Quantitative and qualitative approaches for stock movement prediction

Jordi Petchamé Sala[a] , Àngela Nebot[a] and René Alquézar [a]

[a] *Departament LSI, Universitat Politècnica de Catalunya, Edifici Omega, Campus Nord, Jordi Girona Salgado 1-3, Barcelona 08034, Spain*

Abstract. Knowing about future values and trend of stock market has attracted a lot of attention by researchers, investors, financial experts and brokers. The benefits of having a good estimation of the stock market behavior are well-known, minimizing the risk of the investment and maximizing the profits. In recent years, mathematical and computational models from artificial intelligence have been used for such purpose. This research studies quantitative and qualitative modeling approaches to forecast four indices of the Indian stock market. As quantitative methodologies we use time delay feed forward neural networks, auto regressive integrated moving average and their combination. As a qualitative counterpart we use the fuzzy inductive reasoning methodology. 10-fold cross validation has been used to evaluate the generalization capacity of each predictive model developed. The best results are obtained with the time delay feed forward neural networks models and the worst with the fuzzy inductive reasoning models. No significant enhancement is obtained with the approaches proposed when compared with the simple random walk method.

Keywords. artificial intelligence, stock movement prediction, neural networks, autoregressive integrated moving average, fuzzy inductive reasoning.

1. Introduction

The price variation of stock market is a non-linear dynamic system that deals with non-stationary and volatile data. This is the reason why its modeling is not a simple task. In fact, it is regarded as one of the most challenging modeling problems due to the fact that prices are stochastic.

Two main tendencies exist when dealing with stock market time series prediction. The first one is based on the efficient market theory and it explains that the session price gathers all events and therefore the best possible prediction is the last value. It is called random walk (RW). The second one believes in the existence of patterns in financial time series which make them predictable. The researchers that believe in the second line have centered their work mainly in two different approaches: statistical and artificial intelligence (AI). The statistical techniques most used in financial time series modeling are the autoregressive integrated moving average (ARIMA) [1-2], the generalized autoregressive conditional heteroskedasticity (GARCH) [3-4] and the smooth transition autoregressive (STAR) [5-6].

On the other hand, artificial intelligence provides sophisticated techniques to model time series and search for behavior patterns: genetic algorithms, fuzzy models, artificial neural networks (ANN), support vector machines (SVM), hidden markov models, multi-agents and expert systems, are some examples. Unlike statistical techniques, they are capable of obtaining adequate models for nonlinear and unstructured data. Neural networks are the most widely used method for this task, because of their superior performance in some applications. There exists a huge amount of literature that uses AI approaches for time series forecasting [7-11].

There are interesting researches that apply AI techniques to predict the stock market movement. One of these works is the one of O'Connor and Madden that evaluates the effectiveness of neural networks using external indicators, such as commodity prices and currency exchange rates, for predicting movements in the Dow Jones industrial average index[12] and involving the use of trading simulations to assess the practical value of predictive models. Their models bet the buy-hold strategy (simple benchmark that responds to underlying market growth). In [13] the authors studied the importance of the neural network architecture for predicting stock prices obtaining as result that the most accurate neural network was the one with two hidden layers. Zorin compared Kohonen self-organizing maps and error backpropagation algorithm on Ventspils Nafta stock prices, concluding that the Kohonen approach has better performances in this application [14]. More complex systems have also been developed recently. In [15], the adaptive neuro-fuzzy inference system (ANFIS) has been used to predict the trend of the stock price of the Iran khodro corporation at Tehran stock exchange, concluding that ANFIS is capable of forecasting the stock price behavior. In [16], the authors investigate whether a hybrid approach combining different stock prediction techniques, i.e. ANFIS, back-propagation network and SVM, could outperform the single approach on S&P 500 index. Their results show better performances by combining approaches.

The research presented here is mainly based on the paper written by Merh et al. [17], that illustrates a comparison between hybrid models for the prediction of four Indian stock indices, i.e. BSE 30, BSE IT, S&P Nifty, BSE 100. The hybrid models were developed using the combination of ANN (a non-linear multi-layer perceptron) and ARIMA techniques. They tested ARIMA_ANN and ANN_ARIMA combinations and concluded that the ANN_ARIMA model had better prediction results for the BSE 30, BSE IT and S&P Nifty stock indices, whereas in the case of BSE 100 the ARIMA_ANN got higher accuracy. The best results presented in the paper are quite good if compared with the random walk option. In fact in almost all the indices the errors obtained are surprisingly low. However, these results are based on a unique partition between training and test sets. The size of the only test data set used is of 30 samples (days) whereas the training set contains 1218 days. We consider that the test set is not statistically significant and, therefore, we do not think that the results obtained in that study can be considered relevant and, definitely, do not show the generalization power of the models developed.

In order to study the generalization power of quantitative and qualitative approaches to the same stock indices in a rigorous manner, we have decided to use 10-fold cross validation. We compare quantitative techniques like TDFN, ARIMA and their combination TDFN_ARIMA (unlike the ANN used in [17], TDFN adds delayed inputs as well), and a qualitative technique like FIR, versus the random walk option. The hypothesis is that the predictions can be improved applying hybrid techniques.

2. Methodologies

As mentioned before, in this research we want to study the usefulness of quantitative and qualitative methodologies for modeling Indian stock indices. The quantitative methodologies chosen are time delay feed forward neural networks (TDFN), auto regressive integrated moving average (ARIMA) and their combination TDFN_ARIMA. As a qualitative counterpart we use the fuzzy inductive reasoning (FIR) methodology. In this section, the main characteristics of TDFN and FIR methodologies are presented, as well as the combination of TDFN_ARIMA. For an insight related to the ARIMA technique the reader is referred to [18].

TDFN belongs to dynamic networks. Its main characteristic is that the output depends not only on the current input but also on the previous inputs. This is solved by adding delays in the configuration of the network. A straightforward method for implicit representation of time is to add a short-term memory structure in the input layer of a static neural network (e.g., multilayer perceptron). The resulting configuration is sometimes called a focused time-lagged feed-forward network (TLFN) or focused time delay network (FTDN). A short-term memory structure can be implemented as a tapped delay line (TDL) [19], although there exist other ways to do it like gamma memory [20]. However, the TDL is the most commonly used form of short-term memory. It consists of *p* unit delays with *p + 1* terminals, as shown in Figure 1, which might be seen as a single input multiple output (SIMO) network. The memory depth of a TDL is fixed at *p*, and its memory resolution is fixed at unit *y*, giving a depth resolution constant of *p*.

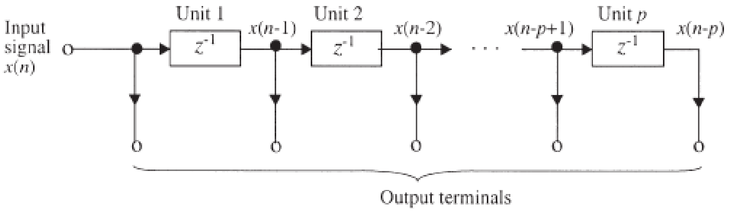


Figure 1. ordinary tapped delay line memory of order *p*

In this research we define dynamic networks by means of a TDFN with a tapped delay line at the input. Figure 2 illustrates the structure of such a network that will be used in this study for the application at hand. These kinds of structures are available in the Matlab framework.

As hybrid model, we have applied TDFN_ARIMA, which has two stages. In the first stage TDFN is used to forecast the future value of the close price, and then, the residual generated is provided to the ARIMA, which will forecast the error. In the second stage the predicted close price by TDFN is added to the error forecast generated by ARIMA, thus yielding the final forecasted value.

With respect the qualitative approach we use the FIR methodology [21]. The conceptualization of the FIR methodology arises of the general system problem solving (GSPS) approach proposed by Klir [22]. This methodology of modeling and simulation is able to obtain good qualitative relations between the variables that compose the system and to infer future behavior of that system. It has the ability to describe systems that cannot eas-

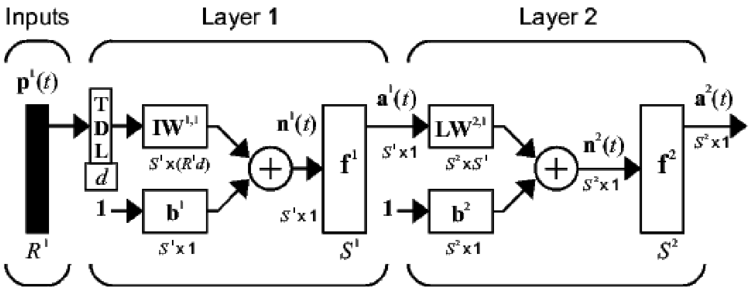


Figure 2. two-layer TDFN

ily be described by classical mathematics or statistics, i.e. systems for which the underlying physical laws are not well understood. FIR offers a model-based approach to predicting either univariate or multi-variate time series [23]. It is a qualitative, non-parametric, shallow model based on fuzzy logic.

In a first step, the available measurement data are fuzzified. Thereby, the real–valued quantitative data values are mapped onto qualitative triples, consisting of a class value, a fuzzy membership value, and a side value. The process is illustrated in Fig.3 by means of a variable, called ambient temperature.

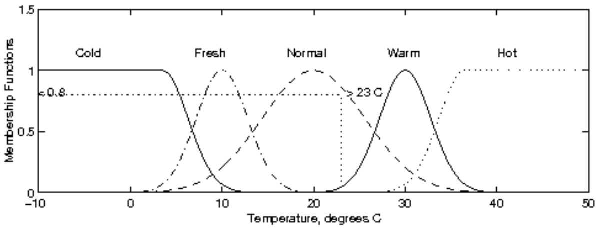


Figure 3. Fuzzification in FIR

The ambient temperature is mapped onto five discrete classes, called *cold*, *fresh*, *normal*, *warm*, and *hot*. Each of these classes is associated with its own fuzzy membership function, a function with values in the range $[0.0, 1.0]$. The fuzzy membership functions can either be gaussian distributions or triangular distributions. The side function can assume values of either *left*, *center*, or *right*. Within each fuzzy class, values to the left of the maximum of the fuzzy membership function have associated a side value of *left*, etc. Hence, an ambient temperature of 23 degrees centigrade is classified as *normal*, with a fuzzy membership value of 0.8, and a side value of *right*. No information is being lost in the process of fuzzification. The qualitative triple can be mapped unambiguously onto a single quantitative value by means of de-fuzzification.

The FIR modeling engine reasons only about the class values. In the case of a univariate time series, the next value of the variable, $x(t)$, must be a function of previous recordings of that same variable:

$$x(t) = f(x(t - \Delta t), x(t - 2\Delta t), \ldots) \tag{1}$$

The FIR modeling engine does not try to identify the function, f. It only determines, which subset of previous recordings is most useful in determining the next value of the variable, x, e.g.

$$x(t) = f(x(t-5\Delta t), x(t-2\Delta t), x(t-\Delta t)) \tag{2}$$

which would be represented as a so-called *optimal mask*:

$$\text{mask} = [-1, 0, 0, -2, -3, +1] \tag{3}$$

where the $+1$ element denotes the position of the output within the time series, whereas the negative values in the mask denote the relative positions of the three inputs.

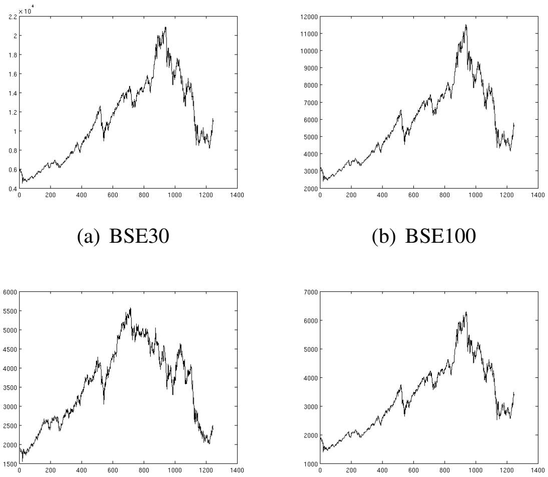
The FIR modeling engine searches through all possible masks up to a given mask depth, creating for each mask an input/output table of class values. The optimal mask is the one that makes the map from the set of input classes to the single output class as deterministic as possible. The FIR modeling engine optimizes the information content of the map by minimizing the Shannon entropy measure. Once the optimal mask has been found, FIR stores the training data for later retrieval in an *experience data base* consisting of an alpha-numerically sorted list of input/output data, whereby each quantitative input/output data record is converted into a record of qualitative triples.

The FIR simulation engine predicts values of the output variable beyond the end of an episode of recorded data values. It uses the previously found optimal mask. The inputs to the mask are inside the known episode, i.e., have known values. They are fuzzified, and a qualitative input record is created that can be compared with the records in the experience data base. The five nearest neighbors are retrieved, and the output value is predicted as a qualitative triple representing a weighted average of the output values of the five nearest neighbors in the experience data base.

3. Results

This research is focused on forecasting four indices:

1. The BSE SENSEX, also called the BSE 30 or simply the SENSEX, is a free-float market capitalization-weighted stock market index of 30 well-established and financially sound companies listed on Bombay stock exchange.
2. The BSE 100 is a free-float market capitalization-weighted stock market index of 100 companies listed on Bombay stock exchange.
3. The BSE IT, also called the BSE-TECk is a a free-float market capitalization-weighted stock index and it is constituted of companies in the Information Technology, Media and Telecom sectors.
4. The S&P CNX Nifty is a a free-float market capitalization-weighted stock market index. It is also called the Nifty 50 or simply the Nifty, and this is one of several leading indices for large companies which are listed on National stock exchange of India.



(a) BSE30 (b) BSE100

(c) BSEIT (d) S&P Nifty

Figure 4. Close price of the 4 indices from 6th April 2004 to 6th April 2009

For each index, four variables are available, i.e. open, high, low and close daily prices. These data are comprehended from the 6th of April 2004 to the 6th of April 2009 (1248 trading days). Figure 4 shows the close daily process variables for each index studied.

In this research TDFN, TDFN_ARIMA and FIR models have been developed for each of the four Indian stock indices mentioned earlier. For each model, the inputs considered are yesterday open, high, low and close daily prices and their previous delays. The output variable is today's close price. The performance of these models has been evaluated by computing the errors between the current and the predicted close prices. The previous models results are also compared with classical models like RW and ARIMA.

Therefore, all the models obtained in this study represent a function as the one described in equation 4, although each method uses their own mathematical tools to define their own relation or function *f*.

$$C_t = f\{C_{t-1}, H_{t-1}, L_{t-1}, O_{t-1}, ..., C_{t-p}, H_{t-p}, L_{t-p}, O_{t-p}\} \tag{4}$$

where C is the Close price, H the High price, L the Low price, O the Open price, t the instant to predict and p the number of lags.
In addition, we introduce 10 fold cross-validations in order to estimate how accurately the predictive models will perform in practice. With the data available, i.e. 1248 data values, test sets of 125 values are derived.

The characteristics and structure of the different models developed in this study are:

1. **TDFN**: The data were normalized to mean 0 and variance 1. Different combinations were tested introducing from 1 to 5 lags in the input layer. Then, for each input, different sets of neurons in the hidden layer were tested, i.e. 1, 5 and 15 hidden neurons. As activation function a linear one was finally chosen in both layers, due to the fact that when a sigmoid transfer function, hyperbolic tangent, was used

in the hidden layer the performance of the network was reduced due to a saturation problem, Table 1. Although a linear two-layer TDFN might be compressed into a linear one-layer TDFN, we kept the original two-layer architecture also in this case. The resilient backpropagation was used as learning algorithm. This is a first-order optimization algorithm which takes into account only the sign of the partial derivative over all patterns (not the magnitude), and acts independently on each weight. The resilient backpropagation is one of the fastest weight update mechanisms.

2. **Hybrid TDFN_ARIMA**: The TDFN was obtained as has been explained above. Once the predictions of the TDFN are available, the residues, which are stationary series, are fitted to different ARIMA structures and the one with better performance is kept. The goodness of an ARIMA model is defined by means of Akaike information criterion (AIC):

$$\mathrm{AIC} = \log V + 2d/N \tag{5}$$

   where V is the loss function, d the number of estimated parameters and N the number of values in the estimation data set. The most accurate model has the smallest AIC. Then, the final prediction is the sum of TDFN and ARIMA predictions.
3. **FIR**: The data were fuzzified into three classes each, using the equal frequency partition method. A depth of 5 has been used in this study in order to find the optimal mask. The optimal masks obtained for each index are shown in the fourth column of Table 1.
4. **ARIMA**: These models are developed in an univariant form, thereby they are only applied on close price, as described in equation 6.

$$C_t = C_{t-1} + ... + C_{t-p} \tag{6}$$

   where C is the Close price, t the instant to predict and p the number of lags. The input value (close price) is transformed in log(C$_t$ / C$_{t-1}$), in order to get stationary series. At this point, some models combining AR and MA from 1 to 5 delays have also been tested (up to 25 models).
5. **RW**: The prediction is simply given by the last known value, i.e. $C_t = C_{t-1}$.

Table 1 summarizes the cross-validation performance of the best models obtained for each of the methodologies studied, with respect to its performance. The performance is measured by means of the mean absolute error (MAE).
The mean absolute error (MAE) measure is defined by equation 7.

$$\mathrm{MAE} = \sum_i \frac{\|y_i - y_i'\|}{n} \tag{7}$$

where, y_i is the target value in time i, y_i',the predicted value in time i and n the number of observations.

Table 1 shows that the linear TDFN reaches the highest accuracy, being 4 lags and 1 neuron in the hidden layer the best architecture for that kind of time series. Comparing

	TDFN$_{sigmoid}$	TDFN$_{linear}$	FIR	TDFN_ARIMA	ARIMA	RW
BSE30	4 lags, 5 h.	4 lags, 1 h.	[0, 0, 0, -1, -2, +1]	-	(5,1,2)	-
MAE	156.605	148.926	201.116	199.320	149.144	150.024
BSEIT	2 lags, 5 h.	5 lags, 1 h.	[0, -1, 0, -2, -3, +1]	-	(0,1,5)	-
MAE	53.714	49.634	67.820	67.735	49.716	50.751
BSE100	4 lags, 15 h.	4 lags, 1 h.	[0, 0, 0, -1, -2, +1]	-	(5,1,0)	-
MAE	78.487	77.768	105.564	114.676	77.537	77.915
S&PNifty	2 lags, 5 h.	4 lags, 1 h.	[0, 0, 0, -1, -2, +1]	-	(5,1,0)	-
MAE	47.741	44.539	60.430	60.377	44.556	44.800

Table 1. The cross-validation average MAE for each model

all models, we find out that both TDFN and ARIMA beat very slightly the RW prediction but TDFN_ARIMA and FIR obtain worse results. Adding an ARIMA model in order to improve the TDFN predictions increases the complexity of the model and it introduces noise that impairs the quality of the forecast. However, we think that new hybridization approaches should be studied. On the other hand, the FIR qualitative approach also obtains bad results because a k value of 5 in the prediction process, is probably especially large in this application. Notice that the method finds the 5 nearest neighbors and, afterwards, the prediction is obtained as a weighted average of the outputs associated with these neighbors. If we look to Figure 4 it is clear that the behavior of the close variable (and also the rest of the variables) is not a repeated behavior. Therefore, using more than one neighbor in the prediction process implies increasing the noise associated in the forecast. We think that lower values of k can enhance the performance of FIR methodology in the application at hand.

Figures 5(a) and 5(b) show the predictions obtained by the different methodologies in one fold of the cross-validation for two of the indices and their determination coefficient, which is defined by equation 8,

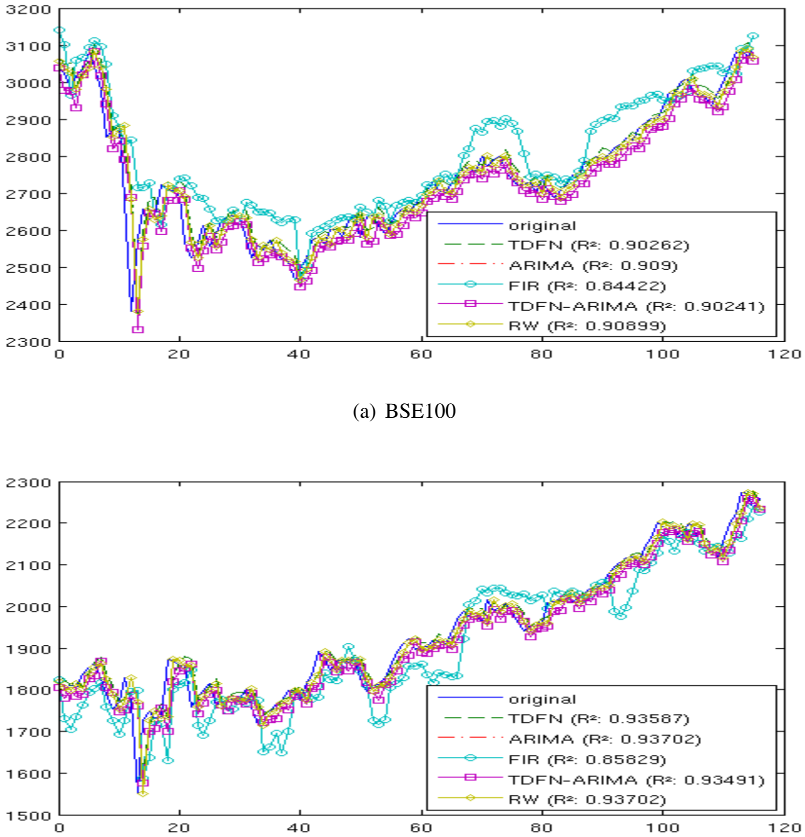
$$R^2(i,j) = \left[\frac{C(i,j)}{\sqrt{C(i,i) \cdot C(j,j)}} \right]^2 \tag{8}$$

where C is the covariance, i and j the time series.

4. Conclusions and future work

Stock price prediction is one of the most challenging problems due to the fact that the series are inherently noisy and non-stationary. In this study we have investigated the performance of different models to forecast the close price stock movement. We have tested both quantitative and qualitative techniques. As quantitative tecniques, we have used both non-linear and linear TDFN, ARIMA and their combination TDFN_ARIMA and FIR as qualitative approach. To assess their goodness we have compared them with the random walk technique.

Summarizing, the best perfomance was obtained by a linear TDFN though its accuracy was only slightly better than ARIMA and RW. Then, we can not conclude at this time that advanced prediction techniques (and, in particular, non-linear techniques) can overcome the performance of the very simple random walk technique for these time se-



(a) BSE100

(b) BSEIT

Figure 5. Predictions and determination coefficient for the BSE100 and BSEIT indices (on the test set of the cross-validation first fold)

ries. In our opinion, the problem does not lie on the prediction techniques, but in the intrinsic difficulty of the stock market time-series prediction task, mainly because much relevant information that actually affects the signal behavior is missing or not easily available. However, we expect to improve the performance taking into account new variables as well as to provide confidence levels for the predictions.

The future research will be focused on the following points:

1. To test other advanced and hybrid techniques: recurrent neural networks, support vector machines, genetic algorithms, neuro-fuzzy systems, etc.
2. To introduce dependent variables: a) macroeconomics variables: GDP, unemployed ratio,etc; b) advanced technical indicators: MACD, MM, RSI, etc; c) correlated indices, daily volume or financial ratios like earnings per share, price-earnings, etc.
3. To study other sample frequencies: minutes, weekly, monthly, etc.
4. To analyse the importance of the trend and the volatility, and to specialize different TDFN for predicting each component

In addition, the study should be extended to predict another kind of financial time series like interest rate, derivative (option and future) and foreign exchange prices, with the final goal of obtaining a useful and powerful tool in financial issues like risk management, asset allocation and automatic trading systems.

References

[1] Ping-Feng and Chih-Sheng Lin, 2005. A hybrid ARIMA and support vector machines model in stock-price forecasting. Omega 33(6), 497–505.

[2] Guillaume Weisang and Yukika Awazu, 2008. Vagaries of the Euro: an Introduction to ARIMA Modeling. CS-BIGS 2(1), 45–55.

[3] Dima Alberga, Haim Shalita and Rami Yosefb, 2008. Estimating stock market volatility using asymmetric GARCH models. Applied Financial Economics 18, 1201–1208.

[4] Him Tang, Kai-Chun Chiu and Lei Xu, 2003. Finite Mixture of ARMA-GARCH Model for Stock Price Prediction. Computational Intelligence in Economics and Finance, 1112–1119.

[5] Kuang-Chung Hsu and Hui-Chu Chiang, 2011. Nonlinear effects of monetary policy on stock returns in a smooth transition autoregressive model. The Quarterly Review of Economics and Finance 51(4), 339–349.

[6] Ahdi Noomen Ajmi and Lanouar Charfeddine, 2011. The tunisian stock market: a regime switching approach. Asian Journal of Business and Management Sciences 1(3), 43–53.

[7] Weigend, A. and Gershenfeld, N., 1994. Time series prediction: forecasting the future and understanding the past. Addison Wesley

[8] Hal S. Stern, 1996. Neural networks in applied statistics. Technometrics 38(3), 205–214.

[9] Brad Warner and Manavendra Misra, 1996. Understanding neural networks as statistical tools. The American Satistician, 50, 284–293.

[10] Aiken and Bsat, 1999. Forecasting market trends with neural networks. Information Systems Management 16(4).

[11] Kuo, R. J., Chen, C. H. and Hwang, 2001. An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. Fuzzy Sets and Systems 118, 21–45.

[12] Niall O'Connor and Michael G. Madden, September 2006. A neural network approach to predicting stock exchange movements using external factors. Knowledge-Based Systems 19(5), 371–378.

[13] Asif Ullah Khan, Mahesh Motwani, Sanjeev Sharma, Abdul Saboor Khan and Mukesh Pandey, 2007. Stock Rate Prediction Using Backpropagation Algorithm: Results with Different Number of Hidden Layers. Journal of Software Engineering, 1, 13–21.

[14] Alexey Zorin, 2003. Stock price prediction: Kohonen versus Backpropagation.

[15] Ebrahim Abbasi and Amir Abouec ,2008. Stock Price Forecast by Using Neuro-Fuzzy Inference System. World Academy of Science, Engineering and Technology 46, 320–323.

[16] Qinghua Wen, Zehong Yang, Yixu Song, 2009. Hybrid Approaches for Stock Price Prediction.

[17] Nitin Merh, Vino P.Saxena and Kamal Raj Pardasani, 2010. A comparison between hybrid approaches of ANN and ARIMA for indian stock trend forecasting. Business Intelligence Journal, 23–43.

[18] Box, G.E.P and Jenkins, G.M., 1970, *Time series analysis: Forecasting and control*, Holden-Day, San Francisco.

[19] Mills, Terence, 1990. Time Series Techniques for Economists.

[20] Jose C. Principe and Jyh-Ming Kuo, and Same1 Celebi, 1994. An Analysis of the Gamma Memory in Dynamic Neural Networks. IEEE Transactions on Neural Networks, 5(2), 331–337.

[21] Antoni Escobet, Angela Nebot, Francois E. Cellier, 2007. Visual-FIR: A tool for model identification and prediction of dynamical complex systems. Simulation Modelling Practice and Theory 16, 76–92.

[22] Klir, G. and Elias, D., 2002, Architecture of Systems Problem Solving, 2 nd ed., (NY: Plenum Press).

[23] Nebot, A., Mugica, F., Cellier, F., Vallverdú, M., 2003. Modeling and Simulation of the Central Nervous System Control with Generic Fuzzy Models. Simulation 79(11), 648–669.

[24] Cellier, F.E., Nebot, A., Mugica, F., De Albornoz, A., 1995. Combined qualitative/quantative simulation models of continuous-time processes using fuzzy inductive reasoning techniques. International Journal of General Systems 24(1–2), 95–116.