CHAPTER – 4

STRATEGIES FOR INDIAN SHARE MARKET INVESTMENT THROUGH ANFIS
CHAPTER – 4
STRATEGIES FOR INDIAN SHARE MARKET INVESTMENT THROUGH ANFIS

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 have also been used in many financial trading systems for investing and predicting stock prices. In recent years, most of the researchers concentrate their research work towards the future prediction of share market prices by using Artificial Neural Networks (ANN). In this chapter we newly propose a methodology that neural network is applied to the investor’s financial decision making to invest in the restricted share (sub-sectors) in a continuous time frame work and further it is extended to establish the fuzzy design rule to design an optimum investment rule that one can earn more return on investment in a share market. Again, the FIS trained through ANFIS and the proposed ANFIS (Artificial Neural Fuzzy Inference System) has been tested with stock data obtained from the Indian Share Market Index BSE. In this chapter, the design, implementation and performance of the proposed neural network and ANFIS are described.
4.1 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. Various technical, fundamental, and statistical indicators have been proposed and used with varying results. However, no one technique or combinations of techniques have been successful enough to consistently “beat the market”. Traditionally, technical analysis approach specified in Maier, H.R. and Dandy, G.C (1996), 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 to expert systems [Stokelj, T, Paravan, D and R. Golob (2002), Garson, G.D.,(1991) and Milne, L.K.,(1995)] to neural networks [Gedeon, T.D,(1997), Kim, C.Y., Bae, G.J., Hong, S.W., Park, C.H., Moon, H.K. and Shin, H.S.(2001), D.E. Rumelhan, J.I. McClelland, el. al.(1986) and D.E. Rumelhart. B. Widrow and M.A. Lehr,(1994), Amilon (2001), Azoff (1994)] have also been used in many financial trading systems for stock prediction. Ultimately, in most of the researchers have been derived the various methodology for predicting future share market prices using Artificial Neural Networks. But in this chapter, the ANFIS is used to identify the proper section of sub-sectors of investment that one can earn the optimum return on investment.
Indian Share Market Index BSE is increased or decreased depending on the performance of the various sub indexes (sectors) namely BSEIT, BSECD, BSEFMC, BSEHC, BSEC, TECK, BSEPSU, BANKEX, AUTO, METAL and OILGAS. During data collection (Figure-4.1 represents the quarter average index value of last 42 slots), we have observed that the fluctuations and variations can happened rapidly for all sub-sectors from time to time. Therefore, the area of investment in the sub-sectors and its identification is very difficult task and ambiguity in nature of investors. The decision making of the identification of the optimum sub-sectors investment for one or more scripts are vital role. The combinations are some portion of the shares are more profit oriented but the risk is more and some of the shares are very less risk but the profit is some what good, but not that much of first one. In this chapter, we proposed a new methodology for optimizing share profit by the way of ANFIS. The proposed network has been tested with BSE data.

![Fig-4.1 BSE Index Vs Average of 42 quarter.](image)
The following ways the chapter was arranged. Section 4.2 provides the information related to neural network and its learning rule (back propagation) for predicting the minimum number of iterations required to sub-sectors investment in stock a market. Section 4.3 covers the introduction of fuzzy design rule and the evaluation procedure for ANFIS. This will help how ANFIS is useful for predicting the script (shares) selection of the sub-sectors investment. In both the sections details on how ANFIS has been designed to outperform current techniques. The conclusion is given in section 4.4.

4.2 Artificial Neural Network

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, Menderhall et al.
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. (1995, 1996, 1997) 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. (1995, 1996, 1997) “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. (1993) 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.
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.

4.3 DESIGN OF ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM

Applying a neuro-fuzzy inference system to model this real life problem is quite natural, because of the similarity to real life human decision-making. Some design issues were set up based on information from share market experts, while others were learned or experimented. The initial investment and portion of investment on various sub-sectors of share market were fuzzified for initial training. Later on we may use linguistic descriptions (low, medium, and high) from investing details for various sub-sectors of the share market. Some of the design issues were the following:

1. Number of input membership functions: The fuzzy membership functions were set up based on knowledge of investment of sub-sectors decisions. We determined that three membership functions (low, medium, high) for each of the six soft constraints model BSE index. Due to increased computational time, we take the average data for recent 42 months BSE index data collection.

2. Type of input membership functions: Based on the properties of the soft constraints we primarily consider triangular membership functions,
yet trapezoid, Gaussian curve and generalized bell-shaped membership functions were tested as well. The membership function of this type is given by

\[
\mu_a(x) = \begin{cases} 
0 & \text{for } x < a \\
\frac{x-a}{b-a} & \text{for } a \leq x \leq b \\
\frac{c-x}{c-b} & \text{for } b < x \leq c \\
0 & \text{for } x > c.
\end{cases}
\]

where \(a, b\) and \(c\) are the extreme points of the triangular model.

3. Type of output membership functions: We used a single output, obtained using weighted average defuzzification. All output membership functions had the same type and were either constant or linear (zeroth and first order Sugeno type system).

4. The number of output membership functions: It ranged from 2 to 243.

5. The number of rules: For a well defined fuzzy system we need to define control actions (fuzzy output) for every possible combination of input membership function values. If the training set doesn’t include examples for given combination of values, but the testing set does then in the testing phase we may still “guess” the output value using the aggregate gain(return on investment) function from all the learned fuzzy rules. This can be done before the actual testing phase begins resulting in a fully defined fuzzy system. In our case six constraints, each with three
membership functions result in 729 fuzzy rules, where the linguistic values were not negated, and they were connected with "and" relation. However, through rule extraction and combination of rules we drastically decrease the number of rules later.

6. Performance function: Some of the widely used where SSE is Sum of Squared Error, \( l \) is a confidence factor and \( w \) is weight. To verify training performance we can also verify the correct classification rate (R.Kozma et al. 1996).

7. Optimization methods: Back-propagation and hybrid (mixed least squares and back propagation) methods had been used as optimization methods.

8. Partitioning the data into training, cross-validation and testing sets: The range of the training data set size was 50% to 90%. The cross-validation and testing data sets each took half of the rest of the data (5%-25%). The use of cross validation is optional but in our implementation is important, to avoid over investment.

9. Number of epochs: In most runs it was set up to 6000. Through an adaptive neuro-fuzzy inference system the range of the membership functions are learned, fuzzy rules are created and their weights are adjusted in order to better model the training data. The performance function values are calculated, and classification is provided.
4.4 RESULTS AND RULE EXTRACTION

For implementation, we used Matlab 7.0 environment with fuzzy logic toolbox. Various types of input membership functions have been tried, and as we expected the triangular membership functions performed best closely followed by trapezoid, Gaussian curve and generalized bell-shaped membership functions. Linear output membership functions performed better than constant ones.

For optimization method, the back-propagation and hybrid optimization method performed similarly regarding the performance function and classification, but the hybrid method running time was about 5 times as long as that of the back-propagation. To avoid over-investment we used cross-validation. In some cases the minimum cross-validation error occurred within the first epoch. This meant that the given set of membership functions were not a good choice for modeling the training data. This also indicated that either more data need to be selected for training, or we need to modify the membership functions (both the number of membership functions and their types). After training, the FIS parameters were set to be associated with the minimum cross-validation error.
Layer 2. Input membership functions: Three triangular membership functions for each input neuron may be established and each node of layer 2 is connected with all inner hidden nodes. The number of hidden layers may be generated dynamically while using Matlab 7.0. The output of layer 2 is getting by applying fuzzy BSE index value to the triangular membership function and this membership function will give the output of layer 2.

Layer 3. Fuzzy rule left hand sides: Each connected to 4 input membership functions.

Layer 4. Output membership functions (right hand sides): The right hand side rules are in one to one relation with the left hand side rules.

Layer 5. Aggregated output: Each output membership function gets aggregated along with the maximum weight they carry.

Layer 6. Output (decision): The input and output membership functions and its relations are giving below:
Figure 4.3 : ANFIS model structure.

Figure 4.3 depicts the 6- layered architecture of single output ANFIS and the functionality of each layer is as follows:

**Layer 1.** Input neurons: 11 input neurons, namely BSEIT, BSECD, BSEFMCG, BSEHC, BSECG, TECK, BSEPSU, BANKEX, AUTO, METAL and OILGAS and each of them have three soft constraints (low, medium and high). The fuzzified BSE index value will be the input and output of this layer 1. The output of this layer 1 will be the input of layer 2. For reducing the time complexity, we select six sub-indexes with high return on investment with low risk by using K.S.Ravichandran and et al. (2006 a).
Layer 2. Input membership functions: Three triangular membership functions for each input neuron may be established and each node of layer 2 is connected with all inner hidden nodes. The number of hidden layers may be generated dynamically while using Matlab 7.0. The output of layer 2 is getting by applying fuzzy BSE index value to the triangular membership function and this membership function will give the output of layer 2.

Layer 3. Fuzzy rule left hand sides: Each connected to 4 input membership functions.

Layer 4. Output membership functions (right hand sides): The right hand side rules are in one to one relation with the left hand side rules.

Layer 5. Aggregated output: Each output membership function gets aggregated along with the maximum weight they carry.

Layer 6. Output (decision): The input and output membership functions and its relations are giving below:
Layer 2. Input membership functions: Three triangular membership functions for each input neuron may be established and each node of layer 2 is connected with all inner hidden nodes. The number of hidden layers may be generated dynamically while using Matlab 7.0. The output of layer 2 is getting by applying fuzzy BSE index value to the triangular membership function and this membership function will give the output of layer 2.

Layer 3. Fuzzy rule left hand sides: Each connected to 4 input membership functions.

Layer 4. Output membership functions (right hand sides): The right hand side rules are in one to one relation with the left hand side rules.

Layer 5. Aggregated output: Each output membership function gets aggregated along with the maximum weight they carry.

Layer 6. Output (decision): The input and output membership functions and its relations are giving below:
The surface diagram for the above is given below.

Figure – 4.4: Surface diagram
The following results have been obtained by the training implemented through hybrid method and during its training period the following nodes and parameters have been automatically created except the input and output nodes and its initial BSE index values. The number of nodes are 1503, number of linear parameters are 729, number of non-linear parameters are 54, number of linear parameters are 783, number of training data pairs 12 and the number of fuzzy design rules generated are 729. For three triangular input membership functions, 500 epochs, linear output membership functions, the Sum Squared Error between the actual and desired outputs was 0.177.

Based on 729 fuzzy design rules and its aggregated function value, the optimized design rule has been established (the percentage of amount per 100 rupees and its scripts given below). While making a fuzzy membership function for the various fuzzy sub-indexes the risk factors are also considered. Ranking of the various sub-indexes are arranged for taking both the return on investment and risk factor using K.S.Ravichandran et al. (2005a).

A typical running time on a 3.3 GHz Pentium IV processor was about 96 minutes for 6000 epochs when back-propagation optimization method was used with three triangular membership functions for each input. Other methods took considerably longer time, particularly the hybrid optimization method.
Therefore, the optimum priority rule out of 729, if (METAL is medium) and (BSEPSU is medium) and (BANKEX is high) and (BSECG is high) and (OILGAS is low) and (BSEIT is high) then the output is maximum according to Sugeno aggregation function. The above result is obtained by taking care of high return on investment and less risk. For every 100 rupees investment, 58 rupees should be evenly invested in high profile shares, 33 rupees should be evenly invested in medium profile shares and 9 rupees should be evenly invested in low profile shares. Further enhancement to this work, even among the high profile shares, we can able to investigate the percentage of amount invested.

From 11 sub-sectors of indexes, if we use ranking of the above 11 sub-sectors of indexes with minimum risk (K.S.Ravichandran et al. (2006a)) then we select 6 among them. If we use ANFIS to the above six indexes then we get 729 rules. If we use all the 11 indexes then we get \(3^{11} = 177147\) rules and the optimum rule will be selected among \(3^{11}\) rules is very difficult. Therefore, we need a help of non-traditional techniques called genetic algorithms or the association rules of data mining (R.Agrawal et al. (1996)) and then rules extraction can be obtained by any one of the above said methods.
4.5 Conclusion

The study reveals that a high potential of adaptive neuro-fuzzy inference system predicting the Indian share market investment BSE. Results have been shown that the human-kind decisions can be achieved with the model. The Sugeno FIS rules are extracted from direct verification method and the maximum output was tested. In this work, one can easily select an optimum rule that will give the maximum return on investment with low risk among the selected 729 rules. For further work, if we use either genetic algorithm or association rules in data mining that will help a lot to identify the script selection among the high, medium and low investment group but it will take a lot of complexity including a very large computational time.