KFRI Research Report No. 299 ISSN 0970-8103

Comparison of Prediction Models Developed by Statistical and Neural Network Techniques in Applied Forestry Research

M.Sivaram

Statistics Discipline
Division of Forest Information Management System
Kerala Forest Research Institute
(An Institution of Kerala State Council for Science, Technology and Environment)
Peechi -680 653 Kerala

November 2007

KFRI Research Report No. 299

Comparison of Prediction Models Developed by Statistical and Neural Network Techniques in Applied Forestry Research

(Final Report of KFRI 450/04)

M.Sivaram

Statistics Discipline
Division of Forest Information Management System
Kerala Forest Research Institute
(An Institution of Kerala State Council for Science, Technology and Environment)
Peechi -680 653 Kerala
e-mail: sivaram@kfri.org

November 2007

ABSTRACT OF THE PROJECT PROPOSAL

Project Number	KFRI 450/04
Title of the Project	Comparison of prediction models developed by Statistical and Neural Network Techniques in Applied Forestry Research
Objective	To compare the performance of neural network models with statistical models for certain datasets emerged out in applied forestry research
Date of commencement	April 2004
Scheduled date of completion	March 2007
Funding Agency	Kerala Forest Research Institute (Plan Fund)
Investigator	M.Sivaram

CONTENT

ACKNOWLEDGMENT

The author is thankful to

Dr.K.C.Chacko, Programme Coordinator (Division of Training and Extension) and Dr.N.Sasidharan, Scientist-in-charge (NWFP) and Dr.C.N.Krishnankutty, Scientist for permitting to use the data for this study.

Dr.K.Jayaraman, Programme Coordinator (Division of Forest Information Management System) for his useful comments.

Dr.J.K.Sharma, Former Director for his constant encouragement.

Dr.R.Gnanaharan, Director for his active support and going through the manuscript critically.

Project Fellows Mrs.E.P.Sreedevi, Mr.V.Jilesh, Miss.A.A.Hasina and Miss.Jaisymol Thomas for their research assistance.

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Neural Network or Artificial Neural Network (ANN) is a powerful data modeling tool that is able to capture and represent complex input/output relationships whether it be linear or non-linear. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. ANN acquires knowledge through learning and the knowledge is stored within inter-neuron connection strengths known as synaptic weights.

The most common ANN model is the Multilayer Perceptron (MLP). This type of ANN is known as a supervised network because it requires a desired output in order to learn. In MLP with one hidden layer the inputs are fed into the input layer and get multiplied by interconnection weights (synaptic weights) as they are passed from the input layer to the hidden layer. Within the hidden layer, they get summed, then processed by a nonlinear function (usually the sigmoid/hyperbolic tangent). The processed data leaves the hidden layer and finally again processed one last time within the output layer to produce the neural network output. The MLP and many other ANNs learn using an algorithm called backpropagation. With backpropagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "training. The trained neural network is tested and validated for applications.

In this study, the performance of ANN model is compared with traditional statistical models for certain datasets in forestry. The nature of statistical problems that could be considered for the investigation is of regression type (functional approximation) and time series prediction using Auto Regressive Integrated Moving Average (ARIMA) model. For regression problem, three data sets were used. First two data sets are related to the prediction of bark thickness using diameter measurements of two species *Lagerstroemia reginae* and *Acacia caesia*. The third dataset is related to the prediction of the ratio of germination percentage to the viability percentage at different days of germination of teak seeds (*Tectona grandis*). With regard to time series prediction problem, the prices of teakwood in different girth classes were considered. The architecture of ANN used is MLP with one hidden layer for all the problems. The activation function used in the hidden neuron is sigmoid. The error minimization algorithm used is Levenberg-Marquardit algorithm. While the performance of ANN with regression was assessed by the root mean square error, the performance of ANN with ARIMA was assessed by mean absolute percentage error. The performance statistics suggest that ANN is comparable with that of regression and ARIMA models.

1. INTRODUCTION

Most of the prediction problems that arise in biological sciences are complex and nonlinear in nature. The traditional statistical models are basically linear and therefore may not capture the nonlinearity present in the data. Recently, there has been applications of a new generation of methodologies including artificial neural network (ANN), knowledge based systems and genetic algorithms to nonlinear prediction problems. In the presence of alternative techniques, it is essential to evaluate the performance of various techniques for different situations and inform the user community so that appropriate choice of technique is made.

Application of ANN in modeling problems is based on its ability to approximate nonlinear functions (Qi-Bin Zhang *et al*., 2000). The ANN is independent of model and norm that constructs a function by learning (Bishop, 1995). Despite the rapid spread of ANN technique in many fields such as medicine, business, image processing and atmospheric sciences, there have been limited applications of this emerging technique in natural resource management including forestry (Peng and Wen, 1999). Current applications in natural resource management include i) land mapping and classification ii) soil type classification iii) forest growth and dynamics modeling iv) spatial data analysis and GIS modelling v) plant disease dynamics and insect pest management vi) primary production and ecosystems vii) population, community and evolutionary ecology viii) marine ecosystems and ix) global and climate change research (Liu *et al.,* 2003) . The objective of the present study is to apply ANN technique to certain common non-linear prediction problems in forestry and assess its performance in relation to traditional statistical technique.

2. ARTIFICIAL NEURAL NETWORK

ANN is loosely based on biological neural systems such as brain, which are made up of an interconnected system of neurons (also called nodes/processing elements/ units) each possibly having a small amount of local memory. Neurons are connected each other using synapses which are the elementary structural and functional units that mediate interaction between neurons. Acquired knowledge is stored in interneuron connection strengths known as synaptic weights. These are the free parameters of the network and are modified over time by the process of learning. Although there are many definitions for ANN, the definition by Haykin (1999) is described here.

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use.

It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store the knowledge.

ANN jargons vis-à-vis Statistical jargons

ANN and statistical literatures contain many of the same concepts but usually with different terminology. Sometimes the same term or acronym is used in both literatures but with different meanings. Only in very rare cases is the same term used with the same meaning, although some cross-fertilization is beginning to happen. Terminology in both fields is often vague, so precise equivalences are not always possible and therefore loose correspondences are made in Table 1 to facilitate readers.

Table 1. ANN jargons vis-à-vis Statistical jargons

ANN Jargon	**Statistical Jargon**
Architecture	Model
Training, Learning, Adaptation	Estimation, Model fitting, Optimization
Training set	Sample, Construction Sample
Test set, Validation set	Hold-out sample
Pattern, Case	Observation, Case
Input	Independent variables, Predictors, Regressors, Explanatory variables
Output	Predicted values

Training values, Target values	Dependent variables, Responses, Observed values
Training pair	Observation containing both inputs and target values
Shift register, (Tapped) (time) delay (line)	Lagged variable
Errors	Residuals
Noise	Error term
Generalization	Interpolation, Extrapolation, Prediction
Error bars	Confidence interval
Weights, Synaptic weights	(Regression) coefficients, Parameter estimates
Bias	Intercept
the difference between the expected value of a statistic and the corresponding true value (parameter)	Bias

Basic model of a neuron

A neuron is an information-processing unit which is fundamental to the operations of ANN. There are three basic elements in a neuron model (Figure 1).

i. A set of synapses or connecting links each of which is characterized by its own weight or strength. A signal x_i at the input of synapse is connected to the neuron j is multiplied by the synaptic weight b_{ij}.

ii. An adder for summing the input signals weighted by respective synapses of the neuron. These operations are constituted by a linear combiner.

iii. An activation function for limiting the amplitude. Typically the normalized amplitude range of output of a neuron is written as the closed unit interval [0, 1] or alternatively [-1, 1]. An activation function or squashing function limits the permissible amplitude range of output to some finite value. An externally applied bias a_j may also be included for the net input of activation function.

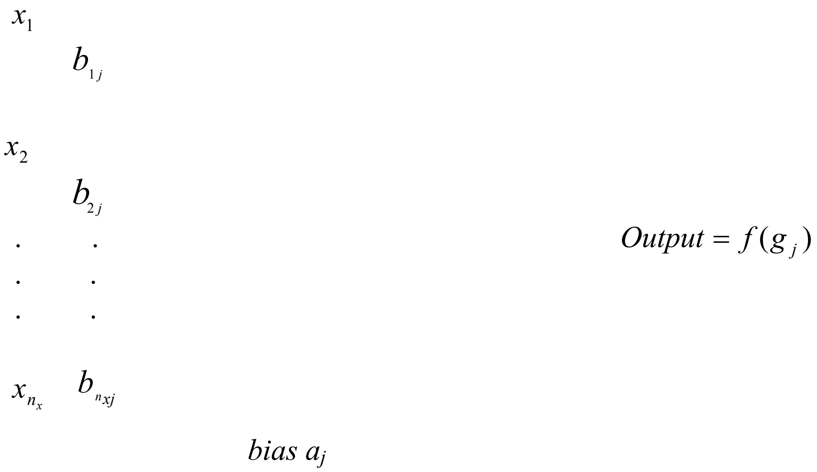


Figure 1. A single artificial neuron contains input -*X*, synaptic weight-*b*, sum-*g*, and transfer function –*f(g)* to produce an output

In mathematical terms a neuron may be represented by an equation

$$g_j = a_j + \sum_{i=1}^{n_x} b_{ij} x_i$$

where n_x is the number of input features (x_i), a_j is a bias for the hidden layer, and b_{ij} is the synaptic weights from the input (i^{th}) layer to the hidden (j^{th}) layer g_j (Haykin,1999; Liu *et al.,* 2003).

Classification of ANN

ANN models can be classified according to the

i) Topology of connection

ii) Learning rule they adopted

iii) Types of data accepted.

Different network topologies

Feedforward and feedback are two important types of network topologies. In feedforward network, connections between neurons do not form cycles. This network responds more quickly to input signals. Here one advantage is that conventional numerical methods can be used for training the network. According to the number of layers in the network there are two types of feedforward networks.

i) Single layer feed forward networks

ii) Multi layer feed forward networks (see Figure 2 in a later section)

In feedback networks connections form cycles or recurrent type.

Learning rules

The most important property of ANN that makes it distinguishable from other data-mining techniques is the ability of the network to learn from its environment and to improve its performance through learning. Learning is defined in the context of ANN, as learning is a process by which the free parameters are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the changes in parameters take place.

Generally, learning is the process by which the network adapts itself to the stimulus (input variable) and eventually after making the proper parameter adjustments to itself, it produces a desired response. A prescribed set of well defined learning rules for the solution of a learning problem is called learning algorithm.

There are different types of learning algorithms which differ from each other in which adjustments to the synaptic weight of a neuron are formulated. Another considerable factor is the manner in which an ANN is made up of a set of interconnected neurons relates to its environment. The fundamental learning paradigms are supervised learning (learning with a teacher) unsupervised learning (learning without a teacher).

Supervised learning

In supervised learning during the training session of the network an input stimulus is applied that results in an output response. This output is compared with a priori desired output signal (target response). If the actual response differs from the target response, the ANN generates an error signal, which is then used to calculate the adjustments that should be made to the network's free parameters (synaptic weights), so that the actual output matches the target output. Here one may think of the teacher as having knowledge of the environment, with that knowledge being represented by a set of input-output examples. The error minimizes possibly to zero. Error correction learning, Competitive learning and Boltzman learning are some of the learning rules that come under supervised learning.

Unsupervised learning

In an unsupervised learning rule there is no target output (no labeled examples of the function) to be learned by the network. This type of learning is needed when the training data lack output values corresponding to the input patterns. The network must group input pattern into clusters based on some input variables. This type of learning in ANN is used for classification purposes. If an input stimulus does not belong to any of the existing groups, a new group may be formed. Even though unsupervised learning does not require a teacher, it requires guidelines to determine how it will form groups. If no guidelines have been given as to what type of features should be used for grouping objects, the grouping may or may not be successful. Memory based learning rule is an example of unsupervised learning (Kartopoulos, 1996). Some of the learning methods and learning algorithms are listed below.

Different learning methods

i. Error correction learning
ii. Memory based learning
iii. Hebbian learning
iv. Competitive learning
v. Boltzman learning
vi. Reinforcement learning
vii. Markovian learning

Different learning algorithms

i. Back propagation algorithm
ii. Winner-Takes-All algorithm
iii. Boltzman algorithm
iv. ART (Adaptive Resonance Theory) algorithm
v. Hopefield learning algorithm

Types of data

Different data types available are categorical data and quantitative data. Categorical variables take only a finite number of possible values and usually there are one or more classes belonging to each category. Both supervised learning with categorical target variables and unsupervised learning with categorical outputs are called classification. Quantitative variables are numerical measurements of some attribute. Supervised learning with quantitative data values is called regression. It is clear that different learning methods and different learning algorithms can be used for different networks. Table 2 presents ANN models analogous to commonly used multivariate statistical techniques.

Table 2. ANN models vis-à-vis Statistical techniques

ANN model	Statistical technique
Supervised learning	Regression, Discriminant analysis
Unsupervised learning, Self-organization	Principal components, Cluster analysis, Data reduction
Competitive learning	Cluster analysis
Hebbian learning	Principal component analysis

Multi-layer Perceptrons (MLP) network

One of the most commonly used supervised ANN is the back propagation feed-forward network. It is also referred as multi-layer perceptrons (MLP) network. This study is mainly concerned with this particular ANN model. This section provides a brief introduction to the major features of the MLP with one hidden layer (but can be extended to two more hidden layers). (Peng and Wen, 1999; Liu *et al*., 2003).

Structure

After the introduction of Rosenblatt's original perceptron, many other perceptron models have been developed. MLP is one such network. MLP nets are composed of many simple perceptrons in a hierarchical structure (Figure 2). There are one or more hidden layers between the input and output layers. All these layers are connected using feedforward topology. In particular, an MLP with no hidden layer is basically a general linear model. The number of hidden layers and number of neurons per layer are not fixed in advance. Each layer may have different number of neurons depending on the applications. The function of hidden neurons in the network is to intervene between the external input and network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher order statistics. In a rather loose sense the network acquires a global perspective despite its local connectivity due to the extra set of synaptic connections and extra dimensions of neural interactions. In MLP there is no lateral connections within any layer and no feed back connections.

In MLP, the number of source nodes in the input layer corresponds to the number of input features (independent variables). The input layer supply respective elements of activation pattern, which constitute the input signals, applied to the neurons (*i.e.* computation nodes) in the second layer. The output signals of the second layer are used as inputs to the third layer and so on. The number of neurons in the output layer corresponds to the number of target classes (*i.e.* dependent variables).

Operation

In the MLP algorithm, the propagation of data through the network begins with an input pattern stimulus at the input layer. The data then flow through and are operated by the network until an output stimulus is yielded at the output layer (Figure 2).

The input features$\left(X_i\right)$ and output classes $\left(Y_k\right)$ are known patterns (*i.e.* observations) in the data. When they are presented to the input nodes and output nodes respectively, the net output to the hidden nodes (g_j) is calculated by

$$g_j = a_j + \sum_{i=1}^{n_x} b_{ij} x_i \qquad (1)$$

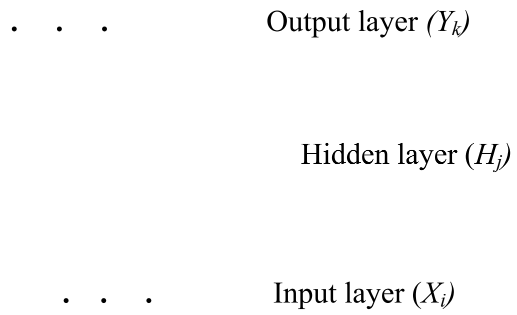


Figure 2. Typical architecture of Multi Layer Perceptron

where n_x is the number of input features (X_i), a_j is a bias (*i.e.* intercept) for the hidden layer, and b_{ij} is the weight (*i.e. coefficient*) from the input (i^{th}) layer to the hidden (j^{th}) layer. Then an activation function is applied to g_j to compute the output from the hidden nodes h_j as

$$h_j = f(g_j) = \frac{1}{1 + e^{-g_j}} \qquad (2)$$

Next, h_j becomes the net input to the output nodes (q_k) which is calculated by

$$q_k = c_k + \sum_{j=1}^{n_h} d_{jk} h_k \qquad (3)$$

where n_h is the number of the hidden nodes, c_k is a bias (*i.e.* intercept) for the output layer, and d_{jk} is the weight from the hidden (j^{th}) layer to the output (k^{th}) layer. Again an activation function is applied to q_k to compute the predicted output p_k:

$$p_k = f(q_k) = \frac{1}{1 + e^{-q_k}} \qquad (4)$$

Learning and training for optimization of weights

Learning and training are fundamental to almost all ANN. Training is the procedure by which the network learns and learning is the end result of that procedure. Learning consists of making systematic changes to the synaptic weights to improve the performance of the network's response to acceptable levels. The aim of the training is to find a set of synaptic weights that will minimize the error.

To accomplish the goal of optimizing the weights in Equations (1) and (3), an error back-propagation algorithm minimizes the objective function defined as

$$E = \frac{1}{2}\sum_{i=1}^{n_x}\sum_{k=1}^{n_k}(y_k - p_k)^2 \qquad (5)$$

where n_k is the number of output classes. This is the sum of squared difference between the predicted output (p_k) and the observed output (y_k) averaged over all input and output patterns. Evaluating the partial derivatives of Equation (5) with respect to the weights in Equations (1) and (3) (*i.e.* b_{ij} and d_{ij}), two error terms are derived for the output layer (δ_k) and for hidden layer (δ_j) as follows:

$$\delta_k = (y_k - p_k)f(q_k) \qquad (6)$$

$$\delta_j = f'(g_j)\sum \delta_k d_{jk} \qquad (7)$$

If the two errors are not sufficiently small, iterative techniques are used to find the optimal set of synaptic weights for the network. Each iteration is considered a training period. By updating the weights, the ANN is said to be learning.

The connection weights are adjusted using the error back-propagation algorithm based on the generalized delta rule such that the weights of two layers are computed iteratively as:

$$\Delta d_{ij}(t+1) = \eta\delta_k h_j + \alpha\Delta d_{jk}(t) \qquad (8)$$

$$\Delta b_{ij}(t+1) = \eta\delta_j x_i + \alpha\Delta b_{ij}(t) \qquad (9)$$

where $\Delta d_{jk}(t+1)$ is the change of the weights between the hidden layer and the output layer at the $(t+1)^{th}$ iteration, $\Delta b_{ij}(t+1)$ is the change of the weights between the input layer and hidden layer at the $(t+1)^{th}$ iteration, $\Delta d_{jk}(t)$ is the change of the weights between the hidden layer and the output layer at the t^{th} iteration, $\Delta b_{ij}(t)$ is the change of the weights between the input layer and hidden layer at the t^{th} iteration, η is the learning rate (analogous to the step-size in a gradient-descent-based optimization), and α is the momentum parameter. The above procedure is repeated for all the training samples until the network errors are less than a predefined threshold or stabilized.

Levenberg-Marquardt algorithm

Although the error backpropagation algorithm has been a significant milestone in neural network research, it has been known as an algorithm with a very poor convergence rate. Among many of the optimization algorithms, the Levenberg-Marquardt (LM) algorithm is widely accepted as the most efficient one in the sense of realization accuracy. LM algorithm is efficient and designed specially for minimizing a sum-of squares error (Bishop, 1995).

Consider the sum-of squares error function in the form

$$E = \frac{1}{2}\sum_n \left(\varepsilon^n\right)^2 = \frac{1}{2}\|\varepsilon\|^2 \qquad (10)$$

where ε^n is the error for the n^{th} pattern, and ε is a vector with elements ε^n. Consider a point b_{old} in weight space is moved to a point b_{new}. If the displacement $b_{new} - b_{old}$ is small then the error vector ε can be expanded to the first order in a Taylor series

$$\varepsilon(b_{new}) = \varepsilon(b_{old}) + \mathbf{Z}(b_{new} - b_{old}) \qquad (11)$$

where the matrix **Z** with elements is defined.

$$(\mathbf{Z})_{n_i} = \frac{\partial \varepsilon^n}{\partial b_i} \tag{12}$$

The error function (Equation 10) can then be written as

$$E = \frac{1}{2}\left\|\varepsilon(w_{old}) + \mathbf{Z}(b_{new} - b_{old})\right\|^2 . \tag{13}$$

If this error is minimized with respect to the new weights w_{new} then

$$b_{new} = b_{old} - \left(\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{T}\varepsilon(b_{old}) \tag{14}$$

By minimizing the sum-of-squares error function (10), the elements of the Hessian matrix take the form

$$(\mathbf{H})_{ik} = \frac{\partial^2 E}{\partial b_i \partial b_k} = \sum_n \left\{ \frac{\partial \varepsilon^n}{\partial b_i}\frac{\partial \varepsilon^n}{\partial b_k} + \varepsilon^n \frac{\partial^2 \varepsilon^n}{\partial b_i \partial b_k} \right\}. \tag{15}$$

If the second term is neglected, then the Hessian matrix can be written in the form

$$\mathbf{H} = \mathbf{Z}^{\mathbf{T}}\mathbf{Z} \tag{16}$$

In the Levenberg-Marquardit algorithm a modified error function is considered as

$$\widetilde{E} = \frac{1}{2}\left\|\varepsilon(b_{old}) + \mathbf{Z}(b_{new} - b_{old})\right\|^2 + \lambda\left\|b_{new} - b_{old}\right\|^2 \tag{17}$$

where the parameter λ governs the step size. For large values of λ, the value of $\left\|b_{new} - b_{old}\right\|^2$ tends to be small. If the modified error (Equation 17) is minimized with respect to b_{new}, then

$$b_{new} = b_{old} - (\mathbf{Z}^{\mathbf{T}}\mathbf{Z} + \lambda\mathbf{I})^{-1}\mathbf{Z}^{\mathbf{T}}\varepsilon(b_{old}) \qquad (18)$$

where, **I** is the unit matrix. In practice, a value must be chosen for λ and this value should vary approximately during the minimization process.

Activation functions in MLP networks

Researchers have used several activation functions in MLP networks.

Logistic: $f(x) = \dfrac{1}{1+e^{-x}}$

Gaussian: $f(x) = e^{-\frac{x^2}{2}}$

Linear: $f(x) = x$

Hyperbolic tangent: $f(x) = tanh(x)$

Threshold: $f(x) = 0$ if $x < 0$,

= 1 otherwise

Usually the same activation function is used for both $f(g_j)$ and $f(q_k)$, although it is possible to use different ones. A commonly used activation function in the MLP network is logistic function, which introduces nonlinearity into the network (Liu *et al.,* 2003).

3. ANN AND REGRESSION MODELS FOR CERTAIN NON-LINEAR PREDICTION PROBLEMS

3.1 Statement of the Problem

In forestry research, prediction equations are often developed relating biometric measurements such as diameter at breast height, wood volume and bark thickness. So far, the primary means of developing prediction equations have involved regression analysis/curve fitting. The mathematical models associated with these methods for developing equations, however, are linear and may fail to predict the turning points that exist in the nonlinear relationship between the variables. Application of ANN in modeling problems is based on its ability to approximate nonlinear functions (Qi-Bin Zhang *et al.*, 2000). This section deals with the application of

ANN technique to certain common non-linear prediction problems in forestry and assesses its performance in relation to traditional regression technique.

3.2 Methodology

Three datasets were used as examples for the comparison of ANN models with regression models.

Data Sets - 1 and 2: Prediction of bark thickness

Developing regression equation for predicting bark thickness is very important in the non-destructive estimation of the availability of bark from certain Non-Wood Forest Product species. The datasets for the prediction of bark thickness (output) using diameter at breast height (dbh) with respect to two species *Lagerstroemia reginae* and *Acacia caesia* considered for evaluation were taken from the research project 'Quantitative Inventory of Non-wood Forest Products in Northern Kerala State' conducted at the Kerala Forest Research Institute. The dbh of the trees was measured for those trees mostly having girth of 10 cm and above. The bark pieces of size 10 cm x 10 cm were collected (pealed out by cutting) at breast height level from the trees and bark thickness measured.

Data Set - 3: Prediction of the ratio of germination percentage to the viability percentage

The dataset for predicting the ratio of germination percentage to the viability percentage at different days of germination of teak seeds (*Tectona grandis*) was obtained from the experiments conducted at the Kerala Forest Research Institute. A teak fruit (botanically a 'drupe') with stony endocarp and felty mesocarp is regarded as a seed. Such a seed may contain a maximum of four mature ovules (true seed) and rarely up to six seeds. Samples, each containing 30 seeds (fruits), were drawn from seed lots obtained from different plantations. Each seed (fruit) was cross cut to see the condition of the seed (true seed) visually. If a seed had at least one locule (out of the 4 locules) filled with fresh creamy true seed as revealed by the colour of the cut surface of the cotyledon, then the seed was regarded as viable and construed as 'one viable seed'. Using this data, viability percentage of seeds was calculated.

$$\text{Viability percentage} = \frac{\text{Number of viable seeds in the sample}}{\text{Total number of seeds in the sample for cutting test}} \times 100$$

Another sample from the same lot was simultaneously put in the germination medium for germination test. The viable seeds started germinating in about 10 days. As in the cutting test if at least one seedling (maximum possible is 4) emerged, germination was construed as one. Even if more than one seedling emerged from a seed (fruit), it was regarded as one count. The germination count was recorded daily for 45 days. From this data the germination percentage was worked out.

$$\text{Germination percentage} = \frac{\text{Number of germinated seeds in the sample}}{\text{Total number of seeds in the sample for germination test}} \times 100$$

The ratio of germination percentage to viability percentage at different days of germination (up to 45 days) was worked out. Thus, this ratio, hereafter, called as germination ratio would have the lowest value at the beginning and highest on 45th day.

Development of ANN and Regression models

Prediction of bark thickness (output) using dbh as the input is considered for evaluating the performance of ANN in relation to traditional regression models. This was done for two species *Acacia caesia* and *Lagerstroemia reginae*. In the case of prediction of ratio of germination percentage to viability percentage, the days of germination is the input. In each of the prediction problem, about 80 per cent of the dataset were used for testing and 20 per cent for the validation of the models. In the regression analysis, all the possible regression equations were tried to find out the best fitting equation (Table 3) and the best one was chosen based on adjusted R^2 value. The nonlinear regression equations such as Logistic model, Gompertz model, Richards model and monomolecular model were also tried but not found to be better fit. Recently, fit statistics like Akaike's Information Criterion (AIC) and Bayesian's Information Criterion (BIC) have been in use. However, these could not be used as the models considered involve only a very few parameters. Burnham and Anderson (2002) do not recommend using these measures unless the ratio of the number of observations to parameters is greater than 40:1. Residual analysis was carried out before finalizing the best model.

With respect to ANN, its structure is represented as follows for the purpose of presentation. ANN (1,1,1) represents ANN with one input, one hidden neuron in the hidden layer and one output. ANN(1,2,1) represents ANN with one input, 2 hidden neurons in the hidden layer and one output. ANN (1,1,1) was applied to the prediction of bark thickness with dbh as input for species *Lagerstroemia reginae* and *Acacia caesia* (Figure 3). ANN (1,1,1) and ANN (1,2,1) were applied each separately to the prediction of germination percentage of teak seeds (Figure 4). In all the three prediction problems input as well as output layer contained only one neuron and logistic function was used as an activation function in the hidden layer. No activation function was used in the output layer. LM algorithm was used for minimizing the error sum of square and optimizing the synaptic weights.

Table 3. **Regression equations tried to find out the best fitting equation**

$Y = a + bD + cD^2$
$ln(Y) = a + b\,D$
$ln(Y) = a + bln(D)$
$ln(Y)=a+bln(D)+c(ln(D))^2$
$Y^{0.5} = a + bD$
$Y = a+ bD^2h$
$ln(Y) = a + bD^2h$
$Y^{0.5} = a+bD^2H$
$lnY = a + bln(D) + cln(H)$
$Y^{0.5} = a + bD + cH$
$Y^{0.5} = a + bD^2 + cH + D^2H$

Y- dependent variable (eg. bark thickness)
D- diameter
H- Height

b_{11}
d_{11}
a_1
c_1

Figure 3. Diagrammatic representation of ANN (1,1,1)

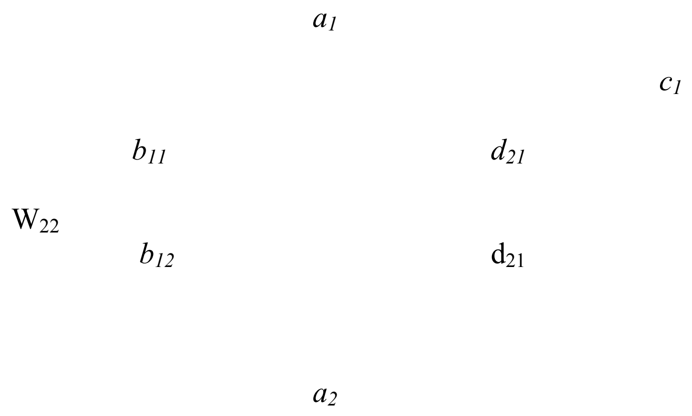


Figure 4. Diagrammatic representation of ANN (1,2,1)

ANN(1,1,1) - ANN with one input, one hidden neuron in the hidden layer and one output.
ANN(1,2,1) - ANN with one input, 2 hidden neurons in the hidden layer and one output.
b_{11} - weight connecting input and first hidden neuron
b_{12} - weight connecting input and second hidden neuron
d_{11} - weight connecting first hidden neuron and output
d_{21} - weight connecting second hidden neuron and output
a_1 - bias of first hidden neuron
a_2 - bias of second hidden neuron
c_1 - bias of output neuron

Assessment of model performance

The performance of ANN and regression models was evaluated based on the root of the mean square error (RMSE).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{n}}$$

where y_i is the observed value, $\hat{y}_i$ is the predicted value and *n* is the number of cases.

All the modeling exercises were performed using Enterprise Miner Module of the SAS software (SAS Institute Inc. 1999-2003).

3.3 Results

Table 4 presents the summary statistics of the variables used for modeling with respect to all the three data sets. The regression co-efficients and the adjusted R^2 values for the best fitting regression models are presented in Table 5. The second degree polynomial with logarithmic transformation on both bark thickness and dbh was found to be the best fit explaining over 98 per cent variation in thickness for both the species *Lagerstroemia reginae* and *Acacia caesia* while the third degree polynomial was found to fit well for the germination ratio of teak seeds explaining 96 per cent variation. Comparative statistics showing the performance of regression and ANN models for all the three datasets are presented in Table 6. With respect to both test and validation datasets, ANN has lower RMSE than regression models in all the prediction problems considered with an exception for validation dataset of *Lagerstroemia reginae*. While the relative gain on applying ANN was less (approximately 1 to 8 per cent) in the case of prediction of bark thickness of *Lagerstroemia reginae* and germination ratio of teak seeds, it was noticeable with regard to prediction of bark thickness of *Acacia caesia* by 39 per cent for training dataset and 64 per cent for validation dataset. Table 7 shows the estimated synaptic weights of the chosen ANN models for all the three datasets. Figures 5 to 7 indicate that the predictions were better using ANN than regression model especially turning points were closely predicted by ANN in the case of *Acacia caesia*. As far as the ANN pertaining to germination ratio of teak seeds is concerned 1-2-1 type provided slightly lower *RMSE* than 1-1-1 type with a relative gain of only about 3 per cent.

Table 4. Descriptive statistics on the variables used in different datasets for the prediction problems

Dataset	Variables	Minimum	Maximum	Mean	SD
Dataset -I *Lagerstroemia reginae*					
Training (*n* =91)	dbh (cm)	3.82	51.91	21.14	11.11
	Thickness (cm)	0.10	0.90	0.50	0.19
Validation (*n* =23)	dbh (cm)	7.96	38.22	18.90	8.55
	Thickness (cm)	0.30	0.90	0.50	0.15
Dataset –II *Acacia caesia*					
Training (*n* =98)	dbh (cm)	1.59	38.55	8.58	5.03
	Thickness (cm)	0.10	1.00	0.49	0.18
Validation (*n* =23)	dbh (cm)	1.91	28.66	9.22	5.31
	Thickness (cm)	0.30	9.22	0.47	0.14
Dataset –III Germination Test					
Training (*n* =368)	Days for germination	1.00	45.00	23.32	13.03
	Germination Ratio	0.00	0.69	0.27	0.25
Validation (*n* =92)	Days for germination	1.00	45.00	13.03	14.24
	Germination Ratio	0.00	0.60	0.19	0.20

Table 5. Parameter estimates of Regression model applied to predict bark thickness of *Lagerstroemia reginae* and *Acacia caesia* and the germination ratio of teak seeds

Regression model	Parameter estimates						$Adj.R^2$
	b	$SE(b)$	c	$SE(c)$	d	$SE(b)$	
Dataset 1- ***Lagerstroemia reginae*** $\log(T+2) = b(\log D) + c(\log D)^2$	0.5803	0.0053	-0.0888	0.0166	-	-	0.9934
Dataset 2- ***Acacia caesia*** $\log(T+2) = b(\log D) + c(\log D)^2$	0.7892	0.0233	-0.1589	0.0101	-	-	0.9877
Dataset 3- Germination Ratio $R = b(D) + c(D)^2 + d(D^3)$	-0.0027	0.0013	0.0010	0.0001	1.6E-5	1.421E-6	0.9600

D-dbh (in cm); T-Thickness (in cm); R-Ratio of germination of teak seeds.

Table 6. Comparison of fit statistics obtained on applying Statistical and Neural network models for the prediction of bark thickness of *Lagerstroemia reginae* and bark thickness of *Acacia caesia* and germination ratio of teak seeds

Dataset	Model	Training data	Validation data
Dataset 1- ***Lagerstroemia reginae***	Regression model	0.184662	0.128062
	Neural Network model (1-1-1)	0.169411	0.129228
Dataset 2- ***Acacia caesia***	Regression model	0.217025	0.269072
	Neural Network model (1-1-1)	0.131529	0.095917
Dataset 3- **Germination ratio**	Regression model	0.06558	0.053257
	Neural Network model (1-1-1)	0.065111	0.052467
	Neural Network model (1-2-1)	0.064622	0.051022

Table 7. Parameter estimates of Neural Network model (1-1-1) applied to predict bark thickness of *Lagerstroemia reginae* and *Acacia caesia* and the germination ratio of teak seeds

Neural Network model with one neuron in hidden layer	b_{11}	a_1	d_{11}	c_1
Dataset 1- ***Lagerstroemia reginae***	-0.0873	-0.7309	-1.3107	0.6256
Dataset 2- ***Acacia caesia***	-0.5143	-0.4191	-2.4390	1.2059
Dataset 3- **Germination ratio**	-0.1374	2.5860	-0.5622	0.4993

b_{11} - weight connecting input and hidden neuron; a_1 - bias of hidden neuron

d_{11} - weight connecting hidden neuron and output; c_1 - bias of output neuron

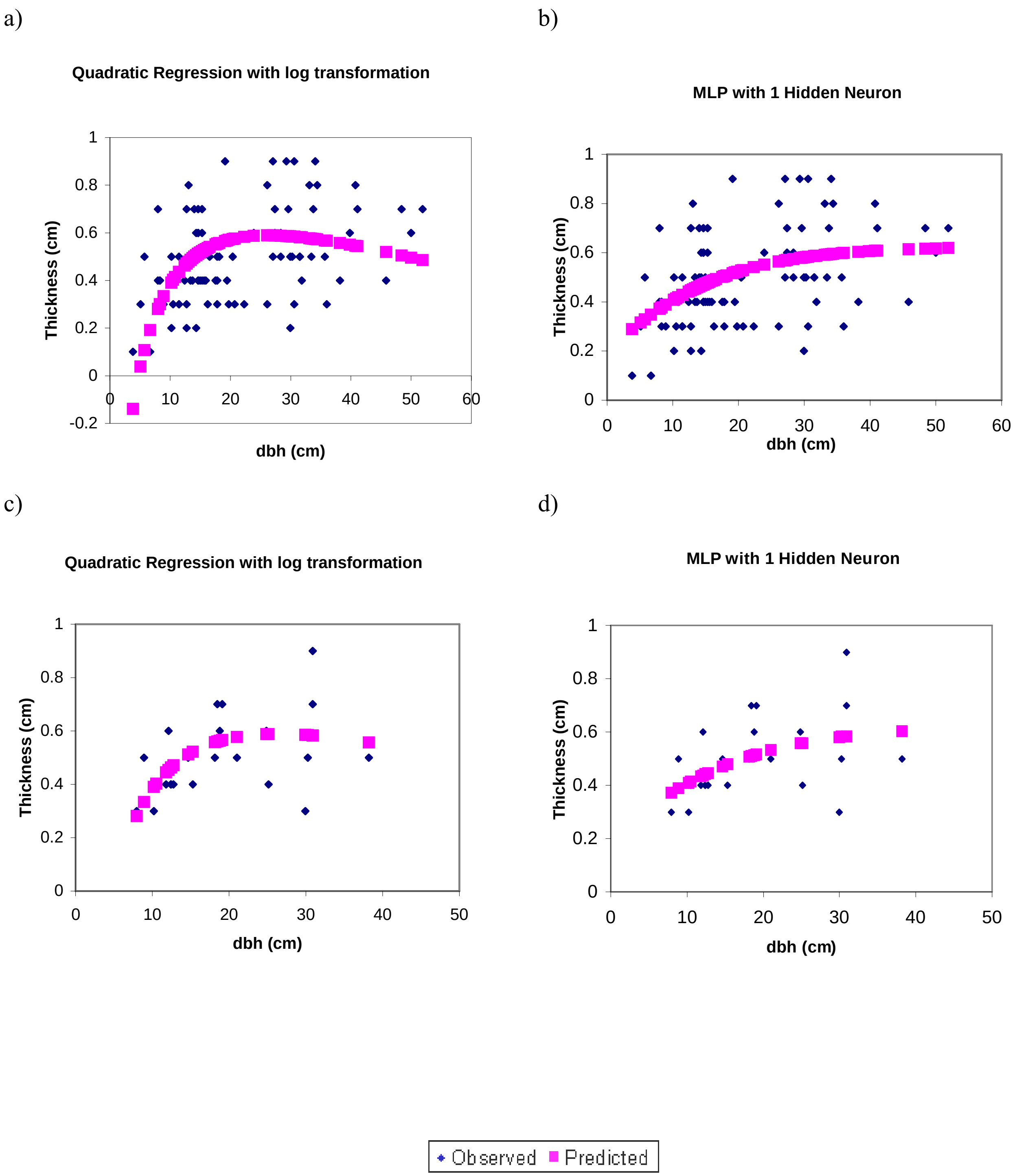


Figure 5. Comparison of predicted bark thickness obtained using Regression and Neural Network models with that of observed bark thickness of *Lagerstroemia reginae*

a) Regression model (training) b) Neural Network model (training)

c) Regression model (validation) d) Neural Network model (validation)

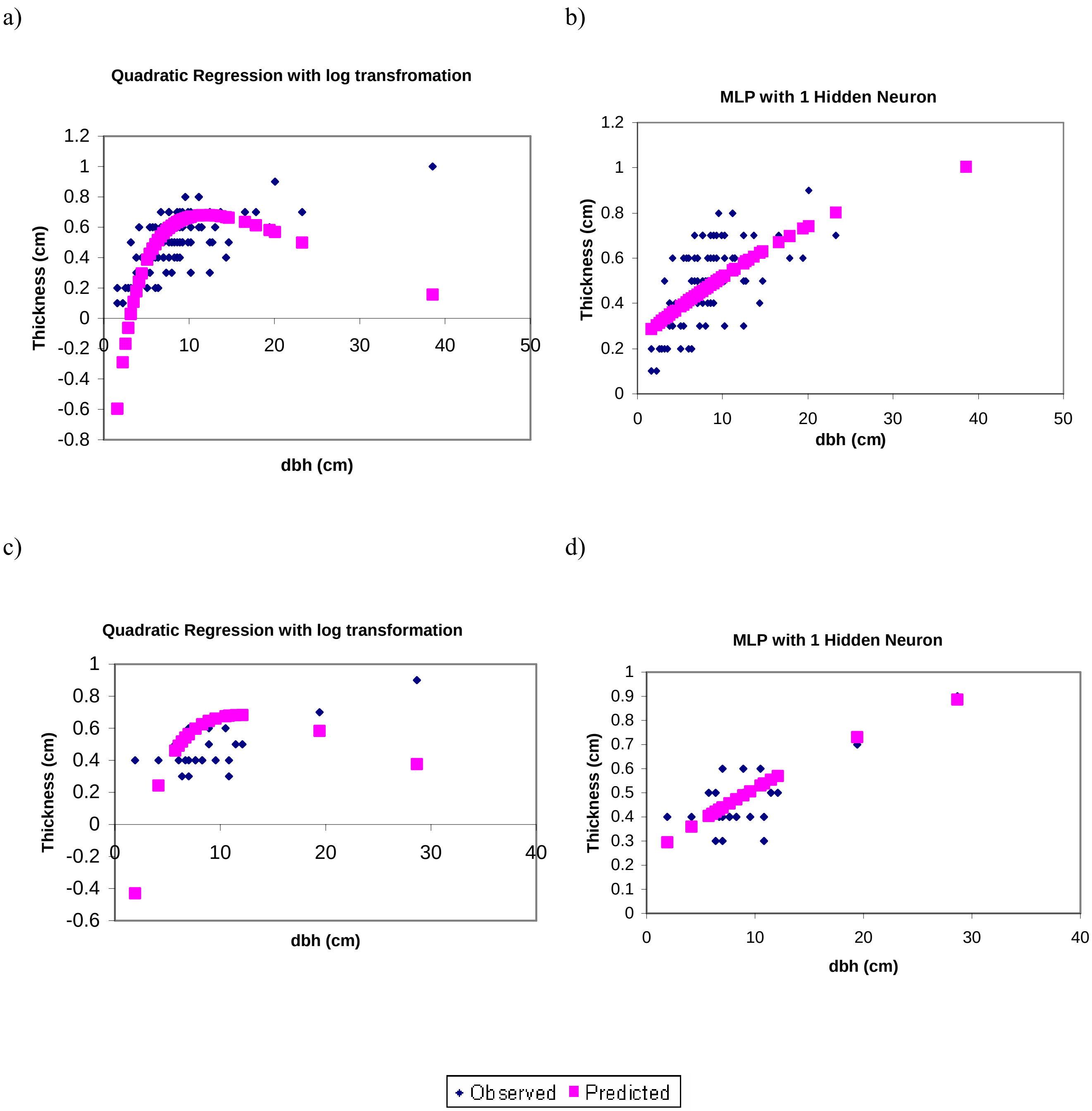


Figure 6. Comparison of predicted bark thickness using Regression and ANN models with that of observed thickness of *Acacia caesia*
a) Regression model (training) b) ANN model (training)
c) Regression model (validation) d) ANN model (validation)

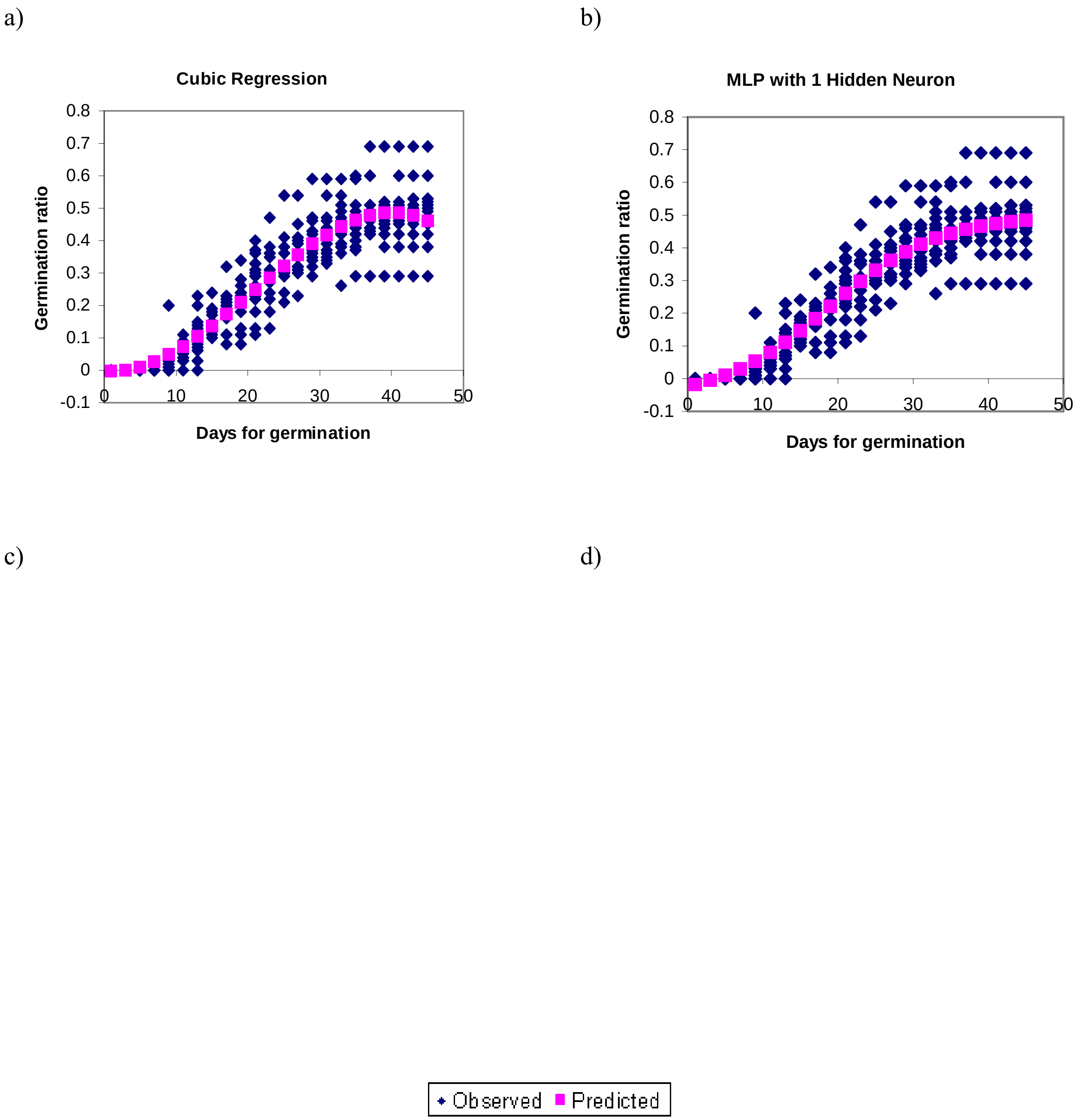


Figure 7. Comparison of germination ratio of teak seeds using Regression and Neural Network models with that of observed germination ratio
a) Regression model (training) b) Neural Network model (1-1-1) (training)
b) Regression model (validation) d) Neural Network model (1-1-1)(Validation)

4. ANN AND ARIMA MODEL FOR FORECASTING PRICES OF TEAKWOOD

4.1 Statement of the Problem

A time series is a sequence of observations taken sequentially in time. The succession of values in a time series is usually influenced by some external factors. If the information on the influencing factors is not known, only the past values of the time series itself can be used to build a mathematical model for forecasting future values. In traditional statistical forecasting, several models emerged from time to time. The most popular forecasting model is Auto regressive integrated moving average (ARIMA) model due to Box and Jenkins (1994). These mathematical models, however, are linear and may fail to forecast the turning points because in many cases the data they model may be highly non-linear. Recently, there have been applications of artificial neural network (ANN) to time series forecasting problems in variety of fields ranging from forecasting of rainfall to stock market prices (Lin, 1995; Rech, 2002; Guhathakurta, 2006). This is because ANN is free from assumptions including linearity and robust to missing observations. In this section, an attempt has been made to compare the performance of ANN with ARIMA in forecasting prices of teakwood in different girth classes.

4.2 Methodology

ARIMA model

ARIMA model is a powerful model for describing stationary and nonstationary time series. The application of the ARIMA methodology for the study of time series analysis is due to Box and Jenkins (1994). The basic concepts involved in ARIMA are described in Appendix. ARIMA model is usually denoted as ARIMA (p, d, q), which can be expressed as

$$z_t = \phi_1 z_{t-1} + ... + \phi_p z_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_p a_{t-q}$$

where $z_t = \nabla^d y_t$

p = order of the autoregressive process

d = degree of differencing

q = order of the moving average process

The parameters involved in the above model can be estimated by Maximum Likelihood Method (MLE). The details are given in Box and Jenkins (1994).

In Box-Jenkins methodology of ARIMA modeling, it must be first established that a given time series is stationary before trying to identify the order of autoregressive and moving average processes. This is done by usual analysis of autocorrelation and partial autocorrelation. The next step is to examine the sample autocorrelation and partial autocorrelation of the first differenced series to determine the number of parameter values (p, d, q) to be involved in the models. The autocorrelation function of an autoregressive process of order p tails off while, its partial autocorrelation function has a cut off after lag p. Conversely, the autocorrelation function of a moving average process of order q has a cut off after lag q, while its partial autocorrelation tails off. If both the autocorrelations and partial autocorrelations tails off, a mixed process is suggested. Furthermore, the autocorrelation function for a mixed process, contain p^{th} order autoregressive component and q^{th} order moving average component, is a mixture of exponentials and damped sine waves after the first $q-p$ lags. Conversely, the partial autocorrelation function for a mixed process is dominated by a mixture of exponentials and damped sine waves after the first $p-q$ lags.

Feed Forward Neural Network models

A variety of neural net architectures has been examined for addressing the problem of time series prediction. These architectures include: MLP, recurrent networks and radial basis functions (Kajitani, 2005). In this study, we apply MLP to solve the forecasting problem. We can take a set of $k-1$ values $y_{t-1}, y_{t-2}, ..., y_{t-k+1}$ to be the input to a feed-forward network, and use the next value y_t as the target for the output of the network, as indicated in Figure 8. By stepping along the time axis, we can create a training data set consisting of many sets of inputs values with corresponding target values. Once the network has been trained, it can be presented with a set of observed values $y_{t-1}, y_{t-2}, ..., y_{t-k+1}$ and used to make prediction for y_t. This is called one step ahead prediction. If the prediction themselves are cycled around to the inputs of the network, then prediction can be made to further points y_{t+1} and so on. This is called multi-step ahead prediction, and is typically characterized by a rapidly increasing divergence between the predicted and observed values as the number of steps ahead is increased due to accumulation errors. The above approach is easily generalized to deal with several time – dependent variables in the form of a time-dependent vector $y(t)$. As in the case of ARIMA modeling, detrending is required and without it, a trained network would have poor performance.

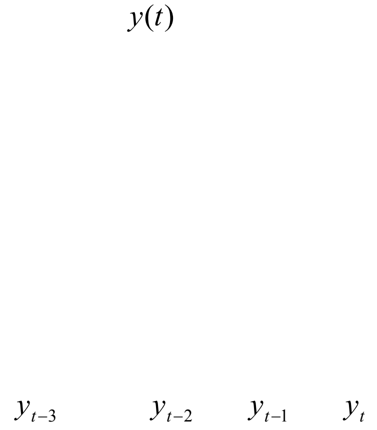


Figure 8. Sampling of a time series at discrete steps can be used to generate a set of training data for a feed-forward network. Successive values of the time-dependent variable $y(t)$, given by $y_{t-1}, y_{t-2}, ..., y_{t-k+1}$, form the inputs to a feed-forward network, and the corresponding target values is given by y_t.

Dataset

The data set considered for the study is on current prices of teakwood in different girth classes for the period from 1943 to 2006. These prices are weighted average prices for the quantity of timber sold. The current prices were collected from different Timber Sales Divisions of the Kerala Forest Department in Kerala State. Five different girth classes were considered based on mid girth measurements *viz*., Export Class (185 cm and above), Girth Class I (150-184 cm), Girth Class II (100-149 cm), Girth Class III (75-99 cm) and Girth Class IV (60-74 cm). The data relating to the period 1943 to1994 is from Krishnankutty (1998) and data for the period from 1994 to 1998 from Krishnankutty *et al* (2003). Data for the period from 1998 to 2006 were collected and compiled during the study period.

Identification of ARIMA model

In order to find the suitable ARIMA model, first the sample autocorrelation of prices were examined for different girth classes. The estimated autocorrelation function does not die out rapidly suggesting that the underlying process should be treated as nonstationary. Therefore, the first differencing was done to remove the trend. The autocorrelation and partial autocorrelation were again worked out for the differenced series. Figures 9-13 reveal that the autocorrelations for the differenced data are found to fall within 2 times of standard error. This indicates that the autocorrelations are not found statistically significant. Because the sample autocorrelation or partial autocorrelation function neither tails off nor cuts off, it appears that the mixed model is required. Therefore, the parameter values of p, d and q were varied and different combinations of ARIMA models examined. Based on the model selection criteria such as Mean Absolute Percentage Error (MAPE) the best ARIMA model was arrived at.

Identification of ANN structure

The appropriate MLP network was identified by trial and error method, as there is no established procedure. The neural network model of the following form was conceived.

$$\text{ANN}(i, j, k) = \text{ANN}\ (y_{t-1}, y_{t-2}, \ldots, y_{t-k+1})$$

where *i* is the number of inputs, *j* number of neurons in the hidden layer and *k* lag period. The input values of these parameters for ANN modeling are broadly based on autocorrelation co-efficient between successive year prices. For illustration, the architecture of ANN (3,3,3) is depicted in Figure 14. The ANN was applied to actual prices, log transformed prices, prices after first order differencing and to prices obtained after linear detrending. The synaptic weights were optimized using Levenberg algorithm. All the modeling exercises were performed using Enterprise Miner module of SAS software (SAS Institute Inc. 1999-2003).

9a) Autocorrelations for prices of teak in Export Class

```
LAG      COVARIANCE      CORRELATION      -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1           STD ERROR

     0         19782206           1.00000
  ********************                         0
     1         -7679364           -.38820                   ********
.                         0.166667
     2          3586094           0.18128                   .
  ****     .                         0.190131
     3         -3440761           -.17393                   .     ***
.                        0.194873
     4          3587825           0.18137                   .
  ****     .                         0.199138
     5          -822844           -.04160                   .       *
.                        0.203675
     6           570667           0.02885                   .
  *        .                         0.203911
```

```
. MARKS TWO STANDARD ERRORS
```

9b) Partial Autocorrelations for prices of teak in Export Class

```
                    LAG    CORRELATION      -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1

                      1     -0.38820                      ********
.
                      2      0.03601                      .
  *     .
                      3     -0.10860                      .     **
.
                      4      0.09024                      .
  **    .
                      5      0.08626                      .
  **    .
                      6      0.01405                      .
.
```

Figure 9a) Autocorrelations of prices of teakwood after first differencing (Export Class)
b) Partial Autocorrelations of prices of teakwood after first differencing (Export Class)

10a) Autocorrelations of prices of teakwood in Girth Class I

```
LAG      COVARIANCE      CORRELATION      -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1            STD ERROR

     0        7134965         1.00000
  ********************                         0
     1       -1394415        -.19543                        .****
.                          0.125000
     2        1251287         0.17537                       .
  ****.                           0.129686
     3        -290222        -.04068                        .    *
.                          0.133341
     4         916650         0.12847                       .
  ***  .                          0.133534
     5         460550         0.06455                       .
  *    .                          0.135452
     6        -223654        -.03135                        .    *
.                          0.135932
```

```
      7      -315201      -.04418            .    *
.                       0.136044
      8       821372      0.11512            .
** .                          0.136268
      9       735705      0.10311           .
**  .                         0.137780
     10      1859529      0.26062           .
*****.                        0.138980
     11      -348834      -.04889           .    *
.                       0.146418
     12       402464      0.05641           .
*   .                         0.146672
     13       706885      0.09907           .
**  .                         0.147011
     14       334246      0.04685           .
*   .                         0.148051
     15      -140917      -.01975           .
.                       0.148282

                            . MARKS TWO STANDARD
ERRORS
```

10b) Partial Autocorrelations of prices of teakwood in Girth Class I

```
            LAG   CORRELATION    -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1
              1   -0.19543                        .****
.
              2    0.14263                        .
 ***   .
              3    0.01760                        .
.
              4    0.10559                        .
 **    .
              5    0.11416                        .
 **    .
              6   -0.03751                        .   *
.
              7   -0.08918                        .  **
.
```

```
              8        0.09790                      .
** .
              9        0.15148                      .
*** .
             10        0.31029                      .
******
             11        0.06000                      .
* .
             12       -0.05455                      .   *
.
             13        0.04432                      .
* .
             14        0.00186                      .
.
             15       -0.04653                      .   *
.
```

Figure 10 a) Autocorrelations of prices of teakwood after first differencing (Girth Class I) b) Partial Autocorrelations of teakwood after first differencing (Girth Class I)

11a) Autocorrelations of prices of teak in Girth Class II

```
     LAG       COVARIANCE          CORRELATION          -1 9 8 7 6 5 4 3 2 1
0 1 2 3 4 5 6 7 8 9 1                    STD ERROR

       0         3903815            1.00000
  ********************                           0
       1          595491            0.15254                            .
  ***  .                                0.125000
       2       71030.744            0.01820                            .
.                                0.127876
       3      -35011.517            -.00897                            .
.                                0.127916
       4          633065            0.16217                            .
  ***  .                                0.127926
       5          432826            0.11087                            .
  **   .                                0.131098
       6      -58667.377            -.01503                            .
.                                0.132555
```

```
    7        -530533        -.13590                    . ***
.                        0.132582
    8        -155469        -.03982                    .   *
.                        0.134741
    9         628582        0.16102                    .
 *** .                          0.134925
   10        1642299        0.42069                   .
 ********                       0.137895
   11         212514        0.05444                   .
 *     .                        0.156670
   12         192154        0.04922                   .
 *     .                        0.156966
   13         462633        0.11851                   .
 **    .                        0.157207
   14         607672        0.15566                   .
 ***   .                        0.158596
   15      46357.311        0.01187                   .
.                       0.160966

                                       . MARKS TWO STANDARD
ERRORS
```

11b) Partial Autocorrelations of prices of teak in Girth Class II

```
LAG   CORRELATION   -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1

1     0.15254       .
***   .
2     -0.00519      .
.
3     -0.01123      .
.
4     0.16928       .
***   .
5     0.06382       .
*     .
6     -0.04649      .  *
.
7     -0.12829      .  ***
.
```

```
            8     -0.02445                    .
 .
            9      0.15655                    .
  ***   .
           10      0.41423                    .
  ********
           11      0.00059                    .
 .
           12      0.05495                    .
  *     .
           13      0.06491                    .
  *     .
           14     -0.04420                    .   *
 .
           15     -0.09741                    .  **
 .
```

Figure 11 a) Autocorrelations of prices of teakwood after first differencing (Girth Class II) b) Partial Autocorrelation for teakwood prices after first differencing (Girth Class II)

12a) Autocorrelations of prices of teak in Girth Class III

```
LAG      COVARIANCE      CORRELATION      -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1            STD ERROR

    0        2588888         1.00000
 ********************                             0
    1         710807         0.27456                    .
 *****                          0.125000
    2        -359908         -.13902                    .  ***
.                           0.134092
    3        -237723         -.09182                    .   **
.                           0.136326
    4         408129         0.15765                    .
 ***  .                         0.137289
    5         358770         0.13858                   .
 ***   .                        0.140089
    6        -511912         -.19773                   .  ****
.                          0.142215
```

```
    7          -244103         -.09429                      .    **
.                          0.146447
    8          -262890         -.10155                      .    **
.                          0.147393
    9           791541         0.30575                      .
 ******                           0.148482
   10           748566         0.28915                      .
 ******                           0.158013
   11           235751         0.09106                    .
 **      .                        0.166075
   12           113693         0.04392                    .
 *       .                        0.166853
   13           161774         0.06249                    .
 *       .                        0.167034
   14           344813         0.13319                    .
 ***     .                        0.167398
   15         8952.639         0.00346                    .
.                        0.169046

                                            . MARKS TWO STANDARD
ERRORS
```

12b) Partial Autocorrelations for Girth Class III

```
              LAG      CORRELATION      -1 9 8 7 6 5 4 3 2 1 0 1 2
     3 4 5 6 7 8 9 1

                1         0.27456                                  .
   *****
                2        -0.23188                                  *****
 .
                3         0.02156                                  .
 .
                4         0.17152                                  .
   *** .
                5         0.01718                                  .
 .
                6        -0.23339                                  *****
 .
```

** .

.

****

* .

*** .

* .

* .

.

****.

7	0.11222	.
8	-0.21532	.****
9	0.44780	.
10	0.05075	.
11	0.13334	.
12	0.07205	.
13	0.03331	.
14	-0.12382	. **
15	0.21711	.

Figure 12a) Autocorrelations of prices of teakwood after first differencing (Girth Class III) b) Partial Autocorrelations for teakwood prices after first differencing (Girth Class III)

13a) Autocorrelations of prices of teak in Girth Class IV

```
LAG      COVARIANCE      CORRELATION      -1 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 1            STD ERROR


   0         1735334         1.00000
 ********************                  0
   1       54295.394         0.03129                        .
 *     .                           0.166667
   2          146228         0.08427                        .
 **    .                           0.166830
```

```
    3          -271817          -.15664                       .     ***
.                        0.168008
    4           131932          0.07603                       .
 **     .                           0.172017
    5          -146221          -.08426                       .      **
.                        0.172947
    6          -247593          -.14268                       .     ***
.                        0.174084


                         .  MARKS  TWO  STANDARD  ERRORS
```

13b) Partial Autocorrelations of prices of teak in Girth Class IV

```
                    LAG     CORRELATION      -1  9  8  7  6  5  4  3  2  1  0
1  2  3  4  5  6  7  8  9  1

                      1         0.03129                       .
 *      .
```

```
                    2          0.08337                    .
   **      .
                    3         -0.16295                    .     ***
 .
                    4          0.08318                    .
   **      .
                    5         -0.06719                    .       *
 .
              6          -0.18145                         .    ****
        .
```

Figure 13 a) Autocorrelations of prices of teakwood after first differencing (Girth Class IV) b) Partial Autocorrelations of pries of teakwood after first differencing (Girth Class IV)

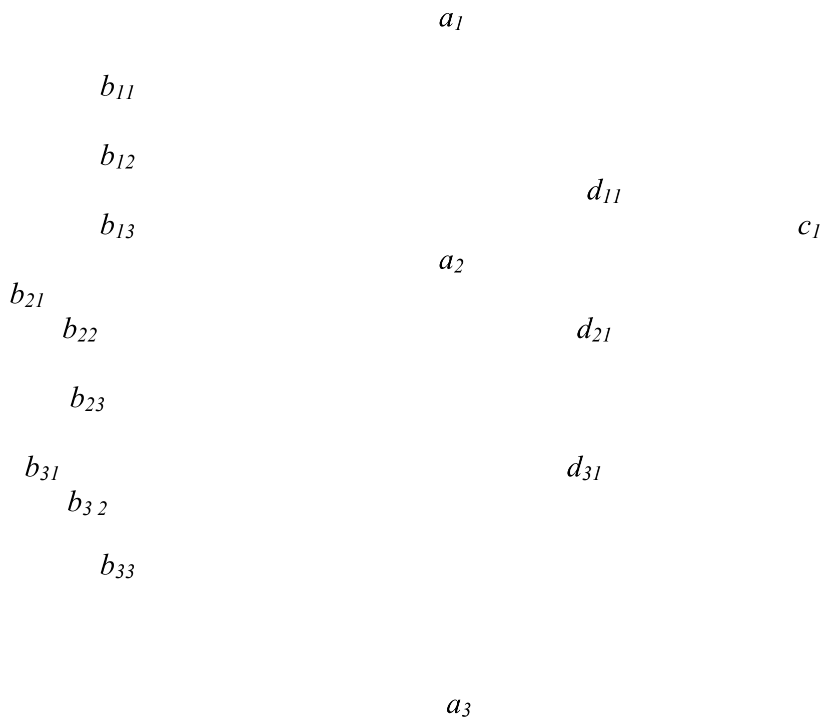


Figure 14. Diagrammatic representation of ANN (3,3,3)

H1, H2 and *H3* are hidden neurons in the hidden layer
b_{11} - weight connecting y_{t-1} and *H1*
b_{12} - weight connecting y_{t-1} and *H2*
b_{13} - weight connecting y_{t-1} and *H3*
b_{21} - weight connecting y_{t-2} and *H1*
b_{22} - weight connecting y_{t-2} and *H2*
b_{23} - weight connecting y_{t-2} and *H3*
b_{31} - weight connecting y_{t-2} and *H1*
b_{32} - weight connecting y_{t-3} and *H1*
b_{33} - weight connecting y_{t-3} and *H1*
d_{11} - weight connecting *H1* and y_t
d_{21} - weight connecting *H2* and y_t
d_{31} - weight connecting *H3* and y_t
a_1 - bias of *H1*
a_2 - bias of *H2*
a_3 - bias of *H3*
c_1 - bias of y_t

Performance Evaluation of ARIMA and ANN models

This section explains the goodness-of-fit statistics used to measure how well different models fit the data. Statistics of fit were computed using the actual and predicted or forecasted values for observations in the evaluation domain. We could not have holdout samples for validation, as the price data was available for only limited number of years. The various statistics of fit reported for model selection are as follows. In these formulae, *n* is the number of non-missing observations and *m* is the number of fitted parameters in the model.

Mean Square Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Square Error (RMSE)

$$RMSE = \sqrt{MSE}\ .$$

Mean Absolute Percent Error (*MAPE*)

$$MAPE = \frac{100}{n}\sum_{t+1}^{n}\left|\frac{(y_t - \hat{y}_t)}{y_t}\right|.$$

The summation ignores observations where $y(t) = 0$

Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - \hat{y}_t\right|$$

Akaike's Information Criterion (AIC)

$$AIC = n\ln(MSE) + 2m$$

4.3 Results

ARIMA models

The possible ARIMA models developed for prices of teakwood in different girth classes are presented in Table 8 along with fit statistics. The best ARIMA models of the form ARIMA (p,d,q) based on fit statistics are ARIMA (1,2,2) for export glass and girth class III. ARIMA(1,2,1) for girth class I, girth class II and IV. It was also attempted to implement

ARIMA models after log transformation. But such an attempt did not yield better results. The functional form of the chosen ARIMA models and values of the estimated parameters are shown in Table 9.

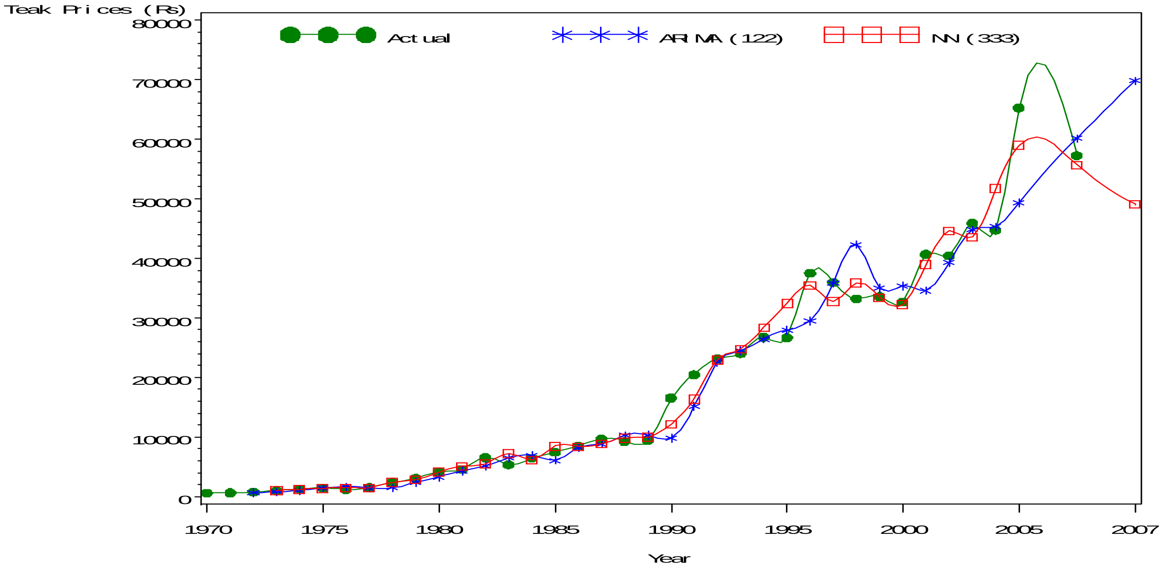
ANN models

The various combinations of the ANN models for different girth classes are presented in Table 10 along with fit statistics for raw price data and log transformed price data. Neither the linear differencing nor first order differencing improved the ANN predictions. The best ANN models based on RMSE and MAPE is ANN (3,3,3) for all girth classes. The synaptic weights of these best models are presented in Table 11. However, with respect to girth class IV, ANN could not predict next year price sensibly. Therefore, ANN (2,2,2) was considered for forecasting next year price.

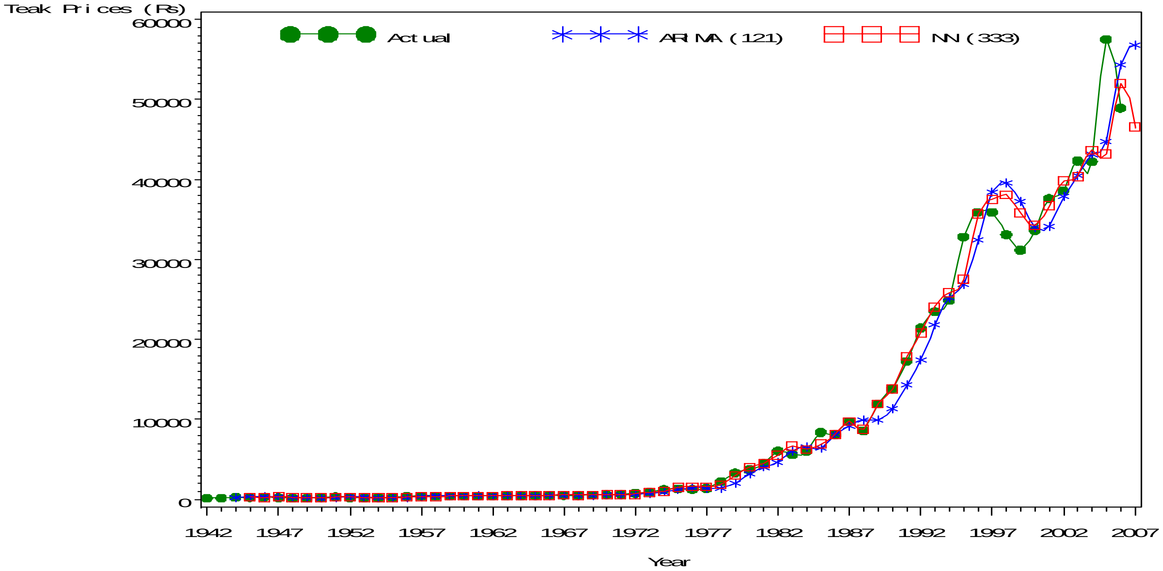
Comparison of ARIMA and ANN models

Based on the best possible models, the simulation was undertaken. The simulated prices obtained using the chosen ARIMA and ANN models are depicted in Figure 15. When the MAPE was compared among the models, the models developed by ANN technique appears to perform better than the ARIMA models, for the log transformed prices (Figure 16). However, with respect to short term price prediction (next year price prediction), the ANN technique appears to produce under estimated prices when compared to previous year price and the price as estimated from ARIMA models (Table 12). Because the short term prices were under estimated, long term simulation under such condition would further under estimate the prices.

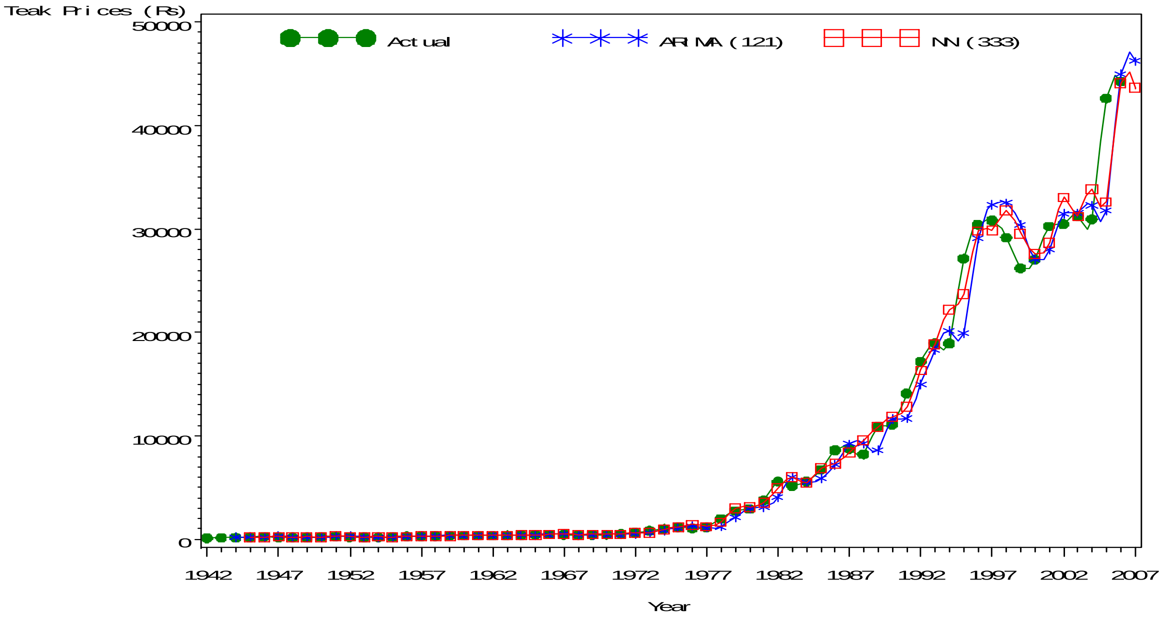
15a) Prices of teak in Export Class



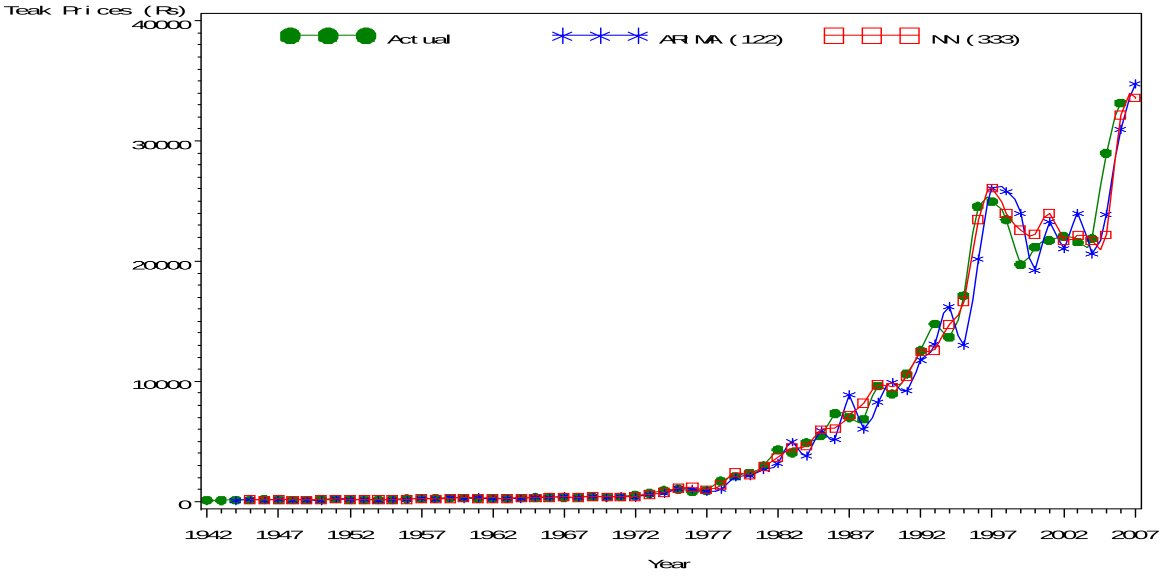
15b) Prices of teak in Girth Class I



15c) Prices of teak in Girth Class II



15d) Prices of teak in Girth Class III


Teak Prices (Rs)
40000
30000
20000
10000
0
Actual
ARIMA (122)
NN (333)
1942
1947
1952
1957
1962
1967
1972
1977
1982
1987
1992
1997
2002
2007
Year

15e) Prices of teak in Girth Class IV

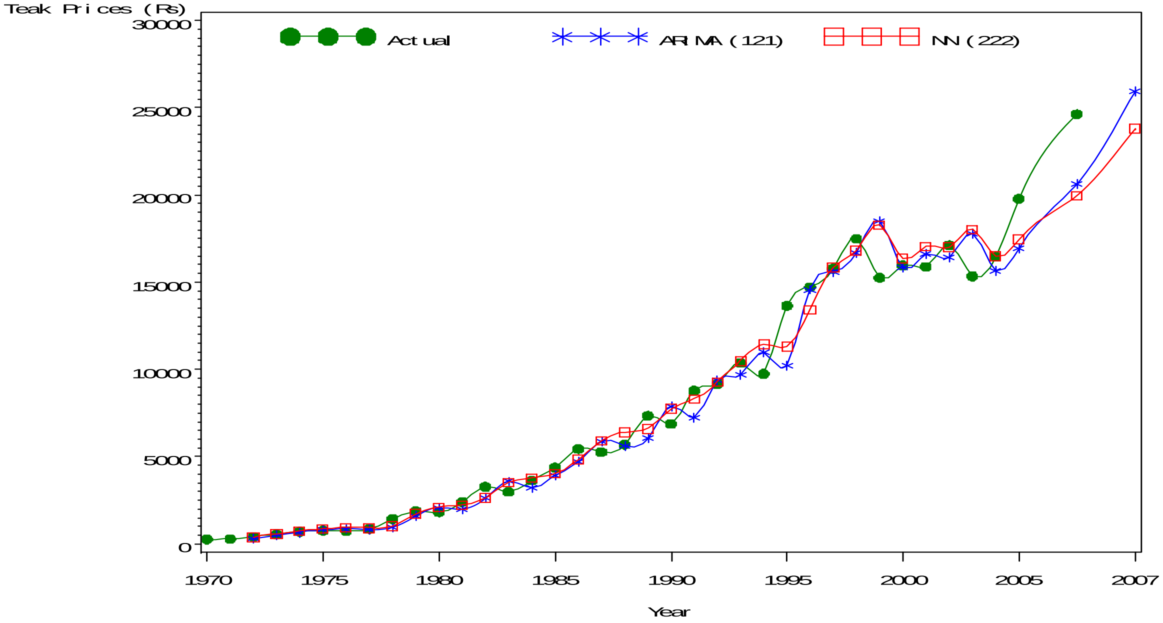


Figure 15 . Forecasted prices of teakwood per m^3 using ARIMA and ANN models for different girth classes a) Export Class b) Girth Class I c) Girth Class II d) Girth Class III e) Girth Class IV

Table 8. Comparison of ARIMA models for forecasting teakwood prices of different girth classes

ARIMA (*p,d,q*)	RMSE	MAE	MAPE	AIC
Girth Class Export				
(1,2,1)	4048.0	2334.5	13.93	585.418
(1,1,2)	4227.1	2344.5	14.64	607.148
(2,1,1)	4200.2	2171.8	13.89	606.695
(2,2,1)	4011.6	2235.0	13.62	586.786
(1,2,2)	3939.0	2136.7	13.22	585.508
Girth Class I				
(1,2,1)	2482.1	1184.8	14.42	988.923
(1,1,2)	2481.7	1166.2	14.41	1006.5
(2,2,1)	2425.1	1166.4	14.49	987.997
Girth Class II				
(1,2,1)	1940.3	843.79	11.06	957.894
(2,2,1)	1931.4	832.93	11.12	959.315
(2,1,2)	1934.8	824.59	11.02	976.673
Girth Class III				
(1,1,1)	1565.6	784.97	13.84	945.57
(1,2,2)	1459.1	825.35	15.49	923.98
(2,2,1)	1493.6	751.97	13.51	926.93
Girth Class IV				
(1,2,1)	1358.2	881.21	12.08	508.976
(1,1,2)	1386.0	925.61	13.53	526.859
(2,1,1)	1374.8	929.56	13.29	526.276
(1,2,2)	1352.8	857.88	11.86	510.696

p- number of autoregressive terms; *d*- order of differencing;
q- number of moving average terms

Table 9. The functional form and co-efficients of the ARIMA models chosen for forecasting prices of teakwood in different girth classes

Girth Class	ARIMA (p,d,q)	Functional form of the chosen prediction equation	Values of th Coefficient		
			ϕ_1	θ_1	θ_2
Export	(1,2,2)	$y_t = (2 + \phi_1) y_{t-1} - (1 + 2\phi_1) y_{t-2} + \phi_1 y_{t-3}$ $+ a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}$	-.90812 (.3032)	.38726 (.3802)	.40601 (.3903)
I	(1,2,1)	$y_t = (2 + \phi_1) y_{t-1} - (1 + 2\phi_1) y_{t-2}$ $+ \phi_1 y_{t-3} + a_t - \theta_1 a_{t-1}$	-.49757 (.1600)	.85056 (.0859)	
II	(1,2,1)	$y_t = (2 + \phi_1) y_{t-1} - (1 + 2\phi_1) y_{t-2}$ $+ \phi_1 y_{t-3} + a_t - \theta_1 a_{t-1}$	.03254 (.1529)	.90156 (.0887)	
III	(1,2,2)	$y_t = (2 + \phi_1) y_{t-1} - (1 + 2\phi_1) y_{t-2}$ $+ \phi_1 y_{t-3} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}$	-.62596 (.1539)	-.05393 (.1446)	.89158 (.1115)
IV	(1,2,1)	$y_t = (2 + \phi_1) y_{t-1} - (1 + 2\phi_1) y_{t-2} + \phi_1 y_{t-3}$ $+ a_t - \theta_1 a_{t-1}$	.02265 (.2363)	.89004 (.1797)	

Standard errors are given in parentheses

Table 10. Comparison of performance of neural network models for forecasting prices of teakwood in different girth classes

Girth Class	ANN (i,j,k)	MAPE	
		Raw data	Log transformed data
Export Class ($n = 37$)	111	118.82	12.40
	121	103.79	16.93
	131	101.81	18.58
	221	102.38	12.48
	222	99.46	11.41
	231	101.81	16.66
	232	20.27	15.67
	331	18.35	12.71
	332	63.95	10.19
	333	42.22	8.77
Girth Class I ($n = 65$)	111	16.97	11.74
	121	25.18	16.88
	131	372.28	20.22
	221	13.73	11.91
	222	490.28	11.59
	231	27.05	16.55
	232	34.32	14.40
	331	15.68	11.31
	332	21.64	11.09
	333	102.83	9.54
Girth Class II ($n = 65$)	111	14.84	10.923
	121	189.08	16.22
	131	209.70	18.57
	221	14.02	10.59
	222	27.75	10.43
	231	32.43	16.31
	232	349.91	14.53
	331	12.75	10.22
	332	14.16	9.62
	333	19.80	8.91

Girth Class	ANN (i, j, k)	MAPE	
		Raw data	Log transformed data
Girth class III ($n = 65$)	111	17.61	12.62
	121	55.33	18.81
	131	68.72	21.44
	221	14.40	12.68
	222	32.74	12.44
	231	65.15	19.09
	232	127.72	17.68
	331	19.33	12.45
	332	208.35	12.33
	333	158.13	10.50
Girth class IV ($n = 37$)	111	15.24	11.20
	121	17.54	13.53
	131	93.26	12.91
	221	79.53	10.72
	222	72.69	10.26
	231	90.99	11.99
	232	16.58	10.79
	331	13.53	10.83
	332	79.67	7.98
	333	158.13	6.12

i- number of inputs; *j* - lag period and; *k*-number of hidden neurons
hyphen (-) indicates that the model was not appropriate.

Table 11. The synaptic weights of the feed forward neural network models chosen for forecasting prices of teakwood in different girth classes

From	To	Notation	Synaptic weights				
			Export Class	Girth Class I	Girth Class II	Girth Class III	Girth Class IV
			ANN (333)	ANN (333)	ANN (333)	ANN (333)	ANN (2,2,2)
y_{t-1}	H1	b_{11}	-3.957	-0.583	-1.065	-0.937	6.397
y_{t-2}	H1	b_{21}	3.306	0.619	1.435	1.347	-4.709
y_{t-3}	H1	b_{31}	-0.241	-2.068	-2.441	-2.431	-
y_{t-1}	H2	b_{12}	-5.746	0.556	4.279	-0.072	0.136
y_{t-2}	H2	b_{22}	4.255	-0.804	-13.011	-64.60	-0.395
y_{t-3}	H2	b_{32}	2.389	5.823	36.475	92.948	-
y_{t-1}	H3	b_{13}	-32.107	356.149	-3.155	-2.678	-
y_{t-2}	H3	b_{23}	11.521	-588.280	-4.038	-2.748	-
y_{t-3}	H3	b_{33}	16.368	-424.530	0.005	1.643	-
Bias	H1	a_1	-1.677	-0.877	-0.852	-0.970	-0.078
Bias	H2	a_2	-0.046	-7.629	-41.104	-39.659	-0.465
Bias	H3	a_3	3.529	508.997	6.876	3.711	-
H1	Output	d_{21}	-3.750	-4.737	-4.655	-4.670	0.724
H2	Output	d_{21}	3.226	1.070	0.442	0.413	-15.882
H3	Output	d_{21}	-0.728	-0.578	-1.095	-1.445	-
Bias	Output	c$_1$	8.938	9.983	10.326	10.226	14.313

Notes for notations used are presented in Figure 14.

Table 12. Forecasted prices of teakwood in different girth classes for the year 2007 using the best ARIMA and ANN models

Model	Predicted price for the year 2007 (Rs/m^3))
Girth Class Exp	
ARIMA(1,2,2)	69,830
ANN(333)	49,096
Girth class I	
ARIMA(1,2,1)	56,834
ANN(333)	46,597
Girth class II	
ARIMA(1,2,1)	46,231
ANN(333)	43,651
Girth class III	
ARIMA(1,2,2)	34,783
ANN(333)	33,618
Girth Class IV	
ARIMA(1,2,1)	25,949
ANN(222)	23,821

5. DISCUSSION

ANN is a totally different approach to data analysis compared to statistical technique. Instead of conceptualising the problem as a mathematical one, ANN uses human brain and its structure to develop a processing strategy. There is substantial motivation for using ANN due to drawbacks of statistical models in which without expertise, it is possible to mis-specify the functional form relating the independent and dependent variables and fail to make necessary data transformations. Outliers can lead to biased estimates of model parameters. Statistical models are often linear, in the case of nonlinear equations they are converted to linear form by mathematical transformation, and solutions are arrived at and thus may not capture exactly the nonlinear behavior. Although about 30 different ANN models have been developed since the first prototype proposed in 1943, MLP is the most commonly used ANN in natural resource management (Peng and Wen, 1999).

In the case of predicting bark thickness using the traditional regression modeling approach, the second degree parabola is found to provide better predictions. Usually, the growth parameters such as height and volume of the trees, increase along with the increase in dbh up to a certain point and thereafter get stabilized. In the case of thickness prediction, after stabilization there is a declining tendency of bark thickness for the increasing dbh. It appears that this has been captured well by the second degree polynomial equation without a constant term. The logarithmic transformation of both input and output variables improved the predictions greatly because of high variation. The high variation may be due to different habitats from where samples were collected. The declining tendency on maturity of tress (increasing dbh) may be partly due to reasons such as dead outer bark and shedding of bark.

This study indicates that although ANN produced better predictions than the regression models, the gain in performing ANN is less for *Lagerstroemia reginae* and ratio of germination percentage of teak seeds. It is observed from our results that in general one hidden neuron in the hidden layer of ANN is sufficient for one input variable. The increase in the number of hidden neurons does not provide better predictions. However, in the case of germination ratio of teak seeds, ANN with two neurons in the hidden layer type is slightly better than ANN with one neuron in the hidden layer. It seems that the inherent presence of non-linear relationships in ANN models allow them to predict better than the regression models especially at the critical turning points of the curves. Comrie (1997) made similar observations in his studies on weather based ozone forecasting using MLP and multiple regression models. Modeling tree-ring growth responses to climatic variables Qi-Bin Zhang *et al*. (2000) demonstrated that the ANN models, if designed and trained properly, could perform better than traditional linear regression approaches currently being used. Bishop (1995) observed that in most instances ANN produced comparable results.

In this study, it was found that the regression model underestimates the bark thickness for the low values of dbh. This might be due to fewer samples available in this range and decimal correction on retransformation of logarithmic scale to original scale. However, this problem appears to be not seen with respect to ANN models and found to be advantageous as it picks up the trend very well throughout the range of data. The performance of ANN was examined by considering examples having one input variable. Efforts should continue to assess the performance of ANN in complex multivariate non-linear situations like studies reported earlier.

The studies conducted so far elsewhere on comparing statistical and ANN models for forecasting problems have drawn mixed conclusions. As far as this study is concerned on the whole, though ANN performed better than ARIMA, it did not sensibly forecast short-term prices as there appeared to be an underestimation. The reasons for this could not be clearly identified. On the other hand, ARIMA could smooth out the trend and appeared to provide better short-term predictions.

Adya and Callopy (1998) examined the application of ANN to business forecasting and prediction. Of the 48 studies evaluated, 22 contributed to the knowledge regarding the applicability of ANN for forecasting and prediction. Nineteen of these produced results that were favorable to ANN, three produced results that were not. They concluded that ANN, when it is effectively implemented, shows potential for forecasting and prediction.

Faraway and Chatfield (1998) found that neural networks often gave poorer out-of-sample forecast for the airline data. Further, they concluded that there is more chance for going badly wrong with ANN modeling. Without careful choices of the architecture, the activation functions and appropriate starting values for the weights, fitting routines may not converge, may converge to a local minimum or may lead to forecasts which are not sensible.

In a comparison of ANN with Box-Jenkins and Holt-Winters exponential smoothing, Zhang (2001), using both 240 simulated linear series and three actual time series, concluded that ANN was able to outperform Box-Jenkins' ARMA(p,q) in all but one of their cases. They found that simple ANN was often adequate in forecasting linear time series. Goh (1998) found that ANN outperformed the univariate Box-Jenkins' approach and the multiple log linear regression on quarterly data. Hwarng (2001) compared ANN with ARMA(p,q) structure on 320 generated time series. He concluded that ANN trained with a normal level of noise tend to perform better than ARMA(p,q) structures.

In a forecasting exercise on 30 time series, ranging on several fields, from economy to ecology (Rech, 2002) it was found that the linear models outperformed the ANN. Moshri and Cameron (2000) found that ANN models were able to forecast as well as all the traditional econometric methods, and to outperform them in some cases. In a study of long term range monsoon rainfall prediction of 2005, the performance of neural network model was far superior to the regression models (Guhathakurta, 2006). Stern (1996) found that the results of the Generalized Additive model were comparable to those of ANN for certain time series data.

Linear detrending is suggested as a preprocessing technique prior to modeling with ANN (Bishop, 1995). However, in this study, improved results could be obtained neither with linear detrending nor with first order differencing. The logarithmic transformation of price data dramatically improved the MAPE in the case of ANN models. Therefore, it is essential that the necessary transformation be made before applying ANN models. A simple ANN architecture with only one hidden layer was used and found to be sufficient. In building-up ANN for the time series data, one has to construct an input data with several lags. The way to select a number for the lag can be arbitrary, but a reasonable idea is to select a lag for which the correlation between the original data and lagged data become large. When the number of input (previous observations) is equal to the lag period the prediction error was less. Similarly, when the number of input is equal to the number of hidden neurons the prediction error was less. By and large ANN with prices of previous 3 years as inputs and 3 neurons in the hidden layer was found better in predicting next year price except for girth class IV. The increase in number of previous year prices and the number of hidden neurons did not improve the MAPE.

The focus primarily is on one-step-ahead forecasting. One justification is that the one-step-ahead forecasts are most useful in practice, since one should automatically recalibrate a model and generate new forecasts as more data become available. Moreover, for many models, one-step-ahead forecast errors are independent of one another. Hence, one-step forecasts are best for discriminating among competing models. Additionally, in most previous forecasting experiments with time series, one-step forecast have been used. Given the pattern of results obtained, it appears that neural networks are an appropriate extrapolation technique for nonlinear time series. ANN is important as a check for other methods and when it is difficult to forecast successfully with other methods. For series that can be identified as nonlinear and discontinuous, it may be possible to gain significant improvement over other methods even by using very simple neural network architectures.

Among the chosen ARIMA and ANN models, ARIMA model was preferred for forecasting because forecasting by ANN model was not sensible from the practical point of view despite ANN with log transformed prices showing lesser MAPE value than the ARIMA model (Figure 16). The forecasts indicated that the higher girth classes *viz*., Export class (185 cm and above), Girth class I (150-184 cm) would fetch high prices than the lower girth classes might be due to higher demand for quality teak wood (Table 13).

The primary difference of ANN from most of the statistical techniques is the absence of any statistical inference tests and construction of confidence bounds for model weights of overall fit. While interpretations can be drawn on regression co-efficient as to know the extent and directionality of relationship between input and output variables, synaptic weight in ANN is not interpretable. One advantage often cited with ANN is that the most ANN can learn to generalize effectively from noisy data and can tolerate missing observations. It is claimed that ANN, unlike statistical models, requires no distributional assumptions. On the other hand, some observed that ANN involves exactly the same sort of distributional assumptions as statistical models, but statisticians study the consequence and importance of those assumptions while many ANN researchers ignore them (Bishop, 1995). From a statistical perspective, ANN is a wide class of flexible modeling algorithm and robust to the problems such as non-gaussian distributions, non-linear relationships, outliers and noise presented in data. However, ANN technology will not replace traditional quantitative techniques completely but it does offer an alternative to traditional quantitative techniques (Peng and Wen, 1999; Lin *et al*., 2002).

Figure 16. Comparison of performance of ARIMA and ANN models using MAPE

Table 13 Forecasted percentage increase in Teak wood prices in Kerala using ARIMA model

Girth Class	Current Price (Rs/cu m) -2006	Forecasted Current Price (Rs/ cum) -2007	Percentage increase
Export	57,270 (1437)	69,830 (1753)	21.9
Girth Class I	48,937 (1228)	56,834 (1426)	16.1
Girth Class II	44,295 (1112)	46,231 (1160)	4.4
Girth Class III	33,174 (833)	34,783 (873)	4.9
Girth Class IV	24,638 (618)	25,949 (651)	5.3

US dollar equivalent is provided in parentheses (1 US $ = 39.845 INR)

6. REFERENCES

1. Adya, M.S. and Collopy, F. (1998). How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*. 17: 481-495.
2. Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University, Press, Oxford.
3. Bose, N.K. and Liang, P. (1998). *Neural Networks Fundamentals with Graphs, Algorithms and Applications*. Tata McGraw-Hill, New Delhi.
4. Box, G.E.P. and Genkins, G.M. (1994). *Time Series Analysis Forecasting and Control*. Pearson Education Inc., Delhi.
5. Burnham, K.P. and Anderson, D.R. (2002). *Model Selection and Multimodal Inference: A Practical Information-Theoretic Approach* (2nd edition). Springer-Verlag, New York, USA. 488 pp.
6. Comrie, A.C. (1997). Comparing neural networks and regression models for Ozone forecasting. *J.Air & Waste Manage. Assoc*. 47: 653-663.
7. Draper, N.R. and Smith, H. (1966). *Applied Regression Analysis*. John Wiley & Sons, Inc., New York.
8. Faraway, J and Chatfield, C. (1998). Time series forecasting with neural networks: a comparative study using airline data. *Applied Statistics*. 47: 231-250.
9. Goh, B.H (1998). Forecasting residential construction demand in Singapore: a comparative study of the accuracy of time series, regression and artificial neural network techniques. *Engineering Construction & Architectural Management*. 5(3): 261-275.
10. Guhathakurta, P. (2006). Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network. *Current Science*. 90(6): 773-779.
11. Haykin, S. (1999). *Neural Networks- A Comprehensive Foundation*. Pearson Education Inc., Delhi.
12. Hwrang, H.B (2001). Insight into neural-network forecasting of time series corresponding to ARMA (p, q) structure. *Omega-International J. Management Science*. 29(3): 273-289.

13. Kajitani, Y., Hipel K.W. and Mcleod A.I. (2005). Forecasting nonlinear time series with Feed-forward neural networks: A case study of Canadian Lynx data. *Journal of Forecasting*. 24: 105-117.

14. Kartopoulos, S.V. (1996). *Understanding Neural Networks and Fuzzy Logic-Basic Concepts and Application*. IEEE Neural Network Council. Prentice Hall of India Private Ltd., New Delhi.

15. Krishnankutty, C.N. (1998). *Timber price trends in Kerala*. KFRI Research Report No. 160. Kerala Forest Research Institute, Peechi.

16. Krishnankutty, C.N. Chundamannil, M and Sivaram, M (2003). Teak wood price projections for Kerala State, India up to 2020. *International Conference on Quality Timber Products from Sustainable Forest management*. 2-5, December 2003. Kerala Forest Research Institute, Peechi.

17. Lin, F, Yu, X.H, Gregor, S and Iron, R (1995).Time series forecasting with Neural Networks. *Complexity International*. Vol. 2.

18. Liu, C., Zhang, L., Davis, C.J., Solomon, D.S., Brann, T.B and Caldwell, L.E. (2003). Comparison of neural networks and statistical methods in classification of ecological habitats using FIA data. *Forest Science*. 49(4): 619-631.

19. Moshiri, S Cameron, N (2000). Neural Network versus Econometric models in forecasting inflation. *Journal of Forecasting*. 19: 201-217.

20. Peng, C. and Wen, X. (1999). Recent applications of artificial neural networks in forest resource management: An overview. In: *Environmental decision support systems and artificial intelligence.* Ulises Corte and Miquel Sanchez-Marre, Cochairs (eds.). Pp. 15-22. Technical Report WS-99-07, AAAI Press, Menlo Park, CA. 1999.

21. Rech, G (2002). *Forecasting with neural network models*. SSE/EFI working paper series in Economics and Finance. Department of Economics Statistics, Stockholm school of Economics, Stockholm, Sweden.

22. SAS Institute Inc. (1999-2003). SAS Enterprise Miner Release 4.2. SAS Institute Inc., Cary, North Carolina. USA.

23. Stern, H.S. (1996). Neural networks in applied statistics. *Technometrics*. 38: 205-214.

24. Qi-Bin Zhang, Richard, J. Hebda, Qi-Jun Zhang and Alfaro R.I. (2000). Modeling tree ring growth responses to climate variables using artificial neural network. *Forest Science*. 46(2): 229-239.

25. Zhang, G.P (2001). An investigation of neural networks for linear time series forecasting. *Computers & Operational Research*. 28(12): 1183-1202.

7. APPENDIX

Autoregressive Model

Autoregressive model is one of the important models for representing certain practically occurring series. In this model, the current value of the process is expressed as a finite, linear aggregate of previous values of the process and a shock *a_t*. Mathematically, an autoregressive model of order *p* can be represented as

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + a_t$$

where μ , $\phi_1,...,\phi_p$ are parameters.

Moving Average Model

A moving average model of order *q* can be expressed mathematically as

$$y_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \theta_2 a_{t-3} - ... - \theta_q a_{t-q}$$

where $\theta_1, \theta_2,..., \theta_p$ are parameters*,* a_t is the error residual and $a_1, a_2,..., a_{t-q}$ are previous values of error. The above equation implies that the dependent variable y_t depends on the values of error term $(a_t, a_{t-1},..., a_{t-q})$ rather than variable itself. In the same way we talked about correlation among successive values of y_t we can talk about the autocorrelation among successive values of error or residuals. According to above equation the future values could be predicted by utilizing the error of each of several past periods.

Mixed Autoregressive Moving Average Model

To achieve greater flexibility in fitting of actual time, it is sometimes advantageous to include both autoregressive and moving average terms in the model such models can be represented as,

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - .....\theta_q a_{t-q}$$

which contains *p+q+2* unknown parameters μ ; $\phi_1,..., \phi_p$; $\theta_1,..., \theta_q$ $\sigma^2{}_q$ that are estimated from data.

Differencing

If the data series contains a trend (non-stationary), it can be removed and made stationary by taking successive differences of the data. Consider the simple series 2,4,6,8…20 consisting of a linear trend and no randomness. Subtracting consecutive values, give the first differences, *i.e.* series, 2,2…2. This series has no trend component. Tendency for the autocorrelation function not to die out quickly is taken as an indication of nonstationarity. Therefore failure of the estimated autocorrelation function to die out rapidly might logically suggest that we should treat the underlying process as nonstationary but possibly as stationary in first or higher differences. In practice, order of differencing (*d*) is either 0,1, or 2.

Autocorrelation Coefficient

Autocorrelation coefficient measures the degree of correlation between neighboring data observation in a time series. Assuming the time series is y_i, i=1,2,3…. The autocorrelation coefficients estimated from sampling observations is as follows.

$$r_k = \frac{\sum_{i+1}^{n-k} (y_{i+k} - \bar{y}_{i-1})(y_i - \bar{y}_k)}{\left[\sum (y_i - \bar{y}_i)^2\right]^{\frac{1}{2}} \left[\sum (y_{i+k} - \bar{y}_{i+k})^2\right]^{\frac{1}{2}}}$$

r_k describes the autocorrelation of y_i and y_{i+k}. The first autocorrelation will indicate how successive values of y relate to each other.

The sampling distribution of autocorrelation coefficient is normal with μ_{r_k} =0, $\sigma_{r_k} = \frac{1}{n^{\frac{1}{2}}}$

Where μ and σ stand for the means and variance of r_k respectively

Partial Autocorrelation Coefficient

Partial Autocorrelation measures the degree of association between y_i and y_{i+1} when the effects of other lags on y are held constant. Their singular purpose in time series analysis is to help identify an appropriate ARIMA model forecasting.