

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/235920219>

Indian Stock Market Prediction Using Differential Evolutionary Neural Network Model

Article · July 2012

CITATIONS

16

READS

1,931

3 authors, including:



Puspanjali Mohapatra

International Institute of Information Technology, Bhubaneswar

30 PUBLICATIONS 247 CITATIONS

[SEE PROFILE](#)



Tapas Kumar Patra

College of Engineering and Technology

12 PUBLICATIONS 54 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Data Classification and Prediction [View project](#)



Time series data mining using soft computing techniques [View project](#)

Indian Stock Market Prediction Using Differential Evolutionary Neural Network Model

Puspanjali Mohapatra (Asst. Prof)
Dept of Computer Science and Engineering
IIIT Bhubaneswar
Bhubaneswar, India

Alok Raj
Dept of Computer Science and Engineering
IIIT Bhubaneswar
Bhubaneswar, India

Tapas Kumar Patra (Reader)
Dept of Instrumentation & Electronics Engineering
College of Engineering Technology
Bhubaneswar, India

Abstract— This paper presents a scheme using Differential Evolution based Functional Link Artificial Neural Network (FLANN) to predict the Indian Stock Market Indices. The Model uses Back-Propagation (BP) algorithm and Differential Evolution (DE) algorithm respectively for predicting the Stock Price Indices for one day, one week, two weeks and one month in advance. The Indian stock prices i.e. BSE (Bombay Stock Exchange), NSE, INFY etc. with few technical indicators are considered as input for the experimental data. In all the cases, DE outperforms the BP algorithm. The Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are calculated for performance evaluation. The MAPE and RMSE in case of DE are found to be very less in comparison to BP method. The simulation study has been done using Java-6 and NetBeans.

Keywords-Stock Market Prediction; Functional Link Neural (FLANN); Differential evolution (DE); Back Propagation (BP) Algorithm; Least Mean Square (LMS) method.

I. INTRODUCTION

Due to the recent financial crisis, the world economy has gone down drastically. When the major contributor of the global economy i.e. US economy is suffering from the debt crisis, Indian stock market is also showing clear signs of slow down. Financial time series[1][2][3][4] is highly noisy, irregular, random, non-linear, non-seasonal and chaotic in nature. So it has always remained as a challenge for the common investors, stock buyers/sellers, policy makers, market researchers and capital market role players to gain knowledge about the daily stock market price values. Making money and gaining high profit is the dream of every investor, but it requires proper financial knowledge, analytical capability and ability for discovering the non-linear pattern hidden[5] within the particular stock market data. As a lot of risk is involved in the stock market, the investors become highly insecure to invest their amount. Though a lot of research has already been done on US Stock market, European Stock market, China and Japan Stock markets but less attention is given for Indian Stock

market in comparison to them. Due to rapid growth in IT, Telecom sectors in the last decade Indian Stock market is setting its upcoming platform not only in Asia as well as globally. Asia's oldest stock exchange i.e. Bombay Stock Exchange (BSE) along with NSE is contributing a lot to the global economy. It has been found that not only the economic factors but also the non-economic factors like political scenario, continuous terrorist attacks and the moods of typical individual investors have become the major role players in contributing the uncertainty in Indian stock market. Researchers are highly motivated to develop more and more efficient and advanced models for making huge profits.

Stock market analysts have adopted many statistical techniques like autoregressive moving average (ARMA) [6][7], autoregressive integrated moving average (ARIMA) [7], autoregressive conditional heteroscedasticity (ARCH) [6], generalized autoregressive conditional heteroscedasticity (GARCH)[8], ARMA-EGARCH [6], Box and Jenkins approach[4] along with various soft computing and evolutionary computing methods[9][10]. Artificial Neural Network (ANN), fuzzy set theory, Support Vector Machine (SVM) etc. are considered under soft computing techniques where as the various evolutionary learning algorithms include Genetic algorithm (GA) [9], Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) [11][12][13], Differential Evolution (DE)[14][15], Bacterial Foraging Optimization (BFO)[16] etc. It has been found that the Soft computing and Evolutionary computing methods outperforms over statistical techniques in Stock market prediction. As Statistical method can handle only linear data, they become unable to follow the non-linear pattern hidden within the stock data. The Literature survey reveals that different types of ANN's like Radial Basis Function (RBF) [5], Multi Layer Perceptron (MLP) [17], Recurrent Neural Network (RNN)[18], Time delay Neural Network (TDNN) [18], Machine Learning techniques [1], Functional Link Artificial Neural Network (FLANN) [2][17][19], Local Linear Wavelet Neural Network (LLWNN) [20], Evolutionary

Neurofuzzy NN[7][10], and various Neurofuzzy hybrid models have been used for stock prediction. In this paper, a simple FLANN network is proposed whose randomly chosen weights are optimized with BP training algorithm and DE algorithm respectively. Initially the FLANN weights are updated using LMS method. Whereas later on they are updated with DE and results are compared. The FLANN model is designed to improve the information processing capability and uncertainty modeling.

Our paper is organized as follows. Section II deals with the basic principle of FLANN and BP [12] training algorithm. The basic principle of Differential Evolution (DE) is dealt in section III. The application of DE based training to the FLANN model is proposed in section IV. The simulation study is carried out in section V. The training and testing results of the proposed model is discussed in section VI. Finally conclusions are given in section VII.

II. BASIC PRINCIPLE OF FUNCTIONAL LINK ARTIFICIAL NEURAL NETWORK AND BACK PROPAGATION

A FLANN network provides large reduction in computational requirement in comparison to Multi Layer Perceptron (MLP). It possesses high convergence speed. The multilayer perceptron has one or more hidden layers. Sometimes due to the large number of nodes in the hidden layer, it suffers from slow convergence rate and high computational complexity. So to reduce the computational burden on the network, the single layer ANN known as FLANN was originally proposed by Pao [22]. The FLANN is a single layer, single neuron architecture which has the capability to form complex decision regions by creating non-linear decision boundaries. An artificial neural network (ANN) can approximate a continuous multivariable function $f(x)$.

In this model, each component of the input vector is subjected to a functional expansion to yield the enhanced representation of the original pattern. The function used in this may be a subset of orthonormal basis function spanning over an n-dimensional representation space like $\{\cos(\pi x), \sin(\pi x), \cos(2\pi x), \sin(2\pi x), \dots\}$ and so on. Besides this trigonometric function other orthogonal function such as Legendre, Chebyshev can be used, but the trigonometric function represents the Fourier series expansion of the time series. So the trigonometric function given by $\{1, \cos(\pi x), \sin(\pi x), \cos(2\pi x), \sin(2\pi x), \dots, \cos(N\pi x), \sin(N\pi x)\}$ provide a compact representation of the function in the mean square sense. When suitable trigonometric polynomials are used after training, the FLANN weights represent a multi-dimensional Fourier series decomposition of a periodic of the desired response function. Different application of trigonometric expansion can be found by Giles and Maxwell[21].

In our proposed FLANN model the final output “Y” is represented as the sum of Y_1 and Y_2 . At first a set of six inputs including moving average input as the technical indicator are getting functionally expanded through $\{\cos(\pi x_1), \sin(\pi x_1), \dots, \cos(\pi x_6), \sin(\pi x_6)\}$ respectively. Then they get multiplied with their corresponding randomly chosen weights and the sum is represented as “ Y_1 ”.

$$Y_1 = [w_1 * \cos(\pi x_1) + w_2 * \sin(\pi x_1) + \dots + w_{11} * \cos(\pi x_6) + w_{12} * \sin(\pi x_6)] \quad (1)$$

Now the set of original six inputs are directly multiplied with their weights and summed to generate “ Y_2 ”.

$$Y_2 = [w_{13} * x_1 + w_{14} * x_2 + \dots + w_{18} * x_6] \quad (2)$$

Finally, the FLANN output

$$Y = \tanh(Y_1 + Y_2) \quad (3)$$

Now the error “e” is calculated by calculating the difference between output “Y” and the desired output “D”.

Then the weights $[w_1, w_2, \dots, w_{17}, w_{18}]$ are updated using BP learning algorithm.

$$W_j(\text{new}) = W_j(\text{old}) + \mu * e(i) * \frac{\partial(Y)}{\partial W_j} \quad (4)$$

Where “ μ ” represents the learning rate, “ $e(i)$ ” represents the error during i^{th} iteration, and $\frac{\partial(Y)}{\partial W_j}$ represents the change in weight.

The structure of the proposed FLANN model is shown in figure 1(a) and figure 1(b).

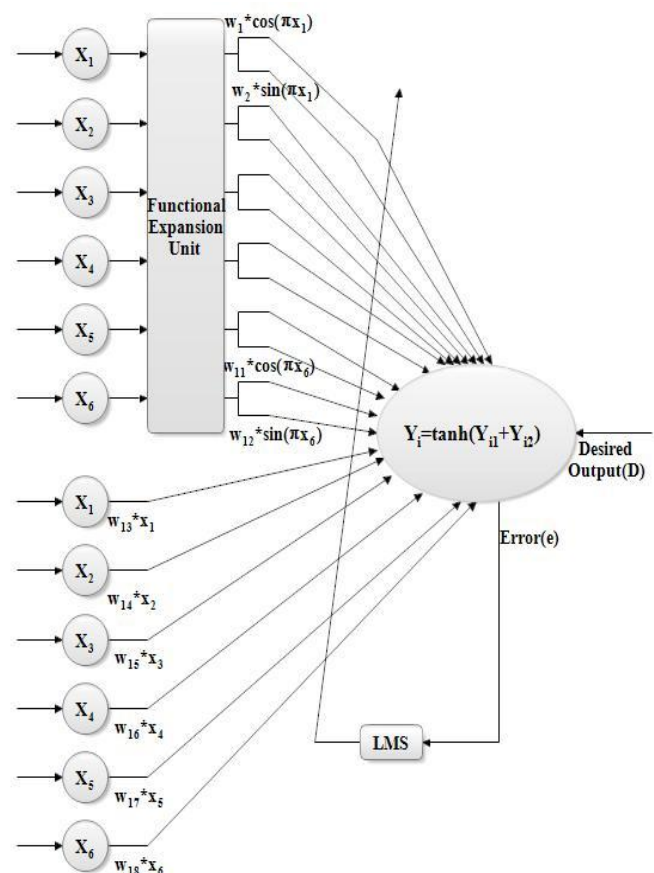


Figure 1.(a) FLANN model with LMS

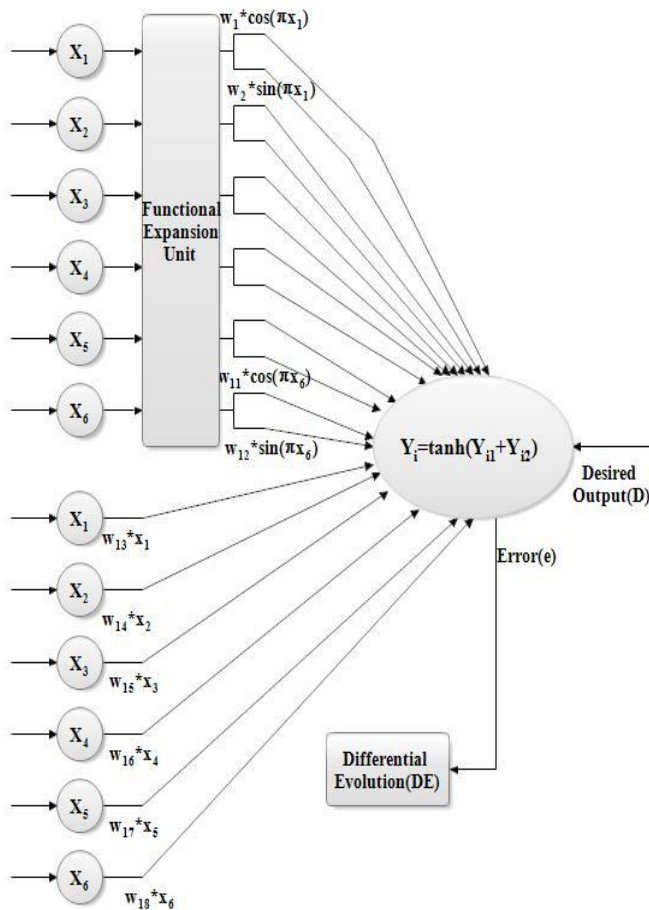


Figure 1 (b) FLANN model with DE

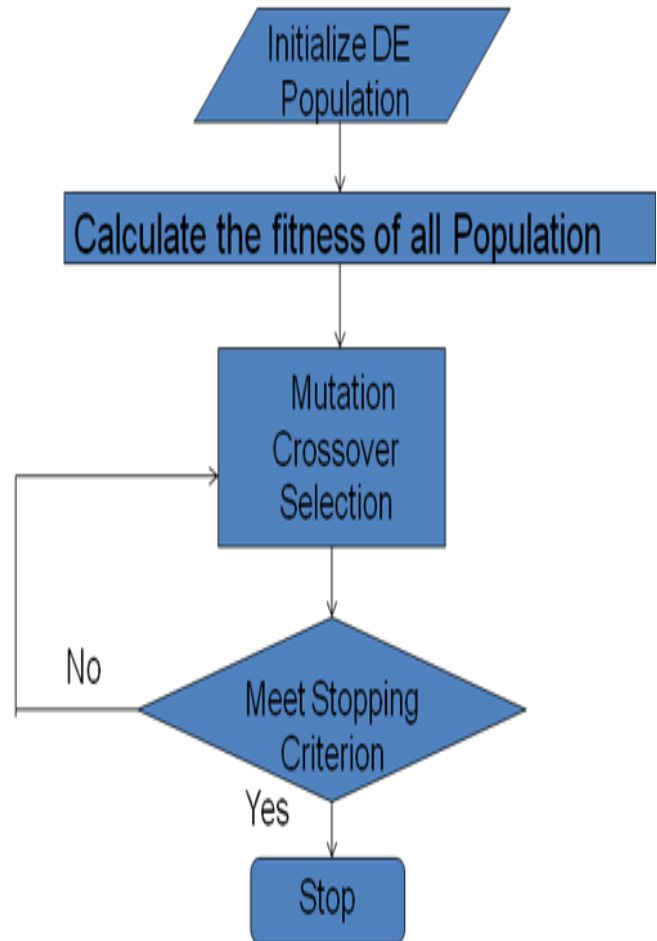


Figure 2. Flow Chart of Differential Evolution method

III. BASICS OF DIFFERENTIAL EVOLUTION ALGORITHM

Differential Evolution is introduced by Kenneth Price and Rainer Storn in 1995[14] is a population based stochastic search which behaves as an efficient global optimizer in the continuous search domain. DE has been successfully applied in global numerical optimization, pattern recognition, image pixel clustering, text document clustering, large scale power dispatch problem etc. DE algorithm is like genetic algorithm which uses similar operators: crossover, mutation and selection. DE can find the true global minimum regardless of the initial parameter values. It has other advantages such as fast convergence and requires few control parameters. The main difference in constructing better solution between GA and DE is that GA relies on crossover while DE relies on mutation operation. The mutation operation is based on the differences of randomly sampled pairs of solution in the population. The algorithm uses mutation operation as a search mechanism and the selection operation directs the search towards the prospective region in the search space. The DE algorithm also uses a non-uniform crossover which takes child vector parameters from one parent more often than it takes from others. When the components of the existing population members are used to construct the trial vectors, the recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space.

A. INITIALIZATION

DE generates an initial population size of NP, D-dimensional parameter vectors called individual which encode the candidate solution.

$$X_{i,G} = [x_{i,G}^1, \dots, x_{i,G}^D] \quad (5)$$

Where $i=1, \dots, NP$ represents the total population. Each parameter values is selected randomly and uniformly between X_{min} and X_{max} . The lower and upper parameters bound are $X_{min} = [x_{min}^1, \dots, x_{min}^D]$ and $X_{max} = [x_{max}^1, \dots, x_{max}^D]$ respectively. The initial value of j^{th} parameter in case of i^{th} individual at generation $G=0$ is generated by

$$x_{i,0}^j = x_{min}^j + \text{rand}(0,1) * (x_{max}^j - x_{min}^j) \quad (6)$$

where, $\text{rand}(0,1)$ represents uniformly distributed random number lying between 0 and 1, $j=1, 2, \dots, D$.

B. MUTATION OPERATION

For each target vector, one mutant vector $V_{i,G}$ has to be produced by using the mutation operation. For each target vector $X_{i,G}$ at the generation G, its associated mutant vector is

$$V_{i,G} = [v_{i,G}^1, \dots, v_{i,G}^D] \quad (7)$$

The mutant vector can be generated using any one of the following mutation strategies. Most frequently used five mutation strategies are given below

“DE/rand/1”

$$V_{i,G} = X_{i,G}^{r_1} + F * [X_{i,G}^{r_2} - X_{i,G}^{r_3}] \quad (8)$$

“DE/best/1”

$$V_{i,G} = X_{best,G} + F * [X_{i,G}^{r_1} - X_{i,G}^{r_2}] \quad (9)$$

“DE/rand to best/1”

$$V_{i,G} = X_{i,G} + F * [X_{best,G} - X_{i,G}] + F * [X_{i,G}^{r_1} - X_{i,G}^{r_2}] \quad (10)$$

“DE/best/2”

$$V_{i,G} = X_{best,G} + F * [X_{i,G}^{r_1} - X_{i,G}^{r_2}] + F * [X_{i,G}^{r_3} - X_{i,G}^{r_4}] \quad (11)$$

“DE/rand/2”

$$V_{i,G} = X_{i,G}^{r_1} + F * [X_{i,G}^{r_2} - X_{i,G}^{r_3}] + F * [X_{i,G}^{r_4} - X_{i,G}^{r_5}] \quad (12)$$

The indices r_1, r_2, r_3, r_4, r_5 are mutually exclusive integers randomly generated within the range 1 to NP. These indices are randomly generated once for each mutant vector. “F” represents the scaling factor.

C. CROSSOVER OPERATION

In this phase a trial vector $U_{i,G}$ is generated from each pair of the target vector $X_{i,G}$ and its corresponding mutant vector $V_{i,G}$ after going through the crossover operation.

$$U_{i,G} = [u_{i,G}^1, u_{i,G}^2, \dots, u_{i,G}^D] \quad (13)$$

Generally DE uses binomial crossover defined below

$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } (\text{rand}_j[0,1] \leq CR) \text{ or } (j=j_{rand}) \\ x_{i,G}^j, & \text{otherwise} \end{cases} \quad (14)$$

$j=1,2,3,\dots,D$ and CR is crossover rate. j_{rand} is a randomly chosen integer. The crossover operator the j th parameter of $V_{i,G}$ to the corresponding element in the trial vector $U_{i,G}$ if $(\text{rand}_j[0,1] \leq CR)$ or $(j=j_{rand})$ otherwise it is copied from the corresponding target vector $X_{i,G}$.

D. SELECTION OPERATION

In selection operation phase DE checks the upper and lower bounds of each parameter of the newly created trial vector. When they exceed, again they have to be re-initialized within pre specified range. The objective function values of all trial vectors are evaluated and selection operation is carried out. The objective function values of all trial vectors $F(U_{i,G})$ is compared to that of its corresponding target vector $F(X_{i,G})$ in the current population. If the vector has less or equal objective function values than corresponding target vector, the trial vector will replace the target vector and enter the population of next generation. Otherwise the target vector will remain in the population for the next generation. It can be defined as

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, & \text{if } F(U_{i,G+1}) \leq F(X_{i,G}) \\ X_{i,G}, & \text{Otherwise} \end{cases} \quad (15)$$

The above steps are repeated until some specific criteria are satisfied.

IV. APPLICATION OF DE BASED TRAINING TO THE (FLANN) MODEL

(1) The coefficients or weights of the proposed FLANN model are chosen from a target population of ‘NP’ number of Individuals. Each member of the population constitutes ‘D’ number of parameters. Here we have considered NP=20, D=18.

(2) ‘P’ number of patterns each containing six features including moving average (technical indicator) is obtained from the stock market price data.

(3) For each of the ‘P’ no. of patterns, X_p are passed through functional expansion unit multiplied with the weights to generate partial sum

$$Y_{i1} = \{w_1 * \cos(\pi x_1) + w_2 * \sin(\pi x_1) + w_3 * \cos(\pi x_2) + w_4 * \sin(\pi x_2) + \dots + w_{(2p-1)} * \cos(\pi x_p) + w_{(2p)} * \sin(\pi x_p)\} \quad (16)$$

and the same “P” number of patterns ‘ X_p ’ is passed through the linear combiner and multiplied with the weight

$$\text{generate the partial sum } Y_{i2} = \sum_{n=1}^P w_n * x_n \quad (17)$$

(4) Finally the resultant sum is passed through hyperbolic tangent activation function to generate

$$Y_i = \tanh(Y_{i1} + Y_{i2}) \quad (18)$$

(5) Now the Final calculated output ‘ Y_i ’ is compared with the actual output ‘ Act_i ’ to calculate the error ‘ e_i ’. At the completion of all the training sample inputs, (I) no of errors are produced. The mean square error (MSE) for a set of parameters using the relation

$$MSE(I) = \frac{\sum_{i=1}^I e_i^2}{I} \quad (19)$$

This is repeated for I number of times.

(6) The perturb vector is then calculated using equation (8) and the mutant population is obtained.

(7) Then the crossover operation is carried out using equation (13) and the trial population is created.

(8) The selection of members of the target population for the next generation is performed using Equation (15) and the whole process is repeated for some generations.

(9) In each generation the MSE is determined. MSE is plotted over number of generations. This is used to show the convergence speed of the DE model specifying its learning characteristics.

V. SIMULATION STUDY

A. Experimental Data for training and testing

The daily closing price of various Indian Stock market data i.e. BSE, INFY and NSE are considered here as the experimental data. All these data are obtained from www.finance.yahoo.com and all the models are predicting for 1-day, 1-week, 2-weeks, and 1-month ahead. All the inputs are

normalized within a range of [0, 1] using the following formula.

$$X_{\text{norm}} = \frac{X_{\text{orig}} - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \quad (20)$$

Where X_{norm} → normalized value, X_{orig} → current daily closing price, X_{min} and X_{max} are the daily minimum and maximum prices of the stock data respectively.

Details of datasets and Technical Indicators are given in the following tables.

TABLE I. STOCK DATA SETS

STOCK DATA SETS	Total Sample	Data Range	Training Sample	Testing Sample
NSE NIFTY	1 st -March-2000 30 th -March-2012	3019	1200	400
INFY	30 th -March-2000 23 rd -March-2012	9028	2000	400
BSE	26 th -feb-1992 15 th -oct-2008	4000	1600	400

TABLE II. FORMULAS USED

Technical indicators	FORMULA
Simple moving average(SMA)	$\frac{1}{N} \sum_{i=1}^N x_i$ N → No. of days, x_i → today's price
Exponential Moving Average(EMA)	$P \times A + \text{Previous EMA} \times (1-A)$; $A = 2/(N+1)$ P → current price, A → smoothing factor, N → time period

B. Training and testing of the forecasting model

Training of the FLANN model is carried out using the DE algorithm given in Section III and the optimum weights are obtained. Then using the trained model, the forecasting performance is tested using test patterns for 1-day, 1-week, 2-weeks, and 1-month ahead. The Mean Absolute Percentage Error (MAPE) defined in (21) is computed to compare the performance of various models.

$$MAPE = \frac{\sum_{k=1}^K \left| \frac{d_k - y_k}{d_k} \right|}{K} \quad (21)$$

where K is the number of test patterns.

VI. RESULT DISCUSSION

TABLE III. PERFORMANCE OF DE BASED FLANN (TRAINING AND TESTING)

Stock Data		RMSE (train)	MAPE (train)	RMSE (test)	MAPE (test)
NSE NIFTY	1-day	0.0007	1.92%	0.0002	0.527%
	1-week	0.0051	2.11%	0.0027	0.89%
	2-weeks	0.0234	2.72%	0.0162	1.14%
	1-month	0.0969	3.37%	0.0553	1.97%
INFY	1-day	0.0004	1.31%	0.0002	0.61%
	1-week	0.0056	1.77%	0.0041	0.95%
	2-weeks	0.0791	2.59%	0.0065	1.16%
	1-month	0.1134	4.24%	0.0823	2.24%
BSE	1-day	0.0001	1.31%	0.006	0.62%
	1-week	0.0060	1.76%	0.013	1.42%
	2-weeks	0.0081	2.19%	0.0238	1.96%
	1-month	0.020	3.38%	0.0386	2.03%

TABLE IV. PERFORMANCE OF BP BASED FLANN (TRAINING AND TESTING)

Price Data		RMSE (train)	MAPE (train)	RMSE (test)	MAPE (test)
NSE NIFTY	1-day	0.004	6.22%	0.0021	1.727%
	1-week	0.051	7.61%	0.0378	2.86%
	2-weeks	0.088	9.62%	0.0792	3.14%
	1-month	0.169	10.8%	0.0973	4.27%
INFY	1-day	0.003	4.31%	0.0024	0.61%
	1-week	0.008	6.77%	0.0061	1.48%
	2-weeks	0.075	9.49%	0.0524	2.367%
	1-month	0.201	11.2%	0.112	6.24%
BSE	1-day	0.0067	1.34%	0.0026	0.65%
	1-week	0.0630	2.76%	0.0176	2.24%
	2-weeks	0.088	6.39%	0.0367	4.16%
	1-month	0.180	9.68%	0.0586	6.03%

**Stock Market Prediction original plot for Dataset : BSE.txt ,
NIFTY.txt ,and Infy.txt**

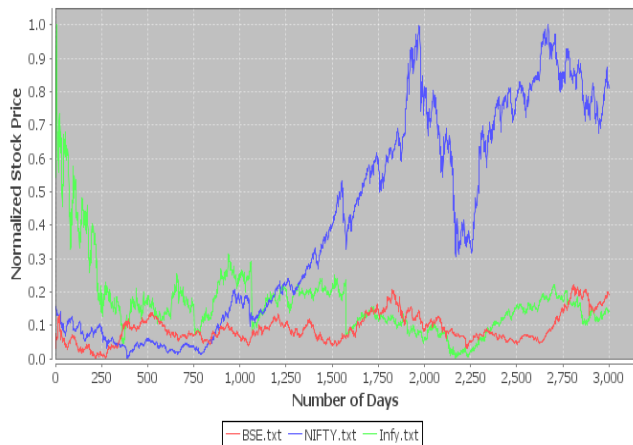


Figure 3. Stock Market Prediction original Dataset: BSE.txt, NIFTY.txt, Infy.txt

A. Performance of FLANN with BP for one day ahead prediction during training

**Stock Market Prediction (Actual VS Predicted) for 1 Day Ahead
Dataset : BSE.txt**



Figure 4. Stock Market Prediction (Actual Vs Predicted) for 1 Day ahead Dataset: BSE.txt

**SMP Mean Square Error (MSE) for 1 Day Ahead
Dataset : BSE.txt**

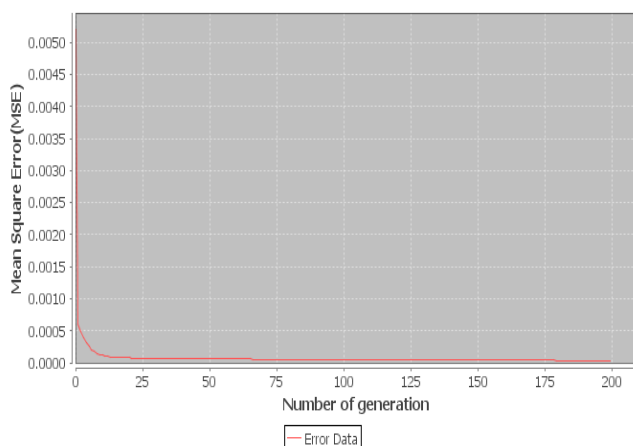


Figure 5. SMP Mean Square Error (MSE) for 1 Day ahead Dataset: BSE.txt

B. Performance of FLANN with BP for one day ahead during Testing:

**Stock Market Prediction (Actual VS Predicted) for 1 Day Ahead
Dataset : BSE.txt**

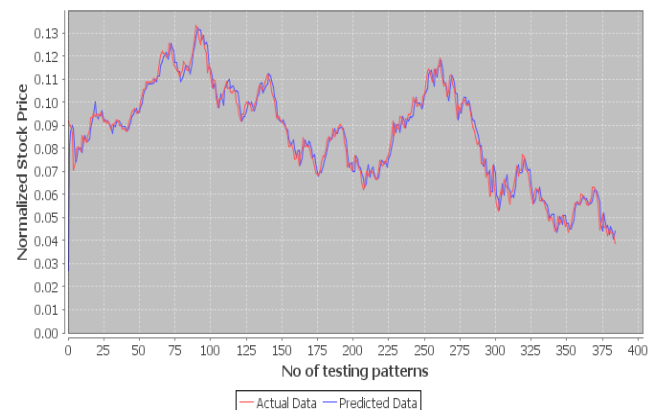


Figure 6. Stock Market Prediction (Actual Vs Predicted) for 1 Day ahead Dataset: BSE.txt

**Optimization Characteristic plot of BP based FLANN Model for
1 day , 7 day , 14 day and 30 day ahead prediction**

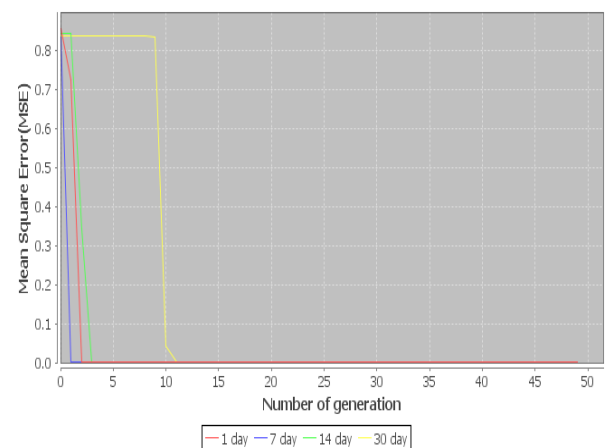


Figure 7. Optimization Characteristic plot of BP based FLANN Model for 1 Day , 7 day, 14 Day and 30 Day ahead prediction

C. Performance of FLANN with DE for one day ahead prediction during training

**Stock Market Prediction (Actual VS Predicted) for 1 Day Ahead
Dataset : BSE.txt , CR : 0.2 , F : 0.1 , NP : 10 , D : 10 (Training)**

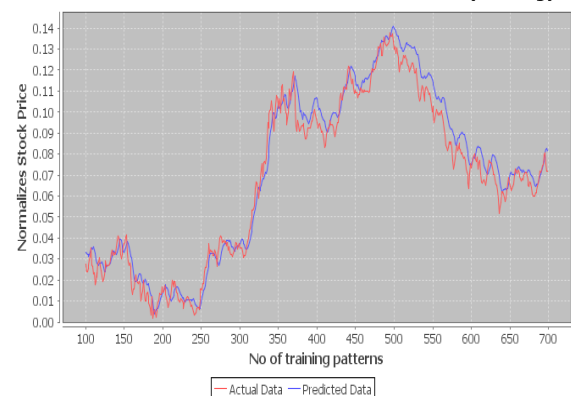


Figure 8. Stock Market Prediction (Actual Vs Predicted) for 1 Day ahead Dataset BSE.txt, CR:0.2, F:0.1, NP:10, D:10 (Training)

**SMP Mean Square Error(MSE)for 1 Day Ahead
Dataset : BSE.txt**

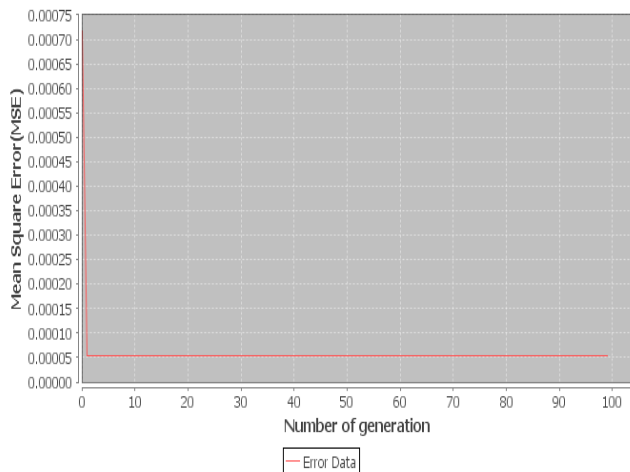


Figure 9. SMP Mean Square Error (MSE) for 1 Day ahead Dataset : BSE.txt

D. Performance of FLANN with DE for one day ahead prediction during testing

**Stock Market Prediction(Actual VS Predicted)for 1 Day Ahead
Dataset : BSE.txt , CR : 0.2 , F : 0.1 , NP: 10 , D: 10 (Testing)**

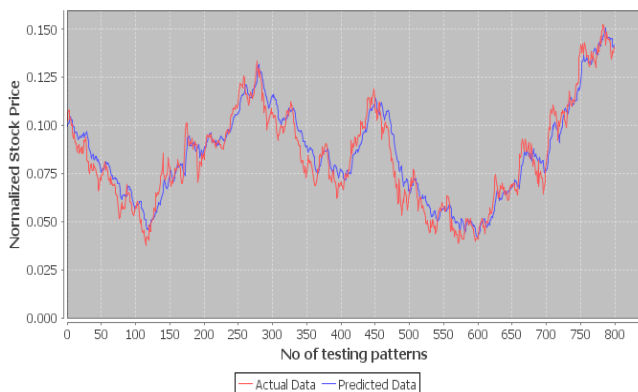


Figure 10. Stock Market Prediction (Actual Vs Predicted) for 1 Day ahead Dataset: BSE.txt, CR:0.2, F:0.1, NP:10, D:10 (Testing)

**Optimization Characteristic plot of DE based FLANN Model for
1 day , 7 day , 14 day and 30 day ahead prediction**

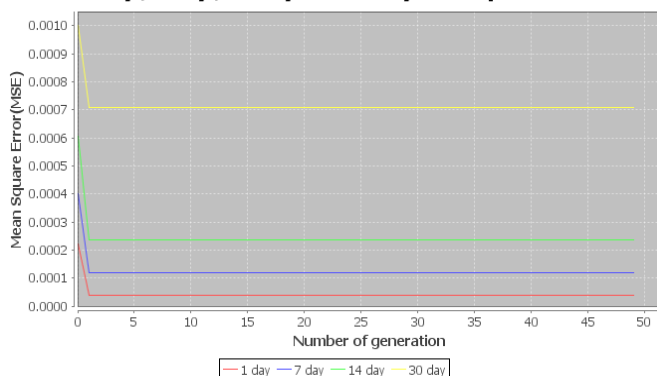


Figure 11. Optimization Characteristics plot of DE FLANN Model for 1 Day, 7 Day, 14 Day and 30 Day ahead prediction

VII. CONCLUSION

Accurate stock prediction is always a very challenging task. The proposed FLANN model trained with back propagation is giving good result as per the recorded RMSE, and MAPE values during testing for one day, one week, two weeks and one month ahead respectively. The DE optimized FLANN is proving it's superiority as far as RMSE and MAPE are concerned. Further the prediction performance of DE is to be compared with PSO.

REFERENCES

- [1] Krollner B., Vanstone B. and Finnie G.(2010), "Financial Time Series Forecasting with Machine Learning Techniques: A Survey" -ESANN 2011 proceedings, European Symposium on Artificial Neural Networks-Computational Intelligences and Machine Learning .Bruges(Belgium),28-30
- [2] Majhi B., Shalabi H. and Fathi Mowafak.(2005), "FLANN based forecasting of S&P 500 index", Information Technology Journal, 4(3):289-292, Asian Network for scientific information
- [3] Nogales F.J., Contreras, Conejo A.J., and Espinota R.(2002), "Forecasting Next-day Electricity prices by Time series Models", IEEE transactions on power systems, vol.17, No.2, pp-342-348.
- [4] Box G.E.P and Jenkins G.M.(1976) , "Time Analysis Forecasting and Control" , Sunfransisco, Holden-day.
- [5] Han J. and Kamber N.(2001), "Data Mining Concepts & Techniques", Sanfransisco, Morgan Kaufmann Publishers.
- [6] Radha S. and Thenmozhi M.(2006), "Forecasting Short term interest rates using ARMA, ARMA-GARCH and ARMA-EGARCH models-Indian Institute Of Capital markets", 9th Capital markets conference paper
- [7] .Ballini R., Luna I., Lima L.M.de and Silveria R.L.F .da.(2010), "A Comparative analysis of Neurofuzzy, ANN and ARIMA models for Brazilian Stock Index Forecasting."
- [8] Angabini A. and Wasiuzzaman S.(2011), "GARCH Models and the Financial Crisis -A study of the Malaysian Stock Market" - The International Journal of Applied Economics and Finance 5(3):226-236.
- [9] Chen.Z.(2000) , "Computational Intelligence for Decision Support", Florida, USA, CRC Press.
- [10] Yu Lixin Zhang Y.Q.(2005), "Evolutionary Fuzzy Neural Networks for Hybrid Financial Prediction", IEEE Transaction on systems, Man and Cybernetics-part-C . Applications and Reviews, vol.35, No.2.
- [11] Das S. and Abraham A.(2006), "Synergy of particle Swarm Optimization with Evolutionary Algorithm for intelligent search and optimization."
- [12] Gudise V.G. and Venayagamoorthy G.K.(2003), "Comparison of particle swarm optimization and BP as training Algorithms for Neural Networks"-07803-7914-3/s10.00 (2003) IEEE
- [13] Zhang X. Chen Y. and Yang J.Y.(2007), "Stock index forecasting using PSO based selective Neural Network Ensemble". International Conference on Artificial Intelligence, Vol.1, pp. 260-264, 2007.
- [14] R. Storm and K. Price, "Differential Evolution-A Simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, March 1995, <http://www1.icsi.berkeley.edu/~storn/TR-95-012.pdf>
- [15] Das S and Suganthan PN, "Differential Evolution- a survey of the State-of-the-art", IEEE Transaction on Evolutionary Computing, Vol. 15.No.1, pp-4-31, Feb. 2011.
- [16] Majhi Ritanjali, Panda G., Manjhi B, sahoo G., "Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques". An International Journal Expert Systems with Applications Volume 36 Issue 6, August, 2009.
- [17] Majhi R., "Stock Market Prediction using Functional Link Artificial Neural Network", 7th consortium of students in Management Research, (COSMAR-2007), IISC, Bangalore.

- [18] Saad E.W. Prokhorov D.V. and Wunsch D.C.(1998), "Comparative study of stock Trend Prediction using Time delay, Recurrent and Probabilistic Neural Networks", IEEE Transactions on Neural Networks, vol.9, No.6, November 1998.
- [19] Chen C.H. , Lin C.J. and Lin C.T.(2008), "A Functional Link Based Neurofuzzy Network for Nonlinear system control" ,IEEE Transactions on Fuzzy systems,vol.,16,No.5.
- [20] Chen Y. and Dong X.(2005), "Stock Index Modeling using EDA based local linear wavelet neural networks", International Conference on neural network and brain,vol.3, pp.1646-1650.
- [21] Giles, C.L. and T. Maxwell, 1987. "Learning invariance and generalization in higher order neural networks", Applied Optics, 26; 4972-4978
- [22] Y-H. Pao, "Adaptive Pattern Recognition & Neural Networks", Reading, MA;Addison-Wesley, 1989.