Stock Picking Using Dynamic Evolving Neural Fuzzy Inference System (DENFIS)

G.Anuradha and Shalini Bhatia

***Abstract*-- This paper investigates the method of forecasting stock price difference on price series data using neuro-fuzzy systems and neural networks. Trading profits in terms of portfolio end values are obtained using dynamic evolving neural fuzzy inference system (DENFIS) forecast model and benchmarked against stock trading without forecast model.**

***Index Terms*--The Forecasting theory, fuzzy neural networks, Stock market, time series prediction, Dynamic Evolving Neural Fuzzy Inference System (DENFIS), online adaptive learning, online clustering.**

I. Introduction

Savings is the essence of life. One of the modes of saving is investment. Amongst various alternate avenues for investments, Share investments are considered to be one of the best possible choices for many. The capital of a company is divided into number of units called as shares. These shares are initially issued to the public by way of IPO(Initial public offer). After the IPO and subsequent listing in the stock exchange the shares are traded in the Stock Market. A **stock market** is a market for the trading of the Shares which are listed on the stock exchanges. The price at which stocks or shares of a company are being traded is known as Stock Price. There are two major approaches to the analysis of stock market price predictions viz. Fundamental Analysis and Technical Analysis. Fundamental Analysis is the analysis of factual information like financial data viz Turnover, income, expenditure, profit, Assets, Liabilities etc. This information is used to derive a fair price of the share of the company. Technical Analysis is the study of past financial market data, primarily through the use of charts. It forecasts price trends and helps investment decisions.Besides trends in prices it helps in evolving trading strategies of entry or exit in a share or scrip.

G.Anuradha is a M.E. (Computer Engineering) student of Tahoma Shaman Engineering College, Mumbai, Maharashtra, India (e-mail:ganusrinu4@yahoo.co.in).

Shalini Bhatia is the Head, Computer Engineering Department, Thadomal Shahani Engineering College, Mumbai, Maharashtra, India (e-mail: shalini.tsec@gmail.com).

The main approach in financial forecasting is to identify trends at an early stage. Popular trading rules like Moving average, Autoregressive Integrated Moving Average (ARIMA), [1] Autoregressive Conditional Heteroskedasticity/Generalized Autoregressive Conditional Heteroskedasticity (ARCH/GARCH) fail to give satisfactory forecast for some series because of their linear structures [2]. Increasingly applications of artificial intelligence (AI) techniques, mainly artificial neural networks, have been applied to technical financial forecasting as they have the ability to learn complex nonlinear mapping and self-adaptation for different statistical distributions [3][4]. Although neural networks possess the properties required for technical financial forecasting, they cannot be used to explain the causal relationship between input and output variables because of their black box nature. [5] Neuro-fuzzy hybridization synergies neural networks and fuzzy systems. It combines the human-like reasoning style of fuzzy systems with the learning structure of neural networks.

NFSs incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. Thus the main strength of NFSs is that they are universal approximators with the ability to solicit interpretable IF-THEN rules.

This paper proposes a new type of fuzzy inference system denoted as DENFIS (Dynamic evolving neural-fuzzy inference system)[6] for adaptive online and offline learning, and its application in time series prediction. DENFIS evolve through incremental, hybrid (supervised/unsupervised) learning and accommodate new input data, including new features, new classes etc., through local element tuning. New fuzzy rules are created and updated during the operation of the system. At each time moment, the output of DENFIS is calculated through a fuzzy inference system based on m-most activated fuzzy rules, which are dynamically chosen from a fuzzy rule set. Evolving Clustering method-ECM [7] was specially designed for DENFIS. ECM dynamically performs scatter partitioning of the input space, creates fuzzy inference rules and evolves a fuzzy system. Section II reviews the two main NFSs and outlines the DENFIS approach. Section III elaborates the principle of Evolving Clustering Method (ECM) and its algorithm. Section IV reviews existing trading models with and without forecast. The trading profits in terms of portfolio end values are presented and compared with trading without forecast model. Section V concludes this paper.

II. NEURO-FUZZY APPROACHES

A. Types of approaches

The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability verses accuracy. The fuzzy modeling research field is divided into two areas: *linguistic fuzzy modeling* that is focused on interpretability, mainly the Mamdani model (1) and *precise fuzzy modeling* that is focused on accuracy, mainly the Takegi-Sugeno-Kang (TSK) model given in (2) and (3).

R_k:IF x_1 is $A_{k,1}$ AND…x_{n1} is $A_{k,n1}$ THEN y is B_k (1)

R_k:IF x_1 is $A_{k,1}$ AND…x_{n1} is $A_{k,n1}$ THEN y_k=a_kx + b_k (2)

$$y = \sum wk(x)yk \quad (3)$$

Where x,y are the input vector $x=[x_1,x_2,\ldots x_n]$ and output value respectively; $A_{k,I}$,B_k are the linguistic labels with fuzzy sets associated defining their meaning; n_1 number of inputs and n_3 are the number of rules. The consequents in (1) are linguistic labels whereas the consequents in (2) are linear functions of the inputs. TSK model has increased representative power compared against the Mamdani model though it has decreased interpretability.

B. General principle of DENFIS

DENFIS use Takagi-Sugeno type fuzzy inference engine. Such inference engine used in DENFIS is composed of *m* fuzzy rules.

if x_1 is R_{11} ……..and x_q is R_{1q} then y is $f_1(x_1 \ldots x_q)$. (4)

if x_2 is R_{21} ……..and x_q is R_{2q} then y is $f_2(x_1 \ldots x_q)$ (5)

……

if x_1 is R_{m1} ……..and x_q is R_{mq} then y is $f_m(x_1 \ldots x_q)$. (6)

"x_j is $R_{i,j}$" i=1,2,…..m: j=1,2,….q are $m*q$ fuzzy prepositions as m antecedents from m fuzzy rules respectively; x_j , j = 1,2,…q are antecedent variables defined over universes of discourse X_j, j =1,2….q and $R_{i,j}$, i =1,2,….m;j=1,2,..q are fuzzy sets defined by their fuzzy membership functions $\mu_{Rij}:X_j \rightarrow [0,1]$, i= 1,2….m;j=1,2…..q. In the consequent parts, y is a consequent variable, and polynomial functions f_i,i=1,2…m are employed.

If the consequent functions are crisp constants, i.e.,$f_i(x_1,x_2,\ldots x_q) = C_i$ i = 1,2,……m the system is called as zero-order Takagi-Sugeno type fuzzy inference system. The system is called a first-order Takagi-Sugeno type fuzzy inference system if $f_i(x_1,x_2 \ldots x_q)$ where i = 1,2,…..m are linear functions. If these functions are nonlinear functions it is called high-order Takagi-Sugeno fuzzy inference system. $y = \sum wk(x)yk$ (7)

C. Learning Process in DENFIS

In the DENFIS model, the first order Takagi-Sugeno type fuzzy rules are employed and the linear functions in the consequences can be created and updated by linear least-square estimator(LSE) with learning data. Each of the linear functions can be expressed as

$$y = \beta 0 + \beta 1x1 + \ldots\ldots\ldots \beta qxq . \quad (8)$$

For obtaining these functions a learning data set, which is composed of p data pairs $\{([x_{i1},x_{i2},\ldots x_{iq}],y_i)$, i = 1,2…$p$, is used and the least-square estimator (LSE) of

β= $[b_0\ b_1\ b_2 \ldots b_q]^T$ are calculated as the coefficients *mbib*=$[b_0\, b_1\, b_2 \ldots b_q]^T$, by applying the following formula:

$b = (A^TA)^{-1}A^Ty$ (9)

where

$$A = \begin{pmatrix} 1 & x11 & . & x1q \\ 1 & x21 & . & x2q \\ . & . & . & . \\ 1 & xp1 & . & xpq \end{pmatrix} \quad (10)$$

and y = $[y_1\, y_2 \ldots y_p]^T$. (11)

In the DENFIS model a weighted least-square estimation method is used where the weights are the corresponding distance between the j^{th} example and the corresponding cluster center. The steps involved in the learning process are as follows

The first n_0 learning data pairs from the learning data set are taken.

Using ECM clustering is implemented with these n_0 data for obtaining *m* cluster centers.

For every cluster center C_i, find p_i data points whose positions in the input space are closest to the center, i = 1,2…m

For obtaining the fuzzy rule corresponding to a cluster center, the antecedents of the fuzzy rule are computed using the position of the cluster center (12)and the consequents are computed using the values of P and b(13)

$$\mu(x) = mf(x,a,b,c) = \begin{matrix} 0 & x \le a \\ x-a/b-a & a \le x \le b \\ c-x/c-b & b \le x \le c \\ 0 & c \le x \end{matrix} \quad (12)$$

$$P_w = (A^TWA)^{-1} \quad (13)$$
$$B_w = P_wA^TWy$$

III. EVOLVING CLUSTERING METHOD (ECM)

A. General Principle

Evolving, online, maximum distance-based clustering method called ECM, is proposed to implement a scatter partitioning of the input space for the purpose of creating fuzzy inference rules. This method has two modes: the first one is usually applied to online learning systems, and the second one is more suitable for offline learning systems. The online ECM is used in the DENFIS online model. The online ECM is a fast, one-pass algorithm for a dynamic estimation of the number of clusters in a set of data, and for finding their current centers in the input space. It is a distance-based connectionist clustering method. In this method, evolved nodes represent cluster centers. In any cluster, the maximum distance, *MaxDist*, between an example point and the cluster center, is less than a threshold value, *Dthr*, which has been set as a clustering parameter and would affect the number of clusters to be estimated. The distance between vectors x and y denotes a *general Euclidean distance* defined as follows:

$$\|x-y\| = \frac{\sqrt{\sum_{i=1}^{q} |x_i - y_i|}}{\sqrt{q}} \qquad (14)$$

where $x,y \in R^q$. In the clustering process, the data examples come from a data stream and this process starts with an empty set of clusters. When a new cluster is created, the cluster center Cc, is defined and its cluster radius Ru, is initially set to zero. With more examples presented one after another, some created clusters will be updated through changing their centers positions and increasing their cluster radiuses. Which cluster will be updated and how much it will be changed, depends on the position of the current example in the input space. A cluster will not be updated any more when its cluster radius, Ru, reaches the value that is equal to a threshold value, *Dthr*.

B.ECM Algorithm

Step 0: Create the first cluster C_1^0 by taking the position of the first example from the input stream as the first cluster center Cc_1^0 and setting a value 0 for its cluster radius $R_{u1.}$

Step 1: If all examples of the data stream have been processed, the algorithm is finished. Else, the current input example, x_i is taken and the distances between this example and all n already created cluster centers Cc_j. $D_{i,j} = \|x_i - Cc_j\|$ j= 1,2 …n are calculated.

Step 2: If there is any distance value, $D_{i,j} = \|x_i - Cc_j\|$ *equal* to or less than at least one of the radii, R_{uj}, j=1,2…n, it means that the current example x_i belongs to the cluster C_m with the minimum distance

$D_{im}=\|x_i - Cc_m\|=min(\|x_i - Cc_j\|)$ subject to the constraint $D_{ij} \leq Ru_j$. $j=1,2\ldots n$.

In this case neither a new cluster is created nor any existing cluster is updated and the algorithm returns to step 1. Else – go the next step.

Step 3: Find cluster C_a from all n existing cluster centers through calculating the values $S_{ij}=D_{ij}+Ru_{j,}$ j= 1, 2, n and then choosing the cluster center Cca with the minimum value of S_{ia}.

$$S_{ia}=D_{ia}+R_{ua}=min(Sij) \qquad (15)$$

j=1, 2…n.

Step 4: If S_{ia} is greater than 2 * *Dthr* the example x_i does not belong to any existing clusters. A new cluster is created in the same way as described in Step 0 and the algorithm returns to Step 1.

Step 5: If S_{ia} is not greater than 2 * *Dthr* the cluster *Ca* is updated by moving its center, Cc_a, and increasing the value of its radius Ru_a. The updated radius Ru_a^{new} is set to be equal to $S_{ia}/2$ and the new center Cc_a^{new} to the point x_i is equal to Ru_a^{new}. The algorithm returns to Step 1.

In this way, the maximum distance from any cluster center to the examples that belong to this cluster is not greater than the threshold value, *Dthr*.

IV. FINANCIAL TRADING SYSTEMS

A. General Principles

Stock Trading is buying and selling of company shares and making profits out of the same. The decision making process in stock trading by analyzing technical analysis, fundamental analysis, volatility of scrip, and economic and global factors. Stock traders mainly use technical analysis, which helps to find out the entry and exit of particular scrip.

In order to arrive at a decision, a trading decision model is required. The action of the trading decision model is assumed to be either short (sell) or neutral (hold), or long (buy). This is represented by F (T) where F ε {-1,0,1}. The return from the trading system in terms of profit is modeled by portfolio end value using multiplicative return given by [8][9]

$$R(T) = \{1+F(T-1)\, r(T)\}\{1-\delta|F(T)-F(T-1)|\} \qquad (16)$$

where $r(T) = y(T)/y(T-1)-1$;

(17)

F (T) is the action from the trading system and δ is the transaction rate.

A number of techniques can be used to generate buy and sell signals .One common aproach is the Moving Average. Identifying trends is one of the key functions of Moving Averages. These are used by most traders . Moving averages are lagging indicators ie, that they do not predict new trends, but confirm trends once they have been established. There is one short coming while following Moving averages for trading. When the market is choppy or moving in a close range there will be frequency crossovers and one may not be in a position to take a decision. To overcome this situation two different types of EMA are used(Long term and short term). Short term MA with shorter duration reacts faster to the changes in price trends than the long term moving average. Thus when there is an intersection of the two moving averages buy and sell signals are generated.

MACD (Moving Average Convergence/Divergence) oscillator is a variant of moving averages is the most popular technical indicator which is used by traders to monitor the relationships between two moving averages. This is represented as a fast signal which is the difference in the EMA of a short and a long term average. The EMA of this difference is the slow signal. The signal of the slow signal is either positive, negative or neutral. When MACD has a positive value it means that the short-term average is above the long-term average which is an indication of upward momentum .This implies that it is wise enough to buy at that spur and vice versa. The exponential moving averags(EMA) of a price series is given by

$EMA_{\tau}z = Kz(T) + (1-K)\, EMA_{\tau}z(T-1)$ (18)

(or)

$EMA_{\tau}z$=(current closing–Previous EMA)*factor + Previous EMA. (19)

Where factor=2/n+1 and n is the number of days for which the average is to be calculated.

In a generic trading decision model based on technical analysis with forecast the forecasting module is trained using supervised learning in a trained data set and is then used to forecast an out-of-sample data set. The forecasts are then used an input to trading module to generate trading signals. This gives rise to forecast bottleneck that leads to suboptimal performance. A neurofuzzy forecast bottleneck free stock trading decision model based on technical analysis approach facilitates the synergy between the time-delayed price difference forecast approach with the use of moving average trading rules.

B. Modules of financial trading systems

There are two modules for prediction of stock prices. One method uses stock trading without forecast and the other uses stock trading with forecast. Both the modules use the trading system with trading rules formed by moving averages.

Stock Trading without forecast module

The steps involved in stock trading without forecast using simple moving averages is as follows

The input is selected for a particular stock or scrip from yahoo finance. The details regarding a stock or scrip in stored in a spreadsheet form and converted either using csvformat or dat file. The price verses the time period is plotted in matlab. Exponential Moving Averages (EMA) is used to compute the EMA for 2 different time windows (one short and one long). F (T) which is the sign of the slow signal is computed. The returns in terms of portfolio end value are computed using (16).

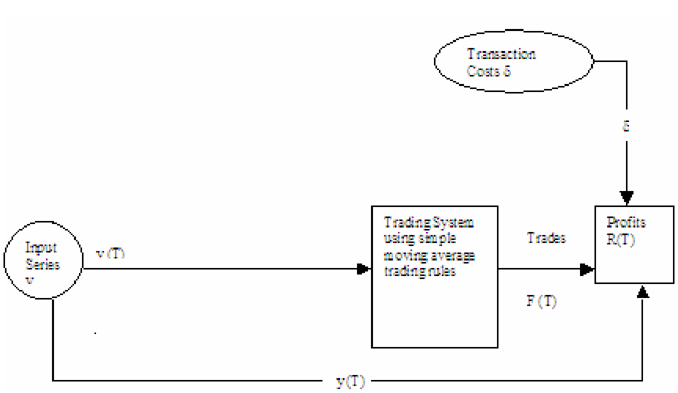

Fig. 1. Trading Decision Model on Technical Analysis approach

Stock trading with forecast.

The steps involved in the Stock trading with forecast are as follows

The differenced price series of the input price values are first computed. From among the given data set (70/30 %) is set aside for training and testing data set. Price difference forecast with DENFIS is used as the forecast model. The output is the forecast value y' (T+1) which is used as the input to the trading system using simple moving average trading rules along with y (T).

The EMA computed uses the forecast price y' (T+1) along with y (T) using the formula

$$EMA_{\tau}y\,(T+1) = Ky'\,(T+1) + (1-K)\, EMA_{\tau}y\,(T) \qquad (20)$$

The fast and slow signals are computed and the trades *F (T)* which is a function of short, neutral, long with $F\varepsilon\{-1,0,1\}$ respectively. It is profitable to sell while in short, retain or hold in neutral and buy while in long. This trades F (T) is used to find the portfolio end value R (T) where R (T) is given by (16) and the transaction cost δ for each scrip. Comparison is made between the portfolio values obtained with forecast using DENFIS and without forecast using simple moving averages and the returns are analyzed.

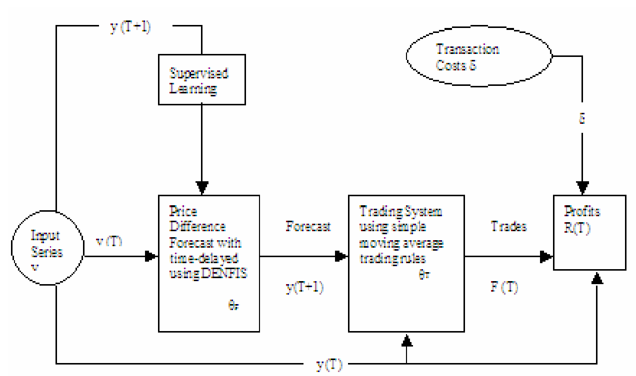

Fig. 2. Overall Architecture of the Proposed system

V. RESULTS AND DISCUSSION

The Stock Trading with non-forecast model was implemented. Inputs were taken from Yahoo finance for three scrips TCS, Indian Oil, Reliance for the periods 12/8/2002 to 20/9/2007 , 2/1/2006 to 30/3/2007, 3/1/2000 to 29/12/2006 respectively.

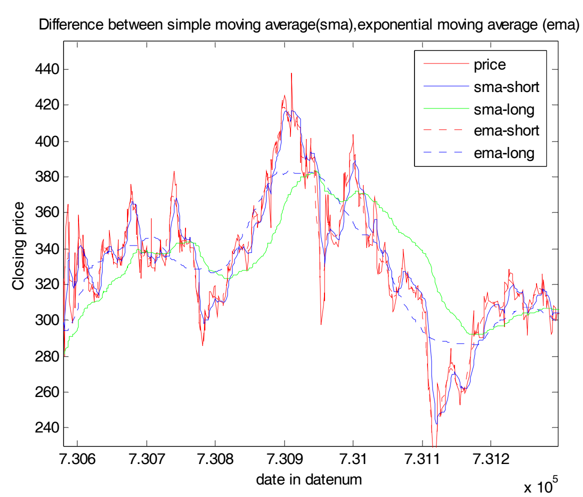

Fig. 3. Plot of Closing Price with SMA and EMA

An average helps to smoothen the jerkiness in a price movement. The market will always fluctuate up and down in waves. The smoothening of the prices by the way of averages helps to gauge the underlying trend. The exponential moving average gives more weight to recent prices in an attempt to make it more responsive to new information. EMA responds more quickly to the changing prices. EMA has higher value when the price is rising and falls faster than simple moving average when the price is falling.

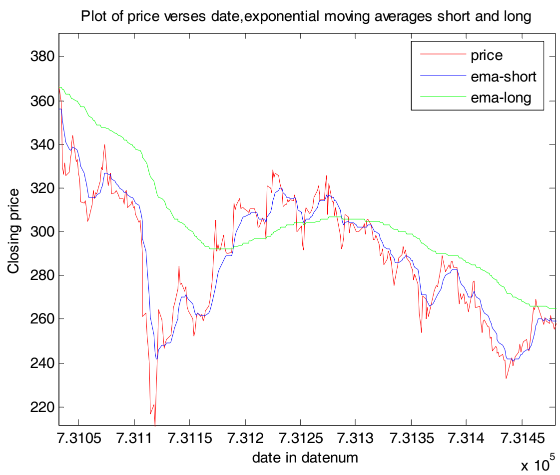

Fig. 4. Plot of Closing Price with EMA – short and EMA-long

The simple rule is when the price cuts the average line from the bottom then a buy signal is generated and when the price cuts the average line from the top a sell signal is generated. When the market is choppy or moving in a close range then there will be frequent crossovers and one may not be in a position to take a decision. To overcome this 2 different types of moving averages one short and other long are used. As the short term moving average is of a shorter duration it will react faster to the changes in price trends than the long term moving average. Thus the intersections of the two averages produce a buy and sell signals. When the shorter term average is cutting the longer term average from the bottom a buy signal is generated and if the short term average cuts the longer term average from top a sell signal is generated.

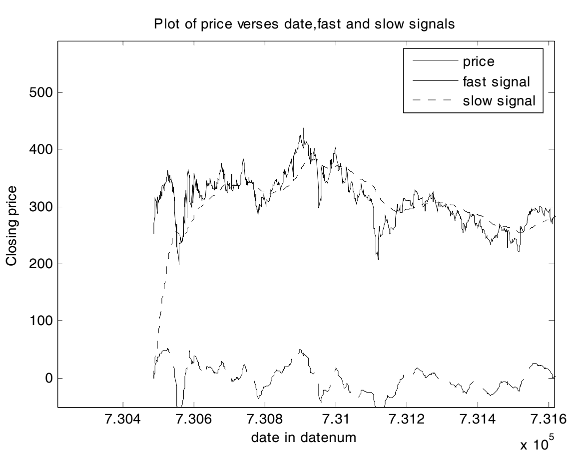

Fig. 5. Plot of Closing Price with Fast and Slow Signals

The simplest form of calculating a trend deviation involves the relationship between the current price and the moving average. The oscillator is constructed by comparing the latest price by the average. When the price and the average are identical, the oscillator is plotted at zero. When it is above the average, the momentum series is in positive territory, and vice-versa. Zero or equilibrium crossovers therefore indicate when the price crosses above and below its moving average. The MACD (moving average convergence divergence) is another way of expressing a trend deviation oscillator. The zero line represents those periods when the two EMAs are identical. When the MACD is above the equilibrium line the shorter average is above the longer one and vice versa. The green line represents the slow signals, which is the EMA of the fast signal (Difference between EMA short and EMA long).

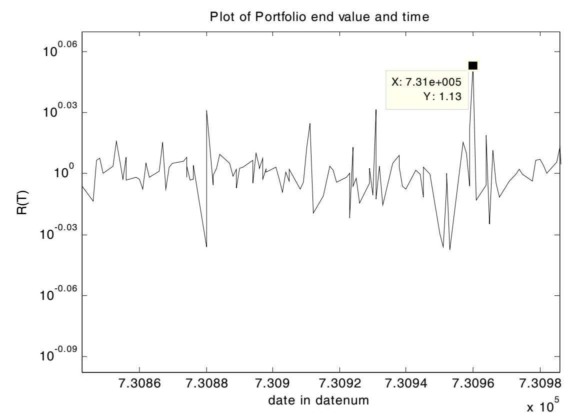

Fig. 6. Plot of Portfolio end value with time

The action of the trading system is assumed to be one of short, neutral, long, and this action is respectively, represented by F (T) where $F \in \{-1,0,1\}$. The sign of slow signal gives the value of F (T). The trading system is modeled by portfolio end value where the transaction cost is assumed to be a fraction δ of the transacted price value. If short i.e F (T)=-1, it means sell first. So if the price goes up, losses are incurred and vice versa. If F (T)=1, it means buy first. So if the price goes up, there is gain and vice versa. If the transaction cost is high, the more traded, the more the losses, because transaction price is a percentage of stock prices.

VI. CONCLUSION

A dynamic Evolving Neuro Fuzzy model called Stock picking using Neuro Fuzzy approach is proposed in this paper. The proposed Stock-picking model circumvents the forecast bottleneck and synergizes the time-delayed price difference forecast approach with the simple moving average rules for generating trading signals.

VI. REFERENCES

[1] http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
[2] http://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity
[3] Y.S. Abu-Mostafu, A.F.Atiya,M. magdon-Ismail, and H.White, “Introduction to the special issue on neural networks in financial engineering”, *IEEE Transaction Neural Networks,* vol.12, no.4, pp.653-656,july 2001.
[4] C.S. Lin, H.A. Khan, and C.C.Huang, “Can the neuro fuzzy models predict stock indexes better than its rivals?” in PROC.CIRJE,Aug 2002, CIRJE-F-165E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propagat.*, to be published.
[5] M. Adya and F.Collopy, “How effective are neural networks at forecasting and prediction? A review and evaluation, “*J.Forecasting*,vol.17,no.5-6, pp.481-495,Sep 1998.*3)*, pp. 876-880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar
[6] E N.K.Kasabov and Q.Song,”DENFIS: Dyanamic evolving neural fuzzy inference systen and its application for time-series prediction,” *IEEE Trans. Fuzzy Systems,* vol.10, no.2, pp.144-154, April 2002.
[7] Quo Song and Nikola Kasabov,”ECM-A Novel On-line Evolving Clustering Method and its applications, Department of Information Science, University of Otago.
[8] J.M.Moody, vol. 17, no.5-6, pp.441-470, Sep. M.Moody, L.Wu, Y.Liao, and M.Saffell, “Performance functions and reinforcement learning for trading systems and portfolios”, *J.Forecasting* 1998.

VII. BIOGRAPHIES

Shalini Bhatia was born on August 08, 1971. She received the B.E. degree in Computer Engineering from Sri Sant Gajanan Maharaj College of Engineering, Amravati University, Shegaon, Maharashtra, India in 1993, M.E. degree in Computer Engineering from Thadomal Shahani Engineering College, Mumbai, Maharashtra, India in 2003.

She has been associated with Thadomal Shahani Engineering College since 1995, where she has worked as Lecturer in Computer Engineering Department from Jan 1995 to Dec 2004 and as Assistant Professor from Dec 2004 to Dec 2005. Since Jan 2006 she is looking after the department as the Head. Her research interests include neural networks, fuzzy systems, bioinformatics, intelligent systems, distributed computing, image processing, and advanced computer architecture. She has published a number of technical papers in National and International Conferences. She is an active member of CSI and also a member of Special Interest Group in Artificial Intelligenge (SIGAI) which is a part of CSI**.**

G.Anuradha was born on February 1, 1970. She received the B.E. degree in Electronics and Tele Communications Engineering from Hindustan College of Engineering , Madras University, Padur, Chennai TamilNadu , India in 1991. She is pursuing M.E. degree in Computer Engineering from Thadomal Shahani Engineering College, Mumbai, Maharashtra, India.

She is working with St.Francis Institute of Technology, Borivli (W) Mumbai from the year 2004 as Lecturer in Information Technology Department. Her research interests include Data Bases, Advanced Databases, Data warehousing and Data mining and Computer Networks.