Artificial Neural Networks: A tool for forecasting stock returns

*Dr. Y. V. Reddy**
*Rebello R. W.***

ABSTRACT

This article examines and analyzes the use of neural networks as a forecasting tool. Specifically an artificial neural network's ability , to predict future trend of prices of stocks against actual price, included in the Sensitive Index (Sensex) of Bombay Stock Exchange (BSE), is tested. Scope of this study is restricted to individual investor. In this study, authors have used 11 probes to forecast the stock returns. Results of the study are encouraging and the deviations are less than 5%. During the training session, it is observed that not the all probes used are relevant for all the stocks and therefore, it is necessary to identify the probes, which are influencing the specific stock. Artificial neural network is not a black box, if it is used judiciously, the results could be amazing. The findings suggest that stock markets do not follow a random walk and there exists a possibility of predicting stock returns. Authors opine that it is possible to capture non-linearities contained in the stock returns by using artificial neural network. If neural network is used astutely, it could benefit the individual investors.

Keywords: Feedforward Neural Network, Hidden layer, Node, Training, Probes, Testing, Generalization, Prediction, Architecture, Simulation, Network Builders.

Introduction

It has been the endeavor of human being to make his life easier. In pursuit of this, he has been struggling to find solutions to the puzzles he has faced during his lifetime. One of the unresolved puzzles he has faced is forecasting the stock prices. A good forecast of stock returns, could make a vast difference to an individual investor as well as to the economy. Investors' confidence in the stock market increases with the ability of making worthy forecasts. Because of this, both practitioners and policy makers are attracted to the forecast of stock returns. However, stock returns are known for their spontaneity, given that they are often characterized by high volatility, noise, non-stationarity, non-linearity and chaos [Abhyankar A. et al. (1997)]. Different techniques such as traditional, modern, simple and sophisticated have been used, but no single technique has given result, which would beat the market repeatedly. The efficient market hypothesis (EMH) reinforces the view of unpredictability of stock returns. However, the recent developments show that, in principle stock returns are predictable from the information gleaned from the past returns [Ingber L. (1996); Lo W. A. at el (1999); Malkiel B. G. (1996) and Taylor S. J. (1994)].

Traditional models such as the market model, capital asset pricing model (CAPM), linear regression model and arbitrage pricing theory (APT) have been used to understand the stock price behaviour, but have been found unsuccessful (Chen N. F. et al. 1986). Since, most of these models were based on linear relationship; it was construed that these models could not approximate for non-linearities contained in the stock price data. Given the failure of traditional linear models, several parametric non-linear models such as autoregressive conditional heteroskedasticity (ARCH), general autoregressive conditional heteroskedasticity (GARCH) and self-exciting threshold autoregression models (SETAR) [Chan at el (2004); Clements M. P. at el (1997)] have been

* *Dr. Y. V. Reddy, Reader, Department of Commerce, Goa University*

***Senior Lecturer, Seth L. U. J. College of Arts and Sir M.V. College of Science and Commerce* proposed and applied to the forecasting of stock returns. While these models may be good for a specific series of data, they do not have a general appeal for other applications. Because there are too many possible non-linear patterns, the pre-specification of the model restricts the usefulness of these parametric non-linear models.

This led to the emergence of non-linear and non-parametric models. Artificial neural network is one such model that has been used in forecasting stock returns in recent past. Using past historical stock prices as explanatory variables, White H. (1988) tries to capture the undetected non-linear regularities that may exist in daily common stock returns by using the neural network. The said author concludes that the neural network is capable of capturing some of the dynamic behaviour of stock returns. Qi and Maddala (1999) used financial and economic variables to forecast excess return. They conclude that the neural network model can improve upon the linear regression model and random walk model in terms of predictability. Shachmurove and Witkowska (2000) applied ordinary least squares, general liner regression and compared their results with the results produced by using artificial neural network models in order to examine the superiority of this technique over others.

Neural Network

An artificial neural network is an information-processing model that is enthused by the way nervous systems process information. The key element of this model is the new structure of the information processing system.It is comprised of a large number of interconnected processing elements (neurons) working in harmony to solve particular problems. An artificial neural network is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons (Stergiou Christos and Siganos Dimitrios).

Importance

Neural networks, with their amazing ability to derive meaning from complicated or loose data, can be used to mine patterns and identify trends that are too complex to be noticed either by humans or by computer techniques. A neural network can be thought, as an expert in the category of information analysis. This expert can also be used to provide projections given new situations of interest. Other advantages include (Stergiou Christos and Siganos Dimitrios):

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training.
2. Self-Organisation: An artificial neural network can create its own organisation of the information it receives during learning period.
3. Real Time Operation: Artificial neural network computations may be carried out in parallel. Special hardware devices are being designed and manufactured for this purpose.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network can even retain some capabilities.

Neural networks do not perform wonders. However, if they are used sensibly they can produce some marvelous results (Stergiou Christos and Siganos Dimitrios).

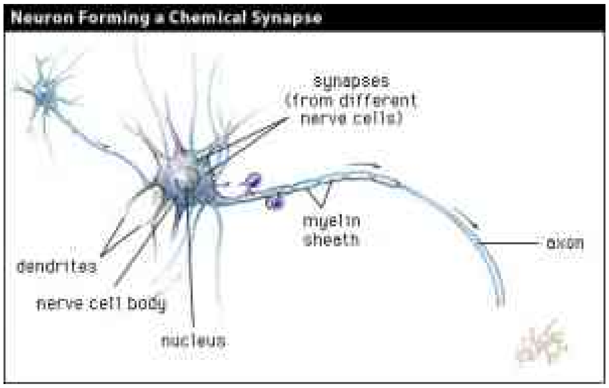
When output is a non-linear function of input, artificial neural network has proved to be a better technique. Artificial Neural Network technique has been put to use in business forecasting, credit and bond rating, business failure forecasting, pattern recognition, image processing etc. A number of studies have been reported using artificial neural network all over the world. However, except few, not much work has been reported in India [Bordoloi (2001); Kamath (2001), Nag at el (2002), Panda at el (2007)].

Artificial Neural Network

Artificial Neural Network works similar to a human brain. Human brain is made up of tiny units called neurons. There are around hundreds of billions of neurons in a human brain. These neurons are connected to each other and they communicate with one another through electrochemical signals. The signals send by one neuron is received by another neuron through junctions. These junctions are called synapses. These synapses are located at the end of the neuron cell. These are called as dendrites. Whatever signals received from other neurons are processed and results are generated by neurons. The output generated by the neurons is in the form of electrochemical signals. These signals are called axon. This has been illustrated through a sample cell (Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k) (Exhibit 1)

EXHIBIT 1

Neuron Forming a Chemical Synapse


Neuron Forming a Chemical Synapse
synapses (from different nerve cells)
myelin sheath
axon
dendrites
nerve cell body
nucleus

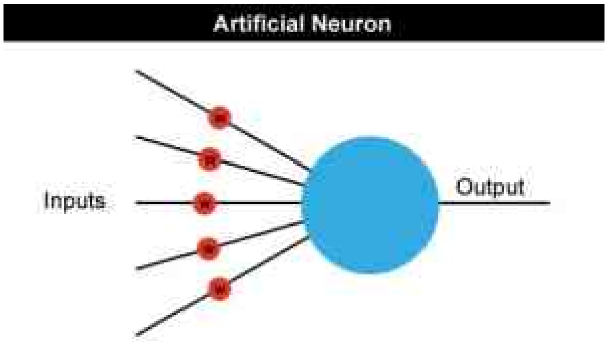
(Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k)

Neural network learns to recognize the things, which brought its notice, as small child is taught to recognize the alphabets. Before training a neural network, it is like a blank black board. It does not carry in pre-notions or prejudices, like a normal grown up adult. So a neural network can be made to recognize all the data brought to its notice and gradually made to develop a pattern. At times, it will not be in a position to recognize the data. Nevertheless, if neural network is shown data repeatedly it will start recognizing the data slowly, but correctly. If the network fails to recognize the data correctly, the procedure of showing the data should be repeated over and again, until it starts recognizing the same correctly. An artificial neural network starts generalizing the data and recognizes only that data, which it has seen before. If new data is brought to its notice, it will learn to compare it with the data it has already studied and provided information, patterns, trends and relationship with one another.

Since artificial neural network is made up of artificial neurons, they are modelled electronically. The number of neurons required to form a network depends upon the task that is required to be carried out. They could be few or as many as several millions. When they connect one another, they form a network. The most common network is called a feedforward network. An illustrative description of a neural network is given hereunder (Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k): (Exhibit 2):

EXHIBIT 2

Artificial Neuron



(Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k)

Each variable directed into a neuron is assigned with its own weight. This is illustrated by a red circle on the input. The weight assigned to an input is simply a floating-point number. These weights are adjusted automatically when network is trained repeatedly. The weights could be positive as well as negative. Therefore, these weights provide excitory or inhibitory influences to each input. As each variable enters the nucleus, blue circle, it is coupled by its weight. The nucleus then sums all these variable values and gives the *activation* along with it weights. If the sum effect is greater than an input value, the neuron generates a signal as output. This is typically called a *step* function (Source: http//:www.ai-junkie.com/ann/ evolved/nnt1.html - 9k).

A neuron can receive any number of input variables. The inputs may be represented therefore as x_1, x_2, x_3 … x_n and the corresponding weights for the inputs as w_1, w_2, w_3… $w_{n.}$ Now, the summation of the weights multiplied by the inputs can be written as x_1w_1

$+ x_2w_2 + x_3w_3 \ldots. \; + x_nw_{n,}$ which is the activation value (Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k). So

$$a = x_1w_1+x_2w_2+x_3w_3+ \ldots +x_nw_n \quad (1)$$

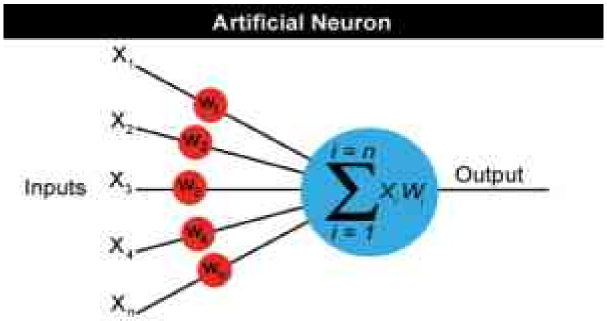
Mathematically this can also be represented,

$$a = \overset{i=n}{\underset{i=0}{3}} x_i\, w_i \qquad (2)$$

Artificial neuron can also exhibited by using the following illustration (Exhibit 3)

EXHIBIT 3

Artificial Neuron

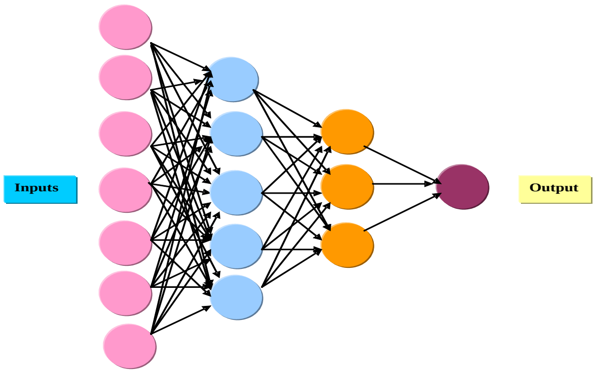


(Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k)

Feedforward network is a design by which several neurons are organizing themselves into another neuron thereby creating a new layer. Similary new subsequent layers are created until final neuron is created, which would deliver desired result. Following is the illustration of simple feedforward net work (Exhibit 4):

EXHIBIT 4

Feedforward Network



(Designed by the authors)

The input sent to neuron is passed on to next layer of neuron as an output. Likewise, there could be several layers in a feed forward network, out of which several could be hidden layers. Generally, one hidden layer within the feedforward network is sufficient to tackle most of the problems. Again, the number of neuron in each network layer depends on the problem in hand. Number of the neurons chosen for the diagram is completely at the discretion of the authors.

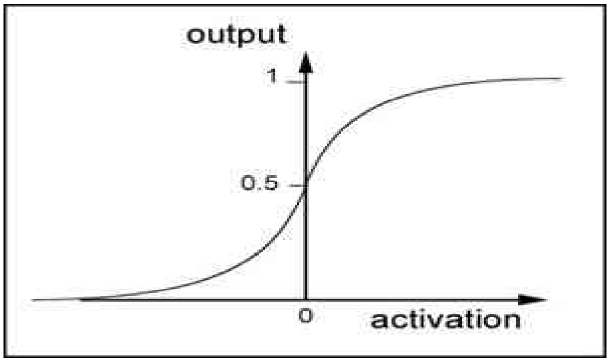
Once the neural network is created, it needs to be trained. While training the networks it initializes itself by random weights. This type of training is called supervised learning. This training should be carried out on the set of data that is earmarked as training set. While training, the weights required to be adjusted. Generally, this adjustment is carried out by back propagation. The weight of the inputs should be changed to soften the output of each neuron to a symmetrical curve. This function is called as sigmoid function (Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k).

$$Output = \frac{1}{1 + e^{-a/p}} \qquad (3)$$

This equation is very simple. The *e* (exponential) is a mathematical constant which approximates to 2.7183, the '*a*' is the activation into the neuron and '*p*' is a number, which controls the shape of the curve. '*p*' is usually set to 1.0. The output of this function is as given below (Exhibit 5):

EXHIBIT 5

Symmetrical Curve



(Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k)

The lower the value of *p*, the more the curve begins to look like a step function. This curve is always centered on 0.5. Negative activation values produce a result less than 0.5, positive activation values produce a result greater than 0.5. (Source: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k)

After explaining inputs and outputs, now there is need to explain hidden layers. There is no rule as to how many layers should be used and how many neurons should be embodied in each layer. However, many researchers have suggested that one layer should have six to eight neurons, but there is no such fixed rule. Generally, artificial neural network uses one minimum hidden layer, though it is possible to use several such hidden layers in the process of training the neural network.

Neural network does not depend on equations and rules, but rather it is a new kind of tool developed for computing. It functions on the basis of correlations and patterns of data supplied to it. During the training, a pattern is developed and network uses such patterns for the purpose of analyzing data. Since different networks are needed for solving different problems, the format remains the same.

Artificial neural network is used to predict the daily stock price of companies referred to above. The mean absolute error is taken as indicators to measure the efficacy of the networks. The results generated by artificial neural network being superior as compared to others, therefore, it is gaining prominence as computation technology in the area of advanced research.

Models

Many tools are used to model the behavior of the stock movements, and many conduct researches with this goal in mind. However, many financial corporations keep them confidential as these are used to guide their financial investments. Another point should be considered when dealing with researchers is their correctness. Since many researchers did not fully investigate the potential of their solutions, it is felt that results generated using neural networks in financial forecasting could not be used properly [Kutsurelis J. (1998), Lo W. A. at el (1999) and Yao J. at el (2001)]. Among all the available techniques, here are the most commonly used:

? *Feed-forward Neural Network* is a design by which several neurons are organizing themselves into another neuron thereby creating a new layer. [Zhang (2003)]

? *Genetic Algorithms*, which help in finding optimal parameters for technical indicators by making them, evolved by combinations and mutations, starting with a population of a given model having established different parameters [Armano at el (2005), Chen S. et al (2005) and Chen S. H. et al, (2007)].

? *Fuzzy Logic Controllers* are for instance used in combination of artificial neural networks and genetic algorithm [Gradojevic (2006)].

? *Fuzzy Neural Network* is also studied and sometimes it demonstrates good performances [Chang et al (2006)].

? *Genetic Programming*, which make different investment strategies evolve and keeps only the most adapted ones [Yu, et al, (2004)].

? *Hidden Markov Models* (HMM) have recently been used for financial forecasting. The objective is to determine the parameters of a Markov process that would fit the stock movements [Hassan, et al (2006)].

Neural Network for Stock Movement Prediction

O'Connor et al (2006) and Kaastra et al (1996) both suggest a step-by-step approach for modeling neural network for financial forecasting, which is used as a guideline to present the most commonly used techniques for building those models. Although they are presented in a chronological way considering the development of a neural network, it is agreed that they should be revisited when significant changes are made to increase the performance of the neural network.

Selection of Input Variables

The selection of input variables is fundamental to forecast accurately the stock movements. It primarily depends on a clear understanding of the economical background of the stock price to forecast. The first value that comes to mind is the evolution of the stock price to forecast over time. Although it is an essential variable, it is often not used directly as a raw variable. The economical context of the studied stock quotes has to be analyzed carefully. These types of input variables are sometimes called external indicators [O'Connor et al (2006)], and are commonly used if available. Once the raw data has been chosen, a set of indicators based on those values might be developed. Of these, at least five indicators are commonly used as input for neural networks [Coupelon (2007)]:

? Relative Strength Index (RSI): A technical momentum indicator that compares the magnitude of recent gains to losses in an attempt to determine over bought and sold conditions of an asset.

? Money Flow Index (MFI): This measures the strength of money in and out of a security.

? Moving Average (MA): This indicates the moving average of a field over a given period.

? Stochastic Oscillator (SO): This compares a security's closing price to its price range over a given time period.

? Moving Average Convergence/Divergence (MACD): This function calculates the difference between a short and a long-term moving average for a field.

Model Construction

There are three types of neural networks, which are commonly used by economists and researchers in stock market prediction [Weckman et al (2003)]:

? *Multi Layer Perceptrons* (MLP): Those are layered feedforward networks.

? *Generalized Feed-forward Networks* (GFN): This is a generalization of MLP, which can jump over one or more layer.

? *Radial Basis Function* (RBF): Hybrid networks containing a single hidden layer of processing elements that uses Gaussian transfer functions rather than the standard sigmoid functions. The type of neural network to use should be decided by determining the most adapted architecture, which is often done by developing and comparing every possibility, and by keeping only the best ones.

The following will discuss the most important points to be kept in mind while modeling a neural network for stock movements forecasting.

? *Number of input neurons*: This number is easy to determine when the previously defined steps have been done, because each input variable will be treated by a single input neuron. This is why it is clearly an important step directly increasing the processing speeds of the neural network.

? *Number of hidden layers*: The hidden layers are composed by the set of all neurons between the input and the output neurons. The number of hidden layer that should be used cannot be clearly defined as it really depends on the amount of input neurons and the properties of the data. Formulas were developed which tried to take into account those parameters, but due to the nature of the stock movements, it cannot be predicted easily [Tsang, et al (2007)]. However, as stated by Chen (2007), it is commonly admitted that one or two hidden layers are enough, and that increasing the amount of those layers increases the danger of over fitting and the computation time.

? *Number of hidden neurons*: The most common technique used today to determine the appropriate number of hidden neurons to include in the model is experimentation. It is a part of the training phase of the neural network development and it might require many computations. The only rule to keep in mind while selecting the most appropriate number of hidden neurons is to always select the network that performs best on the testing set with the least number of hidden neurons [Kaastra et al (1996)].

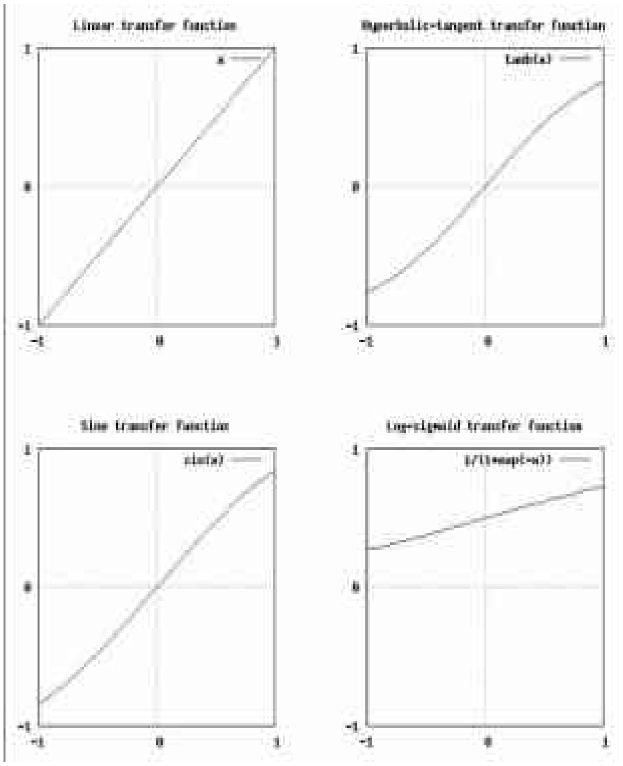
? *Number of output neurons*: Most neural network applications use only one output neuron for both one-step-ahead and multi-step-ahead forecasting [Zhang (2003)]. However, some studies tried different approaches, as in [Zhang, et al (1998)] where multiple output neurons are shown to be of interest.

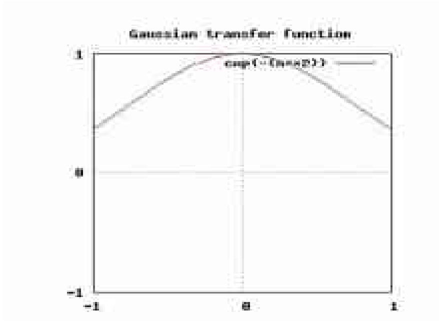
? *Transfer functions*: The transfer function is the formula used to determine the output of a processing neuron. At least five different transfer functions are in use, see exhibit 6 [Coupelon (2007)]. A process of trial and error once again enables selecting the best transfer function. The main selection criteria here is their ability to make the learning phase faster and to increase the neural network accuracy. Neural network that implements heterogeneous transfer function has been studied in [Duch et al (2001)], but despite the fact that they seem to improve the performance of neural networks, no commercial simulators are currently fully implementing them.

EXHIBIT 6

Transfer Function

Coupelon Olivier (2007)





Training Algorithm

The process of training the neural network involves iteratively presenting it the input data so that it is calibrated and can be used later as a forecasting tool. The objective of the training is to minimize a defined error function, which implies that the neural network fit the input data, given the expected results as outputs. There are several kinds of errors functions, used in financial forecasting, namely [Coupelon (2007)]:

- Mean Absolute Deviation (MAD): $\frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$
- Mean Squared Error (MSE): $\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$
- Mean Absolute Percentage Error (MAPE): $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \bar{x}}{x_i}\right|$

(4)

Where, n is the number of error terms. Using those error functions to adjust the weights of the hidden neuron has for long been an insurmountable problem, but [Rumelhart, et al (1986)] introduced the so called Back Propagation Networks (BPN), which consists in neural networks able to issue a backward propagation of errors, even to the hidden neurons. Whenever an error is detected, the weights of the network are slightly modified in order to lower it. This process is generally done using a gradient descent algorithm, which adjusts the weights to move down the steepest slope of the error surface. In the process of back propagation, two degrees of liberty are available to the modeler:

? The learning rate, which determines, how fast the neural network, learns the patterns in the process of training. This value must be carefully chosen as a too small one will make the learning process slow, but a large one might lead to divergence. One way to modify dynamically the learning rate is to increase it as long as the gradient keeps pointing in the same direction, but lowering it when it changes [Jacobs (1987)].

? Momentum terms, which influence the way past weight changes, affect current ones. It helps in preventing the gradient method to be stuck into a local minimum. Once again, this value must be chosen carefully, and should be determined by experimentation. It has to be noted that although the use of a momentum can be omitted, it greatly improves the neural network performances. The back propagation process can suffer of three problems:

a. Reaching the global minimum is not guaranteed by this method, as there can be several local minima, from which the algorithm might possibly not escape.

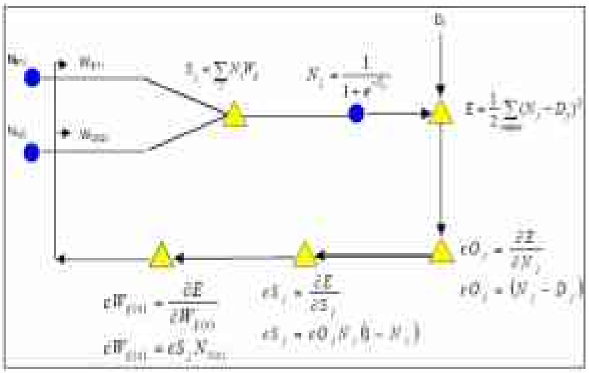
b. This method might lead to over fitting, as it insists on each input data. This point is actually criticized, as it is argued that over fitting is more a problem of having too many neurons in the network [Kaastra et al. (1996)].

c. There is no known configuration for this algorithm, which enables it to find the best solution. Therefore, a process of trial and error must be developed. This is generally done by redoing the process a sufficient number of times, ensuring that the reached minimum is really the global one. The number of iteration needed depends on the complexity of the network, but Kaastra et al. (1996) reports that convergence is generally obtained from 85 to 5000 iterations.

Knowing those problems helps in defeating them and several solutions were advanced to keep back propagation as the algorithm to use [Adya et al. (1998)]. Nonetheless, other training methods are available [Abraham A. (2004), Zhang G. P. (2003)], for instance conjugate gradient descent, quasi-Newton algorithm and Levenberg-Marquardt algorithm, but their use is still marginal, despite the fact that they can achieve better performances.

Another training method that is getting more and more used is based on a genetic approach. The principle is that the weights of the neural network are not adjusted by a back propagation algorithm but with a genetic algorithm (Exhibit 7). At first, a population consisting of randomly generated sets of weights is created. An evolution algorithm is then applied to the population to generate and keep only the most appropriate set of weights. Such a solution generally prevents from falling into local minima and shows good performances. Those types of neural networks are generally called Genetic Algorithm Neural Networks (GANN) [Kim (2006)].

EXHIBIT 7


N_{01}
$W_{(1)1}$
N_{02}
$W_{(2)1}$
$S_j = \sum N_i W_{ij}$
$N_j = \frac{1}{1+e^{-S_j}}$
D_j
$E = \frac{1}{2}\sum (N_j - D_j)^2$
$\varepsilon O_j = \frac{\partial E}{\partial N_j}$
$\varepsilon O_j = (N_j - D_j)$
$\varepsilon S_j = \frac{\partial E}{\partial S_j}$
$\varepsilon S_j = \varepsilon O_j N_j (1 - N_j)$
$\varepsilon W_{ij} = \frac{\partial E}{\partial W_{ij}}$
$\varepsilon W_{ij} = \varepsilon S_j N_{i}$

Kutsurelis Jason E. (Lieutenant) (1998)

Network Training

Training a network involves presenting inputs in such a way that the system reduces its error and brightens its performance. The training algorithm may vary depending on the network architecture. Nevertheless, the most common training algorithm applied when scheming financial neural networks is the back propagation algorithm. This section illustrates some of the training techniques and their challenges associated with it (Ramon Lawrence, 1997).

The most familiar network architecture for financial neural networks is a multilayer feed forward network, trained using back propagation. Back propagation is the process of back propagating errors from the output layer towards the input layer during training. back propagation is necessary because hidden units have no training target value that can be used. Therefore, they must be trained based on errors from previous layers. The output layer is the only layer, which has a target value and which has to be compared (Ramon Lawrence, 1997).

As the errors are back propagated through the nodes, the connected weights are changed. Training occurs pending the errors in the weights is sufficiently small to be accepted. It is interesting to note that the type of activation function used in the neural network nodes can be a factor on what data is being learned. According to Klimasauskas (1993), the sigmoid function works best when learning about average behaviour, while the hyperbolic tangent (tanh) function works best when learning deviation from the average. The major predicament in training a neural network is deciding when to stop training. Since the ability to generalize is fundamental for these networks to predict future stock prices, overtraining is a serious problem. Overtraining occurs when the system memorizes patterns and thus looses the ability to generalize. It is an important factor in these prediction systems as their primary use is to predict on input data that it has never seen. Overtraining can occur by having too many hidden nodes or training for too many times (epochs). Poor results in papers are often blamed on over training. However, overtraining can be prevented by performing test and train procedures or cross-validation (Ramon Lawrence, 1997).

The test and train procedure involves training the network on most of the patterns and then testing the network on the remaining patterns. The network's performance on the test set is a good indication of its ability to generalize and handle data it has not trained. If the performance on the test set is poor, the network configuration or learning parameters can be changed. The network is then retrained until its performance is satisfactory. Cross-validation is similar to test and train except the input data is divided into k sets. The system is trained on k-1 sets and then tested on the remaining set k times. Application of these procedures should minimize over training. It also provides an estimate on network error and determines the optimal network configuration. With these procedures, it is a feeble statement to say that the network's performance was poor because of overtraining, as the experimenter can control overtraining (Ramon Lawrence, 1997).

Each pattern corresponded to the input values on a particular day. It is desirable to have many data available, as some patterns may not be detectable in small data sets. However, it is often hard to obtain many data with complete and correct values. Training on large volumes of historical data is computationally, time intensive and may result in the network learning detrimental information in the data set. Adequate data should be presented so that the neural network can capture most of the trends, but very old data may lead the network to learn patterns or factors that are no longer significant or valuable (Ramon Lawrence, 1997).

Finally, there are varieties of network architectures used for financial neural networks, which are not trained using back propagation. There are different algorithms for some recurrent architecture, modular networks and genetic algorithms, which cannot be adequately covered here. Regardless of the training algorithm used, all prediction systems are very sensitive to overtraining, so techniques like cross-validation should be used to determine the system error (Ramon Lawrence, 1997).

Data Description

In this study, the authors have used daily stock prices (opening price, high price, low price, closing price, weighted average price) to forecast the prices of stocks included in the Sensex Index of Bombay Stock Exchange, India. Other variables used for the study are risk free interest rate (Treasury bill yield rate), quantum of shares traded, forex rate, crude oil price, gold price, consumer price index. Data relating to selected variables were culled for a period from August 2007 to March 2008. Data was gleaned from Energy Information Administration, Canada, World Gold Council, USA and from Reserve Bank of India web sites.

Model Specification

Authors have used Multilayer Perceptrons especially Generalized Feedforward Networks for the purpose of analysis data. Brief explanation about the model used in this study is given below:

Multilayer Perceptrons are an extension of Rosenblatt perception, a device that was invented in the 1950's for optical character recognition. The perceptron only had an input and an output layer (each multiple processing elements). It was shown that the perceptron would only solve pattern recognition problems where the classes could be separated by hyper-planes (an extension of a plane for more than two dimensions). Many problems in practice do not fit this description. Multilayer perceptrons extend the perception with hidden layers.

There are two important characteristics of the multilayer perception. First, its processing elements (PEs) are nonlinear. The nonlinearity function must be smooth (the logistic function and the hyperbolic tangent are the most widely utilized). The second, they massively interconnected such that any element of given layer feeds all the elements of the next layer.

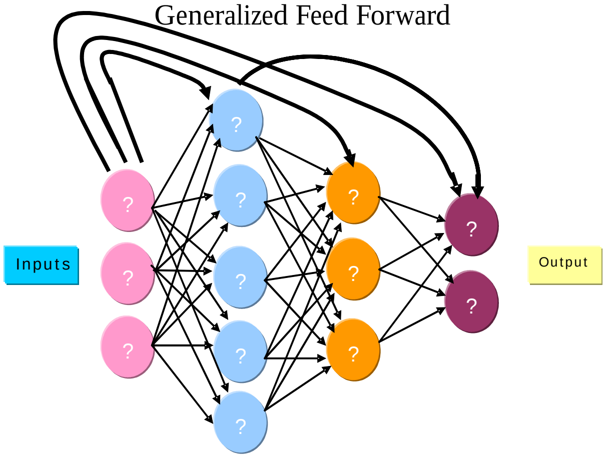
The perceptron and the multilayer perceptron are trained with error correction learning, which means that the desired response for the system must be known. This is normally the case with pattern recognition.

Generalized feed forward nets are a special type of multilayer perceptrons. In these cases, process can jump over one or more layers. In theory, a multilayer perceptron can solve any problem that a generalized feed forward network can solve. In practice, however, the generalized feed forward networks are more efficient. A standard multilayer perceptrons requires several times more epochs for training than the generalized feed forward for the similar size network. The plus point of the generalized feed forward network is its ability to carry out its activities by bypassing several layers. As a result, it requires training lesser layers and the process becomes more efficient (http://aydingurel.brinkster.net/neural/faq.html#5).

As with multilayer perceptrons, the learning rates should be lower towards the output. This needs to be taken into account when setting the learning rates for the synapses within the hidden layers. For example, the learning rate of the synapse, which connects the input directly to the output, should be roughly the same as the synapse, which connects the last hidden layer to the output. The reason is that they are both extracting linear features from the data. Since the learning for the linear portion of the data happens much faster than for non-linear portion, this can skew the performance. A pictorial expression is depicted for ready reference (Exhibit 8).

EXHIBIT 8

Generalized Feed Forward



(Designed by the authors)

Finding And Presentation

The authors have used software of Neural Solutions, in analyzing the data. In the process of the analysis, the raw data is converted into lognormal data, so that non-linearity characteristic of stock price is captured in the analysis. About 50% of above stated data was used for training purpose. The training was carried out by using extended Multi Layer Perceptrons (MLP) model namely Generalized feed forward network. The authors have opted for 1,000 epochs for each training session in these layered feedforward networks. Repeat training was carried out until the least MSE is achieved. Thereafter another 50% of the data was used for testing the architecture created by training process. Then this architecture was used to produce the results.

In the first instance the authors have used all the variables for training, testing and then to derive results. To find out the closing price (used as output variable) all other variables are used as inputs. While to find out high price (used as output variable), all other variables were used as input variables. Similarly, to find out low price (used as output variable) all other variables are taken as input. The summary of derived results is reproduced below:

EXHIBIT 9

Summary of Results Using All Probes

Particulars	Cl. Price	H. Price	L. Price
Average Deviation	3.57%	1.86%	2.92%
Max Deviation of any Company	13.08%	6.55%	12.27%
Min Deviation of any Company	0.17%	0.18%	0.09%
Particulars	**No. of Companies**		
Deviation less than 1%	4	13	6
Deviation between 1% & 2%	6	7	7
Deviation between 2% & 3%	5	5	5
Deviation between 3% & 5%	8	4	7
Deviation between 5% & 7.5%	5	1	3
Deviation between 7.5% & 10%	-	-	1
Deviation more than 10%	2	-	1
TOTAL NO. OF COMPANIES	**30**	**30**	**30**

(for details check Exhibit 12)

From the above table it is observed that in case of closing price there are 23 out of 30 companies whose variation is 5% or less. In the case of high price, there are 29 out of 30 companies, whose variation is 5% and less and for low price, there are 25 companies', whose variation is 5% or less. Despite using all variables, the deviation is relatively small.

In the second instance, the authors have eliminated variables one by one except closing, high, low and opening price, and used all permutations and combinations for the purpose of training and testing and finally used those variables, which have resulted into least MSE, for the purpose of producing results. Even in this case first closing price was ascertained using other identified variables as input. Keeping the same variables as input except the desired output variables again high and low price is ascertained as explained earlier. The summary of derived results is produced below:

EXHIBIT 10

Summary of Results Using Selective Probes]

Particulars	Cl. Price	H. Price	L. Price
Average Deviation	1.97%	1.21%	1.14%
Max Deviation of any Company	6.16%	3.80%	2.68%
Min Deviation of any Company	0.17%	0.01%	0.01%
Particulars	**No. of Companies**		
Deviation less than 1%	8	17	13
Deviation between 1% & 2%	10	6	13
Deviation between 2% & 3%	6	5	4
Deviation between 3% & 5%	4	2	-
Deviation between 5% & 7.5%	2	-	-
Deviation between 7.5% & 10%	-	-	-
Deviation more than 10%	-	-	-
TOTAL NO. OF COMPANIES	**30**	**30**	**30**

(for details check Exhibit 11)

After eliminating some variables one by one, author have retained only those variables, which resulted into least MSE. With these variables, output was derived and these outputs are shown in the above table. From the above table it is observed that there are 28 out of 30 companies, whose variation is 5% and less in the case of closing price. In the case of high price, none of the companies whose variation is more than 5% and for low prices any of the companies' whose variation is more than 3%.

Conclusion

In this article authors employ the artificial neural network to forecast daily prices of stocks included in the Sensitive Index of Bombay Stock Exchange. Our results suggest that the neural network does a better job in forecasting of daily stock returns if variables are identified properly and they are trained adequately. These results appear to be broadly in line with much of the existing work in this area. The general conclusions are that, first that the Indian stock market does not follow a random walk and stock returns appear to be predictable. Further, it is observed that the neural network performs better than the linear regressive model. It recognizes inherent non-linearities in stock returns and suggests that the neural network is able to capture these. However, since the errors are quite close for all the methods, our study is, perhaps, not sufficient to claim superiority of the neural network. It only suggests that neural networks appear to provide a good tool for forecasting, but, of course, more work needs to be done in this area.

EXHIBIT 11

EXHIBIT 11

SUMMARY OF SENSEX STOCKS (USING SELECTIVE PROBES)

Sr. No.	NAME	Regression Coefficient			MSE			Forecasted Price			Actual Price			Deviation (in %)		
		Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price
1	ACC.xls	0.81246	0.78634	0.84782	0.00049	0.00042	0.00047	833.06	836.28	811.73	826.10	844.70	814.00	0.84%	1.00%	0.28%
2	Ambuja Cement.xls	0.45771	0.69728	0.41754	0.00030	0.00017	0.00062	122.11	123.12	120.71	121.05	123.85	119.50	0.87%	0.59%	1.01%
3	Bajaj Auto.xls	0.71473	0.73241	0.72986	0.00080	0.00039	0.00099	2,109.16	2,127.95	2,028.94	2,079.65	2,175.00	1,976.05	1.42%	2.16%	2.68%
4	Bharti Airtel.xls	0.78804	0.84407	0.75780	0.00036	0.00023	0.00056	827.62	843.15	800.07	826.10	840.00	810.20	0.18%	0.38%	1.25%
5	BHEL.xls	0.91600	0.88619	0.87364	0.00023	0.00026	0.00043	2,050.88	2,109.12	2,019.49	2,056.55	2,092.00	1,983.30	0.28%	0.82%	1.82%
6	CIPLA.xls	0.83388	0.81195	0.82581	0.00020	0.00016	0.00024	220.61	220.54	213.92	219.75	224.00	214.00	0.39%	1.55%	0.04%
7	Grasim Industries.xls	0.66010	0.53228	0.72845	0.00040	0.00053	0.00057	2,604.34	2,650.22	2,582.80	2,574.70	2,706.90	2,555.00	1.15%	2.09%	1.09%
8	HDFC.xls	0.84177	0.82520	0.77192	0.00053	0.00037	0.00063	2,530.62	2,522.75	2,381.81	2,383.75	2,611.00	2,365.00	6.16%	3.38%	0.71%
9	HDFC Bank.xls	0.80642	0.81544	0.86661	0.00029	0.00024	0.00035	1,360.94	1,378.73	1,307.72	1,319.95	1,390.00	1,293.05	3.11%	0.81%	1.13%
10	Hindalco.xls	0.83267	0.82597	0.79974	0.00065	0.00046	0.00072	169.58	172.94	166.15	164.75	177.90	163.35	2.93%	2.79%	1.71%
11	HUL.xls	0.84365	0.81798	0.78516	0.00024	0.00016	0.00031	235.38	237.54	223.66	228.70	244.00	225.25	2.92%	2.65%	0.71%
12	ICICI Bank.xls	0.89153	0.89412	0.84574	0.00040	0.00037	0.00054	809.79	812.43	757.41	770.10	825.00	767.00	5.15%	1.52%	1.25%
13	Infosys Technologies.xls	0.84553	0.81089	0.84286	0.00027	0.00029	0.00026	1,446.07	1,538.19	1,407.02	1,430.15	1,510.00	1,406.35	1.11%	1.87%	0.05%
14	ITC.xls	0.82062	0.86133	0.82586	0.00029	0.00017	0.00028	206.69	209.98	203.77	206.35	210.90	203.75	0.17%	0.44%	0.01%
15	L&T.xls	0.83365	0.85829	0.82641	0.00034	0.00018	0.00070	3,070.84	3,156.94	3,014.72	3,024.80	3,148.00	2,988.00	1.52%	0.28%	0.89%
16	M&M.xls	0.82672	0.81597	0.83543	0.00026	0.00021	0.00053	706.23	701.42	661.57	695.65	717.75	675.00	1.52%	2.27%	1.99%
17	Maruti Suzuki.xls	0.79832	0.83162	0.78167	0.00025	0.00019	0.00036	843.48	833.78	802.53	829.55	840.00	817.00	1.68%	0.74%	1.77%
18	NTPC.xls	0.95615	0.92716	0.87612	0.00018	0.00019	0.00046	200.83	204.16	194.51	197.00	204.90	195.10	1.94%	0.36%	0.30%
19	ONGC.xls	0.74782	0.73800	0.81955	0.00056	0.00037	0.00051	1,018.12	1,024.57	971.19	981.35	1,065.00	976.00	3.75%	3.80%	0.49%
20	Ranbaxy.xls	0.82281	0.63612	0.76149	0.00022	0.00043	0.00034	437.22	445.12	434.49	438.75	446.55	431.00	0.35%	0.32%	0.81%
21	Reddy Lab.xls	0.86305	0.69876	0.82283	0.00019	0.00030	0.00025	599.61	593.85	568.31	590.95	600.00	582.00	1.47%	1.03%	2.35%
22	Reliance Comm.xls	0.74304	0.94202	0.82347	0.00073	0.00014	0.00077	527.27	532.97	508.15	508.30	533.90	502.50	3.73%	0.17%	1.12%
23	Reliance Energy.xls	0.81403	0.81331	0.80069	0.00124	0.00080	0.00231	1,278.53	1,325.18	1,241.09	1,251.15	1,325.00	1,241.20	2.19%	0.01%	0.01%
24	RIL.xls	0.75963	0.90743	0.77842	0.00045	0.00015	0.00055	2,314.69	2,342.12	2,238.55	2,264.50	2,340.00	2,251.00	2.22%	0.09%	0.55%
25	Satyam Comp.xls	0.83217	0.75610	0.73016	0.00037	0.00051	0.00069	404.22	408.87	390.62	394.55	408.50	388.05	2.45%	0.09%	0.66%
26	SBI.xls	0.90661	0.91232	0.85200	0.00016	0.00012	0.00032	1,648.17	1,641.84	1,558.46	1,598.85	1,669.00	1,590.00	3.08%	1.63%	1.98%
27	Tata Motors.xls	0.82323	0.83621	0.71308	0.00024	0.00017	0.00085	636.14	641.52	611.28	623.45	645.00	620.00	2.04%	0.54%	1.41%
28	Tata Steel.xls	0.88429	0.88277	0.73069	0.00031	0.00021	0.00123	705.75	722.01	663.19	693.15	715.70	678.65	1.82%	0.88%	2.28%
29	TCS.xls	0.81828	0.80760	0.66598	0.00034	0.00021	0.00069	825.55	860.63	820.73	810.90	869.00	801.00	1.81%	0.96%	2.46%
30	Wipro.xls	0.85353	0.76359	0.64241	0.00028	0.00046	0.00141	428.74	457.43	416.83	425.30	452.00	411.00	0.81%	1.20%	1.42%

EXHIBIT 12

EXHIBIT 12

SUMMARY OF SENSEX STOCKS (USING ALL PROBES)

Sr. No.	NAME	Regression Coefficient			MSE			Forecasted Price			Actual Price			Deviation (in %)		
		Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price	Closing Price	Highest Price	Lowest Price
1	ACC.xls	0.64263	0.72609	0.65039	0.00055	0.00057	0.00178	854.89	820.21	782.38	826.10	844.70	814.00	3.48%	2.90%	3.89%
2	Ambuja Cement.xls	0.42041	0.49782	0.36099	0.00093	0.00040	0.00069	126.23	120.51	121.21	121.05	123.85	119.50	4.28%	2.70%	1.43%
3	Bajaj Auto.xls	0.71473	0.74271	0.66496	0.00080	0.00036	0.00117	2,088.15	2,133.29	2,052.92	2,079.65	2,175.00	1,976.05	0.41%	1.92%	3.89%
4	Bharti Airtel.xls	0.78355	0.83249	0.67917	0.00040	0.00027	0.00120	816.87	832.80	824.27	826.10	840.00	810.20	1.12%	0.86%	1.74%
5	BHEL.xls	0.76205	0.86701	0.84444	0.00130	0.00059	0.00068	2,136.38	2,048.03	1,987.38	2,056.55	2,092.00	1,983.30	3.88%	2.10%	0.21%
6	CIPLA.xls	0.65099	0.77867	0.78377	0.00045	0.00021	0.00030	216.82	222.25	212.14	219.75	224.00	214.00	1.33%	0.78%	0.87%
7	Grasim Industries.xls	0.68232	0.46181	0.61567	0.00040	0.00062	0.00082	2,677.90	2,674.59	2,626.63	2,574.70	2,706.90	2,555.00	4.01%	1.19%	2.80%
8	HDFC.xls	0.65904	0.71543	0.70504	0.00095	0.00150	0.00120	2,518.57	2,597.85	2,487.20	2,383.75	2,611.00	2,365.00	5.66%	0.50%	5.17%
9	HDFC Bank.xls	0.75349	0.64414	0.71376	0.00054	0.00041	0.00143	1,326.36	1,401.00	1,451.65	1,319.95	1,390.00	1,293.05	0.49%	0.79%	12.27%
10	Hindalco.xls	0.80496	0.82818	0.70458	0.00096	0.00054	0.00117	173.14	170.78	172.42	164.75	177.90	163.35	5.09%	4.00%	5.55%
11	HUL.xls	0.81595	0.79528	0.65629	0.00029	0.00029	0.00066	239.60	232.72	223.74	228.70	244.00	225.25	4.77%	4.62%	0.67%
12	ICICI Bank.xls	0.82960	0.90333	0.78500	0.00065	0.00029	0.00096	808.86	803.06	758.04	770.10	825.00	767.00	5.03%	2.66%	1.17%
13	Infosys Technologies.xls	0.79177	0.75997	0.79472	0.00070	0.00067	0.00048	1,495.70	1,505.53	1,403.02	1,430.15	1,510.00	1,406.35	4.58%	0.30%	0.24%
14	ITC.xls	0.78089	0.83183	0.79937	0.00036	0.00022	0.00040	208.14	208.40	198.78	206.35	210.90	203.75	0.87%	1.19%	2.44%
15	L&T.xls	0.69222	0.82259	0.80364	0.00069	0.00064	0.00069	3,086.92	3,122.23	3,081.70	3,024.80	3,148.00	2,988.00	2.05%	0.82%	3.14%
16	M&M.xls	0.84801	0.71265	0.83592	0.00030	0.00082	0.00057	719.83	705.61	660.30	695.65	717.75	675.00	3.48%	1.69%	2.18%
17	Maruti Suzuki.xls	0.74994	0.80117	0.78654	0.00030	0.00028	0.00038	847.05	845.96	803.19	829.55	840.00	817.00	2.11%	0.71%	1.69%
18	NTPC.xls	0.92602	0.82916	0.87272	0.00037	0.00044	0.00064	201.97	200.40	191.59	197.00	204.90	195.10	2.52%	2.19%	1.80%
19	ONGC.xls	0.60441	0.73710	0.77013	0.00084	0.00041	0.00063	1,003.81	1,021.40	1,004.20	981.35	1,065.00	976.00	2.29%	4.09%	2.89%
20	Ranbaxy.xls	0.74189	0.38867	0.73545	0.00033	0.00172	0.00048	439.48	441.44	444.40	438.75	446.55	431.00	0.17%	1.14%	3.11%
21	Reddy Lab.xls	0.69789	0.73484	0.77623	0.00046	0.00026	0.00034	581.53	595.71	560.73	590.95	600.00	582.00	1.59%	0.71%	3.66%
22	Reliance Comm.xls	0.29938	0.77435	0.78468	0.00182	0.00131	0.00108	559.21	498.94	515.97	508.30	533.90	502.50	10.02%	6.55%	2.68%
23	Reliance Energy.xls	0.76289	0.80508	0.76550	0.00166	0.00085	0.00256	1,339.84	1,317.10	1,242.27	1,251.15	1,325.00	1,241.20	7.09%	0.60%	0.09%
24	RIL.xls	0.81465	0.84091	0.40758	0.00038	0.00056	0.00345	2,297.89	2,354.06	2,437.03	2,264.50	2,340.00	2,251.00	1.47%	0.60%	8.26%
25	Satyam Comp.xls	0.73054	0.74857	0.63929	0.00077	0.00052	0.00116	421.96	411.76	413.52	394.55	408.50	388.05	6.95%	0.80%	6.56%
26	SBI.xls	0.87613	0.85269	0.83625	0.00038	0.00055	0.00050	1,663.58	1,589.60	1,539.64	1,598.85	1,669.00	1,590.00	4.05%	4.76%	3.17%
27	Tata Motors.xls	0.70577	0.72561	0.68724	0.00040	0.00036	0.00096	630.92	638.85	613.58	623.45	645.00	620.00	1.20%	0.95%	1.04%
28	Tata Steel.xls	0.76296	0.84376	0.71505	0.00064	0.00028	0.00144	702.40	714.41	657.54	693.15	715.70	678.65	1.34%	0.18%	3.11%
29	TCS.xls	0.68959	0.70212	0.61442	0.00047	0.00041	0.00100	832.14	854.15	806.62	810.90	869.00	801.00	2.62%	1.71%	0.70%
30	Wipro.xls	0.54983	0.81320	0.64398	0.00173	0.00044	0.00147	480.95	460.28	415.97	425.30	452.00	411.00	13.08%	1.83%	1.21%

REFERENCES

Abhyankar, A., L.S. Copeland and W. Wong (1997), 'Uncovering Nonlinear Structure in Real time Stock-Market Indices: The SP 500, the DAX, the Nikkei 225 and FTSE-100', *Journal of Business and Economic Statistics*, 15(1):1-14.

Abraham A. (2004), "Meta-learning evolutionary artificial neural networks," Netherlands, Vol. 56c, p. 1.

Adya M. and Collopy F. (1998): How effective are neural networks at forecasting and prediction? A review and evaluation http://collopy.case.edu/researchArticles/neuralnets.pdf

Armano G., Marchesi M., and Murru A. (2005) A hybrid genetic neural architecture for stock indexes forecasting http://dx.doi.org /10.1016/j.ins..

Aydingurel Brinkster, FAQ. http://aydingurel.brinkster.net/neural/faq.html#5

Bordoloi, S. (2001), 'Genetically Engineered Neural Network: An Application for Predicting Daily Exchange Rate', *37th Indian Econometric Conference*, 1315 April, Surat, India.

BSE, Website http://www.bseindia.com/histdata/stockprc2.asp?

Canadian Crude Oil Producers Association

http://tonto.eia.doe.gov/dnav/pet/pet_pri_wco_k_w.htm

Chan, Wai-Sum and Ng M. (2004), 'Robustness of Alternative Non-linearity Tests for SETAR Models', *Journal of Forecasting*, 23(3): 21531.

Chang P. C. and Liu C. H. (2006) A task type fuzzy rule based system for stock price prediction. //dx.doi.org/10.1016/j.eswa.2006.08.020

Chen, N.F., Roll R. and Ross S.A. (1986), 'Economic Forces and Stock Market', *Journal of Business*, 59(3): 383403.

Chen S. and Liao C. (2005) Agent-based computational modeling of the stock price-volume relation. http://dx.doi.org/ 10.1016/j.ins.2003.03.026

Chen S.H. (2007): Computationally intelligent agents in economics and finance. http://dx.doi.org/ 10.1016/j.ins.2006.08.001

Clements, M. P. and Smith J. (1997), 'A Monte Carlo Study of the Forecasting Performance of Empirical SETAR Models', Working Paper, University of Warwick, UK.

Coupelon Olivier (2007), Neural network modeling for stock movement prediction A state of the art Master's Degree in Computer Science, Blaise Pascal University olivier.coupelon@wanadoo.fr

Duch W. and Jankowski N. (2001) Transfer functions: hidden possibilities for better neural networks. http://www.dice.ucl.ac.be/Proceedings/esann/esannpdf/es2001-400.pdf Neural Network Modeling for Stock Movement Prediction 5

Gradojevic N. (2006) Non-linear, hybrid exchange rate modeling and trading profitability in the foreign exchange market. In Press, Corrected Proof. http://dx.doi.org/ 10.1016/j.jedc.2005.12.002

Hassan M. R., Nath B., and Kirley M. (2006) A fusion model of hmm, ann and ga for stock market forecasting.//dx.doi.org/ 10.1016/j.eswa. 2006.04.007

Ingber L. (1996), 'Statistical Mechanics of Non-linear Non-equilibrium Financial Markets: Applications to Optimized Trading', *Mathematical Computer Modeling*, 23(7): 101-121.

Jacobs R. A., (1987) "Increased rates of convergence through learning rate adaptation," Amherst, MA, USA, Tech. Rep. http://www.bcs.rochester.edu/people/robbie/jacobs.nn88.pdf

Kaastra I. and Boyd M. (1996) Designing a neural network for forecasting financial and economic time series. http://www.geocities.com/francorbusetti/KaastraArticle.pdf

Kamath, M.V. (2001), 'Empirical Investigation of Multifactor Asset Pricing Model using Artificial Neural Networks', NSE Research Initiative, Paper No.10 NSE, India.Kim K. Jae (2006) Artificial neural networks with evolutionary instance selection for financial forecasting http://dx.doi.org/10.1016/j.eswa.2005

Klimasauskas C. (1993) Applying neural networks. In Neural Networks in Finance and Investing, chapter 3, pages 4772. Probus Publishing Company.

Kutsurelis Jason E. (Lieutenant) (1998), Forecasting Financial Markets using Neural Networks: An Analysis of Methods and accuracy, United States Navy B.S., United States Naval Academy, 1991 Master of Science in Management.

Lo, W.A. and Mackinlay A. Craig (1999), *A Non-Random Walk Down Wall Street*. Princeton, NJ: Princeton University Press.

Malkiel, B.G. (1996): A *Random Walk Down Wall Street*. New York: Norton.

Nag, A. and Mitra A. (2002), 'Time Series Modeling with Genetic Neural Networks: Case Studies of Some Important Indian Economic and Financial Series', *Statistics and Applications*, 4(1): 3757.

NeuroDimension, Inc. 3701 NW 40th Terrace, Suite 1 Gainesville, FL 32606

Neural Network in Plain English. Online: http//:www.ai-junkie.com/ann/evolved/nnt1.html - 9k

O'Connor N. and Madden M. G. (2006), A neural network approach to predicting stock exchange movements using external factors. http://dx.doi.org/10.1016/j.knosys.2005.11.015

Panda Chakradhara & Narasimhan V. (2006), 'Predicting stock returns an Experiment of Artficial Neural Network in Indian Stock Market', South Asia Economic Journal 7:2 (2006) Sage Publication, New Delhi

Qi, M. and Maddala G.S. (1999), 'Economic Factors and the Stock Market: A New Perspective', *Journal of Forecasting*, 18(3): 15166.

Ramon Lawrence (1997), Using Neural Networks to Forecast Stock Market Prices, Department of Computer Science, University of Manitoba.

Reserve Bank of India, Mumbai Online http://www.rbi.org.in/scripts/statistics.aspx

Rumelhart D. E., Hinton G. E., and Williams R. J., (1986) “Learning internal representations by error propagation,” pp. 318362.

Shachmurove, Y. and Witkowska D. (2000), 'Utilizing Artificial Neural Network Model to Predict Stock Markets', Caress Working Paper, Series No. 00-11. Pennsylvania: University of Pennsylvania, Center for Analytic Research in Economics and the Social Sciences.

Stergiou Christos and Siganos Dimitrios, Neural Networks, Imperial College of Science, Technology and Medicine, London SW7 2AZ England, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Contents

Taylor, S.J. (1994), *Modeling Financial Time Series*. Chichester: J. Wiley Sons.

Tsang P. M., Kwok P., Choy S., Kwan R., Ng S., Mak J., Tsang J., Koong K., and Wong T. (2007) Design and implementation of nn5 for hong kong stock price forecasting. http://dx.doi.org/10.1016/j.engappai.2006.10.002

World Gold Association http://www.usagold.com/reference/prices/history.html,

Weckman G. and Agarwala R. (2003) Identifying relative contribution of selected technical indicators in stock market prediction htpp://www.engineer.tamuk.edu/departments/ieen/faculty/ranjeet/Publications/IERC-Sens-2003-Upd.pdf

White, H. (1988), 'Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns', *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 2: 45158.

White, H. (1989), 'Some Asymptotic Results for Learning in Single Hidden-layer Feedforward

Network Models', *Journal of American Statistical Association*, 84(408): 100313.

Yao J. and Tan C. L. (2001) Guidelines for financial forecasting with neural networks. http://www2.cs.uregina.ca/_ jtyao/Papers/guide iconip01.pdf

Yu T., Chen S. H. and Kuo T. W. (2004) A genetic programming approach to model international short-term capital flow. http://www.aiecon.org/ staff/shc/pdf/CH02.pdf

Zhang G. P., Patuwo B., and Hu M. (1998) Forecasting with artificial neural networks: The state of the art.

Zhang G. P., (2003) Neural Networks in Business Forecasting. Hershey, PA, USA: Idea Group Publishing. http://books.google.com/books?id=3cukH5eMpMgC