ESTIMATION OF RETURN ON INVESTMENT IN SHARE MARKET THROUGH ANN

K.S.Ravichandran[1], P.Thirunavukarasu[2], R. Nallaswamy[3], R.Babu[4]

[1]School of Computing, SASTRA Deemed University, Thanjavur – 613 402, Tamil Nadu, India.

[2]Department of Mathematics, Sri Angalaman College of Engineering & Technology, Siruganoor, Trichy, Tamil Nadu, India.

[3]Department of Mathematics and Computer Applications, National Institute of Technology, Trichy, Tamil Nadu, India.

[4]Shanmugha Polytechnic College, Thanjavur- 613 402, Tamil Nadu, India

ABSTRACT

The stock market is one of the most popular investing places because of its expected high profit. Traditionally, technical analysis approach, that predicts stock prices based on historical prices and volume, basic concepts of trends, price patterns and oscillators, is commonly used by stock investors to aid investment decisions. Advanced intelligent techniques, ranging from pure mathematical models and expert systems to fuzzy logic networks, have also been used by many financial trading systems for investing and predicting stock prices. In recent years, most of the researchers have been concentrating their research work on the future prediction of share market prices by using Neural Networks. But, in this paper we newly propose a methodology inwhich the neural network is applied to the investor's financial decision making to invest all type of shares irrespective of the high / low index value of the scripts, in a continuous time frame work and further it is further extended to obtain the expected return on investment through the Neural Networks and finally it is compared with the actual value. The proposed network has been tested with stock data obtained from the Indian Share Market BSE Index. Finally, the design, implementation and performance of the proposed neural network are described.

Keywords: Indian Stock Market, Neural Networks, Decision Making, Correlation and Regression analysis.

1 INTRODUCTION

From the beginning of time it has been man's common goal to make his life easier. The prevailing notion in society is that wealth brings comfort and luxury, so it is not surprising that there has been so much work done on ways to predict the markets to increase such wealth. Various technical, fundamental, and statistical indicators have been proposed and used with varying results. However, no one technique or combination of techniques has been successful enough to consistently "beat the market". Traditionally, technical analysis approach [4, 16, 17, 18], that predicts stock prices based on historical prices and volume, the Dow Theory, basic concepts of trends, price patterns and oscillators, is commonly used by stock investors to aid investment decisions. Advanced intelligent techniques ranging from pure mathematical models and expert systems [5, 7] to neural networks [1, 2, 3, 8, 9, 10, 11] have also been used by many financial trading systems for stock prediction. Ultimately, most of the researchers have derived the various methodologies for predicting future share market prices using artificial neural network [. But, in this paper the neural network concept is used to calculate the return on investment and finally it is compared with the actual value.

In Indian Share Market, the index of BSE increases or decreases depending upon the performance of the various sub-indexes, namely, BSEIT, BSEED, BSEFMCG, BSEHC, BSECG, TECH, BSEPSU, BANKEX, AUTO, METAL, and OILGAS. These sub-indexes are denoted by the decimals 1 to 11 and in the later part of this paper, we denote these numbers are script numbers. Based on the available data, [6] the ranking of various sub-indexes is established and based on the ranking, we invest the amount invariably to all type of shares and the expected return on investment is calculated through Neural Networks and finally these values are tested with the actual value. The proposed network has been tested with stock data obtained from the Indian Share Market Index BSE.

The paper is organized as follows. Section 2 provides the information related to neural network and its learning rule (back propagation) for predicting sub sectors investment in stock market. Section 3 covers the determination of the optimum number of iterations needed to get the optimum return on investment through the proposed network. The return on investment in the share market that is obtained through the neural network and compared with the actual values are discussed in section 4. The merits of the present problem using neural networks are discussed in section 5 and finally the conclusion is given in section 4.

Section2.ARTIFICIAL NEURAL NETWORK

Before the age of computers, people traded stocks and commodities primarily on intuition. As the level of investing and trading grew, people searched for tools and methods that would increase their gains while minimizing their risk. Statistics, technical analysis, fundamental analysis, Time series analysis, chaos theory and linear regression are all used to attempt to predict and benefit from the market's direction. None of these techniques has been proved to be the consistently correct prediction tool that is desired, and many analysts argue about the usefulness of many of the approaches. However, these methods are presented as they are commonly used in practice and represent a base-level standard for which neural networks should outperform. Also, many of these techniques are used to preprocess raw data inputs, and their results are fed into neural networks as input. Some of the related work are given below.

A neural network is a computer program that recognizes patterns and is designed to take a pattern of data and generalize from it. An essential feature of this technology is that it improves its performance on a particular task by gradually learning a mapping between inputs and outputs. There are no set rules or sequence of steps to follow in generalizing patterns of data. The network is designed to learn a nonlinear mapping between the input and output data. Generalization is used to predict the possible outcome for a particular task. This process involves two phases known as the training phase (learning) and the testing phase (prediction).

Regression models have been traditionally used to model the changes in the stock markets. Multiple regression analysis is the process of finding the least squares prediction equation, testing the adequacy of the model, and conducting tests about estimating the values of the model parameters, Mendenhall et al. [19]. However, these models can predict linear patterns only. The stock market returns change in a nonlinear pattern such that neural networks are more appropriate to model these changes.

Studies have shown that back propagation networks may be used for prediction in financial market analysis. Refenes et al. [20] compared regression models with a back propagation network both using the same stock data. In comparison with regression models back propagation proved to be a better predictor. The results showed that the Mean Squared Error (MSE) for the neural network was lower than the Multiple Linear Regression (MLR) model. The MSE for the network was 0.044 and the MSE for the MLR model was 0.138 such that the neural net proved to be more effective in learning the training data than the MLR. For the test data, which was different from the training data, the neural network MSE was 0.066 which is also lower than the MLR MSE of 0.128. According to Refenes et al. [20] "neural networks are capable of making better prediction in capturing the structural relationship between a stock's performance and its determinant factors more accurately than MLR models." Kryzanowski et al. [21] using Boltzmann machine trained an artificial neural network with 149 test cases of positive (rise in the stock price) and negative (fall in the stock price) returns for the years 1987-1989 and compared this to training the network with positive, neutral (unchanged stock price), and negative returns for the same 149 test cases for the years 1987-1989. The network predicted 72% correct results with positive and negative returns. However the network predicted only 46% correct results with positive, neutral, and negative returns.

If stock market return fluctuations are affected by their recent historic behavior, Tang [22] neural networks which can model such temporal information along with spatial information in the stock market changes can prove to be better predictors. The changes in a stock market can then be learned better using networks which employ a feedback mechanism to cause sequence learning.

Recurrent networks use the back propagation learning methodology. The main difference between a feed forward back propagation network and a recurrent network is the existence of a feedback mechanism in the nodes of the recurrent network. This feedback mechanism facilitates the process of using the information from the previous pattern along with the present inputs. Copy-back/Context units are used to integrate the previous pattern into the following or a later input pattern, Morgan et al. [23]. This ability of recurrent networks in learning spatiotemporal patterns makes them suitable for the stock market return prediction problem.

Back propagation networks are independent of the sequence in which the inputs are presented whereas the recurrent networks take into account the sequence. Thus the recurrent networks represent the idea of predicting stock market returns on the basis of recent history more closely [24-27]. Since no training occurs during testing, a pattern is matched with its closest learned training pattern (independently) and the corresponding output is generated. Hence, if there was no training after week 48 and we test the network for week 59, it will be matched with the learned data set for weeks l-48 and maybe week 37 will be used to predict the output for week 59 --intrinsically assuming that week 36 before week 37 is a good representative of week 58 preceding week 59. Although, this is ideally what we hope to occur, there is no guarantee that the neural networks will use this relationship. It may use other learned information based on the data.

The prediction accuracy of a network along with additional information available from recent history of a stock market can be used to make effective stock market portfolio recommendations [28].

2.1 NN concepts and its terminology

To model complex process in many physical systems, the use of Artificial Neural Network (ANN) has been extensively in use in recent times. As a branch of artificial intelligence, this robust and versatile tool is being modeled after the human neurological system, consisting of a series of neurons (the basic computing elements), interconnected together to allow recognition of incidents that have had a similar pattern to the current input. Especially for pattern recognition and function approximation, ANN, equipped with parallel distributed processing architecture, is well recognized as a very powerful computational tool, having the ability to learn and to generalize from examples to produce meaningful solutions to problems even in case of erroneous or incomplete data.

Neural networks have widely been used in share market prediction and forecasting of the various share price predictions, as well as for time series modeling. Most often feed-forward networks, which employ a sliding window over a sequence of data (i.e., to induce the function in ANN architecture, using a set of older records as inputs and a single output as the target value of the network), are used for time series modeling. Although, in general, non-linear, auto-regressive time series modeling is difficult than linear models, yet with the ANN approach such a restriction does not apply. Similarly, in contrast to the auto-regressive and moving average methods, ANNs are nonparametric data driven approaches that can capture nonlinear data structures without prior assumption about the underlying relationship in a particular problem. Besides, ANNs are more general and flexible modeling and analysis tools for forecasting applications, capable of finding nonlinear structures, as well as linear ones. In fact, linear autoregressive (AR) models are special cases of ANNS without hidden nodes.

For an explanatory or casual forecasting problem, the inputs to an ANN are usually the independent or predictor variables [11]. The functional relationship estimated by the ANN can be written as:

$$Y = F(x_1, x_2, x_3, \ldots x_n) \tag{1}$$

Where $x_1, x_2, x_3, \ldots x_n$ are n independent variables and y is a dependent variable. In this sense, the neural network is functionally equivalent to a nonlinear regression model. For an extrapolative or time series problem, on the other hand, inputs are typically the past observations of the series and the output is the future value. The function mapping performed by the ANN is as follows:

$$Y_{t+1} = F(y_t, y_{t-1}, y_{t-2}, \ldots, y_{t-n}) \tag{2}$$

Where y_t is the observation at time t. Thus the ANN is equivalent to the nonlinear autoregressive model along the series. For a time series problem, a training pattern consists of a fixed number of lagged observations of the series. There are N observations y1, $y_2, y_3, \ldots, y_N$

in the training set and if one-step-ahead forecasting is required, then using an ANN with n input nodes, we have N-n training patterns. The first training pattern will be composed of $y_1, y_2, y_3, \ldots, y_n$ as inputs and Y_{n+1} as the target output. The second training pattern will contain $y_1, y_2, y_3, \ldots, y_{n+1}$ as inputs and Y_{n+2} as the desired output. Finally, the last training pattern will be $y_{N-n}, y_{N-n+1}, \ldots, y_{N-1}$ for inputs and y_N for the target output. Typically, least- squares based objective function or cost function to be minimized during the training process is:

$$E=\frac{1}{2}\sum_{i=n+1}^{N}(y_i-a_i)^2 \qquad (3)$$

where a_i is the output of the network and ½ is included to simplify the expression of derivatives computed in the training algorithm.

Most of the environmental and water resources applications of ANN have used feed-forward networks for function approximation. While a majority of them used the back-propagation training algorithm, a few of them attempted other algorithms. The general structure of a feed-forward neural network is shown in Fig.1. The nodes in an input layer receive the inputs of the model and they flow through the network and produce outputs at nodes in the output layer. The working principle of feed-forward neural network is available elsewhere [12]. Mathematically, a three-layer neural network with *I* input nodes, *J* hidden nodes in a single hidden layer, and *K* output nodes, can be expressed as:

$$O_{pk}=f_1\left(\sum_{j=1}^{J}w_{jk}^{o}\,f_2\left(\sum_{i=1}^{I}w_{ij}^{h}x_{pi}+b_1^{j}\right)+b_2^{k}\right), \quad \forall\, k\in 1,2,\ldots,K \qquad (4)$$

where O_{Pk} is the output from the kth node of the output layer of the network for the Pth vector (data point); X_{Pi} is the input at the ith node of input layer from pth vector (data point); w_{jk}^{o} is the connection weight *between jth* node of the hidden layer and kth node of the output layer (Fig. 1); w_{ij}^{h} is the connection weight between ith node of the input layer and jth node of the hidden layer; and b_1^{j} *and* b_2^{k} are bias terms; and $f_1(.)$ and $f_2(.)$ are activation functions. The logistic sigmoid function, a commonly used activation function has the form of:

$$f(x)=\frac{1}{1+e^{-x}} \qquad (5)$$

The linear activation function has the form

$$f(x)=x \qquad (6)$$

In this study sigmoid function is used for f_2 and the linear function is applied for f_1. The sigmoid functions (which plots like curves) normally have a tendency to push the value *of f(x)* to the extremes (binary in the case of logistic sigmoid; bipolar in the case of tanh function). Thus the sigmoid functions are more suitable for classification problems. When continuous outputs are expected, as in the case of time series modeling, sigmoid functions are not a good choice. There are several other activation functions used in many other studies, however, this work did not analyze the suitability of activation functions for share market price predictions.

Linear autoregressive models assume the prediction equation to be a linear combination of a fixed number of previous data in the time series. Including a noise term, it can be written as

$$x(t)=\sum_{i=1}^{p}\alpha_i x(t-i)+\xi(t)$$
$$= F^{L}(x(t-1),x(t-2),\ldots,x(t-p))+\xi(t)$$

`

(7)

If p previous sequence elements are taken, one speaks of an AR(p) model of time series. Finding an appropriate AR(p) model means choosing an appropriate and estimating the coefficients α_i through techniques like least squares optimization procedures. This techniques, although rather powerful, is naturally limited, since it assumes a linear relationship among sequence elements. It becomes clear that a feed forward neural network can replace the linear function F^L in equation (7) by an arbitrary non-linear function F^{NN} as in equation (8).

$$x(t) = F^{NN}(x(t-1), x(t-2), \ldots, x(t-p)) + \xi(t)$$

(8)

This non-linear function can be estimated based on samples from the series, using one of the well-known learning or optimization techniques like back propagation or conjugate gradient. Making F^{NN} dependent on p previous elements is identical to using p input units being fed with p adjacent sequence elements. This input is usually referred to as a time window, since it provides a limited view on part of the series. Non-linear autoregressive models are potentially more powerful than linear ones in that they can model much more complex underlying characteristics of the series.

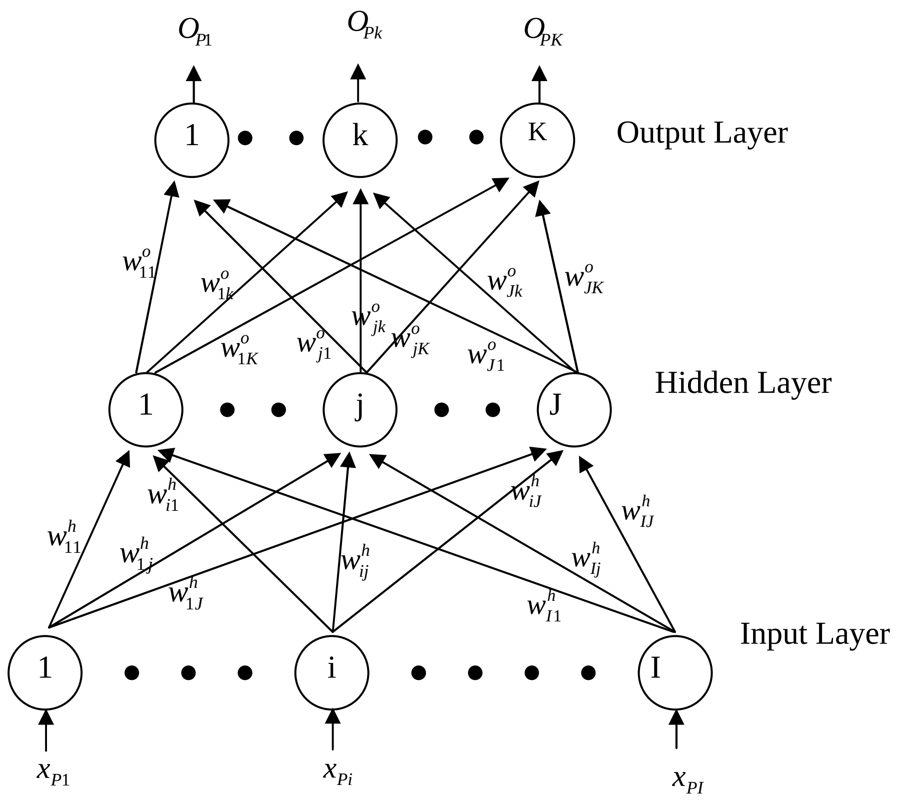

Fig.1 Three-layer feed-forward neural network architecture

2.2 Selection of Activation Function

Suitable activation function for the Hidden Units is needed to introduce non-linearity into the network, which gives the power to capture nonlinear relationship between input and output.

Three commonly used activated functions are logistic, linear, tanh. Since the expected output is a continuous variable (not a classification problem with unbounded function), linear activation function (g(x)) is used (in-stead of logistic sigmoidal or. tanh functions mostly used for classification problems and for bipolar output ranges, i.e., between -1 and +1). The form of the linear activation function is as below:

$$g(x) = x$$
$$g'(x) = 1$$

3. DETERMINATION OF OPTIMUM NUMBER OF ITERATION

Error back propagation with momentum can be viewed as gradient descent with smoothing. To stabilize the weight trajectory by making the weight change a combination of the gradient-decreasing term plus a fraction of the previous weight change, a specific momentum parameter is selected in any ANN architecture. The rate of learning by the network is computed by a factor called learning rate. In standard back-propagation, too low a learning rate makes the network learn very slowly and too high a learning rate makes the weights and objective function diverge, leading to no learning at all. In the present case, since the training is continues (non-batch type), the training rate can be maintained constant throughout the training.

The effect of number of iteration on errors (at different learning rates and momentum parameters) for varying number of hidden nodes were studied and presented in Tables 1 (A to D) for each sub sectors like BSEIT, BSECD and so on.

Table 1: The effect of learning rate, momentum parameter, number of hidden nodes and iteration on errors

A:

Learning Rate(LR)/ Momentum Parameter(MP)	Learning Rate(LR) 0.01									
	Hidden Nodes 1		Hidden Nodes 10		Hidden Nodes 20		Hidden Nodes 30		Hidden Nodes 40	
	Iteration	Error	Iteration	Error	Iteration	Error	Iteration	Error	iteration	Error
Momentum Parameter 0.01	50000	0.009	15000	0.009	9000	0.009	10000	0.009	10000	0.009
Momentum Parameter 0.05	40000	0.009	15000	0.009	8000	0.009	10000	0.009	10000	0.009
Momentum Parameter 0.1	40000	0.008	15000	0.009	7500	0.009	10000	0.009	10000	0.009
Momentum Parameter 0.5	15000	0.008	15000	0.008	7000	0.008	8000	0.008	10000	0.008
Momentum Parameter 0.9	1000	0.0554	6000	0.0554	6000	0.0554	4000	0.009	3000	0.008

B:

Learning Rate(LR)/ Momentum Parameter(MP)	Learning Rate(LR) 0.05									
	Hidden Nodes 1		Hidden Nodes 10		Hidden Nodes 20		Hidden Nodes 30		Hidden Nodes 40	
	Iteration	Error	Iteration	Error	Iteration	Error	Iteration	Error	iteration	Error
Momentum Parameter 0.01	50000	0.009	20000	0.009	40000	0.009	70000	0.009	90000	0.008
Momentum Parameter 0.05	75000	0.008	20000	0.009	30000	0.009	70000	0.008	85000	0.009
Momentum Parameter 0.1	90000	0.009	20000	0.008	25000	0.009	60000	0.008	80000	0.008
Momentum Parameter 0.5	1000	0.0054	15000	0.008	8000	0.008	10000	0.008	50000	0.009
Momentum Parameter 0.9	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	5000	0.005

C.

Learning Rate(LR)/ Momentum Parameter(MP)	Learning Rate(LR) 0.1									
	Hidden Nodes 1		Hidden Nodes 10		Hidden Nodes 20		Hidden Nodes 30		Hidden Nodes 40	
	Iteration	Error	Iteration	Error	Iteration	Error	Iteration	Error	iteration	Error
Momentum Parameter 0.01	1000	0.0554	15000	0.009	50000	0.009	25000	0.008	25000	0.006
Momentum Parameter 0.05	65000	0.013	15000	0.008	50000	0.007	20000	0.006	25000	0.007
Momentum Parameter 0.1	65000	0.009	30000	0.005	50000	0.005	20000	0.007	25000	0.006
Momentum Parameter 0.5	1000	0.0554	30000	0.005	50000	0.005	20000	0.005	25000	0.007
Momentum Parameter 0.9	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554

D.

Learning Rate(LR)/ Momentum Parameter(MP)	Learning Rate(LR) 0.7									
	Hidden Nodes 1		Hidden Nodes 10		Hidden Nodes 20		Hidden Nodes 30		Hidden Nodes 40	
	Iteration	Error	Iteration	Error	Iteration	Error	Iteration	Error	Iteration	Error
Momentum Parameter 0.01	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554
Momentum Parameter 0.05	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554
Momentum Parameter 0.1	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554
Momentum Parameter 0.5	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554
Momentum Parameter 0.9	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554	1000	0.0554

Based on the analyses in the tables (1A to 1D) it was found that the minimum number of iteration for different number of nodes should be 1000 for minimum errors. The learning rate is more than 0.7 with respect to higher momentum parameters are diverges. Hence, if we assign the value of the learning parameters is less than 0.7 at any momentum values then the minimum error is obtained through neural networks. And also we predict that at what level each sub sectors to penetrate and influences to increase or decrease of the share market index. The following bar diagram shows the percentage of influences of each sub sector to increase / decrease the share market index.

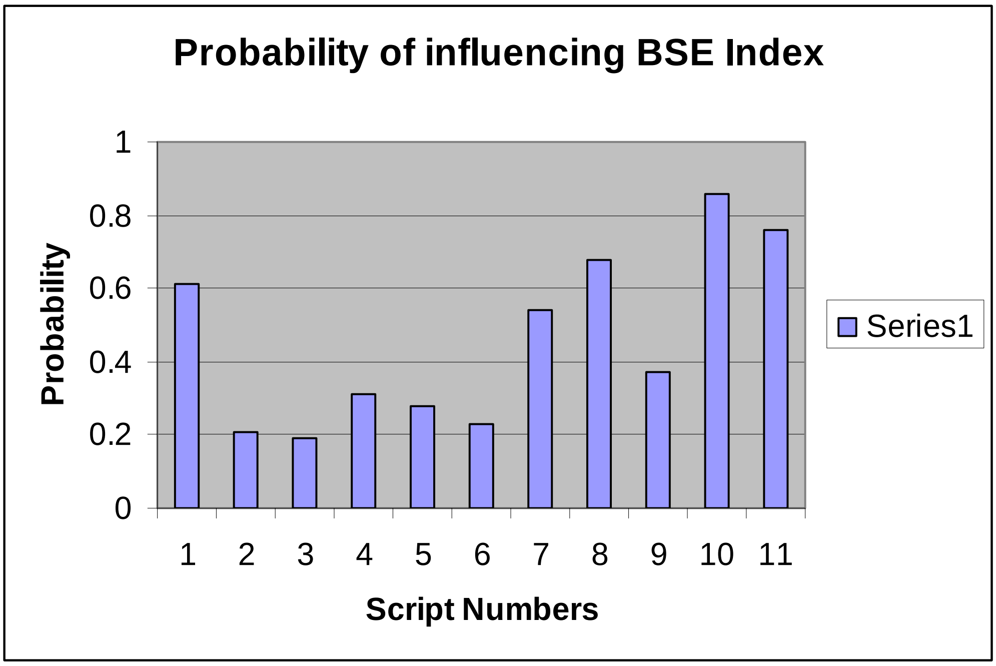

Figure – 2 represents the bar diagram for influencing scripts towards the Increment / Decrement of BSE Index.

4. ANN MODELING OUTPUT

The ANN model (developed based on the training data) with 26 hidden nodes (mono-layer) was found to show the least error, when compared with the testing data, thereby resulting in maximum capture of the actual trend observed in the field with respect to the Indian BSE index. Fig. 3 shows the predictability of the ANN model in predicting the share market price and Fig. 4 the correlation between actual BSE and predicated BSE obtained by the model (showing a significantly high correlation coefficient of about 0.989).

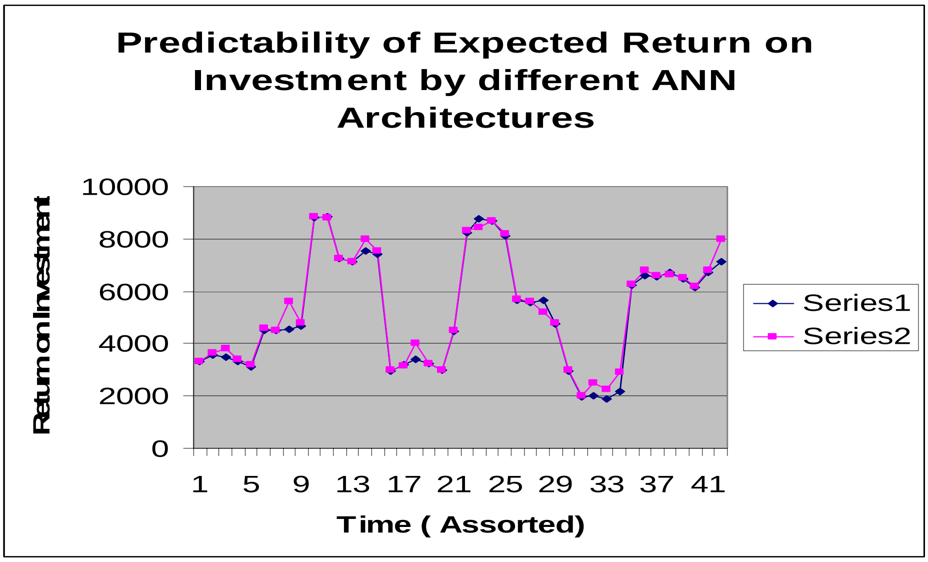

Figure – 3 : Series 1 - Actual and Series 2 - Expected

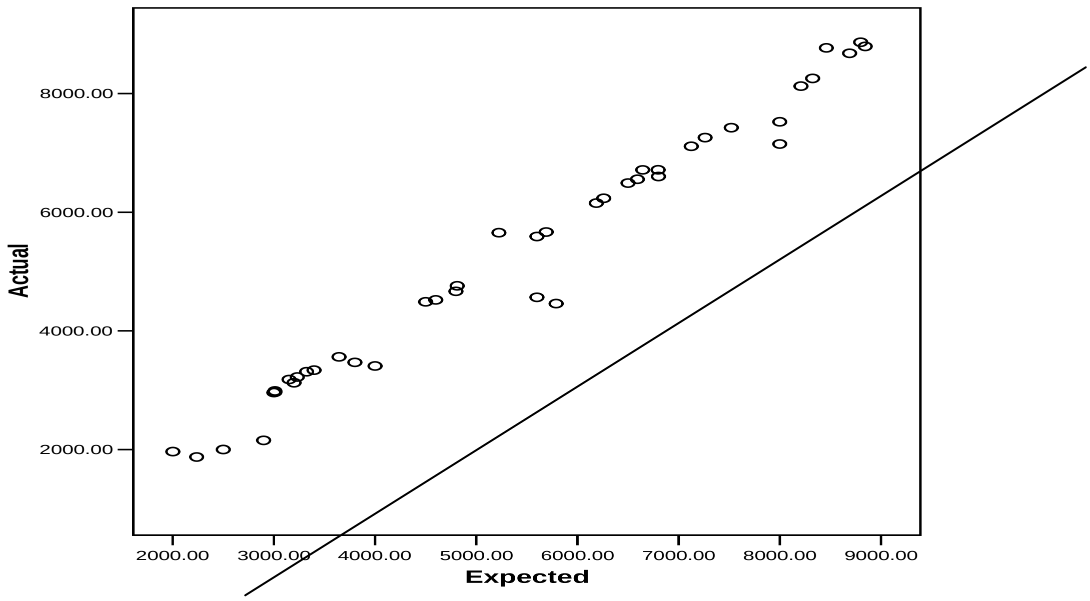

Figure – 4, represents the Correlation between the actual and expected Return On Investment

5. BENEFITS

Most of the benefits in the articles depend on the problem domain and the NN methodology used. A common contribution of NN applications is their ability to deal with uncertain and robust data. Therefore, NN can be efficiently used in stock markets, to predict the expected return on investment. It can be seen from a comparative analysis that the Back propagation algorithm has the ability to predict with greater accuracy than other NN algorithms, no matter which data model was used. The variety of data models that exist in the papers could also be considered a benefit, since it shows NNs flexibility and efficiency in situations when certain data are not available. It has been proven that NN outperform classical forecasting and statistical methods, such as multiple regression analysis [10] and discriminant analysis. When joined together, several NNs are able to predict values very accurately, because they can concentrate on different characteristics of data sets important for calculating the output. Analysis also shows the great possibilities of NN methodology in various combinations with other methods, such as expert systems. The combination of the NN calculating ability based on heuristics and the ability of expert systems to process the rules for making a decision and to explain the results can be a very effective intelligent support in various problem domains [12].

6 CONCLUSIONS

The studies reveal a high potential of ANN in predicting the return on investment in the share market. Already, we know the evaluation of the return on investment in the share market through any one of the traditional techniques [11, 12, 13, 14, 15] (mostly statistical methods like time series analysis, moving averages etc.,) is tedious, expensive and a time-taking process. Again, the return on investment in share market is always uncertain and ambiguity in nature, so that no traditional techniques will give the accurate or appropriate solution. Hence, a non-traditional model would be of immense help for estimating the prediction on the return on investment accurately and this method gives better solution always. This method of predicting return on investment will help further to investigate that the model can be extended to ANFIS (Artificial Neural Fuzzy Inference System), which is based on the linguistic rules that the fuzzy system could design a rule and these rules will be further refined with the help of neural networks.

REFERENCE

[1] Refenes, A.N., Zapranis, A., Francis, G., *Stock Performance Modeling Using Neural Networks: A Comparative Study with Regression Models*, Neural Networks, vol. 7, No. 2, 1994, pp. 375-388.

[2] Schoeneburg, E., *Stock Price Prediction Using Neural Networks: A Project Report*, Neurocomputing, vol. 2, 1990, pp. 17-27.

[3] Yoon, Y., Guimaraes, T., Swales, G., *Integrating Artificial Neural Networks With Rule- Based Expert Systems*, Decision Support Systems, vol. 11, 1994, pp. 497-507.

[4] Murphy J.J. *Technical Analysis of Financial Markets – A Comprehensive Guide to Trading Methods and Applications*, New York Institute of Finance, 1999.

[5] Han J., Lu H., Feng L. "Stock Movement Prediction and N dimensional Inter-Transaction Association Rules". *Proc. of 1998 SIGMOD'96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'98)*, Seattle, Washington, June 1998, pp. 12:1-12:7. [6] John G.H., Miller P. "Stock Selection using RECON". *Neural Networks in Financial Engineering*, 1996, pp. 303- 316.

[6]. Ravichandran K S and et al., " Study on Share Market Situation (Investment) : A fuzzy Approach", Published in the China Journal of Finance, Vol.3, No.2, 2005.

[7] John G.H., Miller P. "Stock Selection Using Rule Induction". *IEEE Expert*, October 1996, pp. 52-58.

[8] Yoon Y., Swales G. "Predicting Stock Price Performance: A Neural Network Approach". *Proceedings of the IEEE 24th Annual Conference on System Science*, 1991, pp.156-162.

[9] Choi J., Lee M., Rhee M. "Trading S&P 500 Stock Index Futures using a Neural Network". *Proceedings of the 3rd Annual Conference on AI Applications on WS*, 1995, pp. 63-71.

[10] Gately E. *Neural Networks for Financial Forecasting*, New York: Wiley, 1996.

[11] Drossu R., Obradovic Z. "Rapid Design of Neural Networks for Time Series Prediction". *EEE Computational Science & Engineering*, 1996, pp. 78- 89.

[12]. Garson, G.D. (1991), "Interpreting neural network connection weights," AI Expert, April 1991, 47-51.

[13] Hansen J.V., Nelson R.D. "Neural Networks and Traditional Time Series Methods: A synergistic Combination in State Economic Forecast". *IEEE Transactions on Neural Networks*, Vol. 8, No. 4, July 1997.

[14] Austin M., Looney C., Zhuo J. "Security Market Timing using Neural Network Models". The New Review of Applied Expert Systems, Vol. 3, 1997, pp. 3-14.

[15] Potts M.A.S., Brromhead D.S. "Time Series Prediction with a Radial Basis Function Neural Network. Adaptive Signal Processing', *SPIE Proceedings* Vol. 1565, 1991, pp. 255-266.

[16] Trippi, R.R., DeSieno, D., *Trading Equity Index Futures With a Neural Network*, The Journal of Portfolio Management, Fall 1992, pp. 27-33.

[17] Kryzanowski, L., Galler, M., Wright, D.W., *Using Artificial Networks to Pick Stocks*, Financial Analyst s Journal, August 1993, pp. 21-27.

[18] Li, E.Y., *Artificial Neural Networks and Their Business Applications*, Information & Management, vol. 27, 1994, pp. 303-313.
[l] Mender&all and Beaver, Introduction to Probabilitv And Statistics, Ninth Edition, International Thomson Publishing, 1994.

[19] Refenes, Zapranis, and Francis, Journal of Neural Networks, Stock Performance Modeling Using Neural Networks: A Comparative Study With Regression Models, Vol. 7, No. 2, 1994. pp. 375-388.

[20] Kryzanowski, Galler and Wright, Financial Analysts Journal, Using Artificial Neural Networks to Pick Stocks, July-August 1993. pp.21-27.

[21] Tang, Almeida and Fishwick, Simulation, Time series forecasting using neural networks vs. Box-Jenkins methodology, November 1991, pp 303-310.

[22] Morgan and Scofield, Neural Networks and Speech Processing, Kluwer Academic Publishers, 1991.

[23] Pao, Adaptive Pattern Recognition and Neural Networks, Addison Wesley Publishing Company, Inc., 1989.

[24] Rumelhart, McClelland, and the PDP Research @OUP, Parallel Distributed Processing Volumel: Foundations, The Massachusetts Institute of Technology, 1988.

[25] Wicirow, Rumelhart, and L&r, Journal of Communications of the ACM, Neural Networks: Applications in Industry, Business and Science, Vol. 37, No. 3, 1994. pp. 93-105.

[26] Elman, TLEARN - simulator program for neural networks. Center for Research in J-aiwxe, C-008, 1990, University of California, San Diego, La Jolla, CA 92093-

[27] Jovina Roman and Akhtar Jameel, “Backpropagation and Recurrent Neural networks in Financial Analysis of MultipleStock Market Returns”, Proceedings of the 29th Annual Hawaii International Conference on System Sciences - 1996