International Journal of Computer Applications (0975 – 8887)
Volume 162 – No 2, March 2017

Inter-comparison of Artificial Neural Network Algorithms for Time Series Forecasting: Predicting Indian Financial Markets

Shilpa Amit Verma
Thadomal Shahani Engineering College
Mumbai-50, India.

G. T. Thampi
Thadomal Shahani Engineering College
Mumbai-50, India.

Madhuri Rao
Thadomal Shahani Engineering College
Mumbai-50, India

ABSTRACT
The financial market prediction is a specialized case of a time series analysis. Although many different methods have been employed by researchers for time series/financial data studies which include statistical techniques like ANOVA (analysis of variances), ARIMA (integrated moving averages), smoothing methods, correlation analysis etc., use of Artificial Neural Network (ANN) methods for financial prediction have become quite popular in the recent times. Though ANN methods like the conventionally used backpropagation method or the recurrent methods have been employed in the past, a complete and detailed investigation of more robust and popular ANN methods incorporated into ANN like the Resilient backprop, Marquardt Lavenberg, Conjugate gradient methods, One Step Secant, Quasi Newton methods, Bayesian learning etc., is missing from the literature. In the present study, a detailed study was undertaken to investigate the potential of these robust methods in the ANN domain for the Indian financial market prediction, more specifically the prediction of the share price of two popular scripts that are traded in the Indian secondary market. In this study, the 1 month ahead opening share price of two scripts namely ICICI bank and L&T have been forecasted. The results of our study indicate that while as for L&T data, Marquardt Lavenberg algorithm is able to give ~85% accurate prediction, it gives ~92% accurate prediction for ICICI bank data. This study therefore attempts to conduct a detailed investigation of many popular methods under the ANN domain to converge on the best possible results instead of just considering one or two methods and comparing them to the backpropagation method- a method being followed conventionally.

General Terms
Artificial Neural Network methods.

Keywords
Artificial Neural Networks, backpropagation, financial data analysis.

1. INTRODUCTION
The share price of a particular company mainly depends on its financial heath and perception of the shareholders and traders. Prediction of the financial market is a complex task, as the parameters on which it depends are not known beforehand. The problem becomes more challenging because this dependence keeps changing with time and thus one cannot predict the real reasons behind its volatility. Although the financial markets are sometimes emotionally driven, in general the parameters which have traditionally been important for financial market analysis include the global economic volatility, local political and economic factors. These factors provide a major challenge, especially in predicting the upcoming economies like India. Moreover what makes financial prediction an extremely challenging task is the presence of uncontrolled parameters i.e., unexpected or sudden changes that influence the market indices. Governed by this necessity, the researchers around the world have started to employ machine learning techniques like the Artificial Intelligence (AI) methods with renewed interest to extract trends/patterns in the financial markets. This has now become somewhat simpler mainly due to advent of high speed and affordable computational power which has become available due to faster computation abilities of present day computers and more importantly at an affordable cost compared to what it was about a decade ago. More so, the application of machine learning paradigms helps in better optimization, reduced cost and more reliable models for trading and doing business. In comparison with the conventional statistical tools like ARIMA, ANOVA etc, the advantage of neural network lies in the fact that it has the ability to model the complex and non linear tasks without any knowledge about the nature of the processes from where these tasks have been generated [1].

AI methods can broadly be classified into Fuzzy logic (FL), Genetic algorithms (GA) and Artificial Neural Networks (ANN). Owing to their characteristics of being extremely powerful in extracting trends and patterns in unknown environments, ANN's have become a preferred tool for prediction of the financial markets compared to FL and GA. ANN's are generally referred to as universal approximator (i.e., it can approximate any function) and a trained ANN is considered an expert in the domain in which it has been used [2]. However within the ANN domain there are a number of algorithms and their sub combinations which cannot be used off the shelf. Therefore prediction of the best suited method for the problem in hand is an important task [3]. It is therefore imperative to use as many combinations for comparison and based on this comparison, the selection of the best algorithm for the problem at hand can be decided. ANN methods can further be classified into the local search methods, the global search methods and the hybrid methods. For the financial prediction, popular methods from the local search and the global search each have been considered in the present study.

2. SURVEY ON FINANCIAL DATA APPLICATIONS
Application of some ANN based methods on the financial data has been attempted elsewhere with some promising results. In the work done by Devadoss and Ligori [4], the closing prices of some selected stocks have been satisfactorily predicted using standard Backpropagation algorithm of ANN. Yetis et al [5], have employed a generalized feed forward

based ANN model to predict NASDAQ's stock values. The results show a satisfactory performance for NASDAQ stock market prediction. In another study by G.Zhang [6] a reduced gradient based algorithm has been used for ANN training. The work done by Narendra Babu et al., [7] also investigates neural network-ARIMA models for financial forecasting. In the work done by J.Z.Wang et al., [8] indices forecasting has been done by employing a backpropagation neural network. In this work the monthly closing price of Shanghai composite index for about 16 year period has been done by employing a backpropagation neural network. The authors have also used a wavelet based method for de-noising the data. In the study done by A.S. Chen [9], the idea about how to predict share market price using feed forward ANN for the Taiwan stock market has been done with good results. In the work done by Mingyue Qiu & Yusone et al., [10], focus of the study employing the ANN techniques to predict the stock price of companies listed under the Japanese stock market. In this work the authors have used backpropagation neural network to predict the returns of the Neikki 225 index. Monthly data was used to make the prediction and results were compared with different models. In the work done by A.H. Moghaddam, [11] ANN models have been employed for forecasting the daily NASDAQ stock exchange rates. In this study the short term and days of the week stock prices are treated as inputs to the neural model and the NASDAQ index for a 6 month period Jan-June 2015 is predicted. The performances were compared for short term and long term forecasting value of the closing price. The study suggests that models perform better in long term forecasting as compared to short term forecasting of the share prices of the data considered by them.

Thus from the literature considered, the majority of the studies conducted have applied a backpropagation based ANN model for prediction of the financial markets. A holistic study of various models contained under ANN domain has been attempted in the present study.

3. BRIEF DESCRIPTION OF ANN METHODS CONSIDERED IN OUR STUDY

For the present study, seven different ANN methods namely the Feedforward Backpropagation algorithm, the Resilient backpropagation method, Conjugate gradient methods, Marquardt Lavenberg (ML) method, One step secant, Quasi Newton methods, Bayesian learning etc. along with different transfer functions like Sigmoid, TanH etc, have been considered. From the results obtained (discussed in next Section) the algorithms which show best prediction capability both in terms of the price index as well as forecasting trend in the data, will be presented in detail. The results of all other algorithms will be compiled separately in one section. A brief description of the four best methods is presented here, readers are however referred to [12] and [13] for a detailed study on these methods.

A multilevel perceptron model or more popularly known as the feedforward backpropagation algorithm is considered as the mother of all training algorithms that exist under the ANN umbrella. For this reason the standard backpropagation method was the first algorithm considered for this study. In the standard backpropagation model, proposed by Rumelhart [14], a set of inputs is applied from outside. These inputs, called input neurons, are problem dependent and user generally has no control on input data. These inputs are then multiplied by weights which are initially chosen as a set of random numbers and then gradually optimized. The product of the inputs above and the weights are then summed. This summation of products, is calculated for each neuron in the network. After this, an activation function is applied to modify it, thereby producing the output signal. Sigmoid activation function is usually applied as the transfer function between the layers.

The network output is subtracted from its corresponding target vector to produce an error. This error is used to adjust the weights of network, where the polarity and magnitude of the weight changes are determined by training algorithm. After enough repetitions of these steps, the error between actual outputs and target outputs should be reduced to an acceptable value, and the network is said to have been trained. At this point the network is used for recognition and the weights are not changed. It may be seen that first two steps constitute a "forward pass" in that the signal propagates from network input to the output. Last two steps which may be termed as the "reverse pass", the calculated error signal propagates backward through the network where it is used to adjust weights.

A major drawback of the backpropagation is the 'contra intuitive' influence of the partial derivative on the size of weight-step. If the error function is shallow, the derivative is small, resulting in a small weight step. On the other hand, large derivatives lead to large weight steps, leading to oscillations. The basic principle of the next method considered, called resilient backpropagation algorithm (RPROP) [15], is to eliminate the harmful influence of the size of partial derivative on the weight step. As a consequence, only the sign of derivative is considered to indicate the direction of weight update. The size of the weight change is exclusively determined by a weight specific, so called "update-value", and given as:

$$\Delta Wij = -\Delta ij(t) \quad if \ \frac{dE(t)}{dwij} > 0$$

$$= \Delta ij(t) \quad if \ \frac{dE(t)}{dwij} < 0$$

$$= 0 \quad otherwise$$

The second step of the RPROP learning is to determine the new update values Δ_{ij}. This is based on a sign dependant adaptation process, i,e

$$\Delta ij(t) = \eta^{+}\Delta ij(t-1) if \frac{dE(t-1)}{dwij}\frac{dE(t)}{dwij} > 0$$

$$= \eta^{-}\Delta ij(-1) \ if \ \frac{dE(t-1)}{dwij}\frac{dE(t)}{dwij} < 0$$

$$= 0 \quad otherwise$$

where $0< \eta^{-}< 1<\eta^{+}$.

The theoretical advantage of employing the RPROP method over the conventional backpropagation method is to expedite the learning/training of the latter method. The next algorithm considered was conjugate gradient method (CGM). The CGM method, developed by Moller [16], is actually a family of methods. There are nearly a dozen or more forms of conjugate gradient algorithms. The methods differ only in their treatment of undetermined systems, accuracy achieved for the problems in hand and their memory requirements etc. The difficulty of using the backpropagation/resilient method is that, a one dimensional minimization in say (A) followed by a minimization in direction (B) does not imply that the function is minimized on the subspace generated by (A) and (B). Minimization along direction (B) may in general spoil a previous minimization along direction (A). This is the main reason why a one dimensional minimization in general has to be repeated a number of times, which is sometimes even larger than the number of variables itself. If however the directions are non-interfering and linearly independent, at the end of N steps the process would converge to the minimum of the quadratic equation. The concept of non interfering directions is the basis of conjugate gradient method.

Backpropagation based methods work well in simple problems but it is too simplistic an approach for real world complex models like the financial data prediction since it can have many free parameters. Convergence therefore can take extremely long time, because of the nature of the gradient descent implementation employed in backpropagation method. For example, when descending the walls of a very steep local minimum bowl, a very small step size must be used to avoid `rattling out' of the bowl. On the other hand when moving along a gentle sloping part of error surface large steps should be taken, otherwise it will take forever to reach minimum. This problem is compounded by the manner in which the backpropagation is implemented. Here, generally a step is taken which is some constant times the negative gradient, rather than a step of constant length in the direction of negative gradient. This means that in steep regions (where large steps are not to be taken) the algorithm moves quickly and in shallow regions (larger steps) the algorithm moves slowly. The problem is also compounded by the fact that the curvature of the error surface may not always be the same in all directions. For example, if there is a long and narrow valley in the error surface the component of the gradient in the direction that points along base of valley is very small while the component perpendicular to the valley walls is quite large, even as, long distance along the base and a small distance perpendicular to the walls has to be moved. Thus one has to use slightly more sophisticated gradient descent algorithms than the simple backpropagation. Using the second order information, in other words using the curvature as well as the gradient of the error surface, can speed up the convergence enormously. This is the basis of Marquardt Lavenberg (LM) method [17] considered next. This is an extremely powerful method for complex problems but one has to be extremely careful about overfitting issues in this method. Being a powerful method, it can sometimes fit noise/outliers present in the training data.

4. FORMULATION OF TRAINING AND TEST DATA

For the present study, an attempt was made to forecast the one month ahead price of two prominent scripts of the Indian financial market namely Larsen and Toubro and ICICI Bank. Data for these scripts was collected from the NSE website [18]. A sufficiently large data window for ~16 years, i.e., from Jan 2000 to Dec 2015 has been considered for this study. The data consists of monthly data for the opening price, high price for the entire month and the monthly close value. About 70% of this data (chosen randomly to avoid any bias) was used for training and the remaining 30% of the data was employed for testing. Training and test data were selected randomly to avoid any bias in the training method. Transfer functions (TF) of the form sigmoid and TanH were considered. However keeping all other parameters same, negligible difference was observed in the prediction capability of the algorithms. Therefore Sigmoid TF was preferred for all algorithms because of its relative ease of implementation. In the section below, the forecasting capability of four algorithms which gave best results on our data namely the Backpropagation, Resilient Backprop method, Conjugate Gradient and the Levenberg-Marquadt (LM) algorithm, is presented. Results of the remaining 3 methods are compiled in the form a table at the end of the next section. For the methods considered MATLAB R 2013a version has been employed for our study.

5. RESULTS AND DISCUSSIONS

For the data under consideration which is in the form of two columns, giving the opening share price at the beginning of the month and the corresponding date, it has been compiled in form of a time series data of monthly opening values of the respective share price. Since the best arrangement for predicting the ahead price is not known beforehand, the data was arranged in 4 different ways, in an attempt to train the ANN model considered: a 12 input series- where the monthly opening share price value of the preceding 12 months, with effect from Jan 2000 is used to predict the monthly opening value for the 13th month; a 7 input series-wherein the month opening share price values for seven months is used to train the ANN model to predict the opening value of share price for the 8th month; a 5 input series, which as above trains on 5 consecutive months data and predicts the 6th month opening share price value and finally a 3 input series where data for 3 consecutive months is used for learning to predict the 4th month opening value. The reason for taking these 4 series, one by one, for training is that there is no way to know beforehand the functional dependence or the trends in the input data as well as the optimum series that can critically affect the output. Thus trial and error method to formulate the best possible configurations for training and testing of these algorithms is attempted. All these four configurations have been tried with the standard backpropagation algorithm. The reason for this, as mentioned earlier, is that backpropagation is the basic ANN method and other methods are some improvement/modification over this conventional algorithm. The assumption here is that once the most optimum series is identified, attempts will then be made to improve upon these results using the more advanced algorithms. Employing this method, it was observed that for configurations 12 input series and 7 input series, large errors were obtained as the convergence in the MSE vs iterations plot of the network was rather large resulting in larger prediction values. Results obtained from 5 input series were reasonably good and improved only slightly for the 3 input series. Thus it was decided to use the 5 input series for further optimization. The reason for poorer results for the 2 higher series could be that the prediction on the basis of 7/12 inputs is too complex than what is expected and the network architecture is not able to extract trend to the desired level of accuracy.

6. DATA SAMPLES CONSIDERED

6.1 Sample data 1(ICICI bank)

The results obtained by employing the standard backpropagation algorithm for a 5 input series for ICICI bank are summarized below in the form of a table. The major challenge before attempting any ANN model is to establish the optimum number of neurons in the hidden layers. Unfortunately there is no standard procedure to establish the number of neurons needed in the hidden layer. In the present study, these have been optimized by trial and error method by changing the number of hidden neurons from 5 onwards (in steps of 5 neurons) till best convergence in terms of Mean Square Error (MSE) is obtained. Though there was hardly any change in MSE after 50000 iterations, about 70000 iterations have been done for all the cases considered. A plot of MSE for 10 neurons in case of a backpropagation algorithm is presented below for representative purpose.

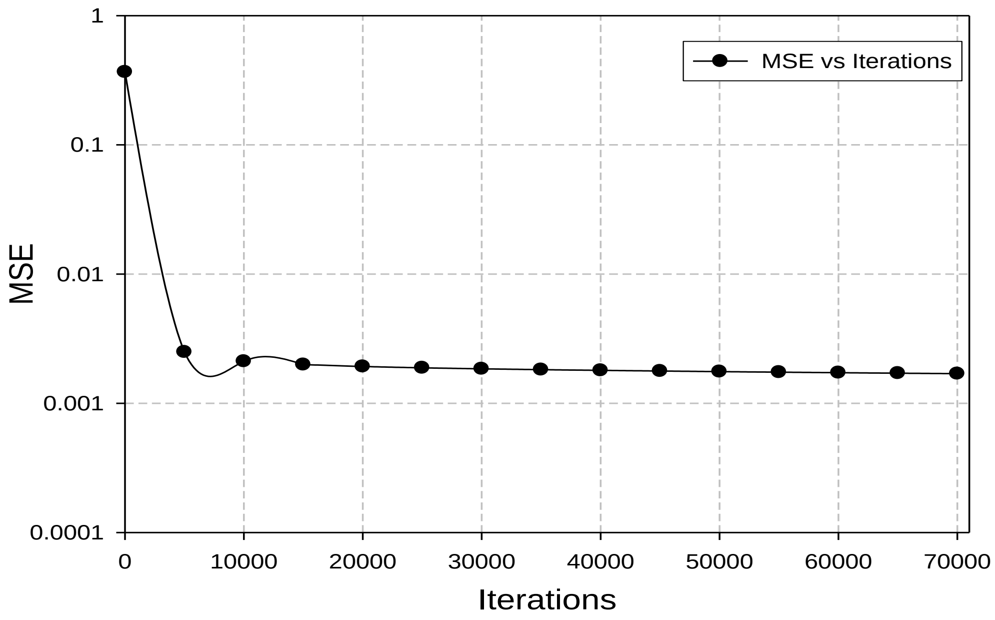
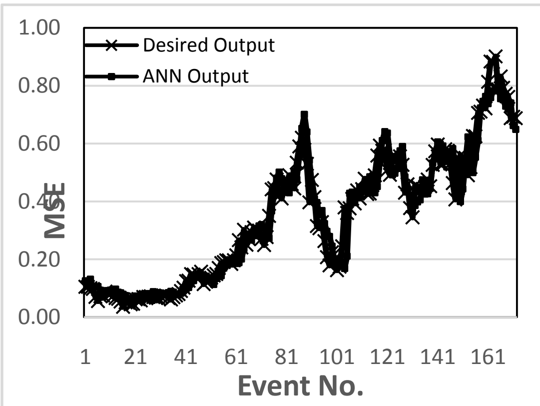


Fig 1: MSE plot for ICICI train data for Backprop 5 with neurons

Table I: Forecasting Performance of Backprop for ICICI

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.0029	84.52	0.0043	88.70
2	10	0.0017	88.97	0.0030	90.14
3	15	0.0016	88.80	0.0023	90.84

As quite accurate results are obtained even with 10-15 neurons, no need was felt to increase the neurons in the hidden layer any further. Graphs below give the desired versus the ANN predicted values for training and testing data for 5, 10 and 15 neurons considered in the table I. Since training data is seen by the network architecture iteratively, it is obvious to have slightly better results for the training data (shown in the graph below) below. The performance of the test data has been shown below in the figure. As close matching is observed between the training and the testing accuracy, it rules out any over fitting of our results (network architecture is generalizing and not remembering). From the plots and the table it is evident that for the backpropagation algorithm, the MSE values for training and testing data is slightly better for 15 neurons compared to the 10 neurons. However the testing accuracy improves only marginally from 90.14% to 90.84%, but at the cost of somewhat higher computation time when 15 neurons are considered. Thus 10 neurons can be considered as optimum for the ICICI bank data considered here. The figures 1 to 6 represent the plot of MSE versus the data point number.



Fg 2: MSE plot for ICICI test data for Backprop with 5 neurons

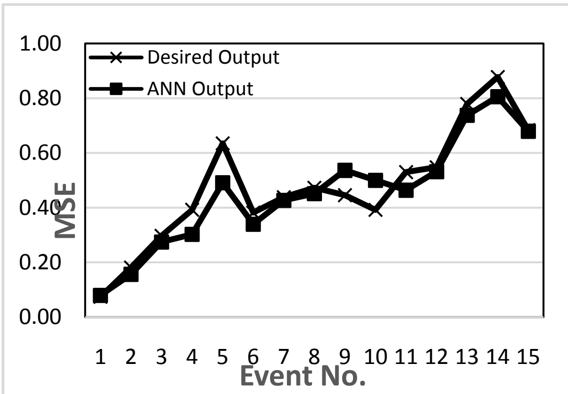


Fig 3: MSE plot ICICI traindata for Backprop with 10 neurons

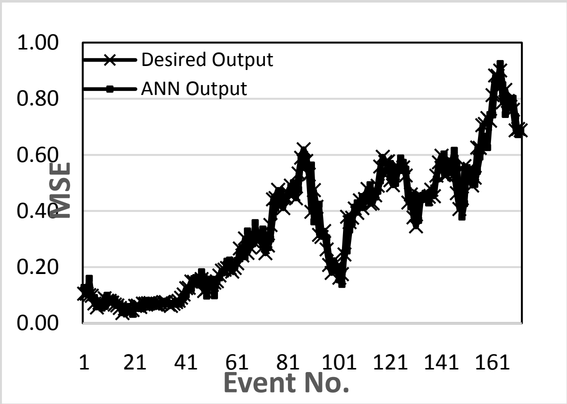


Fig 4: MSE plot ICICI test data Backprop with 10 neurons

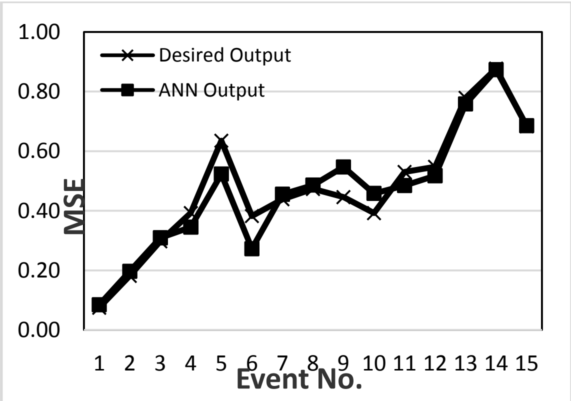


Fig 5: MSE plot ICICI traindata Backprop with 15 neurons

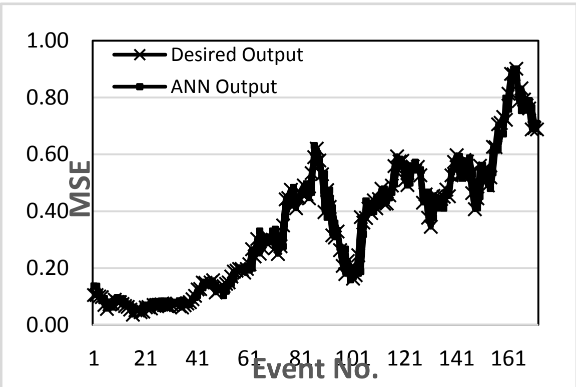


Fig 6: MSE plot ICICI testdata Backprop 15 neuron

Table II: Forecasting Performance of Resilient Backprop Algorithm for ICICI data

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.0009	91.38	0.0024	91.15
2	10	0.0007	92.62	0.0032	90.77
3	15	0.0006	92.84	0.0074	88.02

Graphs below in the figures 7 to 12 again give the desired versus the ANN predicted values for training and testing data for 5, 10 and 15 neurons considered in the table II above. While as close matching is observed between the training and the testing accuracy for 5 and 10 neurons, there is a mismatch between the training and test accuracy for 15 neurons (92.84% compared to 88.02%) suggesting that there is some level of over fitting and hence 15 neurons cannot be used. The results between 5/10 neurons are quite comparable and hence any of them can be used as the testing accuracy is > 90% in either case.

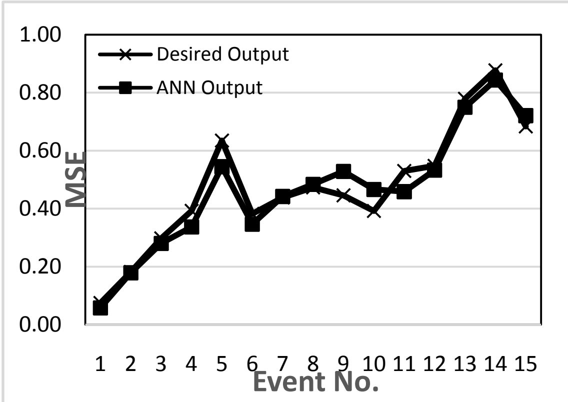


Fig 7: MSE plot for ICICI training data for RP with 5 neurons

e for RP with 10 neurons

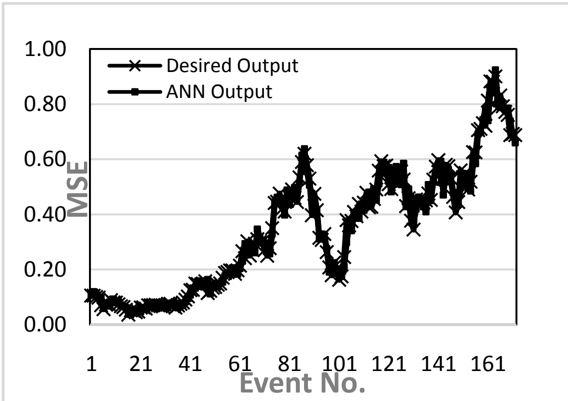


Fig 10: MSE plot for ICICI test data for RP with 10 neurons

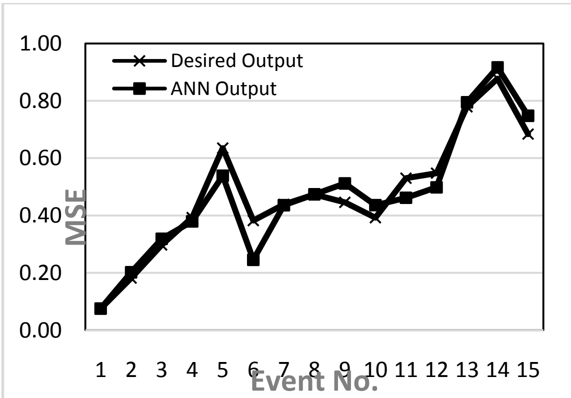


Fig 11: MSE plot for ICICI train data for RP 15 neurons

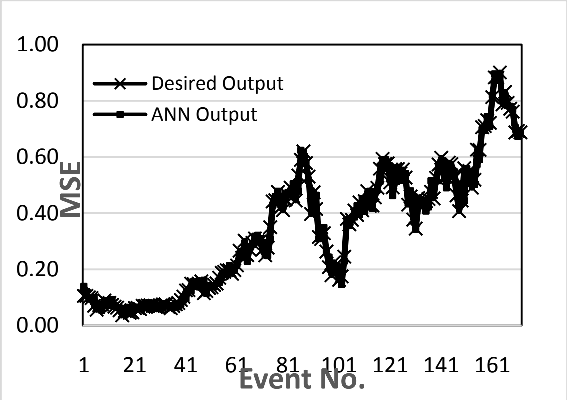


Fig 12: MSE plot for ICICI test data for RP with 15 neurons

Table III: Forecasting Performance of Conjugate Gradient (scaled) Algorithm for ICICI

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.0008	92.35	0.0032	89.85
2	10	0.0005	92.88	0.0075	89.47
3	7	0.0006	92.26	0.0035	91.24

Graphs below in the figures from 13 to 18 give the desired versus the ANN predicted values for train and test data for 5, 10 and 7 neurons considered in the table III above. The reason for employing 7 neurons was that there seems to be an indication of over fitting even for 10 neurons (row 2). Hence it was decided to conduct another training run with lesser (7) neurons. An excellent matching is observed between training accuracy (~92%) and testing accuracy ~ (91%).

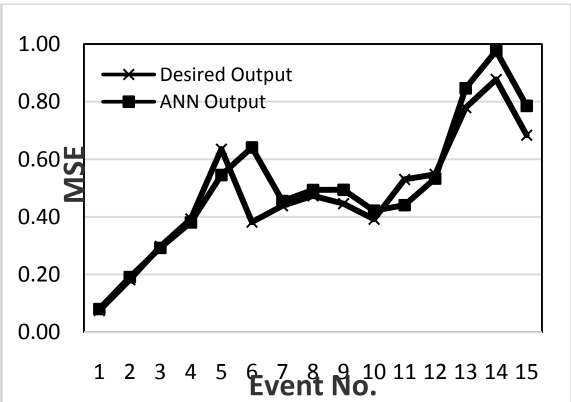


Fig 13: MSE plot for ICICI training data for SCG with 5 neurons

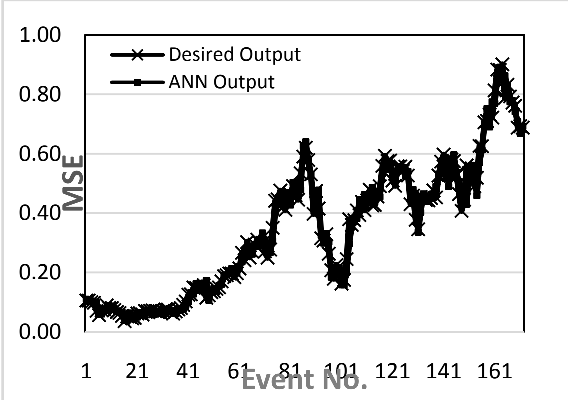


Fig 14: MSE plot for ICICI test data for SCG with 5 neurons

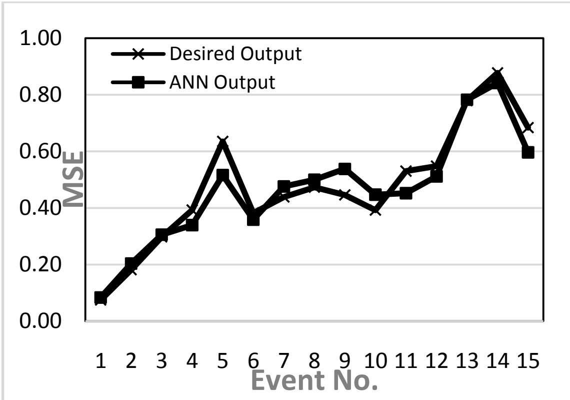


Fig 15: MSE plot ICICI train data for SCG with 10 neurons

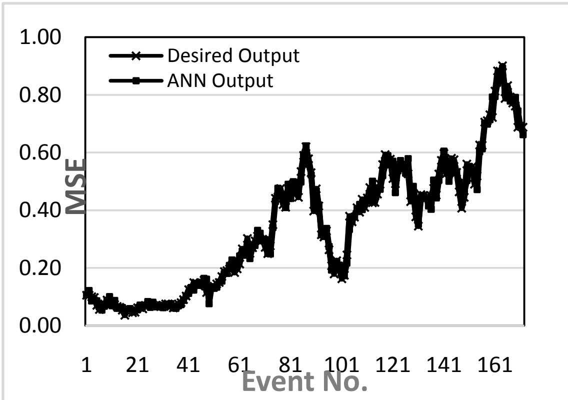


Fig 16: MSE plot for ICICI test data for SCG with 10 neurons

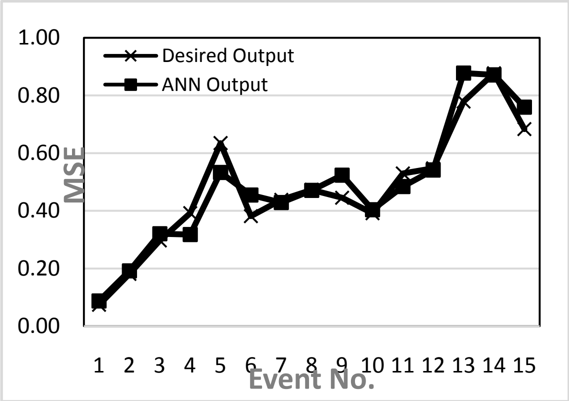


Fig 17: MSE plot ICICI train data for SCG with 7 neurons

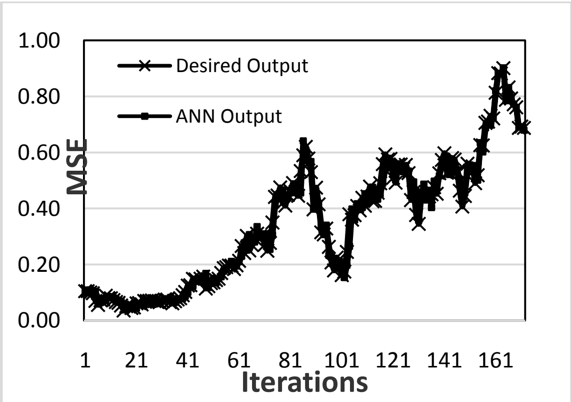


Fig 18: MSE plot for ICICI test data for SCG with 7 neurons

Table IV: Forecasting Performance of LM Algorithm for ICICI

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	4	0.0011	91.12	0.0023	91.39
2	5	0.0008	91.67	0.0019	91.90
3	7	0.0007	91.71	0.0080	87.75

Graphs below in the figures from 19 to 24 give the desired versus the ANN predicted values for train and test data for 4, 5 and 7 neurons considered in the table IV above.

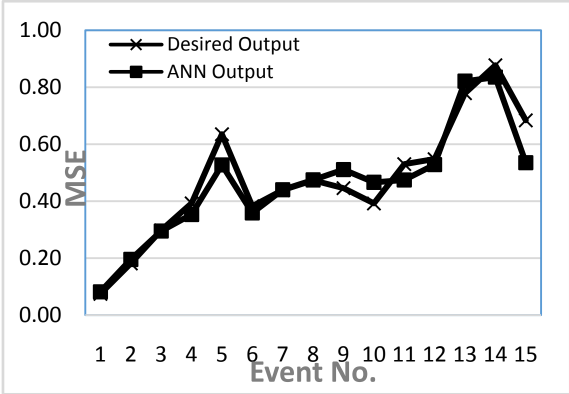


Fig 19: MSE plot for ICICI training data for LM with 4 neurons

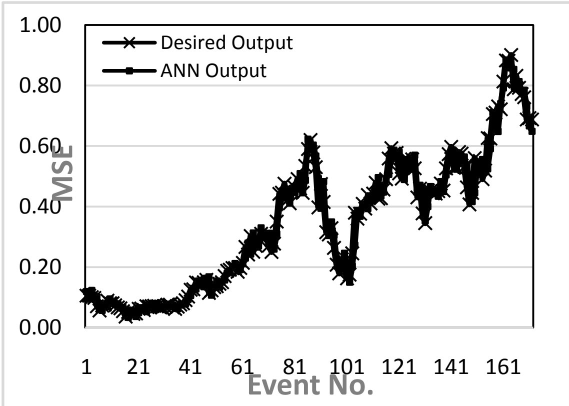


Fig 20: MSE plot for ICICI test data for LM with 4 neurons

Theoretically the way LM method is implemented, it is one of the most powerful algorithms in the ANN domain. Therefore, enough care has to be taken not to employ too many neurons in the hidden layer to avoid over fitting. Close matching is seen between the desired and the ANN generated values for the plots on the RHS side below. There is an indication of over fitting even when 7 neurons are used. However the results for 4/5 neurons are comparable as is expected with a difference of just 1 neuron with slightly better results for 5 neurons compared to 4 neurons. Thus 5 neurons are the optimum choice in this case.

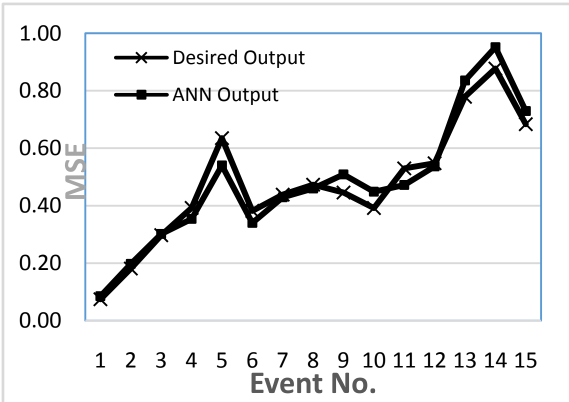


Fig 21: MSE plot ICICI training data for LM with 5 neurons

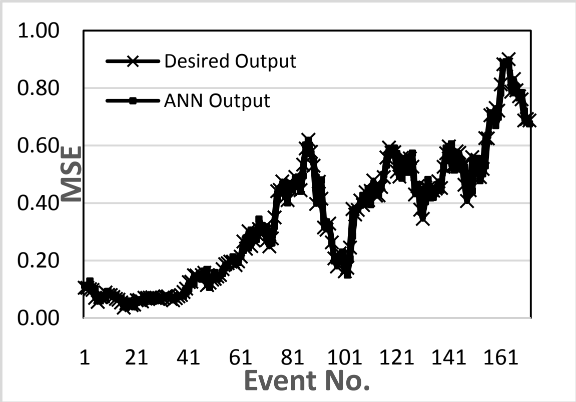


Fig 22: MSE plot for ICICI test data for LM with 5 neurons

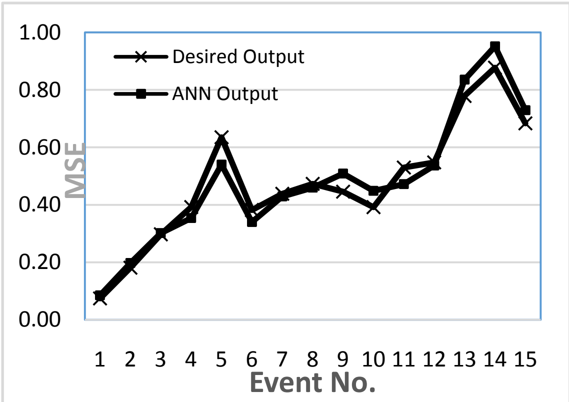


Fig 23: MSE plot ICICI training data for LM with 7 neurons

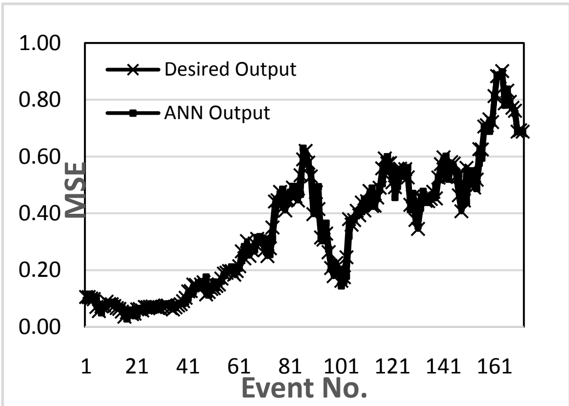


Fig 24: MSE plot for ICICI test data for LM with 7 neurons

6.2 Sample data 2 (L & T):

Table V to Table VIII and the graphs below are for the sample data 2 for L&T data considered. The training and testing files are similar to what has been discussed above for the ICICI bank.

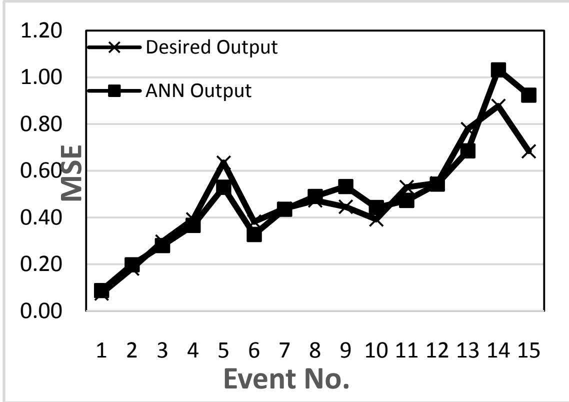


Table V: Forecasting Performance of Backpropagation algorithm for L&T

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.0038	80.80	0.0033	82.31
2	10	0.0026	89.01	0.0046	84.12
3	15	0.0022	83.93	0.0036	84.18

Graphs in the figures from 25 to 30 give the desired versus the ANN predicted values for train and test data for 5, 10 and 15 neurons considered in the table V. Close matching is observed between the desired and the ANN values for the plots on the RHS side below. Since best matching is observed for 15 neurons in table V, thus 15 neurons are considered to be optimum in the present case.

Fig 25: MSE plot for L&T training data for Backprop with 5 neurons

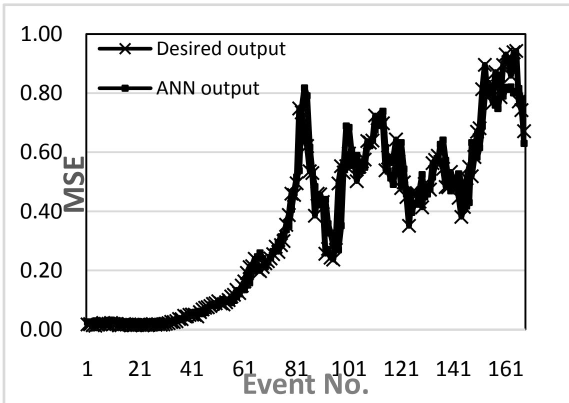


Fig 26: MSE plot for L&T test data for Backprop with 5 neurons

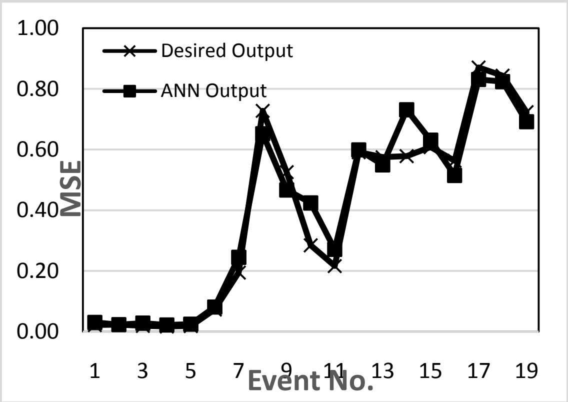


Fig 27: MSE plot L&T train data Backprop with 10 neurons

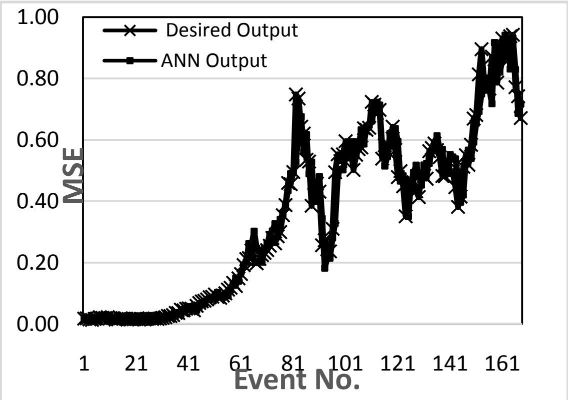


Fig 28: MSE plot L&T testdata Backprop with 10 neurons

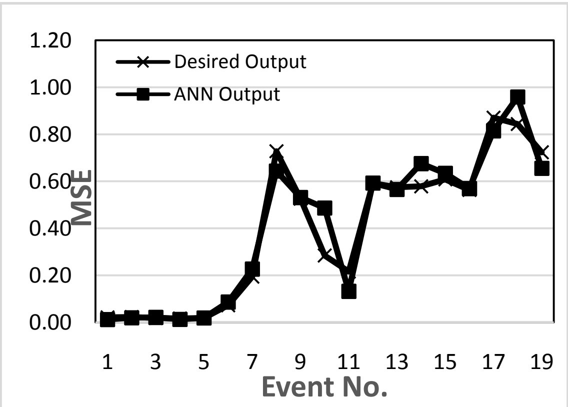


Fig 29: MSE plot L&T train data Backprop with 15 neurons

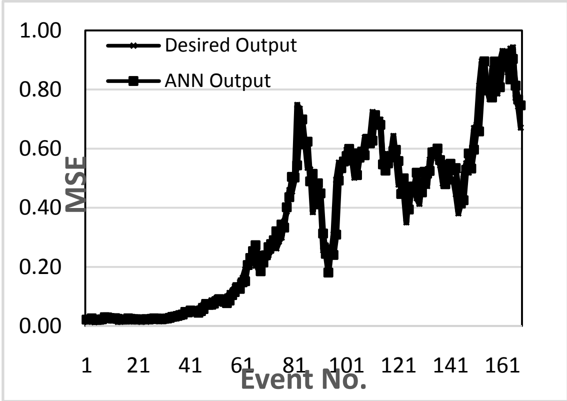


Fig 30: MSE plot L&T testdata for Backprop with 15 neurons

Table VI: Forecasting Performance of Resilient Backprop Algorithm for L&T

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.0012	87.75	0.0041	79.01
2	10	0.0011	90.49	0.0036	81.44
3	15	0.0007	89.01	0.0229	67.03

Graphs in the figures 31 to 36 below, give the desired versus the ANN predicted values for training and testing data for 5, 10 and 15 neurons considered in the table VI above. Good matching is seen between the desired and the ANN generated values for the plots on the RHS side and in the table above for 10 neurons with ~81% matching between training and testing accuracy obtained. Thus 10 neurons are considered optimal for the present case.

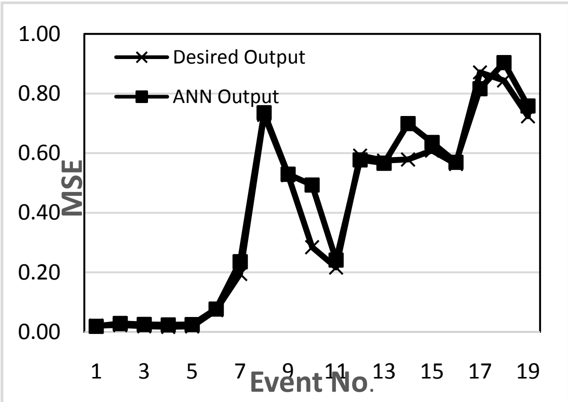


Fig 31: MSE plot for L&T training data for RP with 5 neurons

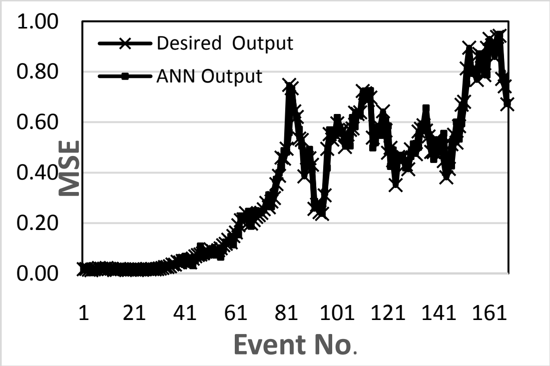


Fig 32: MSE plot for L&T test data for RP with 5 neurons

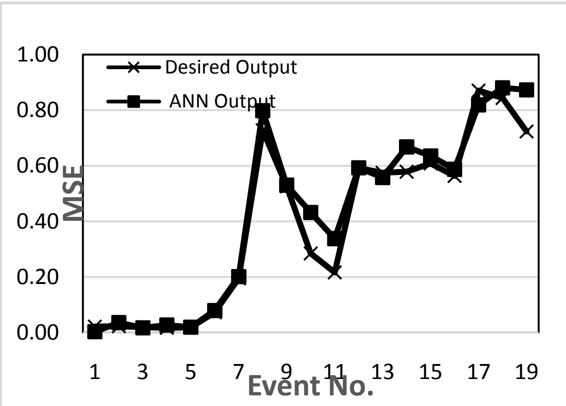


Fig 33: MSE plot L&T training data for RP with 10 neurons

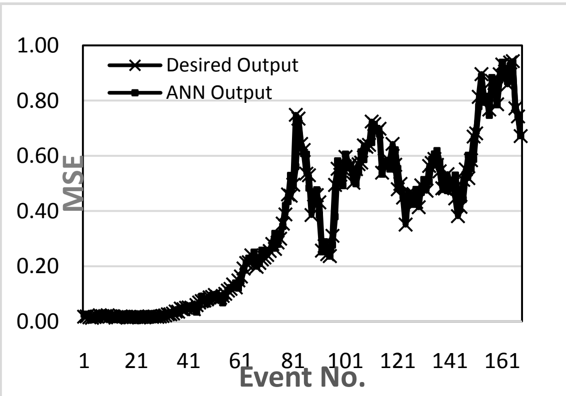


Fig 34: MSE plot for L&T test data for RP with 10 neurons

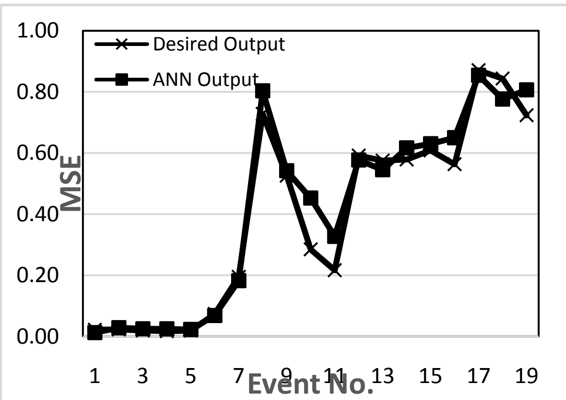


Fig 35: MSE plot L&T train data for RP with 15 neurons

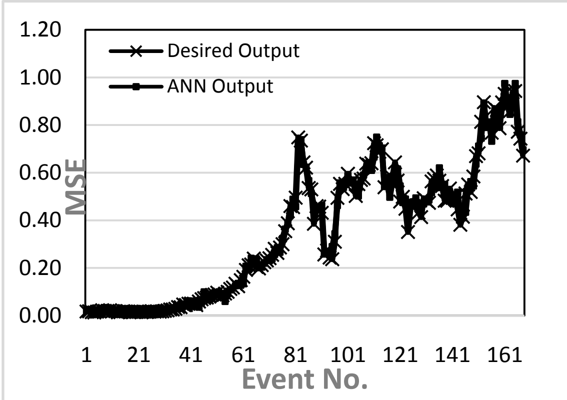


Fig 36: MSE plot for L&T test data for RP with 15 neurons

Table VII: Forecasting of SCG Algorithm for L&T

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.001	89.61	0.0058	80.34
2	7	0.0010	89.98	0.0205	77.58
3	10	0.00045	91.34	0.1268	46.67

Graphs below in the figures 37 to 42 give the desired versus the ANN predicted values for training and testing data for 5, 7 and 10 neurons considered in the table VII. There is an indication of over fitting as neurons are increased to 7 or above as seen in row 2 and 3 for table VII above. Results obtained with 5 neurons were however found to be optimum for the present case.

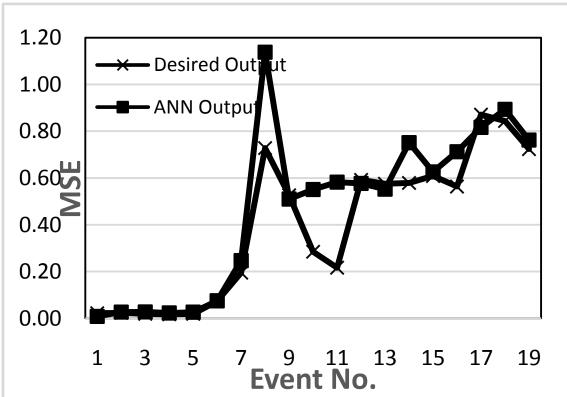


Fig 37: MSE plot for L&T training data for SCG with 5 neurons

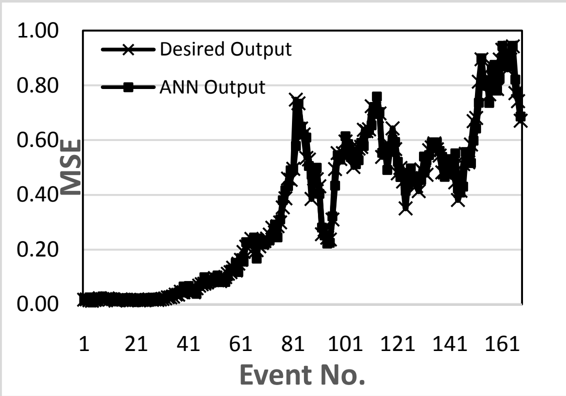


Fig 38: MSE plot for L&T test data for SCG with 5 neurons

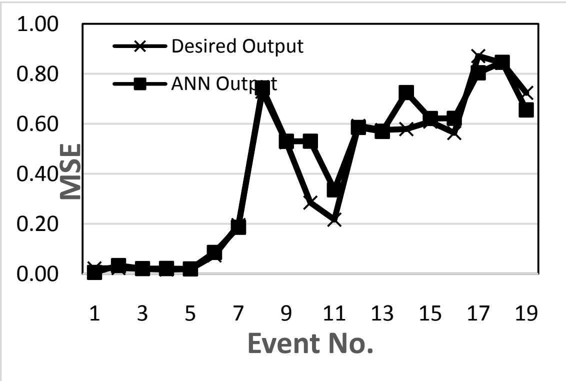


Fig 39: MSE plot L&T train data for SCG with 7 neurons

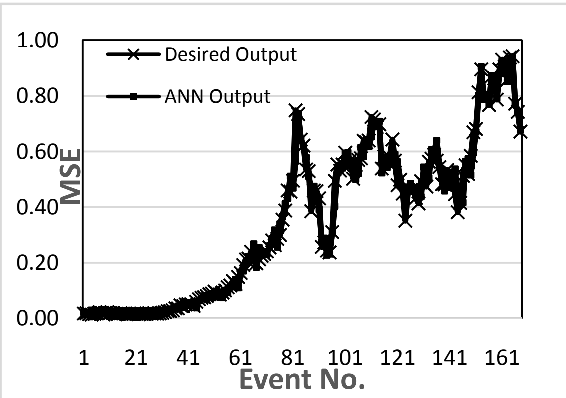


Fig 40: MSE plot for L&T test data for SCG with 7 neurons

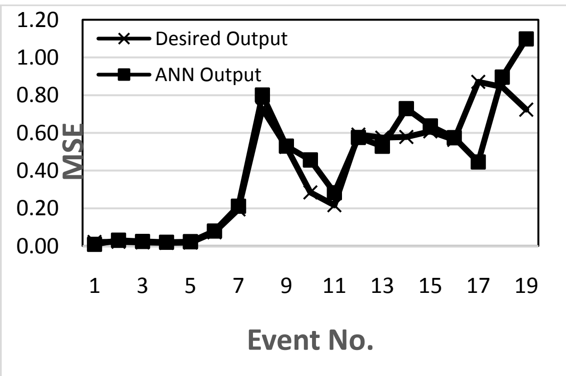


Fig 41: MSE plot L&T train data for SCG with 10 neurons

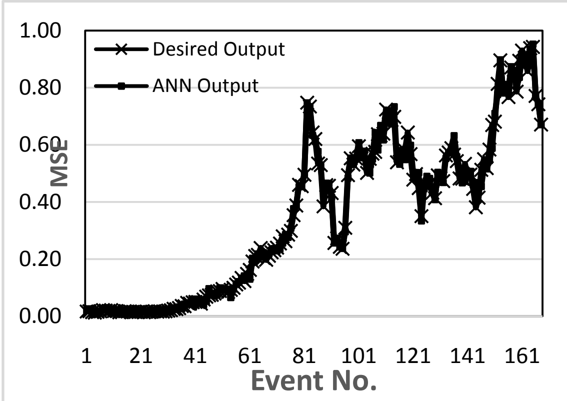


Fig 42: MSE plot for L&T test data for SCG with 10 neurons

Table VIII: Forecasting Performance of Lavenberg Marquardt Algorithm for L&T

Experiment Number	neurons in hidden layer	MSE(Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
1	5	0.0011	89.93	0.0032	82.48
2	7	0.0007	92.76	0.0052	85.35
3	10	0.0004	90.71	1.526	29.01

Graphs below in the figures 43 to 48 again give the desired versus the ANN predicted values for training and testing data for 5, 7 and 10 neurons considered in the table VIII above. Results obtained with 7 neurons were found to be optimum as there is a good matching between the training and the testing data. Results obtained for 10 neurons are clearly remembered resulting in poor generalization of the data.

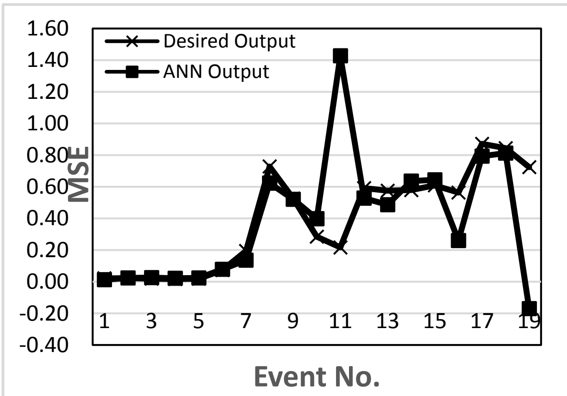


Fig 43: MSE plot for L&T training data for LM with 5 neurons

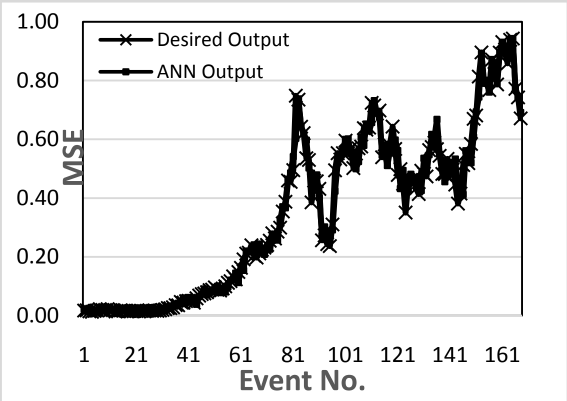


Fig 44: MSE plot for L&T test data for LM with 5 neurons

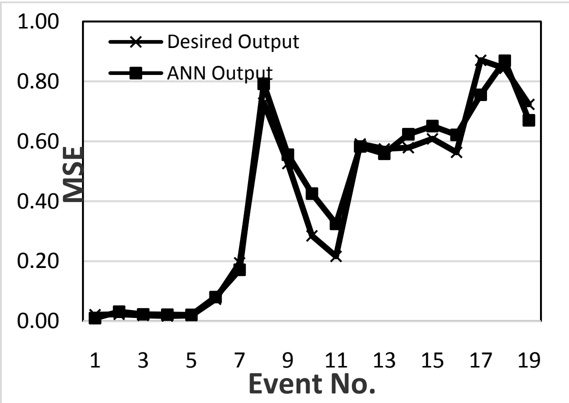


Fig 45: MSE plot L&T training data for LM with 7 neurons

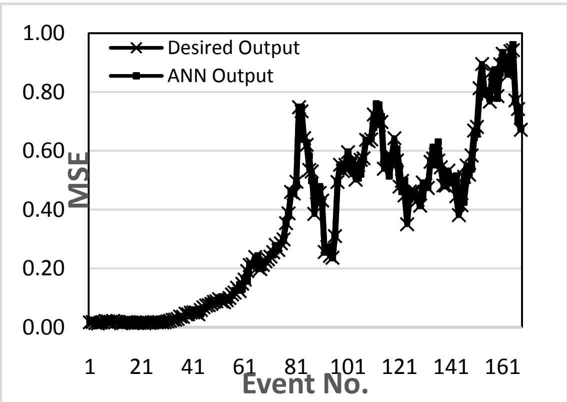


Fig 46: MSE plot for L&T test data for LM with 7 neurons

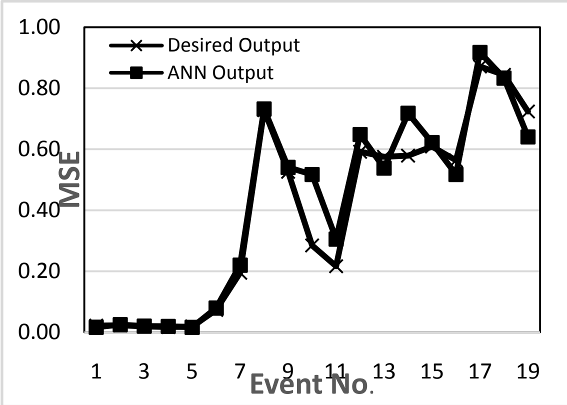


Fig 47: MSE plot L&T training data for LM with 10 neurons

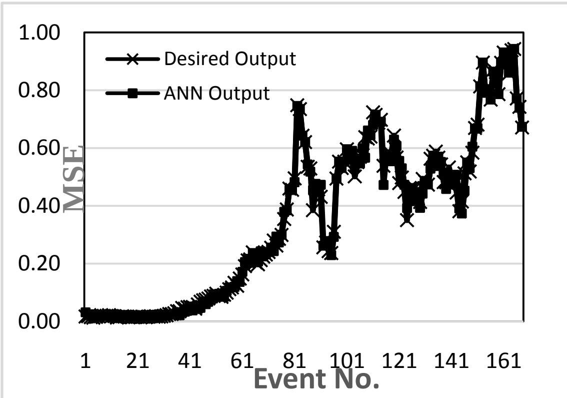
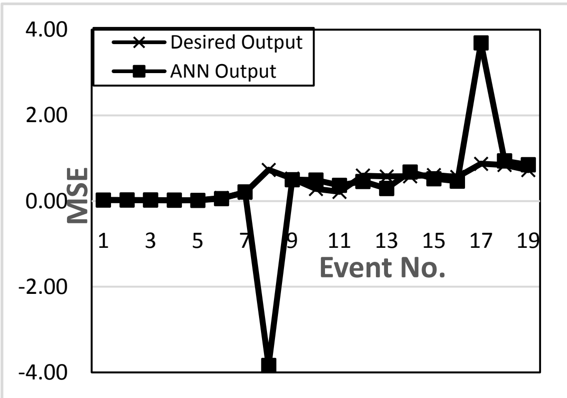


Fig 48: MSE plot L&T testing data for LM with 10 neurons



7. BAYESIAN REGULARIZATION, ONE STEP SECANT AND QUASI NEWTON METHODS

The compilation of results obtained from Bayesian regularization (BR), One step Secant (OSS) and BFGS Quasi Newton (BFG) methods for the L&T data are presented below. Similar training and testing procedure as is discussed in Results and discussion section has been employed for the above methods. It is observed that for the BR method, best results were obtained when 10 neurons were considered which gives ~88% matching in the test data. For the OSS method best results were obtained with 5 neurons which gave ~ 82 % matching for the test data. For the BFG method best results were obtained with 7 neurons which gave ~ 85 % matching for the test data.

Table IX: Forecasting BFG, BR and OSS methods

Method	Optimized neuron number	MSE (Training)	Training Accuracy (%)	MSE(Testing)	Testing Accuracy (%)
BFG	7	0.0008	87.98	0.0035	85.04
BR	10	0.0036	90.44	0.0030	87.75
OSS	5	0.0015	91.63	0.0086	82.20

8. CONCLUSIONS

Application of seven different robust and popular methods within the ANN domain for the financial time series prediction are presented in the present study. The results suggest that the methods available in the ANN domain cannot be used as off the shelf methods as these are problem dependent. There is absolutely no way of predicting beforehand which algorithm works best on a particular problem and a trial and error method has to be applied not only in terms of number of neurons in hidden layer, number of iterations but also for the algorithm to be considered, a fact undermined by many studies undertaken previously where only a one or two methods are tried on a particular problem. For the ICICI bank data our study shows that the results obtained by employing ML method with just 5 neurons in one hidden layer gives an accuracy of ~92% for the test data and is therefore ideally suited to predict the index share price for this data. Similarly for the L&T data best results are obtained by employing the ML method with 7 neurons in one hidden layer which gives nearly 85% accuracy on the test data. It is therefore obvious that for the two data samples considered by us, Marquardt Lavenberg gives best results, though with different architectures. The methods proposed for Indian financial market prediction have the inherent ability for better forecasting as ANN is powerful in recognizing the hidden data trends. In future more training data can be included to improve the prediction capability. Also the methods can be extended to more scripts in future to validate the above claims. The ANN methods considered here will also be compared to other Artificial Intelligence methods like the fuzzy logic and hybrid models like Fuzzy ANN to have a complete study of the models in AI domain.

9.. ACKNOWLEDGEMENTS

Dr. V K. Dhar, and Dr Archana Sharma, from Bhabha Atomic Research Centre (BARC), Mumbai are acknowledged for discussion and suggestions at various stages which has vastly improved the content of this manuscript.

10. REFERENCES

[1] Hayan Mo, Jun Wang et al (2016) Exponent back propagation neural network forecasting for financial cross-correlation relationship. Expert Systems with applications 53: 106-116**.**

[2] K Hornik et al. (1989) Multilayer feedforward networks are universal approximators. Neural Networks 2: 359-366.

[3] VK Dhar et al. (2010) Comparative performance of some popular ANN algorithms on benchmark and function approximation problems. Pramana 74(2): 307-324.

[4] A. Victor Devadoss , T. Antony Alphonnse Ligori (2013) Stock Prediction Using Artificial Neural Networks. International Journal of Data Mining Techniques and Applications 2: 283-291.

[5] Yunus Yetis, Halid Kaplan, and Mo Jamshidi (2014) Stock Market Prediction by Using Artificial Neural Network. In Proc. World Automation Congress, U.S., 718-722. IEEE Computer Society. doi: 10.1109/WAC.2014.6936118

[6] G.Zhang, Time series forecasting using hybrid ARIMA-ANN models , NeuroComputing 50 (2003) 159-175.

[7] C Narendra Babu, B.Eshwara Reddy, A moving averages filter based ARIMA-ANN model forecasting time series data. ; Applied soft computing 23 (2014) 27-38.

[8] Jian Zhou Wang, Ju Zie Wang et al., Forecasting stock indices with backpropagation neural network, Expert systems with applications 38 (2011) 14346-14355.

[9] A S Chen, Leung M F, Application of Neural Networks for emerging financial markets , Computers and Operations Research 30 (2003) 901-923.

[10] Mingyue Qiu, Yusong et al., Application of ANN for prediction of stock market returns : Case of Japanese stock market; chaos, Solitions & Fractals 85 (2016) 1-7

[11] A H Moghaddam, M H Moghaddam et al., Journal of economics, finance & administrative Sience 21 (2016) 89-93.

[12] S N Sivanandam, S N Deepa, Principles of Soft computing 2nd Edition (2011) Wiley Publication.

[13] V.K. Dhar et al. (2013) Artificial Neural Network based gamma-hadron segregation methodology for TACTIC telescope. Nuclear Instruments and Methods in Physics Research A 708: 56–71.

[14] D.E. Rumelhart, et al. (1986) Learning representations by back-propagating errors. Nature 323: 533-536.

[15] M. Reidmiller (1994) Advanced supervised learning in multilayer perceptrons from backpropagation to adaptive learning algorithms. Computer Standards Interfaces 16: 265-278.

[16] M.F. Moller, et al. (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6: 525-533.

[17] D.W. Marquardt(1963)An algorithm for least square estimation of non-linear parameters. Journal of the Society for Industrial and Applied Mathematics 11(2): 431-441

[18] www.nseindia.com