25, were identified for summer months, resulting primarily from the implementation of elevation surcharges of 25% to 50% for two elevation bands (Weber, 1989).

A typical municipal water supply planning problem requires a finely-scaled mathematical model capable of handling uncertainty in both supply and demand to optimize the investment in municipal water supply facilities. Hughes et al., (1973), proposed a branch bound concept of mixed integer approach that overcomes many of the limitations of linear programming. It selects the optimum combination of source-facilities from all possible alternatives.

Auto Regressive Integrated Moving Average (ARIMA) models were studied for daily and monthly demand predictions in the distribution networks of Barcelona, Spain. Though the predictions were initially good, as soon as a large deviation of demand from the predicted value occurred, the situation deteriorated. A periodic seven-day pattern was observed in the prediction errors, which showed a non-decreasing sequence of errors in successive seven-day periods after an abnormally large prediction error. In the stochastic model, a seasonal 12-month pattern with no apparent changes in the variance values between different periods was observed. The mean value tended, however, to increase from one period to the other, a fact that suggested non-stationary of the series confirmed by the study of the autocorrelation functions. Hourly forecasts were derived from daily predictions through the use of average load allocation curves (Quevedo et al., 1988).

Budanaers (1976) proposed a dual set of short-term water demand models, one an extension of the Box-jenkins model and the other a weather component of demand model. These models have the feature of adaptability to changing data: given changes in the data sequence, the models' parameters will self-adjust to provide a better model. The models also have the property of being real-time computer implementable.

A multivariate water forecasting procedure that is simple to implement has been used to estimate the water demand for a proposed subdivision in Barrie, Ontario. For comparison a trend forecasting procedure is also applied. Both the techniques provide accurate results when compared to actual use. However, the multivariate analysis allows more precision (Mitchell et al., 1977).

Smith (1988), developed an autoregressive time series model with randomly varying mean, which dictates that the key step in producing a water use forecast is an updating step in which a revised estimate of current mean water use is computed.

Forecasting: Comparisons between Neural Networks and Traditional Methods

Connor and Atlas (1991) tested recurrent neural networks in forecasting time series. The results also proved that the recurrent networks have advantages over feed-forward networks for prediction of stochastic process. According to their results the recurrent networks performed better when the number of input neurons is more than five.

Comparisons have also been made between feed-forward networks and recurrent networks. According to Su et al. (1992) prediction errors of a feed-forward network increased significantly as prediction horizon becomes large, whereas those of recurrent networks are more constant. They also observed that recurrent networks give consistently better results than a feed-forward network. But feed-forward networks performed better, for one step ahead forecast. They concluded that the RNNs (Recurrent Neural Networks) are better than FFNs (Feed-forward Networks) in long term prediction and multiple step prediction.

Hill et al. (1991), compared time series forecasting ability of neural networks and classical methods. The data is selected from M-competition. The whole 1001 real time series is used to compare the models. The results proved that the neural network models outperformed the classical models in forecasting both monthly and quarterly data series and did about as well as classical models with annual series.

For a single period time series forecasting, Sharda and Patil (1992) have shown that neural networks could be used. Using neural network models and traditional Box-Jenkins forecasting models, a sample of 75 M-Competition data series was tested and compared over annual, quarterly, and monthly time periods. The neural network models performed comparable to Box-Jenkins models.

Tang, Z. et. al., (1990) performed a comparison study in forecasting time series between the Box-Jenkins and feed-forward neural network model. The results proved that with short-term the neural network model outperformed the Box-jenkins model but with long memory both models performed comparably.

Canu et al. (1992) performed a comparison between traditional forecasting methods and a multi-layer perceptron neural network model. The results proved that multi-layer perceptron model containing four layers outperformed the traditional forecasting methods. He observed that the non-linearities of the water demand are followed by the multi-layer perceptron model.

Justification

The recurrent networks are not experimented much in recent studies. Water demand forecasting is not an exception to this. The research of Canu et al. on water demand forecasting is dealing with backpropagation algorithm only. This motivated us to compare various network models. Comparisons are made between two recurrent network models, a backpropagation model, and traditional forecasting models. The main interest is also to compare the performances of two recurrent networks (Jordan net and an Experimental net).

CHAPTER III

DESCRIPTION OF MODELS

Box-Jenkins Model

The Box-jenkins method is well known for time series forecasting. This method consists of four steps: data transformation, model identification, parameter estimation, and diagnostic checking. This method is one of the complex time series modeling methods. The general Box-Jenkins model has the following form:

$$\phi(B)\,\Phi(B^s)\,\nabla^d\,\nabla_s^D\,y_t = \theta(B)\,\Theta(B^s)\,a_t$$

where,

B is the back-shift operator (i.e., $Bx_t = x_{t-1}$)

$\nabla = 1\text{-}B$; s = seasonality; a_t = white noise;

$\phi(B)$ and $\Phi(B^s)$ are seasonal and non-seasonal auto-regressive polynomials respectively.

$\theta(B)$ and $\Theta(B^s)$ are nonseasonal and seasonal moving average polynomials respectively;

y_t is the time-series data, transformed if necessary.

After identification of several alternate models the final model is selected in diagnostics-checking step. Once the final model is selected the forecasting process begins.

Backpropagation

The application of backpropagation algorithm involves two phases. During the first phase the input is presented and propagated forward through the network to compute the output values for each unit. This output is then compared with the target, resulting in an error term δ for each output unit. The second phase involves the backward pass through the network during which the δ term is computed for each unit in the network. The second phase continues until it completes all the hidden layers. Once these two phases are complete the weight changes can be computed and applied for each weight in the network. As there are hidden layers the nonlinear behavior of the data can be followed by the network. The backpropagation can be specified in equation form as:

$\mathbf{W}_{ij}(t+1) = \mathbf{W}_{ij} + \Delta\mathbf{W}_{ij}(t)$

$\theta_j(t+1) = \theta_j(t) + \Delta\theta_j(t)$

$\theta\mathbf{W}_{ij}(t) = (\text{learning rate})\ (\text{error\_derivative})_j\ \mathbf{O}_i$

$\Delta\theta_j(t) = (\text{learning rate})\ (\text{error derivative})_j$

where

$\mathbf{W}_{ij}$ is the weight of connection from neuron i to j,

θ_j is the threshold of the neuron j,

$\Delta\mathbf{W}_{ij}$ is the change in the weight $\mathbf{W}_{ij}$,

$\Delta\theta_j$ is the change in the threshold value of the neuron j.

As there are multiple layers the input to a layer is the output of the previous layer (as in the case of hidden and output layers).

Recurrent Backpropagation

Recurrent backpropagation is an extension of the backpropagation algorithm. In this algorithm the output from the output layer can be fed to any layer and the neurons in the input layer may not necesarily need input from the external sources (may receive input from the output of the output layers). Therefore in this type of network it is difficult to distinguish among input, hidden and output layers.

For example let us consider the Figure 1. In this, the unit 1 and 2 are output units with targets ζ_1 and ζ_2. The units 1, 3 and 5 are input units. The unit 1 acts both as input and output unit while 4 is neither. The error propagation network for this network is shown in Figure 2.

A regularly trained feed-forward neural network responds to a given input pattern with exactly the same output pattern every time the input pattern is presented. A recurrent network may respond to the same input pattern differently at different times, depending upon the patterns that have been presented as inputs just previously. Thus, the sequence of the patterns is as important as the input pattern itself. Recurrent networks are trained just like the regular feed-forward networks except that patterns must always be presented in the same order. The network that we selected has just one extra slab of neurons in the input layer that is connected to the hidden layer just like the other input layer. This extra layer contains one or more of the layers as they existed when the previous pattern was trained. In this, the network is able to see the previous knowledge it had about the previous inputs.

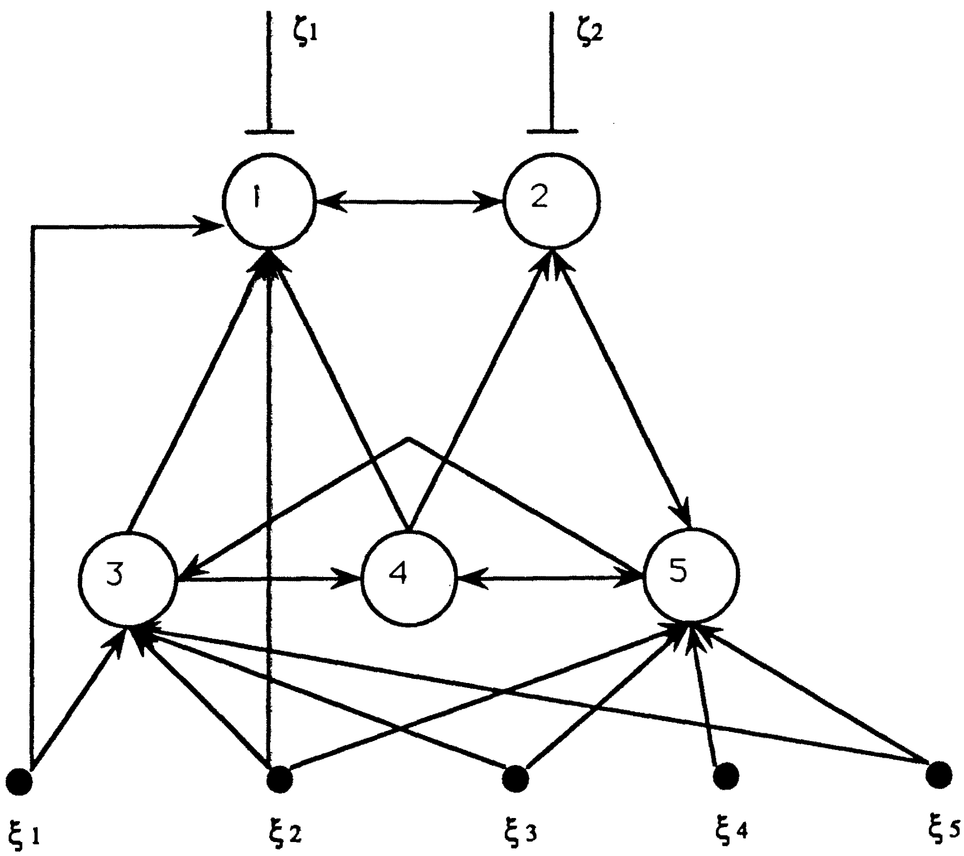


Figure 1

Recurrent Backpropagation Network

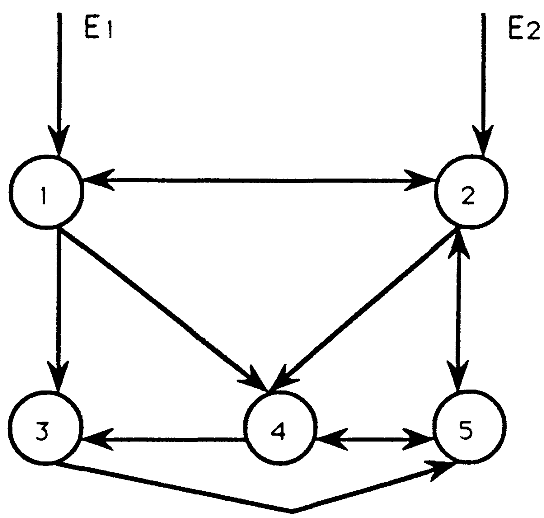


Figure 2

Error Propagation Network For Figure 1

Determinants of Neural Network Performance

Learning:

A neural network is made up of several interconnected processing elements or neurons. Each processing element receives a number of inputs X_i, which are assigned weights W_i. From the weighted total input, the neurons compute a single output signal Y. The following four steps take place when each neuron is activated and processed:

1. Various signals are received from other neurons.
2. A weighted sum of these signals is calculated.
3. The calculated sum is transformed by a function.
4. The transformed signal is sent to other neurons.

Learning process can best be visualized as curve fitting. As the neuron function is fixed, the only way the output from a neuron can be changed due to the input environment is by changing the weights on the inputs.

Factors Influencing Supervised Learning in Neural Networks:

Learning of a network can be broadly classified into two categories. One is supervised learning and the other is unsupervised learning. In supervised learning the network is provided with set of input and target pairs. When the network forecast some value for the present input vector, based on the previous history, the target (actual value to be forecasted) is compared with the predicted value and the error is adjusted. In unsupervised learning the network is supplied with a set of input patterns only. In this case, for example, the neuron that affects the output more will try to change adjust the error in the network.

Training the neural network to find the global minimum can be made easier by using some techniques. One way to find the global minimum is the re-initialization of the weights to a different set. This can be achieved by assigning random weights with different seed value every time. This cannot ensure the global minimum acquisition but may help in finding global minimum.

Too many or too few number of training patterns also confuses the network sometimes. The number of patterns also is very important in deciding the convergence of a network.

Sometimes the network may find a local minimum and keep on oscillating in the same range. The momentum term, if added to delta rule, helps to get rid of such local minimums. But larger momentum term may help the algorithm in skipping the global minimums also.

The neuron transfer function plays a key role in converging the network. If there is non-linearity in the data and the transfer function is a linear one there in no possibility of getting the network converged. The transfer function selection depends on the behavior data also. Some nonlinear transfer functions perform better than the other for some data sets. The sigmoidal transfer function is proving better for most of the problems.

The number of hidden layers also important in getting the network trained rapidly. The number of layers is not fixed for all types of networks. There are some proofs that a network with two hidden layers performs as good as any number of hidden layers. The more number of hidden layers will definitely increase the training time.

The number of neurons in the hidden layer is also an important fact that decides the convergence of a network. There are no formulae to decide the number of neurons depending on the number of the input or output neurons. There is a possibility that a network that can get trained for N neurons may not get trained for M ($\neq$ N) number of neurons. But changing the number of neurons (either decrease or increase in the number of the neurons) may lead to convergence of network.

CHAPTER IV

MODEL DEVELOPMENT AND METHODOLOGY

Data Description

The water pumpage data is gathered from the City of Stillwater and the Oklahoma State University Water Pumping Station. It is important for us to consider the combined water pumpage, given that the OSU Water Pumping Station and the City of Stillwater complement each other to meet local water demand. We have gathered the daily water pumpage data of the past six years (January 1986 - July 1992). The data was later processed to obtain weekly water pumpage data, amounting to 347 data points. To facilitate the analysis of the data we classified the 347 data points into monthly, quarterly, and yearly data values.

The Backpropagation and recurrent backpropagation needs input (or train) and test patterns to train and to evaluate the network respectively. To test a network's performance fifty test patterns are held from each set of patterns. The remaining are fed to the networks as input (or train) patterns. During the training process the patterns are fed in a strict sequential order to the network in every iteration. A maximum of fifty thousand iterations was set to terminate the training and to decide the network's inability to converge. Only one hidden layer is used to train all the models.

In Box-Jenkins (1976) model there is no necessity to classify the data into monthly, quarterly and annual series. The forecast for the n^{th} period is made, based on the history of n-1 data points. There are no input or test patterns in this method. This model tries to understand the data series and determines the coefficients of the forecast equation that it uses in forecasting the next step or steps. This model proved to be performing better than several other approaches in one step ahead forecast in many previous comparisons made with neural network models.

Backpropagation

The internal details of the backpropagation neural network model is shown in Figure 3. The first layer or the input layer consists of processing elements or neurons which can fan out to N processing elements. The elements in this layer take in individual components of the input vector. In this study, the input layer consists of 4, 12 or 52 neurons. The second layer, hidden layer, receives signals from the input layer and after transformation passes them on to the output layer. In all our current models the output layer consists of only one neuron. The hidden layer consists of 4, 15, or 60 hidden neurons for the input neurons of 4, 12 or 52, respectively. The momentum term is set to 0.8 and the learning rate is set to 0.75. The models are tried with only one hidden layer.

Recurrent Backpropagation Networks

The two recurrent backpropagation networks considered are Jordan network and Experimental network. The Jordan network is one of the well known architectures of the

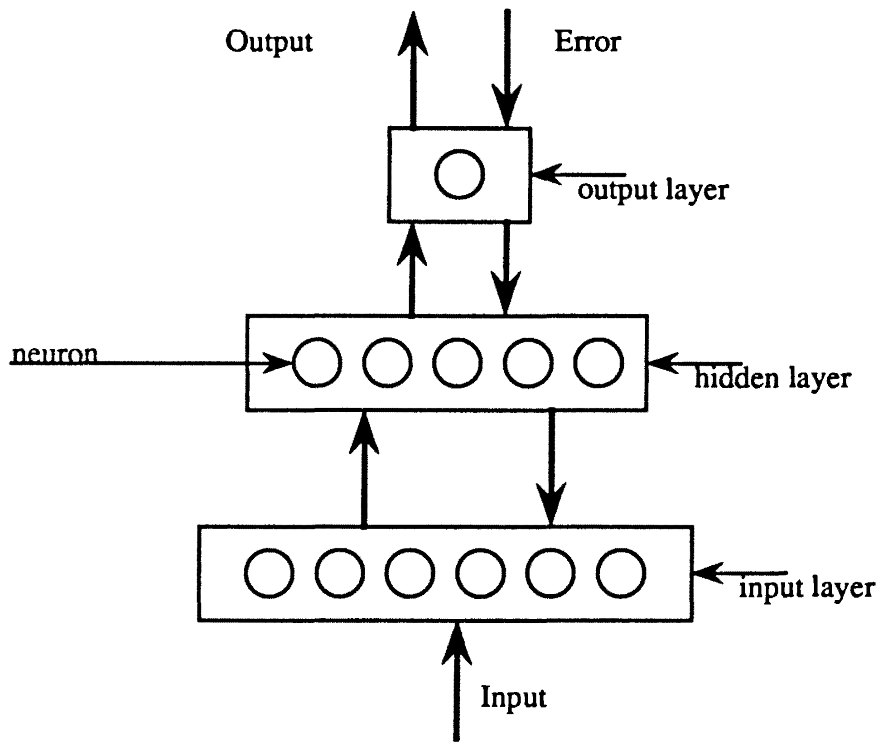


Figure 3

Backpropagation Model

recurrent backpropagation networks. The architecture of the Jordan net is shown in the Figure 4. The Figure 5 shows some more possible recurrent backpropagation architectures. The Experimental net is the architecture that is experimented in this study. The major difference in the architectures is that the output of the hidden or the output layer that is fed back to extra slab of input neurons. In Jordan net output of the output layer is fed back to the extra slab of input neurons, whereas the output of the hidden layer is fed back to the extra slab of input neurons in Experimental network. The number of hidden neurons and the hidden layers is fixed in all the models. The number of extra slab of input neurons is equal to the number of input neurons. Because of the interconnections between the hidden or output layer and the extra slab of input neurons, the recurrent backpropagation can see the data variations ahead of time and accordingly modifies the weights of the interconnections. Figure 6 depicts the architecture of Experimental net.

Method of Analysis

The data is divided, as mentioned earlier, into two parts: one is a set of training patterns, and the second is the set of testing patterns. The minimum of fifty test patterns are held for all models. For the input windows of 4, 12, 52 the leads tested are 1(one step ahead), 4 (monthly), 12 (quarterly), and 52 (annual). The forecasting ability is compared based on two types of statistical parameters: measures of central tendency and the measures of dispersion. There are several methods used to measure the concept of "central tendency." In this theses we considered only the mean and medians. The mean gives equal weight to each observation and may be considered the "balance point" of the

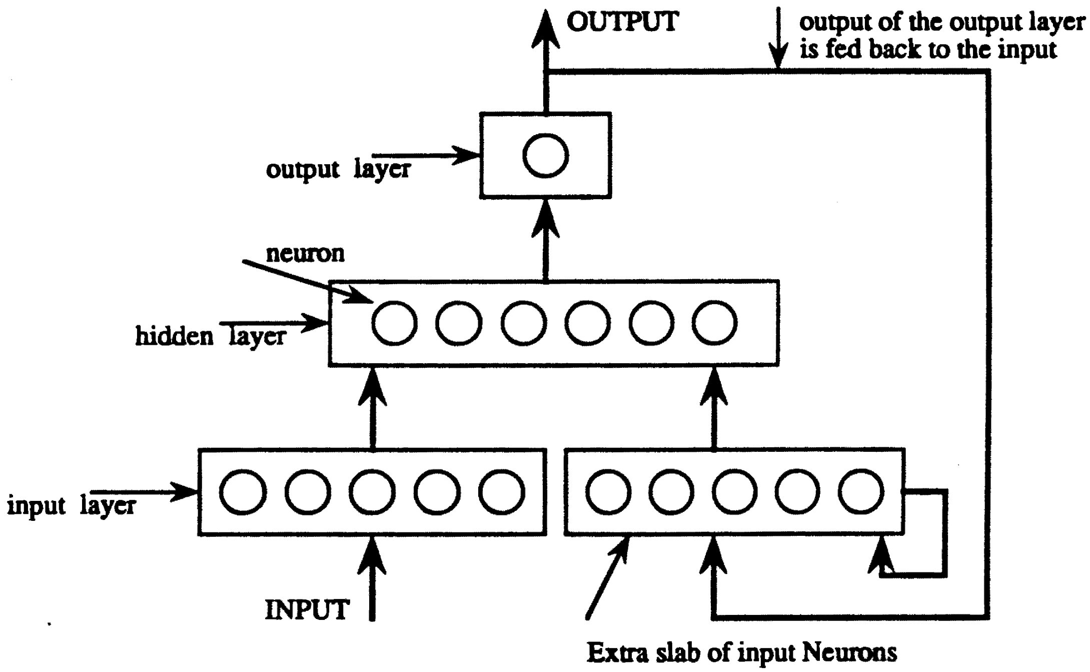


Figure 4

Jordan Network Model

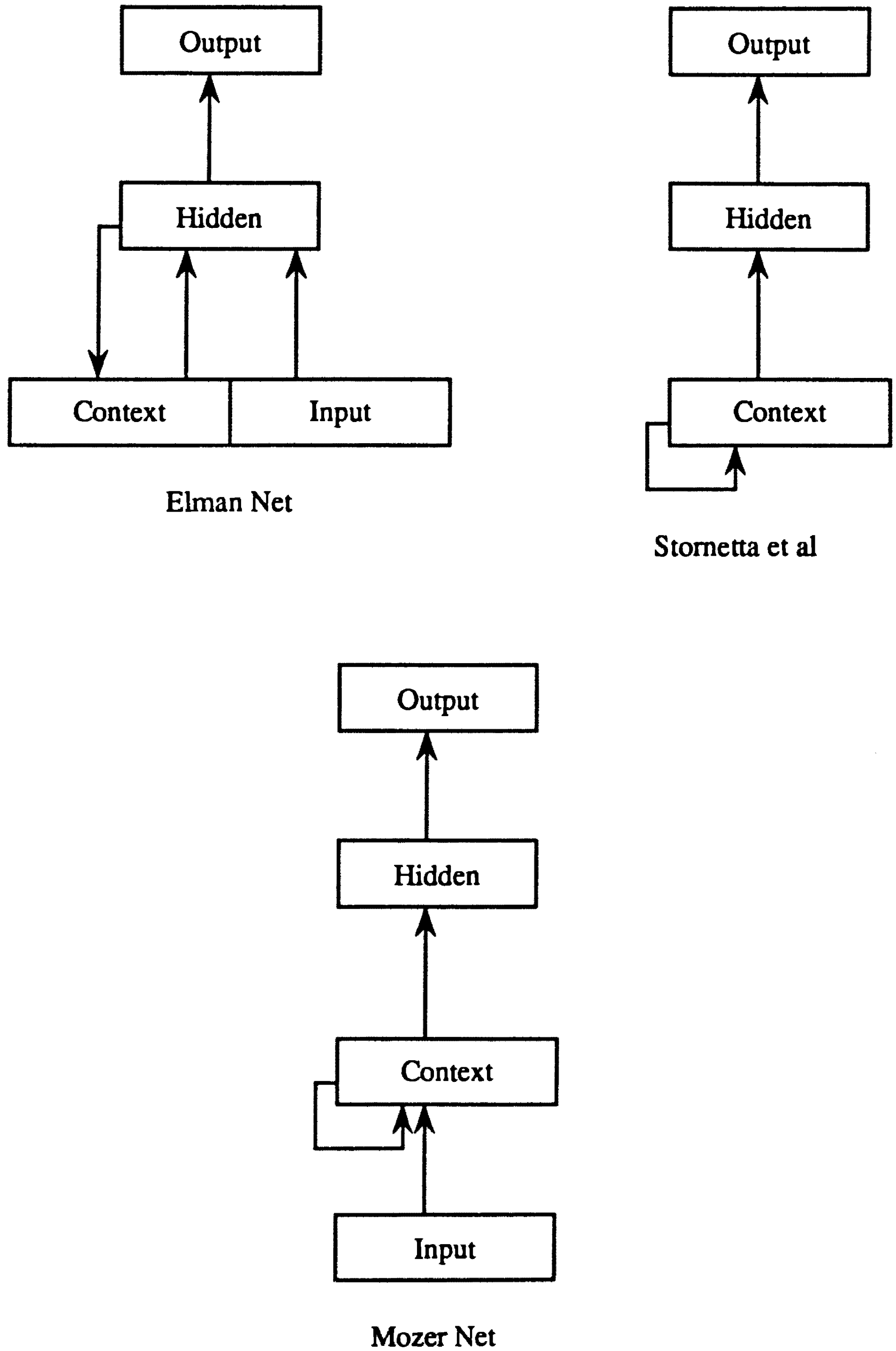


Figure 5

Examples of Some Recurrent Network Models

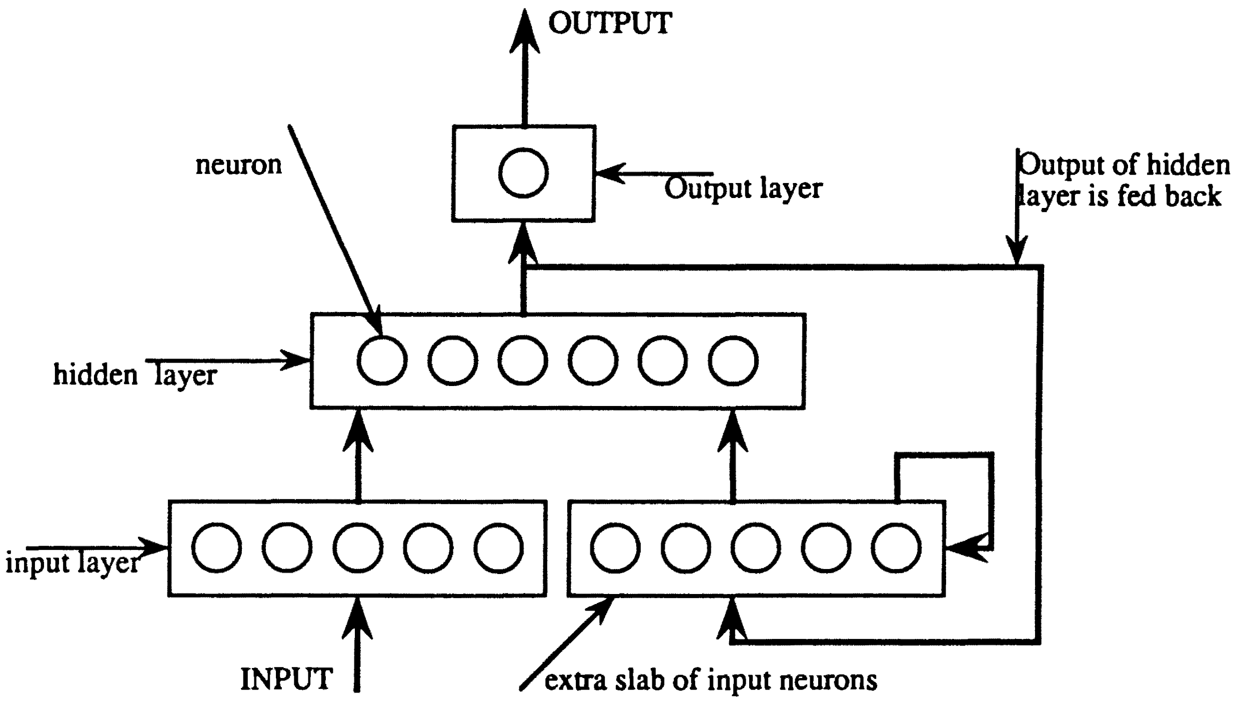


Figure 6

Experimental Network Model

data. The error in the data sometimes influences the mean to an unfair degree. In this situation we would consider median: that is, we arrange the observations in numerical order from the highest to lowest, or vice versa, and then select the midpoint of the arranged data as our average. The measures of central tendency are: MAPE (Mean Absolute Percentage Error), Median APE (Median Absolute Percentage Error), MPE (Mean Percentage Error), MSE (Mean Square Error). The measure of dispersion can be used as the measure of the spread of a distribution. For some distributions (series of data) there is a possibility of having the same mean but the values may be spread over a wider region. In this case the measure of dispersion helps to find which series is better. Definitely, the series that is having lesser dispersion (of observations) is better than the series that disperses more. The measures of dispersion considered are: MD (Mean Deviation), MAD (Mean Absolute Deviation), Variance, Coefficient of Variance, Theil's U Coefficient. The descriptions of these measurements are:

The mean percentage error is relative error expresed as a percentage. Thus to obtain the percentage error we multiply the relative error by hundred.

Mean Percentage Error: $$\frac{\sum_{i=1}^{n}\left(\frac{A_i-F_i}{A_i}\right)}{n}$$

In mean percentage error calculation the negative and positive values cancel out each other and the net error will be reduced to a greater extent. The comparison based on such values may lead to erroneous conclusions. The absolute percentage error overcomes this effect by taking the absolutes to calculate the mean. Thus the Mean Absolute Percentage Error is considered a more robust measure.

Mean APE:
$$\frac{\sum_{i=1}^{n}\left(\frac{|A_i - F_i|}{A_i}\right)}{n}$$

The Mean Square error enhances the error that is appearing in absolute percentage error. As this is a squared error the outliers are given more weight. The comparison based on this measurement is better and accurate.

Mean Square Error:
$$\frac{\sum_{i=1}^{n}(A_i - F_i)^2}{n}$$

The measures of dispersion compare the range of performance of a method. The mean deviation is one such measure. The deviation is computed with respect to the mean of a series. In the equation below the dispersion (of forecasted values) is measured with respect to the actual data series (Autobox Manual 1992).

Mean Deviation (μ):
$$\frac{\sum_{i=1}^{n}(A_i - F_i)}{n}$$

In computing mean deviation the positive and negative deviations may cancel out each other. Comparison based on such value may lead to erroneous conclusions. The absolute deviation overcomes this affect and gives the absolute deviation from the actual series.

Mean Absolute Deviation: $$\frac{\sum_{i=1}^{n} | A_i - F_i |}{n}$$

Because of the modulus sign which was used in the mean absolute deviation and the consequent awkward algebraic manipulation the mean deviation is not easy to use. A far more useful measure is the variance that uses the square deviation, which are all positive and hence de not cancel out each other.

Variance: $$\frac{\sum_{i=1}^{n} \left([A_i - F_i] - \mu \right)^2}{n}$$

The coefficient of variance can be used to give some measure of the relative importance of the standard deviation (square root of variance) referred to mean. For example: (a) A standard deviation of one ft in the measurement of the lengths of planks whose average length is hundred ft, (b) the same variation of one ft in the measurement of planks whose average length is five ft. Obviously the spread about the mean of the lengths in case (a) is less important than the spread in case (b). The coefficient of variance is used in such instances to decide the relative precision.

Coefficient of Variance: $$\frac{1}{\bar{A}} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (F - A)^2}$$

The Theil's U-coefficient computes the goodness of the formal method as compared with naive method. The drawback of this measure is that its interpretation is not straightforward. Mathematical expression for Theil's U-coefficient is

Theil's U Coefficient:

$$\sqrt{\frac{\sum_{i=1}^{n-1}\left(\frac{F_{i+1}-A_{i+1}}{A_i}\right)^2}{\sum_{i=1}^{n-1}\left(\frac{A_{i+1}-A_i}{A_i}\right)^2}}$$

CHAPTER V

RESULTS AND DISCUSSION

Performances of all four models are compared based on their forecasting error. The error is calculated on the basis of measure of central tendency, and the measure of dispersion. In order to have an accurate analysis the measure of central tendency is calculated by four methods and the measure of dispersion is calculated by five methods. The analysis of these methods is summarized in figures 7-15, one for each error calculating method.

In each figure four charts are shown. For the input window values of 4, 12 and 52: the top-left, top-right, bottom-left and bottom-right depict the forecast errors for one, four, twelve and fifty two weeks ahead, respectively.

Figure 7 shows mean percentage error for different models. For one week (Figure 7a) ahead prediction, as the input window grows, the variation in the error for experiment network is consistent, where as the other models vary inconsistently. For four and twelve weeks (Figure 7b, 7c) ahead forecast Jordan and experiment network models have better predictions than other models. For fifty two weeks (Figure 7d) ahead forecast Jordan net provided consistent results, with Experimental network model provided the next better results. The averages of all mean percentage errors for all models (for all combinations

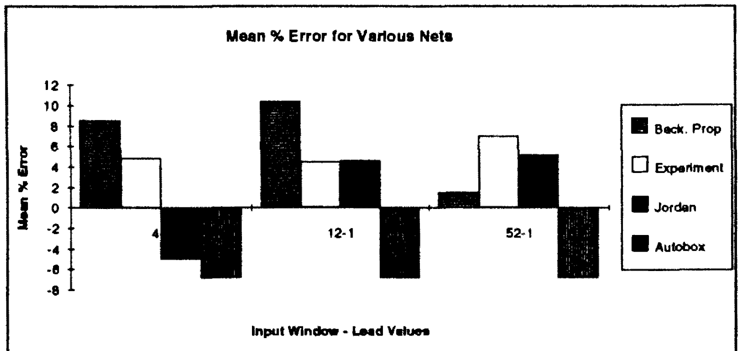


Figure 7 (a)

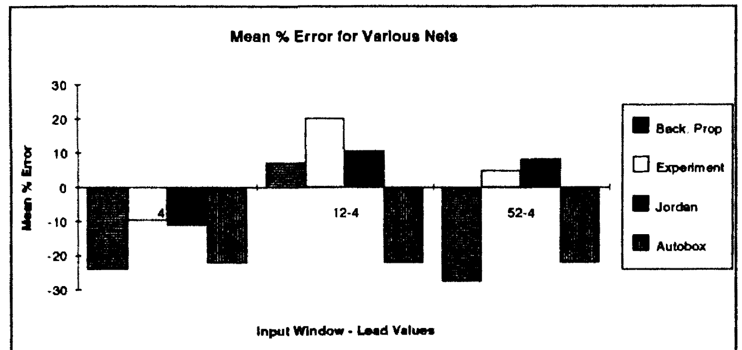


Figure 7 (b)

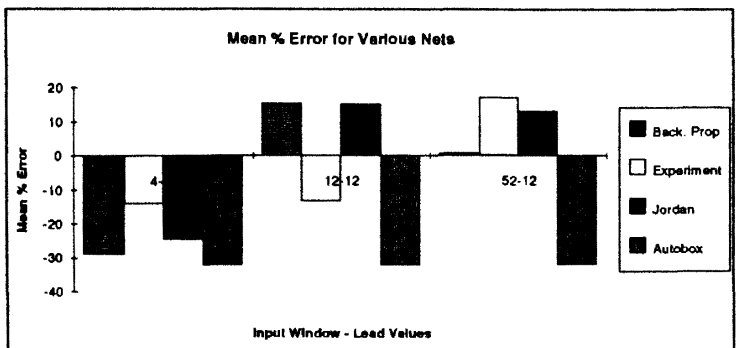


Figure 7 (c)

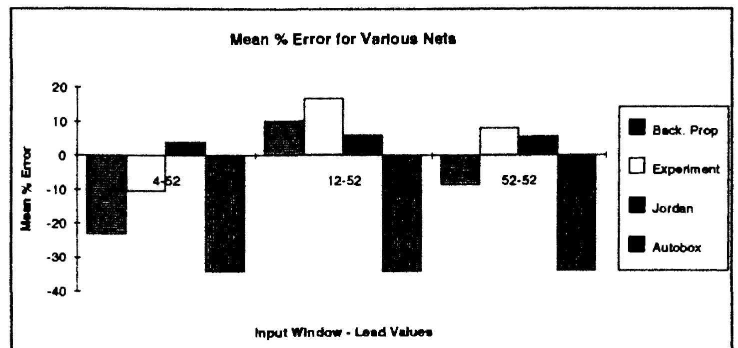


Figure 7(d)

of input window and lead times) are shown in Figure 8(a). Clearly, the experiment and Jordan nets have least prediction percentage errors when compared to other models.

Figure 9 shows the mean absolute percentage errors for all models. For one week, fifty weeks ahead forecast the Jordan net performed better than all the other models (Figures 9a, 9d). For four weeks (Figure 9b) ahead forecast the Experimental and Jordan network models performed almost equally. For twelve weeks (Figure 9c) ahead forecast the Experimental network model performed better than all the other models. On the average, from Figure 8b, the performance of the Experimental and Jordan net is almost the same.

Figure 10 shows the median absolute percentage errors for all models. These figures support the results that are concluded from the mean absolute percentage error graphs (Figures 9a-9d). Though the extremities may affect the mean, the median cannot be affected by these extremities. The median absolute percentage error charts differ from that of MAPE charts only if there are extremities in the forecast. Since the Median APE charts are supporting the MAPE charts, we can conclude that the current forecast does not have any extremities.

Figure 11 shows the mean square error for all models. For short term predictions (i.e., one week ahead, Figure 11a) backpropagation has highly inconsistent forecast. For long term (i.e., fifty two weeks ahead, Figure 11d) the traditional Box-Jenkins method (i.e., Autobox) is too erroneous. The Experimental and Jordan nets outperformed each other in twelve weeks (Figure 11c) and fifty two weeks ahead forecast, respectively. On

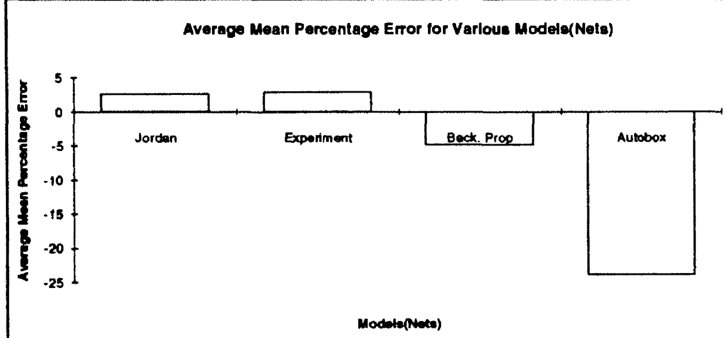


Figure 8 (a)

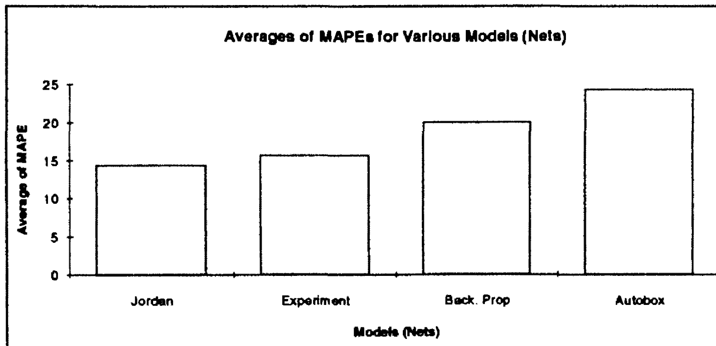


Figure 8 (b)

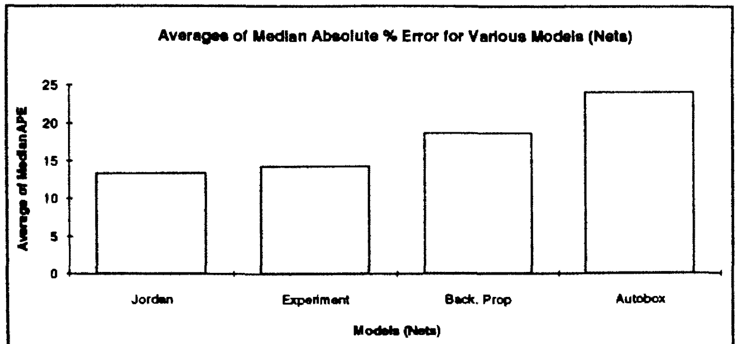


Figure 8 (c)

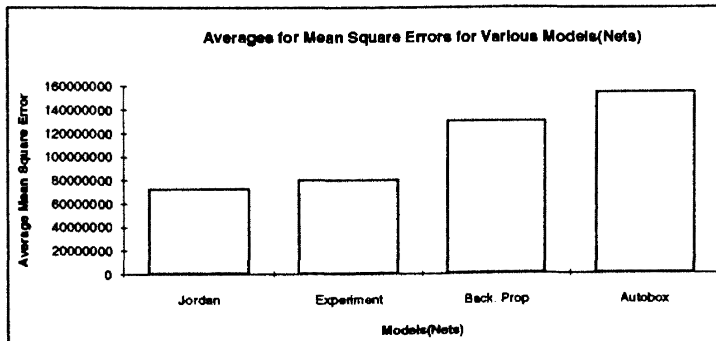


Figure 8 (d)

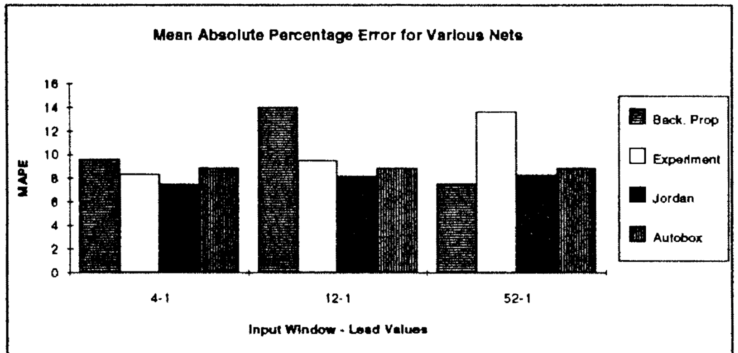


Figure 9 (a)

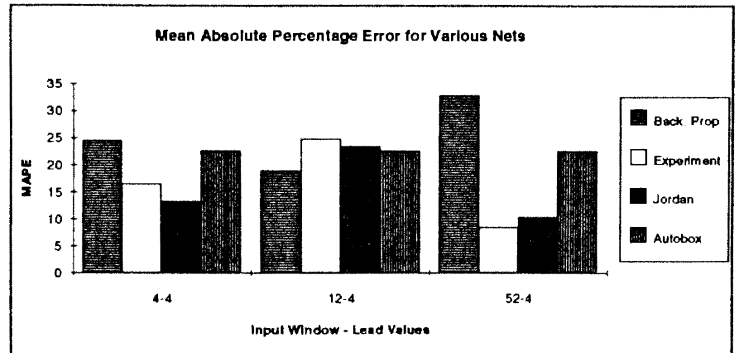


Figure 9 (b)

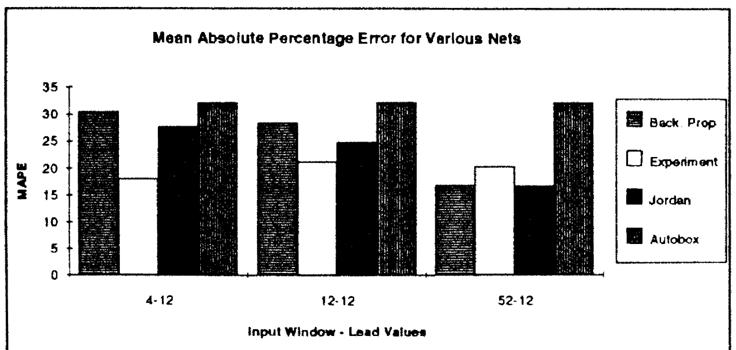


Figure 9 (c)

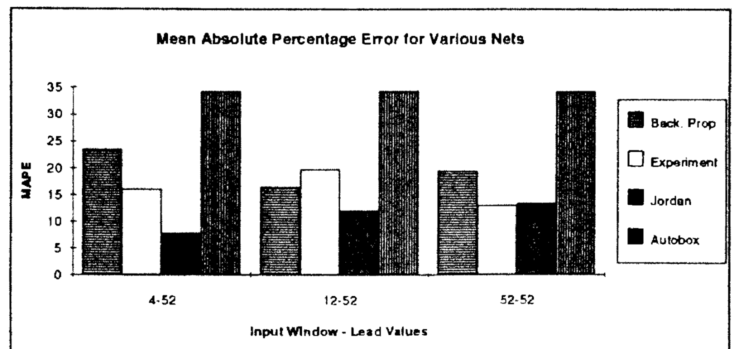


Figure 9 (d)

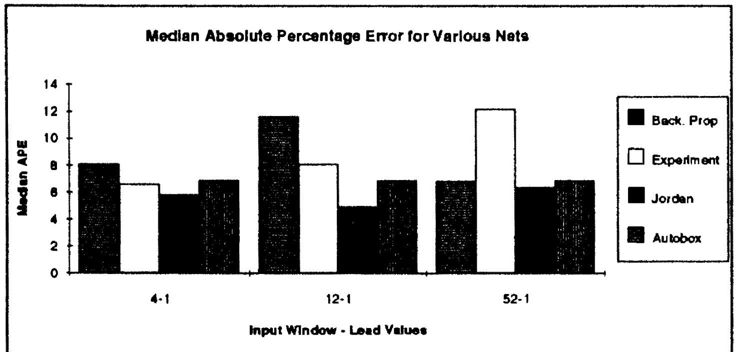


Figure 10 (a)

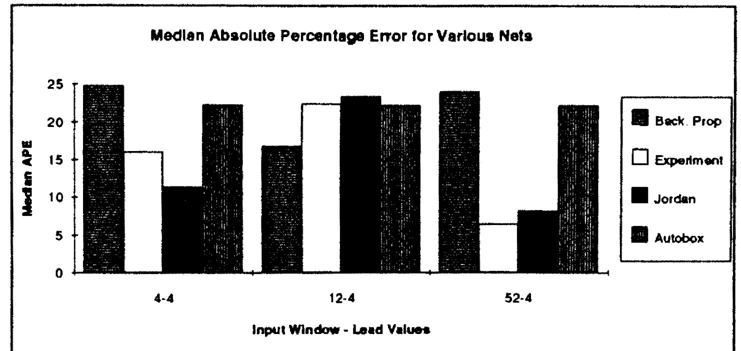


Figure 10 (b)

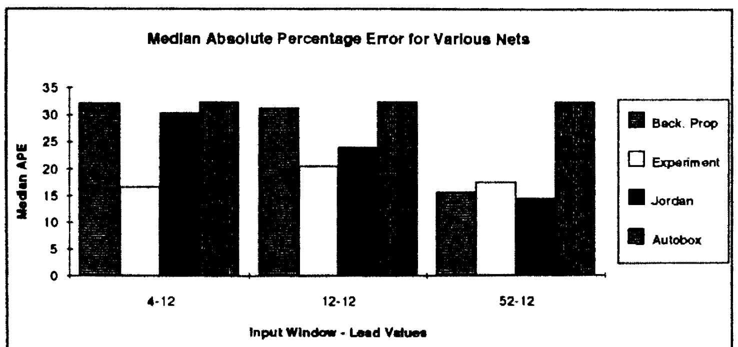


Figure 10 (c)

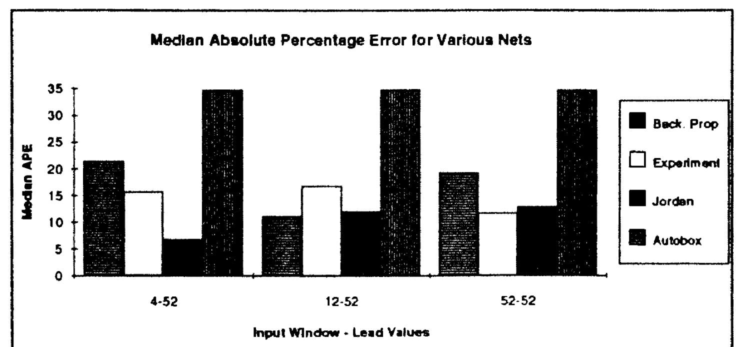


Figure 10(d)

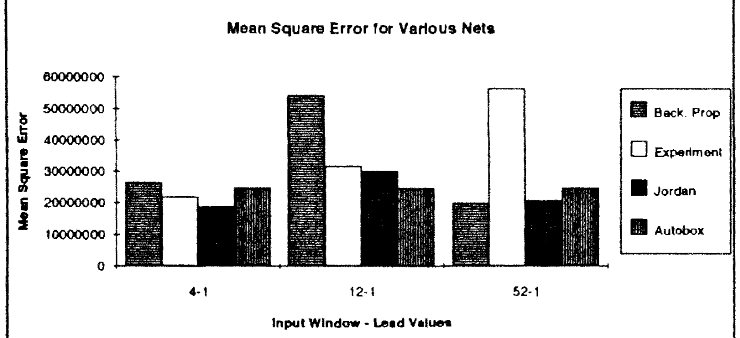


Figure 11 (a)

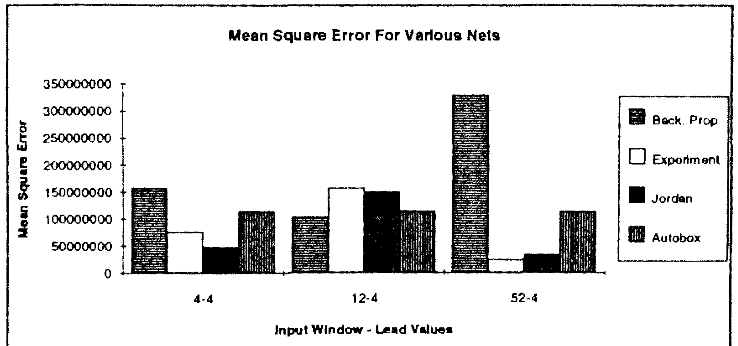


Figure 11 (b)

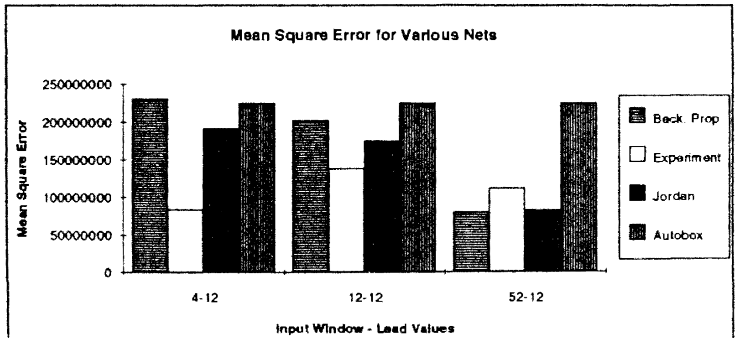


Figure 11 (c)

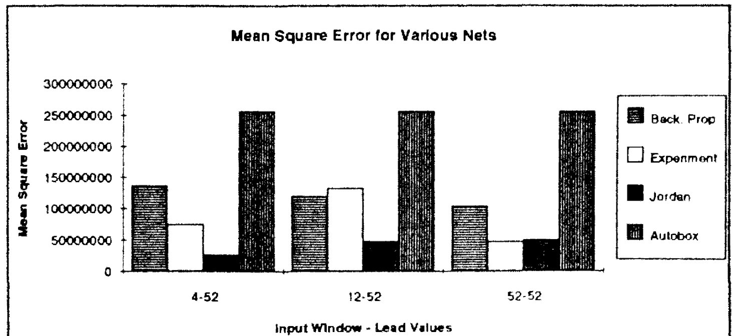


Figure 11 (d)

the average (Figure 8d) the Jordan net outperformed all other models, followed by Experimental net.

The Figure 12 shows the mean deviations for all models. For one week ahead forecast (Figure 12a) the Jordan net performed better than the other models. In four weeks and twelve weeks (Figures 12b, 12c) ahead forecast experiment net deviated lesser than the other models. In fifty two weeks (Figure 12d) ahead forecast the Jordan net outperformed all other models. In all the figures (12a-12d), Box-Jenkins model has negative mean deviation that proves its tendency to under-forecast than to overcast. On the average (Figure 13a) the Jordan net has lesser deviation when compared to other models.

The Figure 14 shows the mean absolute deviation charts for all the models. The lesser deviation of Jordan net in one and fifty two weeks (Figures 14a, 14d) ahead forecast proves its ability of forecast precision. Though the experiment net performed better than other nets in twelve weeks (Figure 14c) ahead forecast, it performed almost similar to Jordan net in four weeks (Figure 14b) ahead forecast. On the average (Figure 13b) the Jordan net outperformed all the other models.

The Figure 15 shows the variance for all the models. The Box-Jenkins model of forecasting has very little variance than the other models. This does not mean that the forecast of this model is very consistent but not followed the variations of the actual data series. This is proved by highest mean square error, mean absolute percentage error, and median absolute percentage error. The average variance chart (Figure 13c) explains that the variance of the Jordan and Experimental nets are better.

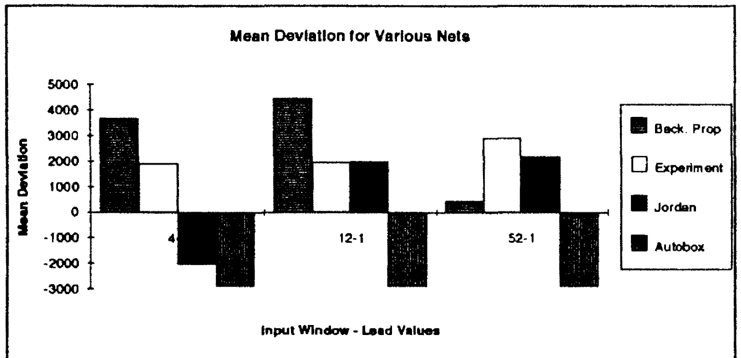


Figure 12 (a)

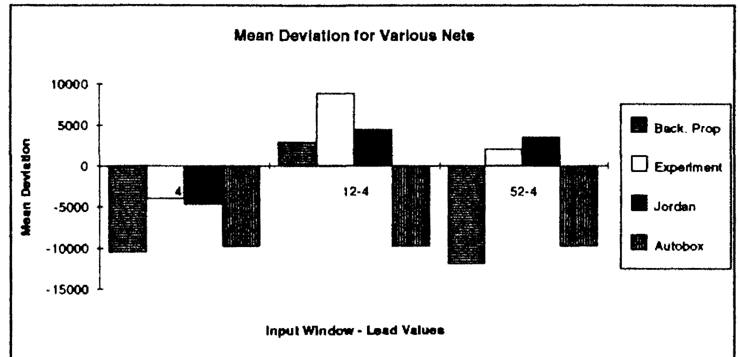


Figure 12 (b)

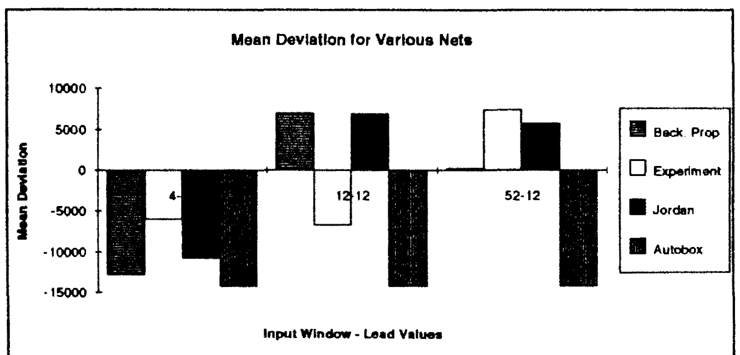


Figure 12 (c)

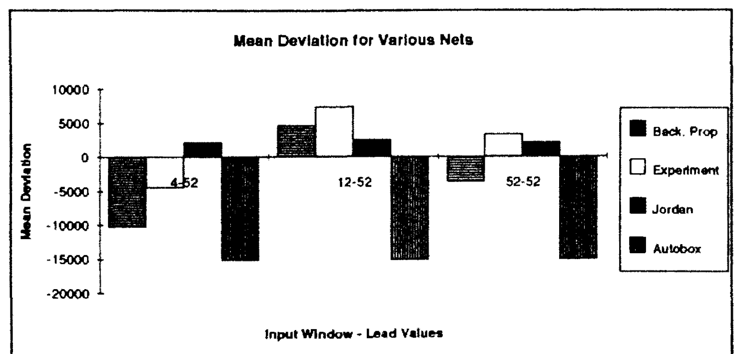


Figure 12 (d)

Average Mean Deviation for Various Models(Nets)

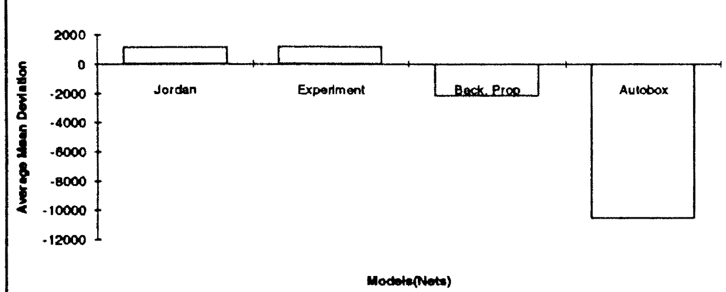


Figure 13 (a)

Average Mean Absolute Deviation for Various Models(Nets)

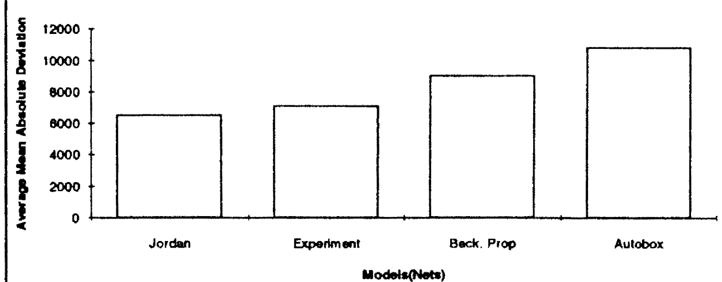


Figure 13 (b)

Average Variance Precision for Various Models(Nets)

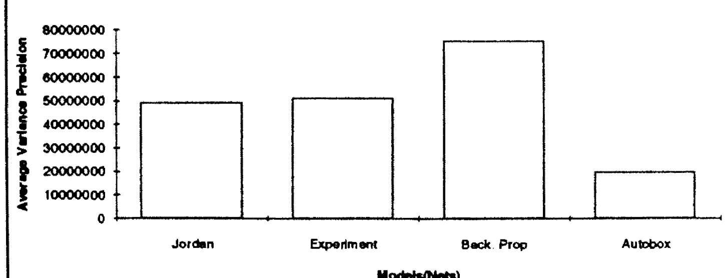


Figure 13 (c)

Average of Coefficient of Variance for Various Models (Nets)

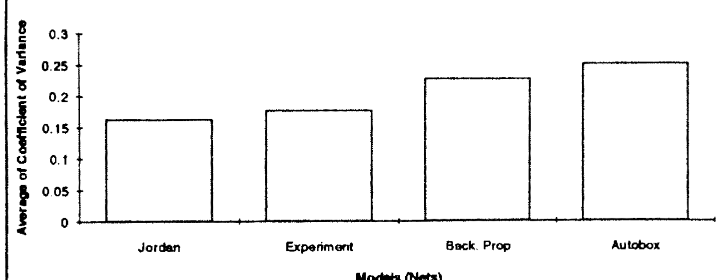


Figure 13 (d)

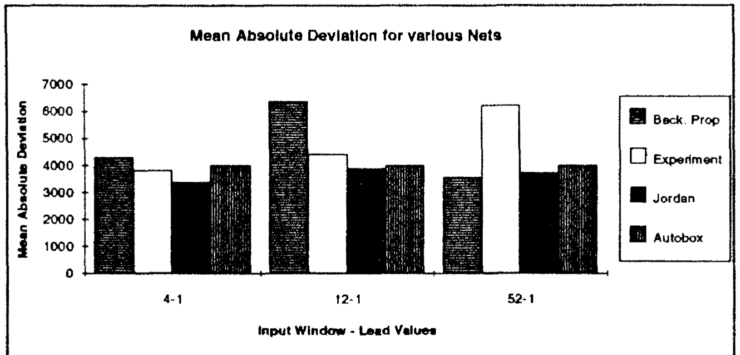


Figure 14 (a)

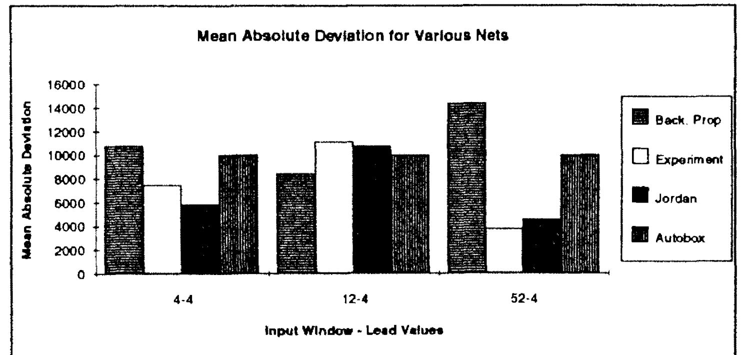


Figure 14 (b)

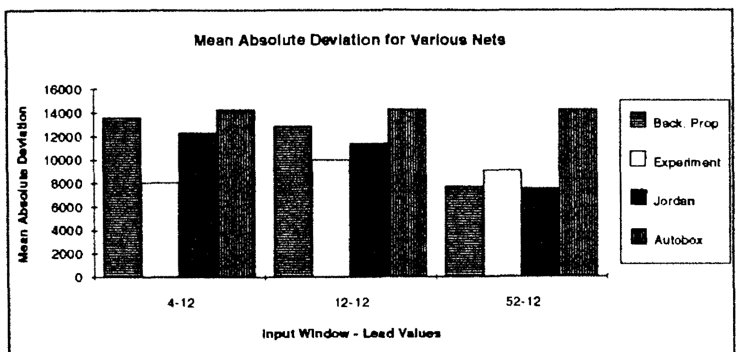


Figure 14 (c)

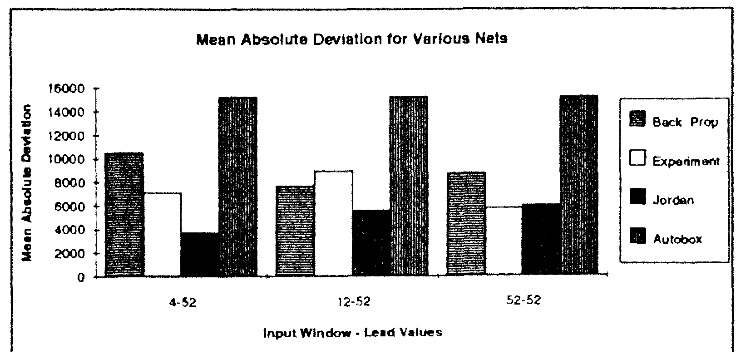


Figure 14 (d)

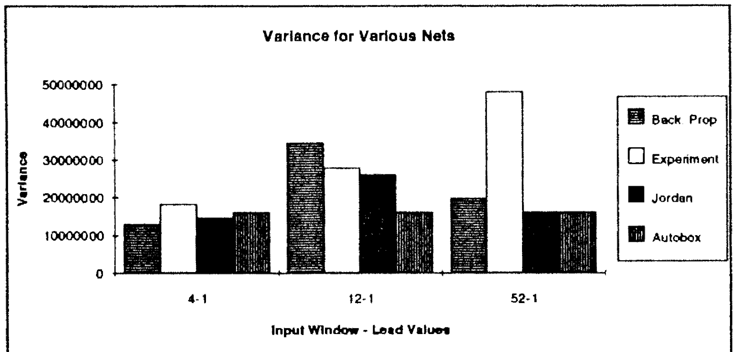


Figure 15 (a)

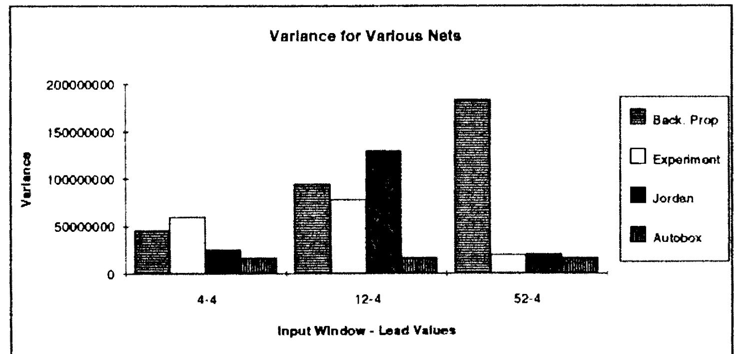


Figure 15 (b)

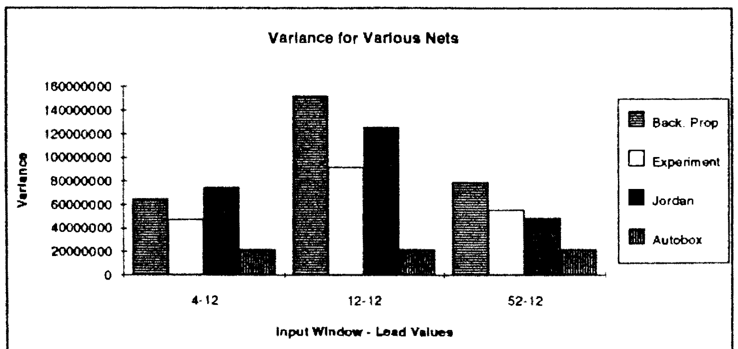


Figure 15 (c)

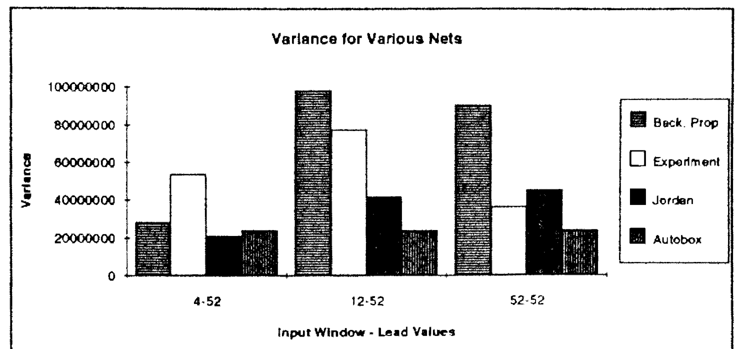


Figure 15 (d)

The Figure 16 shows the coefficient of variance of the models. As explained in the definition of the coefficient of variance, this parameter explains the relative importance of the error with respect to the mean of actual data series. Therefore smaller the coefficient, better the performance. In one, four, and fifty two weeks (Figures 16a, 16b, 16d) ahead forecast the Jordan net outperformed all the other models. In twelve weeks (Figure 16c) ahead forecast the experiment net performed better than the other models. On the average (Figure 13d) the Jordan net performed better than the other models followed by experiment net, backpropagation and Box-Jenkins method.

As explained earlier the interpretation of theil's U coefficient is not straight forward. Briefly, smaller the value of Theil's U Coefficient, better the performance. From the Figures 17a-17d, and 18 it is clearly evident that Jordan and Experimental nets have smaller Theil's U Coefficient values than others, hence they yield better performance.

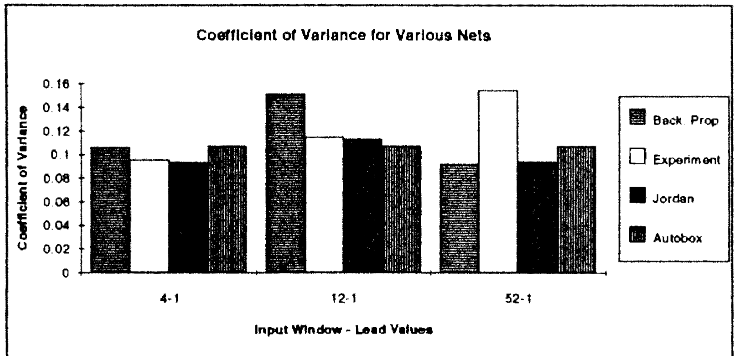


Figure 16 (a)

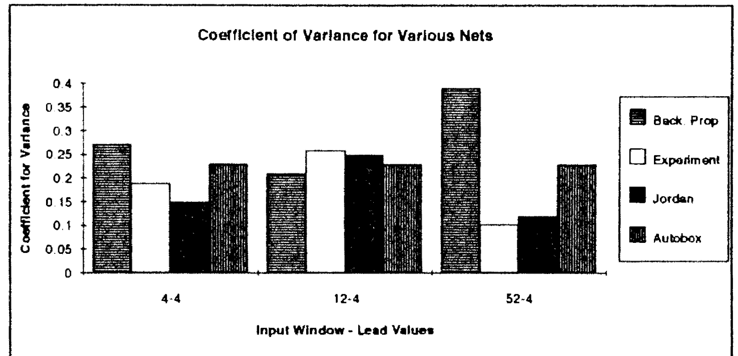


Figure 16 (b)

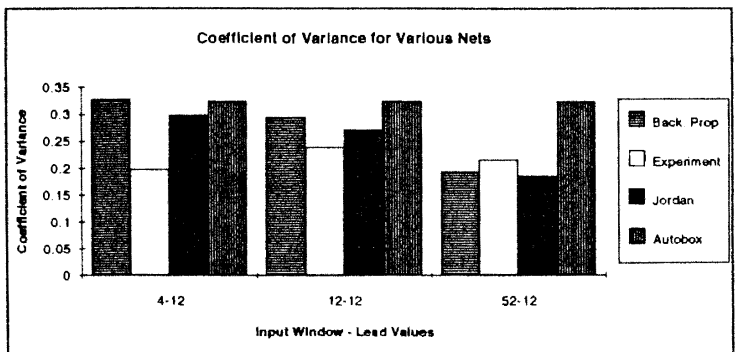


Figure 16 (c)

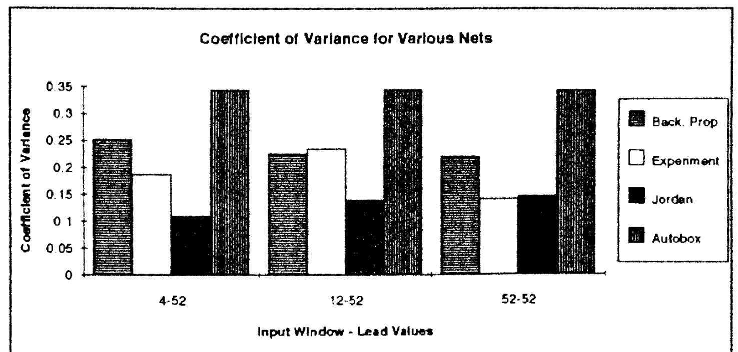


Figure 16 (d)

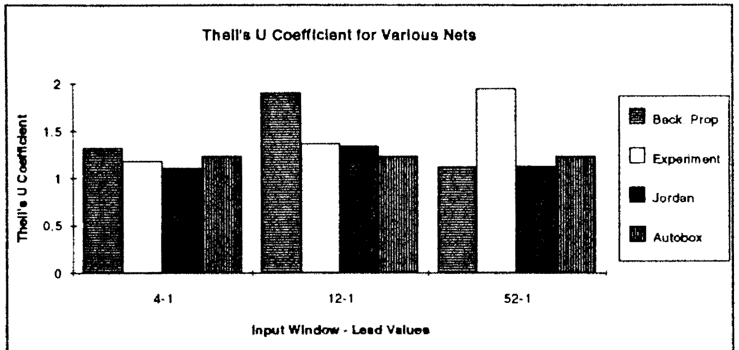


Figure 17 (a)

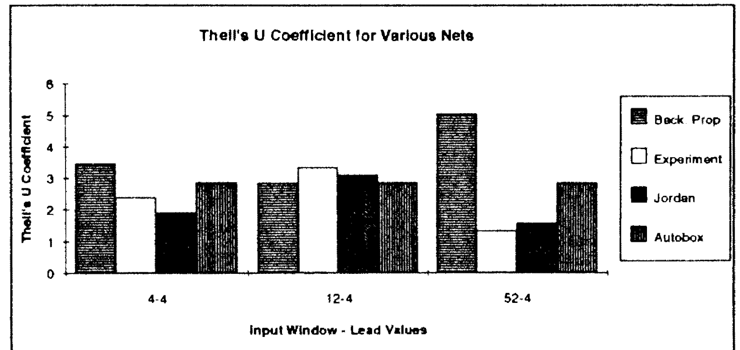


Figure 17 (b)

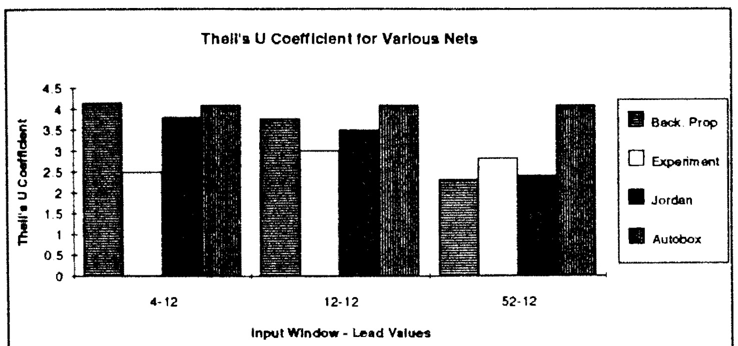


Figure 17 (c)

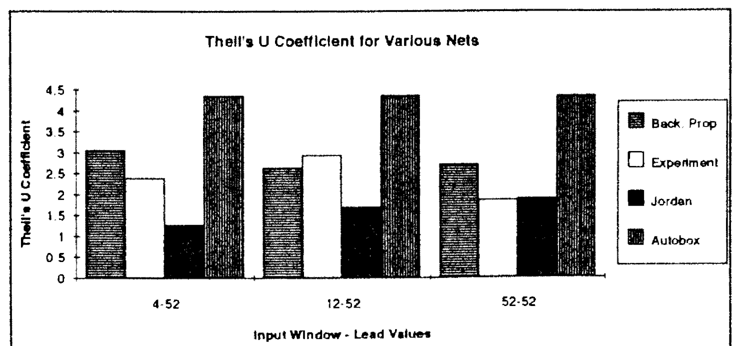


Figure 17 (d)

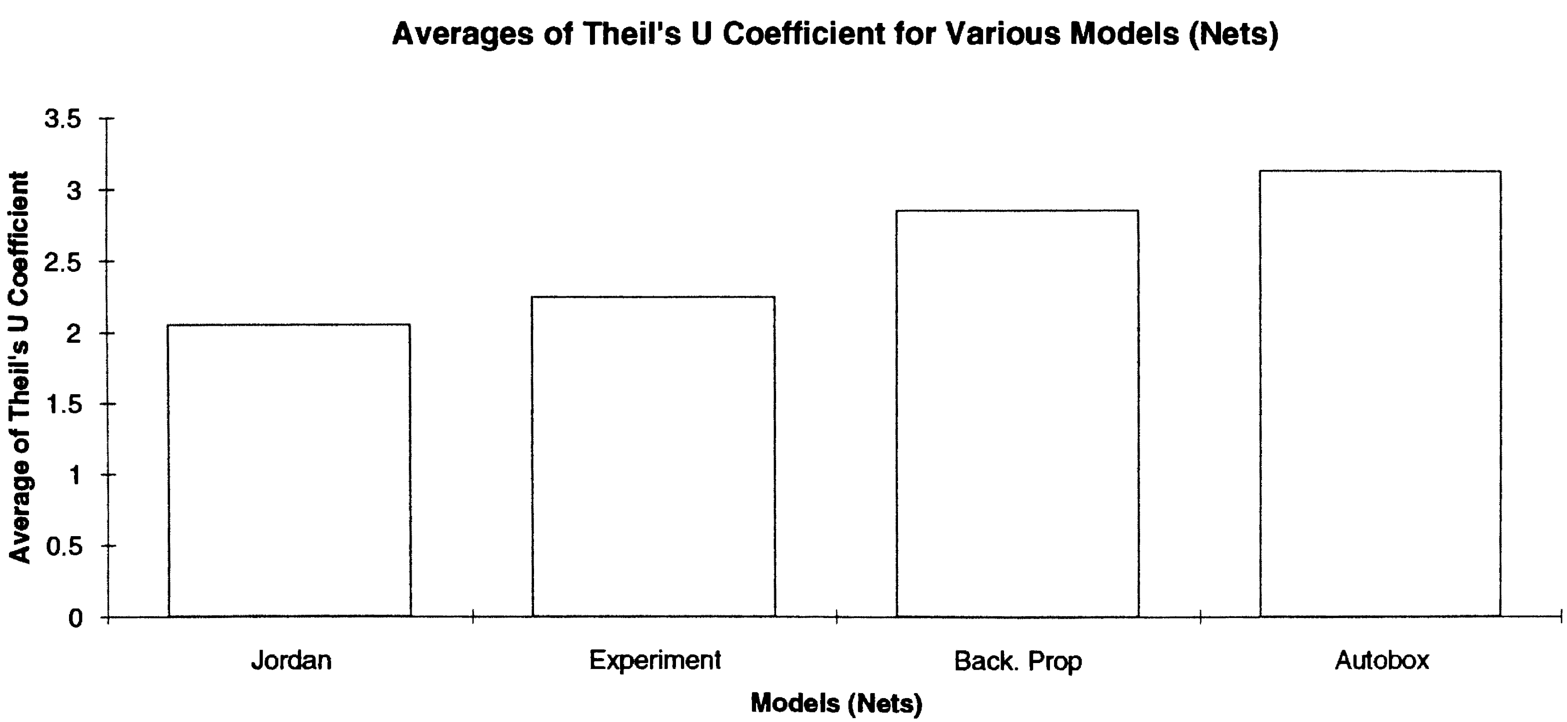

Averages of Theil's U Coefficient for Various Models (Nets)
Average of Theil's U Coefficient
3.5
3
2.5
2
1.5
1
0.5
0
Jordan
Experiment
Back. Prop
Autobox
Models (Nets)

Figure 18

TABLE I

Convergence Details for Various Models

Input Window and Lead Time	Backpropagation	Jordan Net	Experimental Net
4 - 1	23229	*	12637
4 - 4	*	*	*
4 - 12	*	*	*
4 - 52	*	*	*
12 - 1	6055	8534	3912
12 - 4	8015	6197	19226
12 - 12	15378	6198	38095
12 - 52		9424	11366
52 - 1	312	254	294
52 - 4	*	210	296
52 - 12	*	265	287
52 - 52	*	183	172

The above table represents the summary of the convergence of backpropagation and recurrent backpropagation (Jordan and Experimental) nets. The networks that did not converge are represented by a "*". The numbers represent the number of iterations required by the corresponding network. The backpropagation network failed to converge 50% of the time. The Jordan network failed only in 33% of the cases. The important observation is that the Experimental network have the tendency to converge most of the time. In this study the Experimental network converged 25% of the time which is better than the other two networks.

The Figures 19 - 30 represent the rate of convergence of various models considered. For the input window size and lead time values of 4-4, 4-12, 4-52 the rate of convergence is almost the same for the Backpropagation, Jordan net, and Experimental net. The Jordan net could not converge for the input window size and lead time value of 4-1.

For the input window size and lead time values of 12-4, 12-12 the Jordan net and Backpropagation net display a similar convergence rate. In the case of 12-1 model the convergence rate is similar for all the three nets. In the case of 12-52 the convergence rate of Jordan net is appeared to be better than the other two nets.

In the case of input window size and lead time values of 52-1, 52-4, 52-12, and 52-52 the Experimental net and Jordan net have the similar convergence rate. The convergence rate of Backpropagation net is very less for the 52-4, 52-12 and 52-52 models. These Backpropagation net models are interrupted after 72 hours of training. For the 52-1 model the backpropagation network have convergence rate that is similar to that of Experimental net and Jordan net.

Training Performance for Input Window - Lead Values of 4 - 1

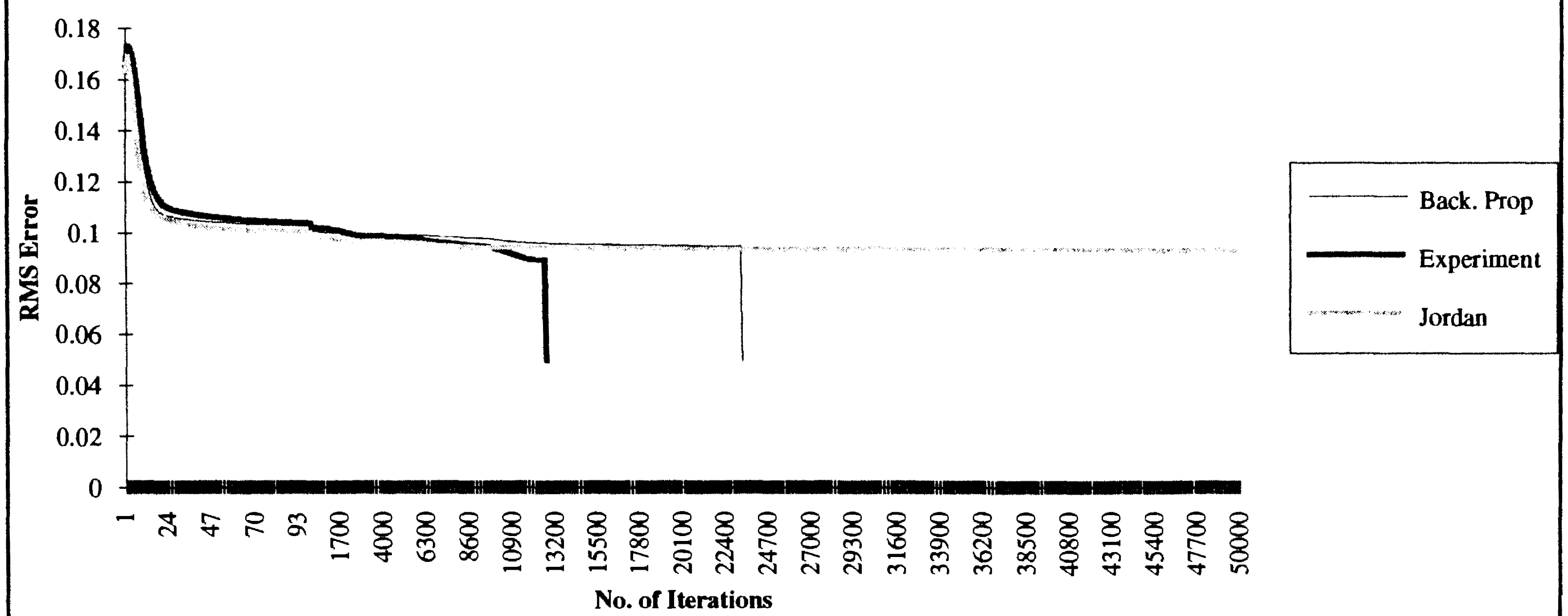
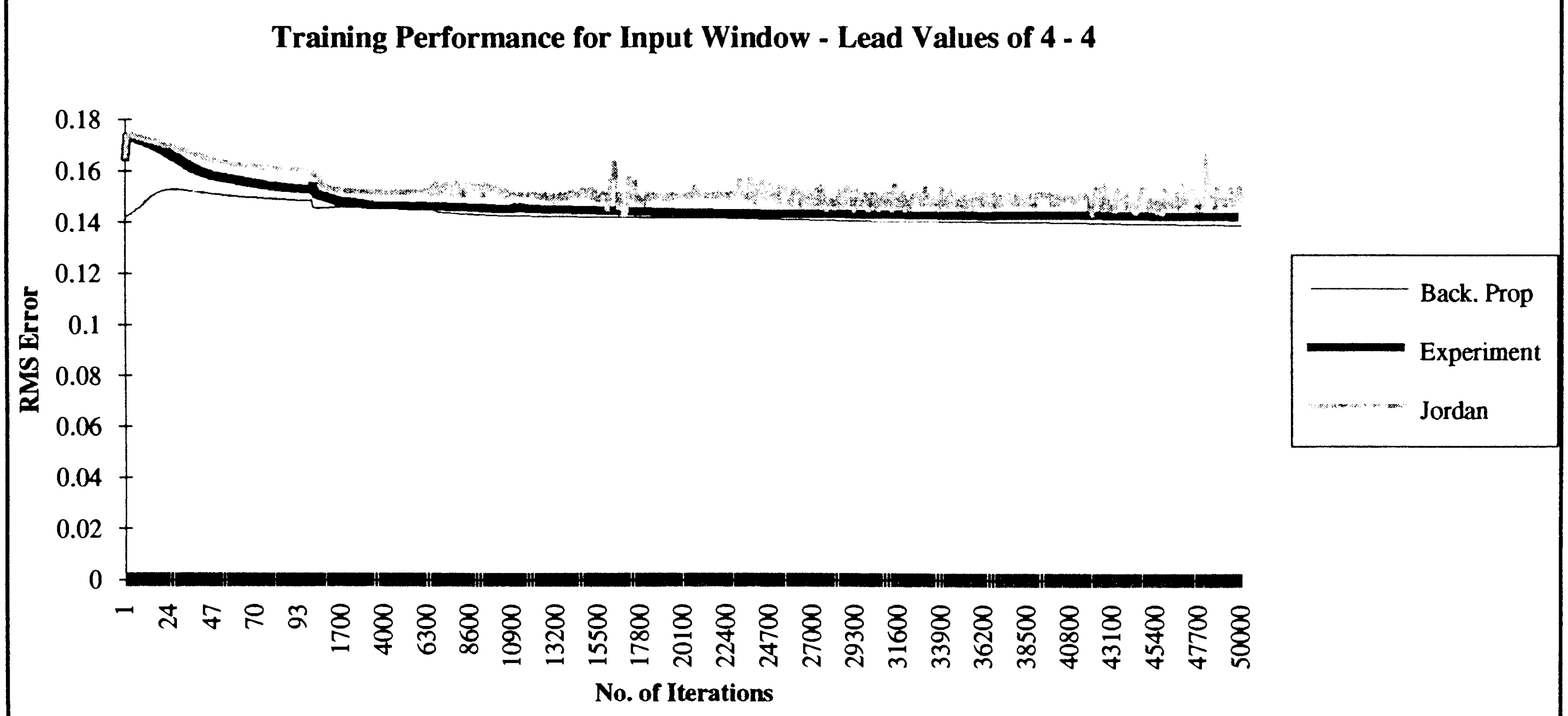


Figure 19



Training Performance for Input Window - Lead Values of 4 - 4
RMS Error
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
24
47
70
93
1700
4000
6300
8600
10900
13200
15500
17800
20100
22400
24700
27000
29300
31600
33900
36200
38500
40800
43100
45400
47700
50000
No. of Iterations
Back. Prop
Experiment
Jordan

Figure 20

Training Performance for Input Window - Lead Values of 4 - 12

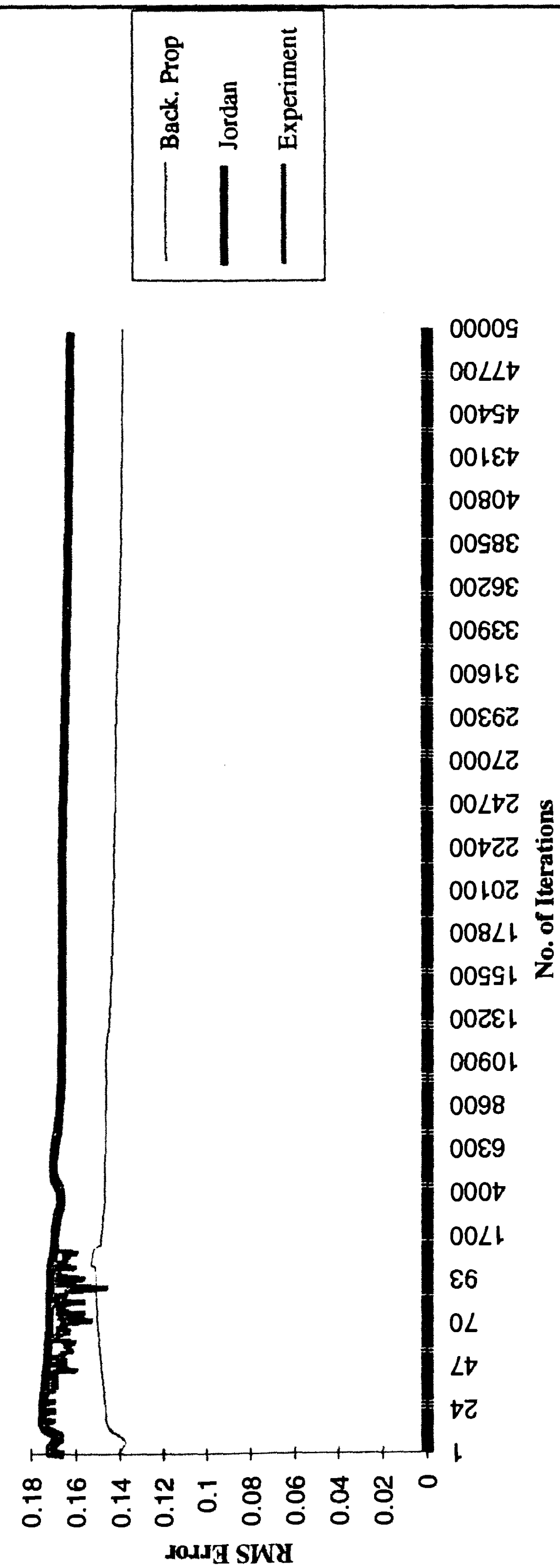


Figure 21

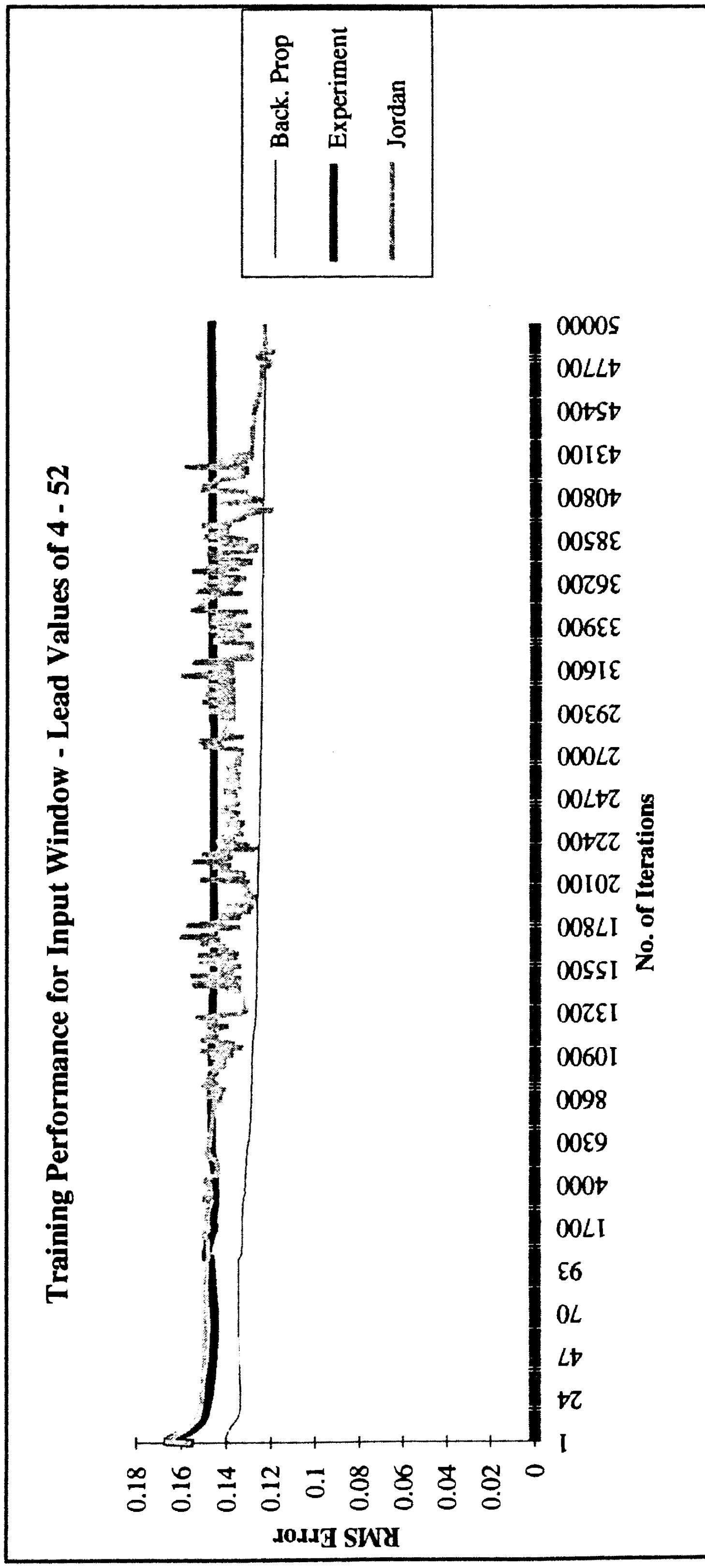

Training Performance for Input Window - Lead Values of 4 - 52
Back. Prop
Experiment
Jordan
RMS Error
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
No. of Iterations
1
24
47
70
93
1700
4000
6300
8600
10900
13200
15500
17800
20100
22400
24700
27000
29300
31600
33900
36200
38500
40800
43100
45400
47700
50000

Figure 22

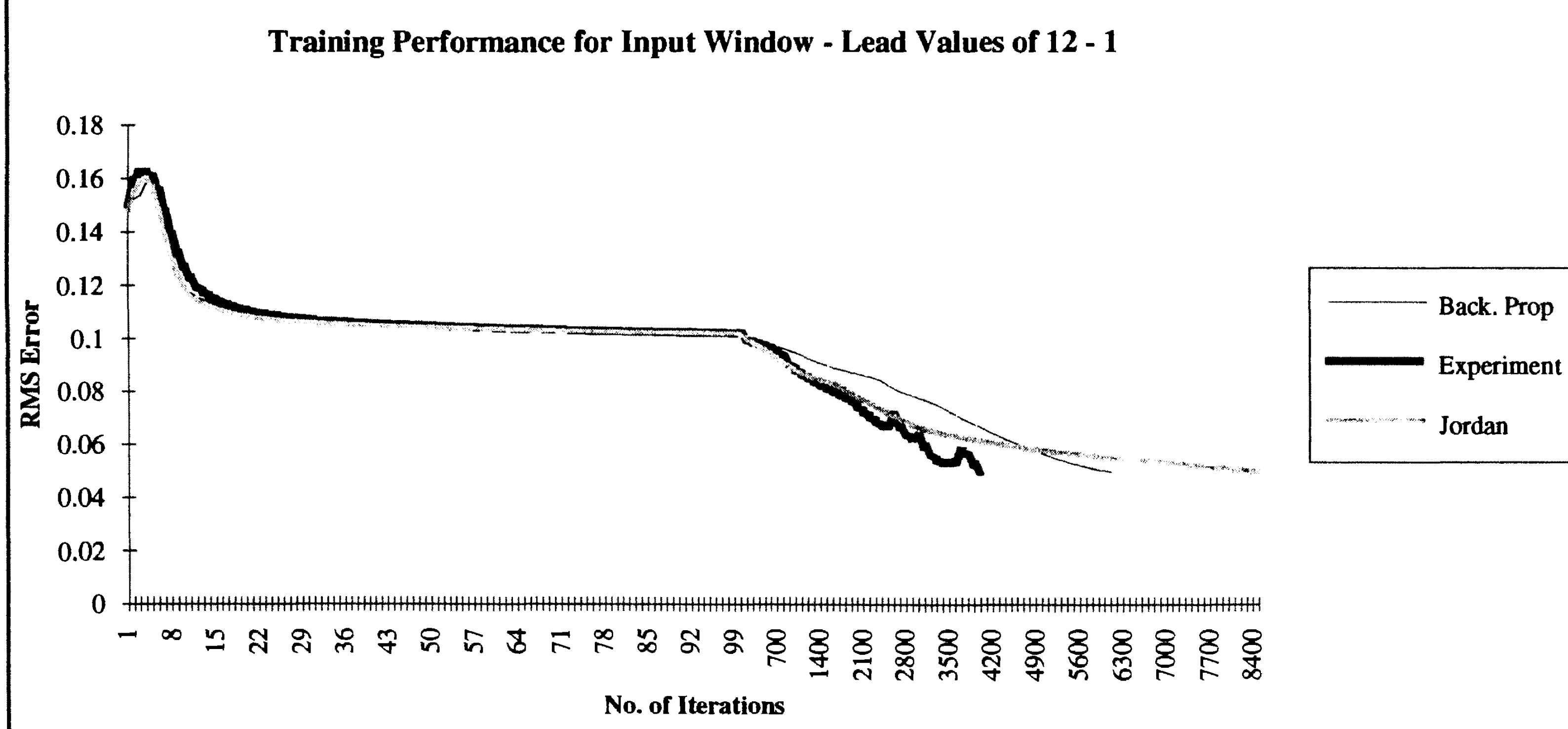
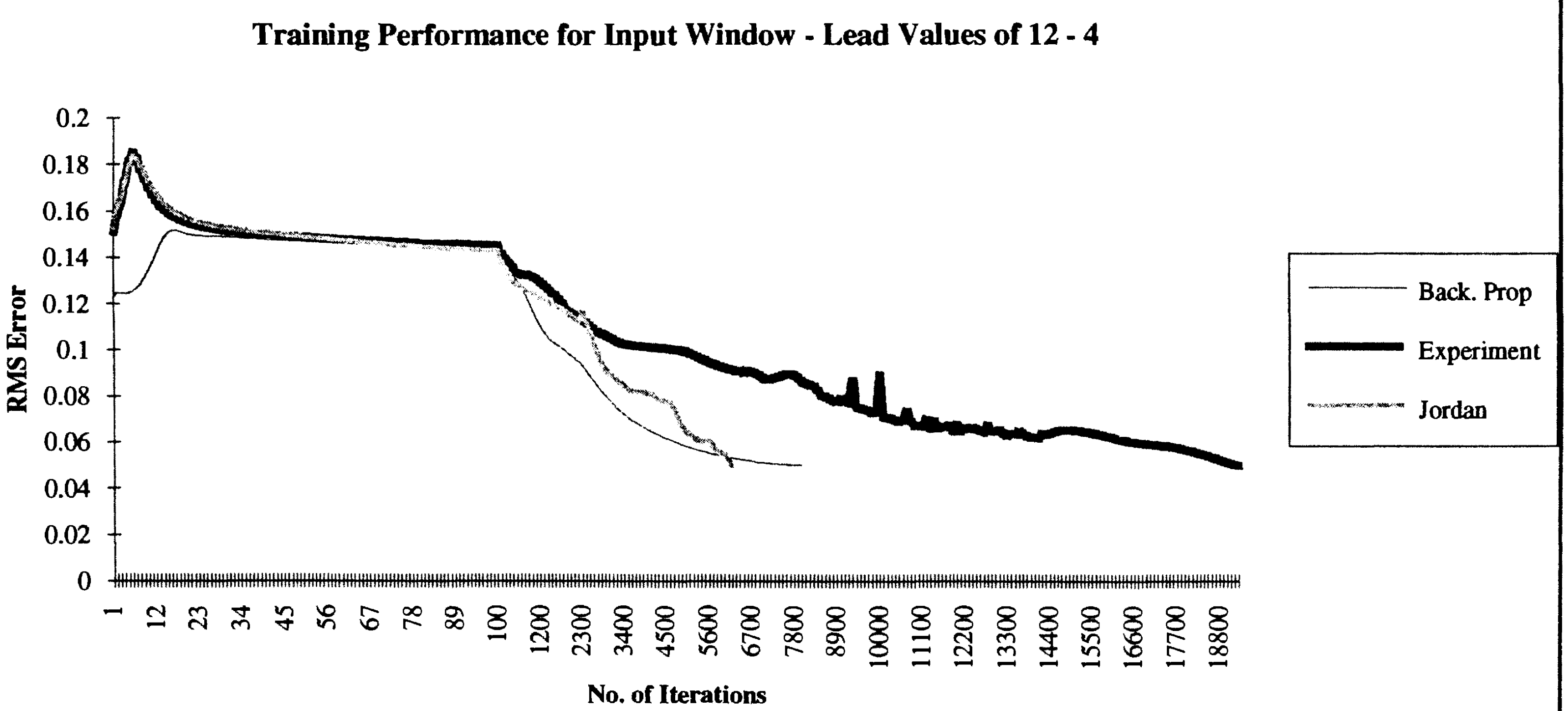

Training Performance for Input Window - Lead Values of 12 - 1
RMS Error
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
8
15
22
29
36
43
50
57
64
71
78
85
92
99
700
1400
2100
2800
3500
4200
4900
5600
6300
7000
7700
8400
No. of Iterations
Back. Prop
Experiment
Jordan

Figure 23



Training Performance for Input Window - Lead Values of 12 - 4
RMS Error
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
12
23
34
45
56
67
78
89
100
1200
2300
3400
4500
5600
6700
7800
8900
10000
11100
12200
13300
14400
15500
16600
17700
18800
No. of Iterations
Back. Prop
Experiment
Jordan

Figure 24

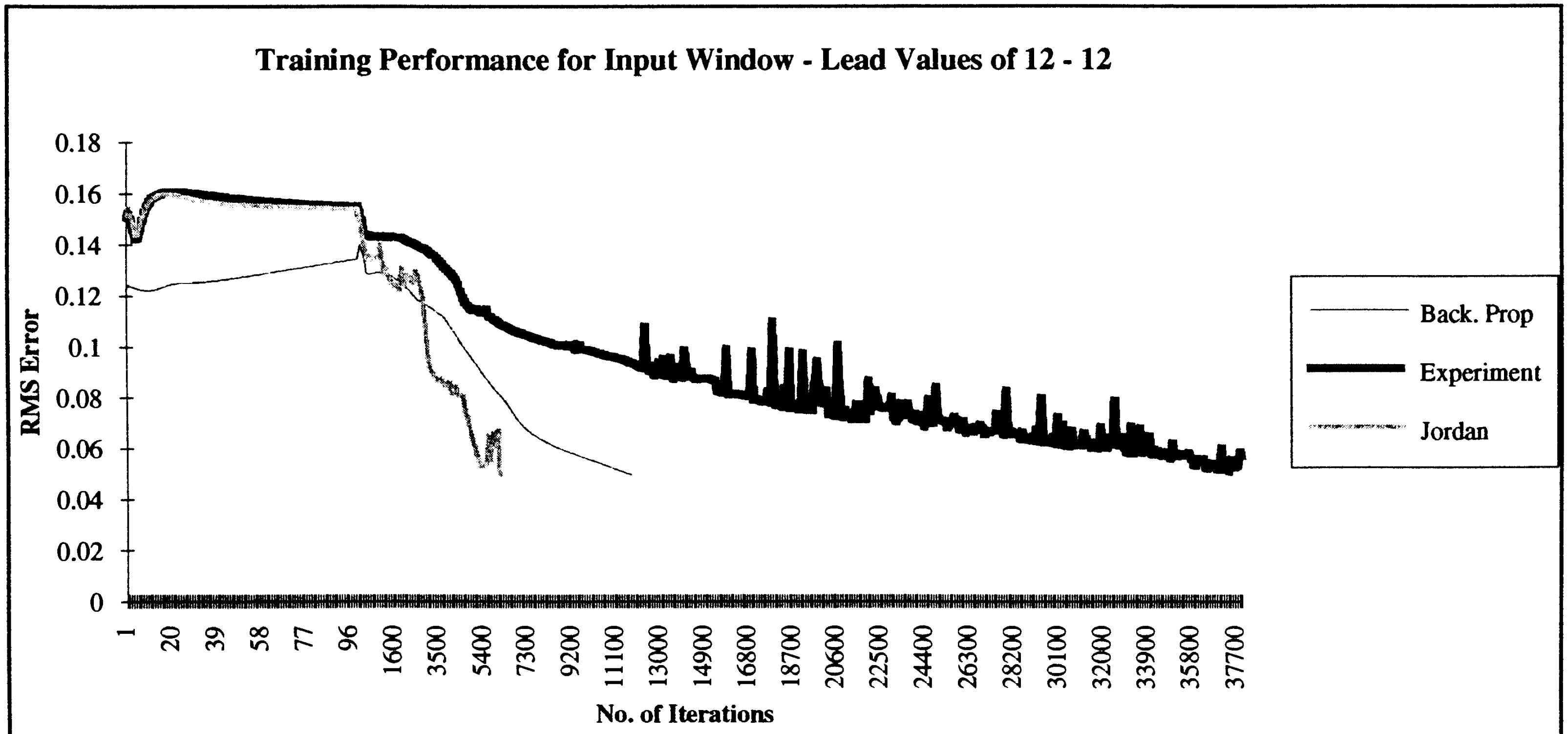

Training Performance for Input Window - Lead Values of 12 - 12
RMS Error
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
20
39
58
77
96
1600
3500
5400
7300
9200
11100
13000
14900
16800
18700
20600
22500
24400
26300
28200
30100
32000
33900
35800
37700
No. of Iterations
Back. Prop
Experiment
Jordan

Figure 25

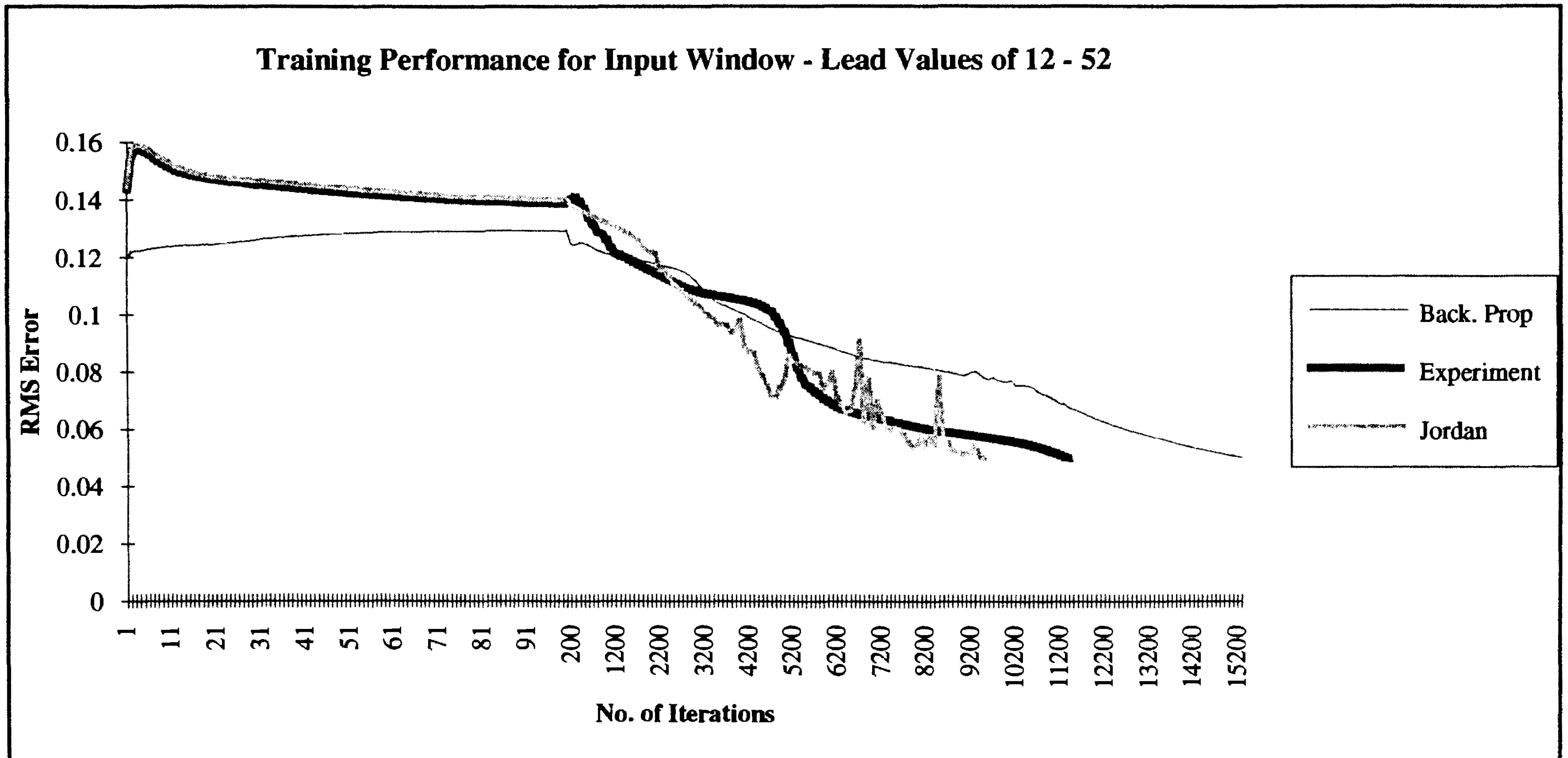

Training Performance for Input Window - Lead Values of 12 - 52
RMS Error
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
11
21
31
41
51
61
71
81
91
200
1200
2200
3200
4200
5200
6200
7200
8200
9200
10200
11200
12200
13200
14200
15200
No. of Iterations
Back. Prop
Experiment
Jordan

Figure 26

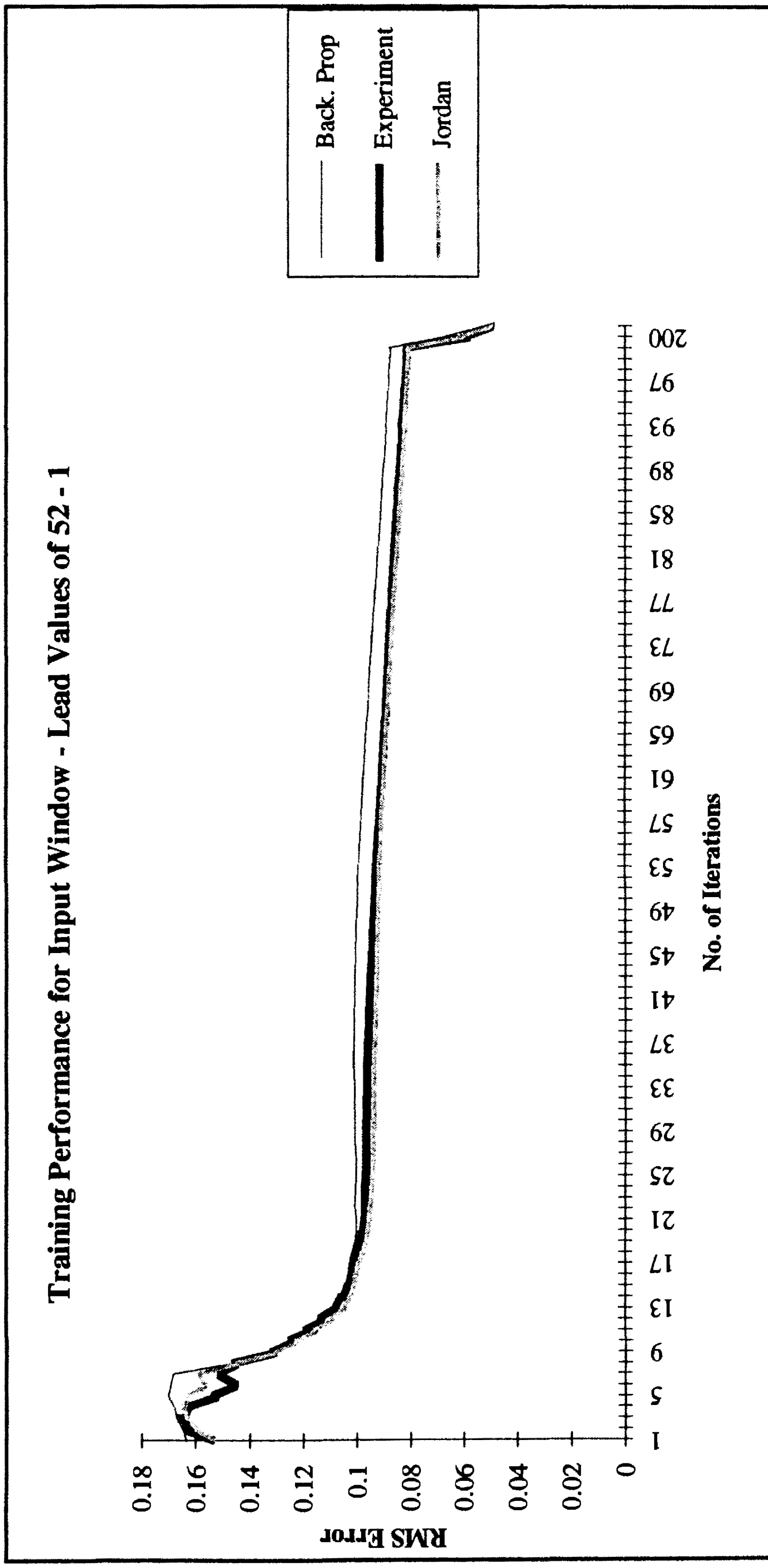

Training Performance for Input Window - Lead Values of 52 - 1
Back. Prop
Experiment
Jordan
RMS Error
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97
200
No. of Iterations

Figure 27

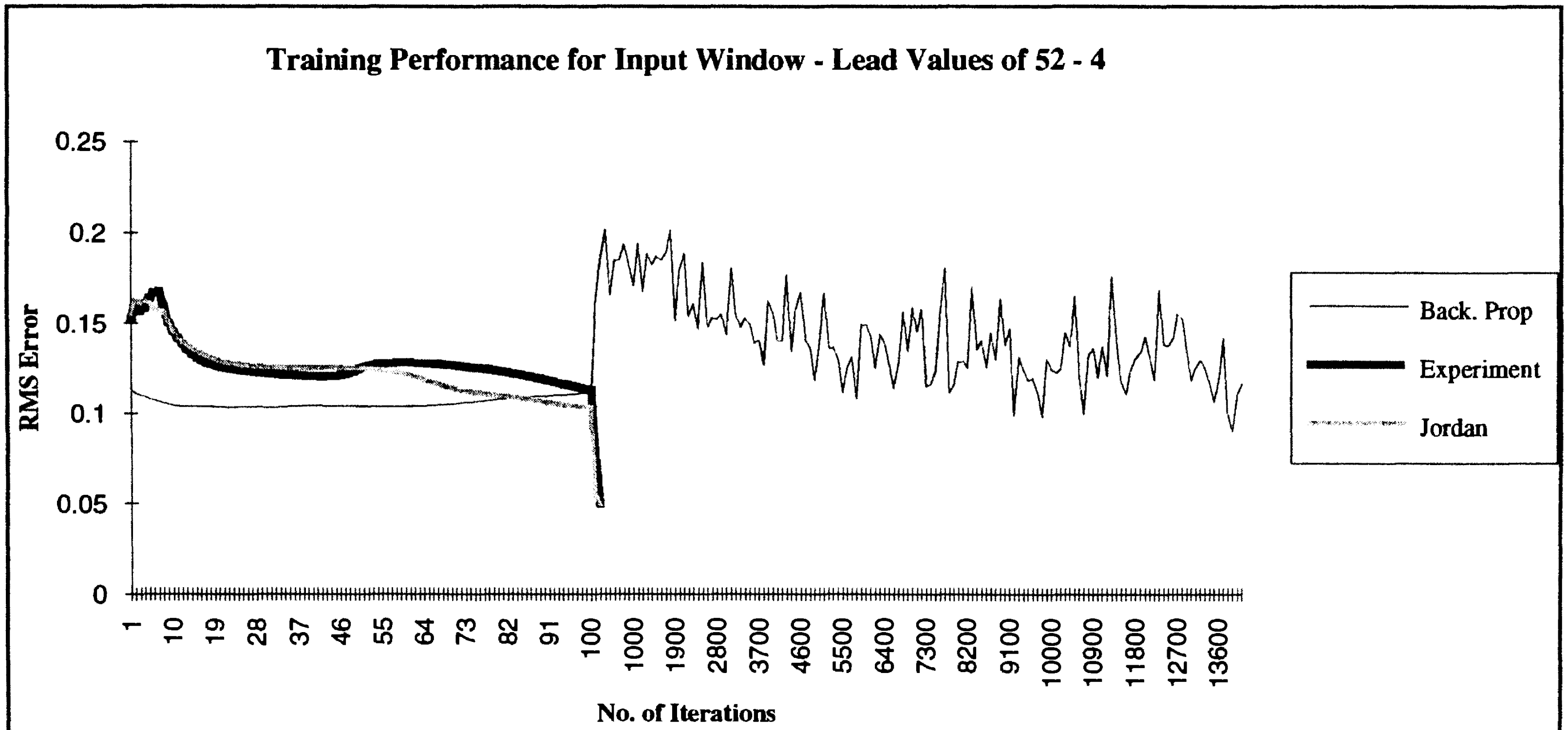

Training Performance for Input Window - Lead Values of 52 - 4
RMS Error
0.25
0.2
0.15
0.1
0.05
0
1
10
19
28
37
46
55
64
73
82
91
100
1000
1900
2800
3700
4600
5500
6400
7300
8200
9100
10000
10900
11800
12700
13600
No. of Iterations
Back. Prop
Experiment
Jordan

Figure 28

Training Performance for Input Window - Lead Values of 52 - 12

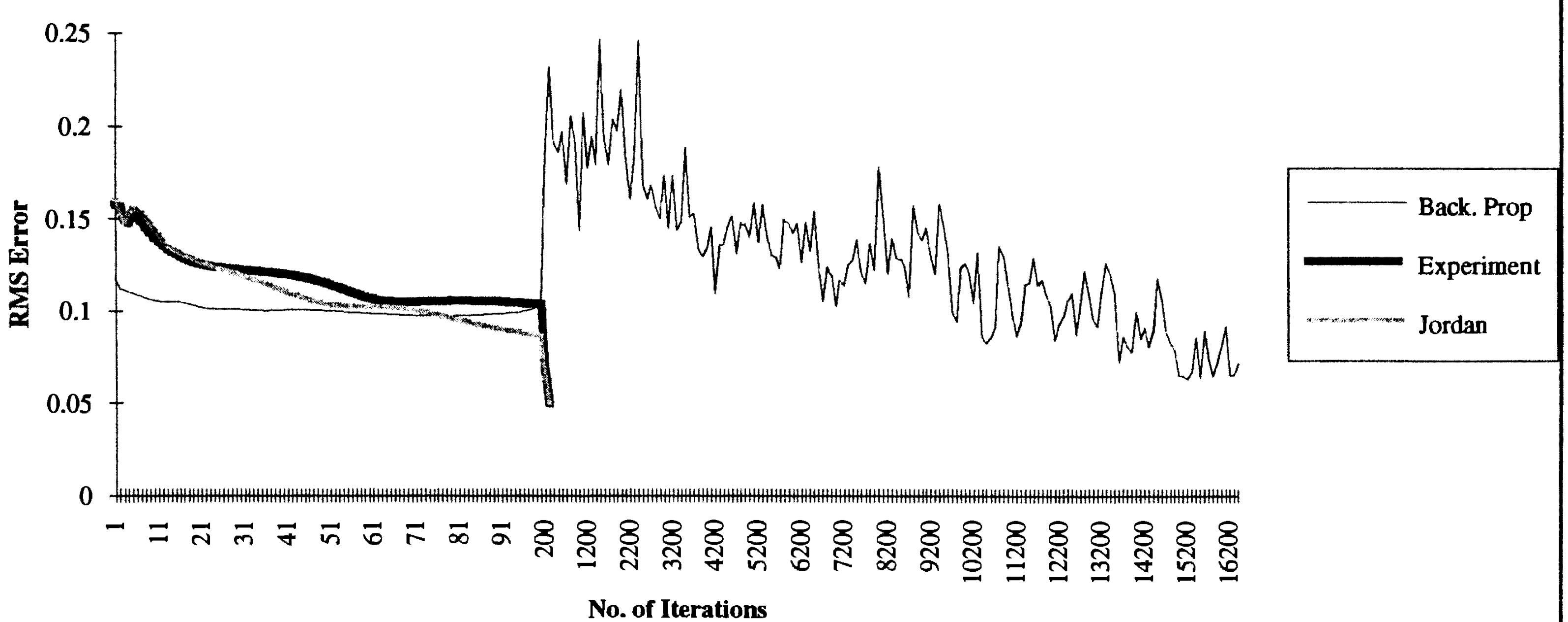


Figure 29

Training Performance for Input Window - Lead Values of 52 - 52

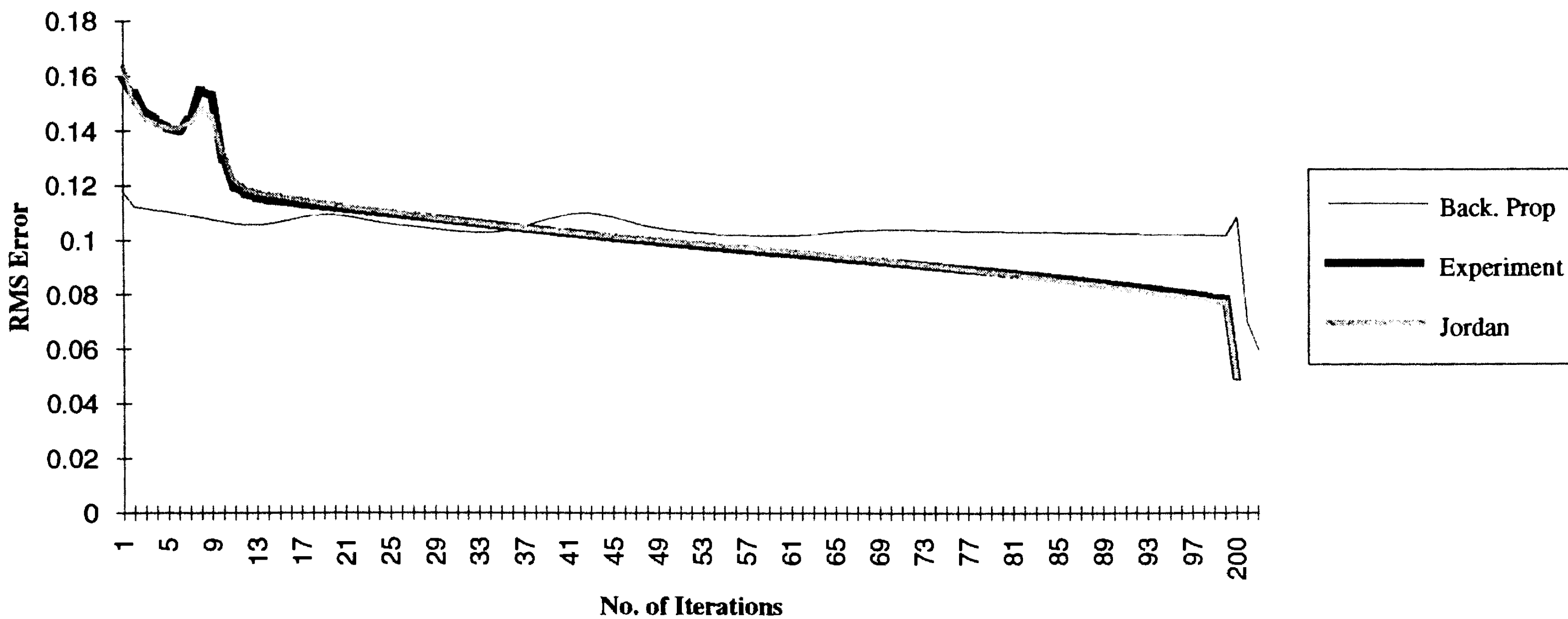


Figure 30

CHAPTER VI

CONCLUSIONS

From the enclosed graphs we can see that the input window does not affect the performance of a network. Though the variation is there in the performance in the network while the input window is growing the final point is again the same for almost all the models. The error for Experimental network is increasing for the lower forecast horizons. But the Experimental net performed better than traditional and backpropagation network for greater forecast horizons. Though the traditional method succeeded in one step ahead forecasting, it failed vary badly for greater forecast horizons. This implies that the short term forecasting can be made using traditional forecasting methods but not the long term. On the average the Jordan network performed better than all the networks. Even for short term forecasting the Jordan network did not produced greater errors when compared to other recurrent or backpropagation networks. The error for this particular network is varying very little. Especially for the long term forecasting the MSE, Theil's U-coefficient and coefficient of variance charts prove that Jordan network is better than all the other models. The Box-Jenkins model of forecasting has very little variance. This does not mean that the forecast of this model is very consistent but not followed the variations of the actual data series in building the initial model. This is proved by highest mean square error, mean absolute percentage error, median absolute percentage error in long term forecasting. The mean deviation charts say that the Box-Jenkins model has the tendency to under-forecast than to over-forecast. The consistency of forecasting can be easily seen in mean absolute deviation and mean deviation charts. From the averages of the each type of error, we can rank the performance of the networks from best to worst order as: Jordan network, Experimental recurrent network, Backpropagation network, Box-Jenkins model forecasting.

BIBLIOGRAPHY

Andreas S. Weigend, Bernardo A. Huberman, and David E. Rumelhart, "*Predicting Sunspots and Exchange Rates with Connectionist Networks,*" Nonlinear Modeling and Forecasting, Addison-Wesley Publishing Company, 1990.

AUTOBOX, "Software User Manual," Automatic Forecasting Systems, Hatboro, PA, 1988.

BrainMaker Professional, "*Neural Network Simulation Software User's Guide and Reference Manual,*" California Scientific Software, 1990.

Egill Masson, and Yih-Jeou Wang, "*Introduction to Computation and Learning in Artificial Neural Networks,*" European Journal of Operation Research, Volume 47, 1990

Gmurman, V. E., "*Fundamentals of Probability Theory and Mathematical Statistics,*" London ILIFFE Books Ltd., 1969.

Hetch-Neilsen, R., "*Theory of Backpropagation Neural Networks,*" International Conference on Neural Networks, Washington, DC, 1989, pp. 177-221.

Hong-Te Su, Thomas J. McAvoy and Paul Werbos, "*Long-Term Predictions of Chemical Process Using Recurrent Neural Networks: A Parallel Training Approach,*" Ind. Eng. Chem. Res. 1992, 31, pp:1338-1352.

Jeannette Lawrence, "*Data Preparation Neural Network,*" AI Expert, November 1991, pp:34-41

Jerome Connor and Les Atlas, "Recurrent Neural Networks and Time Series Prediction," IJCNN, 1991.

Kang, Suh, "*An Investigation of the use of Feedforward Neural Networks for Forecasting,*" Ph.D dissertation, August 1991.

Keith C. Crandall and Robert W. Seabloom, "*Engineering Fundamentals in Measurements Probability Statistics and Dimensions,*" Mc-Graw-Hill Book Company, 1971.

Lapedes, A. and Farber, R., "*How neural net work,*" Neural Information Processing Systems, (Anderson, D. Z., ed.), 442-456, American Institute of Physics, New York.

Lizhong Wu and Mahesan Niranjan, "*Building Energy Data Analysis and Prediction with Recurrent Neural Nets,*" Technical Paper, CUES/F-INFENG/TR.124, University of Cambridge, UK.

ModelWare Professional, "*User's Manual,*" TERANET IA Incorporated, Idaho.

Maureen Caudill and Charles Butler, "*Naturally Intelligent Systems,*" MIT Press, Cambridge, 1989.

McClelland, J. and Rumelhart, D., "*Exploration in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises,*" The MIT Press, Cambridge.

Mulholland, H. and Jones, C. R. "*Fundamentals of Statistics,*" Plenum Press, New York, 1968.

Ranganathan Ramkumar, "*A Test of Noise Filtering Capabilities of Backpropagation Neural Networks,*" Masters Theses, Oklahoma State University, 1986.

Richard P. Lippmann, "*An Introduction to Computing with Neural Nets,*" IEEE Assp. Magazine, April 1987, pp 4-22.

Sharda, R. and Ireland, T., "*An Empirical Test of Automatic Forecasting Systems,*" ORSA/TIMS Meeting, New Orleans, May 1987.

Sharda, R. and Patil, R., "*Neural Networks as Forecasting Experts: An Empirical Test,*" International Joint Conference on Neural Networks, IJCNN-WASH-D.C., Jan 15-19, Vol. II, pp 491-494, 1990

Sharda, R. and Patil, R., "*Connectionist approach to time series prediction: an empirical test,*" Journal of Intelligent Manufacturing, 1992.

Sholom M. Weiss and Casimir A. Kulikowski, "*Computer Systems that Learn,*" Morgan Kauffmann Publishers, Inc., California, 1990.

Smolensky. P., "*Neural and Conceptual Interpretation of PDP Models,*" Parallel Distributed Processing, Vol. 2, McClelland, J. L., and Rumelhart, D. L., and The PDP Research Group Editors, MIT Press, Cambridge, MA, pp: 397, 1986.

Stéphane CANU, Sobral R., Lengelle R., "*Formal Neural Network as an Adaptive Model for Water Demand,*" INNC Paris 1990.

Thomas Miller W., Richard S. Sutton and Paul J. Werbos, "*Neural Networks for Control,*" The MIT Press, Cambridge, 1990.

Wasserman, P. D., "*Neural Computing: Theory and Practice,*" Van Nostrand Reinhold, New York, 1989.

VITA

SIVA RAMA KRISHNA KAVUTURU

Candidate for the Degree of

Master of Science

Thesis: WATER DEMAND FORECASTING AND COMPARISON OF NEURAL NETWORK MODELS

Major Field: Computer Science

Biographical:

Personal Data: Born in Kamareddy, Andhra Pradesh, India, February 20, 1969, son of Prasad Rao K., and Bharathy K.

Education: Graduated from Zilla Parishad High School, Kambhampadu, Andhra Pradesh, India, in May, 1983; received Bachelor of Engineering in Computer Science & Engineering from Osmania University in May, 1990; completed requirements for the Master of Science Degree in Computer Science at Oklahoma State University in December, 1993.

Professional Experience: Software Engineer, Frontier Information Technologies Ltd., Secunderabad, India, August, 1990, to August, 1991; Graduate Research Assistant, Department of Business and Administration, Oklahoma State University, June, 1992 to present.