Volume 5, Issue 2, February 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Stock Price Trend Prediction using Artificial Neural Network and Derived Parameters

Smita Agrawal
Department of Information Technology
Global Institute of Technology,
Jaipur, Rajasthan, India

Dr. P. D. Murarka
Department of Research and Development
Arya College of Engineering and Technology,
Jaipur, Rajasthan, India

Abstract*— This thesis explores derived parameter optimization technique to optimize the performance of forecasting models. This study presents artificial neural network (ANN) based computational approach for predicting the stock market trend of companies from five different sectors such as:- IT Sector (Infosys), Banking Sector (SBI), Consumer Goods Sector (Tata Motors), Industrial Goods Sector (BHEL) and Basic Material Sector (ONGC). A new approach using Derived Parameter (MRDD: Measure of R square value divided by standard deviation) is developed to find out the best forecasting model. Sixty three neural network models were designed and trained using backpropagation training algorithm. Forecasting performance of sixty three neural network models was optimized using derived parameter (MRDD). This paper concludes this research work by proposing new method called Hybrid Parameter Weighted Method using Derived Parameter (HPWMDP).

***Keywords**: artificial neural network, backpropagtion algorithm, derived parameter*.

I. INTRODUCTION

Continued research in financial forecasting has proven the advantages of ANN over statistical and other methods. Financial Engineering is an increasingly popular research area. Trading systems based on computational intelligence techniques has received substantial interest from both researchers and financial traders. Neural networks have been applied extensively to technical financial forecasting because of their ability to learn complex nonlinear mapping [1].This study presents optimization method based on derived parameter (MRDD), which founds to be an efficient method to optimize the forecasting performance of sixty three neural network models. Multi layer Feed forward neural network with backpropagation training algorithm is used to forecast the trend and share prices of Infosys ,SBI, Tata Motors, BHEL and ONGC. Number of networks were designed and trained by varying network parameters, using ten years of trading data. All the networks were trained with 2160 days of trading data and predicted the prices up to next one year (240 days). Predicted output was generated through available historical data. Best forecasting model is identified using optimization method based on derived parameter technique

II. FEED FORWARD NEURAL NETWORK

Multi layer(three layer) feed forward neural network with single hidden layer as shown in Fig. 1(b), with supervised training algorithm is used to train neural network models. A neural network is used to learn patterns and relationship in data. Elements of ANN called neurons, which process information. Architecture of simple neuron is shown in Fig. 1(a), where x_1, x_2,……x_n are inputs, w_{i1}, w_{i2},…….w_{in} are weights of connection links, **b** is called bias, **a** is output from neuron and **y** is output after applying activation function to **a** [2].

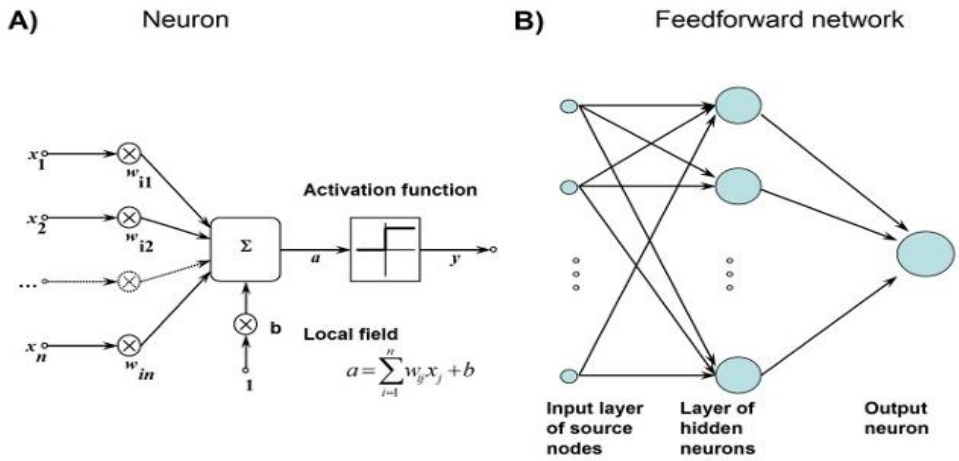

Fig.1 a) A simple neuron Fig.1 b) Feed Forward Neural Network

© 2015, IJARCSSE All Rights Reserved

This study has designed sixty three neural network architecture models by varying neural network parameters as shown in Table I and trained with supervised training algorithms (GDA, BFG and RP).

TABLE I NETWORK PARAMETERS VALUES USED TO TRAIN NEURAL NETWORK

Neural Network Parameters	***Values***
Number of neurons in input layer	***10, 50, 90, 130, 170, 210, 250***
Number of neurons in hidden layer	***5***
Activation function or transfer functions	***Tan-sigmoid, log-sigmoid and purelinear***
Learning rate	***0.01***
Goal	***0.05***
Number of hidden layer	***1***
Training algorithms	***GDA, BFG and RP***
Maximum number of epochs	***2000***

A. Activation Functions

Three types of activation functions : 1) Linear Transfer Function: Neurons of this type are used in the final layer of multilayer networks that are used as function approximators.

2) Log-sigmoid Transfer function: This transfer function is commonly used in the hidden layers of multilayer networks, and squashes the output between 0 and 1.

and 3) Tan-sigmoid transfer function : This is a most common type of activation function used with backpropagation algorithm at hidden layer of ANN, which generates output between -1 to 1.

B. Supervised Training Algorithms:

GDA: Gradient descent with adaptive learning rate backpropagation:-The traingda is a network training function that updates weight and bias values according to gradient descent with adaptive learning rate. It can train any network as long as its weight, net input, and transfer functions have derivative functions. Backpropagation is used to calculate derivatives of performance **dperf** with respect to the weight and bias variables **X**. Each variable is adjusted according to gradient descent:

$$dX = lr*dperf/dX$$

At each epoch, if performance decreases toward the goal, then the learning rate is increased by the factor **lr_inc**. If performance increases by more than the factor **max_perf_inc**, the learning rate is adjusted by the factor **lr_dec** and the change that increased the performance is not made [4].

Training stops when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- Performance is minimized to the goal.

RP: Resilient Backpropagation (trainrp):-The trainrp is a network training function that updates weight and bias values according to the resilient backpropagation algorithm (RP). It can train any network as long as its weight, net input, and transfer functions have derivative functions.

Backpropagation is used to calculate derivatives of performance **perf** with respect to the weight and bias variables **X**. Each variable is adjusted according to the following equation:

$$dX = deltaX.*sign(gX);$$

Where the elements of **deltaX** are all initialized to **delta0**, and **gX** is the gradient. At each iteration the elements of **deltaX** are modified. If an element of gX changes sign from one iteration to the next, then the corresponding element of deltaX is decreased by **delta_dec**. If an element of gX maintains the same sign from one iteration to the next, then the corresponding element of deltaX is increased by **delta_inc** [4].

Training stops when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- Performance is minimized to the goal.

© 2015, IJARCSSE All Rights Reserved

BFG: BFGS Quasi-Newton Algorithm (trainbfg):-The trainbfg is a network training function that updates weight and bias values according to the BFGS quasi-Newton method. It can train any network as long as its weight, net input, and transfer functions have derivative functions.

Backpropagation is used to calculate derivatives of performance **perf** with respect to the weight and bias variables **X**. Each variable is adjusted according to the following:

X = X + a*dX;

Where **dX** is the search direction. The parameter **a** is selected to minimize the performance along the search direction. The line search function **searchFcn** is used to locate the minimum point. The first search direction is the negative of the gradient of performance. In succeeding iterations the search direction is computed according to the following formula [4]:

dX = -H\gX;

Where **gX** is the gradient and **H** is a approximate Hessian matrix.

Training stops when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- Performance is minimized to the goal.

III. METHODOLOGY

This section describes the steps of processing done in this study.

1. Data Collection: Ten years of trading data (from January 2003 to December 2012) of five different companies were downloaded from [3].
2. Data Preprocessing: Trading data is prepared for training by applying data preprocessing steps: like data cleaning and transformation. Data transformation is done in Matlab 2010a using **mapminmax** function.
3. Sixty three neural network model were designed by making various combinations of network parameters, shown in Table 1.
4. These neural network models were trained using backpropagation training algorithms (GDA, BFG and RP). All experiments done in Matlab 2010a.
5. Performance of training models were tested using testing data for five different sectors.
6. Average Forecasting Accuracy in (%), is calculated using following formula :

   Error F_Error = abs(AP-PP)

   Average Forecasting Accuracy F_ACC = (100 – F_Error)/AP*100

   Here AP: Actual Price and PP : Predicted Price

7. The Derived Parameter method is used to optimize the forecasting performance, where derived parameter = (Product of dependent variables to be maximized or 1) / Product of dependent variables to be minimized or 1)
8. To locate a situation that maximizes R-square and minimizes standard deviation. In order to obtain the best combination of impacting variables, a search was made for the highest value of the ratio of R-square and standard deviation. This ratio was termed as "derived parameter MRDD" where

MRDD =Measure of R^2 / standard deviation =COEFFD/AVG. SD

The coefficient of determination (COEFFD), R-square, is used as a guideline to measure the accuracy of the model. A correlation greater than 0.8 is generally described *strong*, whereas a correlation less than 0.5 is generally considered *weak*.

IV. RESULTS

This section illustrates results of experiments done in this study. All experiments were done in Matlab 2010a.

A. Results for IT Sector:

Values of MRDD for different network parameters were shown in Table II for Infosys company.

Table II Showing derived parameter value (mrdd) for it sector.

NNIL	TF1			TF2			TF3		
	GDA	BFG	RP	GDA	BFG	RP	GDA	BFG	RP
10	26.46	167.11	45.76	29.57	51.32	47.93	11.71	14.44	18.01
50	34.35	121.58	58.42	19.71	28.17	43.49	10.71	17.21	2.93
90	15.84	91.21	28.32	20.11	26.0	41.10	8.32	8.32	10.12

© 2015, IJARCSSE All Rights Reserved

130	12.0	65.59	14.78	10.82	16.68	25.28	5.58	13.50	18.38
170	10.53	14.10	37.06	10.91	36.41	46.25	0.901	9.51	14.30
210	11.32	91.23	16.68	14.11	44.0	17.41	6.04	9.28	6.55
250	5.18	77.56	25.91	12.47	14.15	20.87	6.34	11.24	11.25

Here TF1: indicates transfer function tan-sigmoid in hidden layer and purelinear in output layer . TF2: indicates transfer function tan-sigmoid in hidden layer and tan-sigmoid in output layer. TF3: indicates transfer function log-sigmoid in hidden layer and purelinear in output layer of ANN.

B. Results for Banking Sector:

Values of MRDD for different network parameters were shown in Table III for SBI Bank.

Table III Showing derived parameter value (mrdd) for banking sector.

NNIL	**TF1**			**TF2**			**TF3**		
	GDA	**BFG**	**RP**	**GDA**	**BFG**	**RP**	**GDA**	**BFG**	**RP**
10	44.72	136.96	113.94	39.48	28.04	38.46	7.93	10.12	10.75
50	16.83	61.46	53.14	26.63	21.27	25.36	5.49	6.96	7.75
90	14.61	76.82	33.24	26.06	54.94	35.43	8.34	11.41	13.59
130	12.35	161.13	47.30	18.90	29.83	29.93	4.23	4.68	13.33
170	8.39	52.60	24.04	17.47	20.62	19.73	6.64	8.49	8.85
210	16.90	71.11	16.39	26.03	24.35	17.91	4.02	10.06	13.50
250	18.07	55.76	4.91	21.49	19.01	13.82	3.49	6.14	8.65

C. Results for Consumer Goods Sector:

Values of MRDD for different network parameters were shown in Table IV for Tata Motors.

Table IV Showing derived parameter value (mrdd) for consumer goods sector.

NNIL	**TF1**			**TF2**			**TF3**		
	GDA	**BFG**	**RP**	**GDA**	**BFG**	**RP**	**GDA**	**BFG**	**RP**
10	25.25	162.15	61.07	32.79	34.30	19.26	4.11	10.44	12.26
50	15.2	203.27	70.29	12.90	12.83	27.85	9.14	7.28	10.92
90	11.62	103.91	36.96	22.99	25.91	36.60	5.61	7.41	11.22
130	12.25	39.04	41.23	10.80	18.89	28.09	2.75	6.80	4.92
170	16.90	100.30	76.11	16.92	27.08	18.09	2.64	9.76	7.60
210	16.72	139.40	22.91	10.64	51.71	25.77	2.44	11.11	11.64
250	8.18	60.97	5.75	4.29	14.07	3.69	2.93	8.83	4.62

D. Results for Industrial Goods Sector:

Values of MRDD for different network parameters were shown in Table V for BHEL.

Table V Showing derived parameter value (mrdd) for industrial goods sector.

NNIL	**TF1**			**TF2**			**TF3**		
	GDA	**BFG**	**RP**	**GDA**	**BFG**	**RP**	**GDA**	**BFG**	**RP**
10	19.79	69.36	19.79	34.94	17.59	22.40	5.90	5.92	8.10

© 2015, IJARCSSE All Rights Reserved

50	27.61	68.20	60.61	14.25	18.24	31.46	4.05	8.96	9.41
90	17.79	76.23	64.31	13.17	12.82	21.95	2.43	8.17	6.46
130	17.97	40.33	51.51	17.83	13.81	29.75	3.36	6.47	7.76
170	22.87	51.42	22.53	20.59	17.22	8.93	4.84	3.73	7.86
210	13.67	35.73	11.24	20.23	19.27	14.74	3.53	7.55	6.56
250	11.23	34.33	24.59	16.67	11.93	11.93	4..62	8.93	7.62

E. Results for Basic Material Sector:

Values of MRDD for different network parameters were shown in Table VI for ONGC.

Table VI Showing derived parameter value (mrdd) for basic material sector.

NNIL	**TF1**			**TF2**			**TF3**		
	GDA	**BFG**	**RP**	**GDA**	**BFG**	**RP**	**GDA**	**BFG**	**RP**
10	54.04	199.59	188.10	65.27	77.44	123.04	23.58	19.87	23.21
50	33.24	217.33	118.44	46.53	70.0	119.57	20.63	37.51	23.20
90	35.33	183.68	100.89	46.22	112.21	92.88	10.11	23.00	30.00
130	29.0	203.34	42.88	28.19	107.27	141.69	13.79	15.27	23.11
170	147.95	157.76	130.36	39.87	88.28	70.80	8.80	21.24	17.51
210	46.51	118.75	22.59	32.07	108.74	86.99	10.46	15.58	22.48
250	58.68	122.06	4.94	20.72	82.66	101.45	9.48	16.72	22.54

V. CONCLUSIONS

A hybrid parameter weighted method using derived parameters was proposed to find out the best forecasting model. The best forecasting model is achieved by evaluating an optimum value corresponding to each and every network parameter, used in this study. To obtain an optimum value using derived parameters, the following procedure was adopted:

1. Based on the results presented in section IV, the computed values of MRDD have been categorized as shown in Table VII.
2. In next step, the frequency of network parameters corresponding to each category was computed.

Table VII Categories corresponding to derived parameter (mrdd)

S.No.	**Category**	**Range (minimum-maximum)**	**Weights assigned to categories**
1.	C1	0-50	1
2.	C2	51-100	2
3.	C3	101-150	3
4.	C4	151-200	4
5.	C5	201-250	5

3. Then , the following formula was used to compute the probability of occurrence of each network parameter:

   $P = F/TC$

   Where P is the probability, F is the frequency and TC is the total count.

4. Next , an optimum value for each and every network parameter was computed by using the formulas given below:

   $OV1 = P1*W1$
   $OV2 = P2*W2$
   $OV3 = P3*W3$
   $OV4 = P4*W4$
   $OV5 = P5*W5$
   $OV = OV1 + OV2 + OV3 + OV4 + OV5$

© 2015, IJARCSSE All Rights Reserved

Here, P1- is probability corresponding to category C1
P2- is probability corresponding to category C2
P3- is probability corresponding to category C3
P4- is probability corresponding to category C4
P5- is probability corresponding to category C5
W1- is weight assigned to category C1
W2- is weight assigned to category C2
W3- is weight assigned to category C3
W4- is weight assigned to category C4
W5- is weight assigned to category C5
OV1- is optimum value corresponding to category C1
OV2- is optimum value corresponding to category C2
OV3- is optimum value corresponding to category C3
OV4- is optimum value corresponding to category C4
OV5- is optimum value corresponding to category C5
OV- is the sum of optimum values of each category corresponding to a network parameter

5. Optimum value is computed sector wise for each and every network parameter.

A. Results of HPWMDP Method:

Computed optimum values for all sectors using HPWMDP method were shown in Table VIII, IX and X.

Table VIII Optimum value corresponding to nnil for all sectors

	Sector Wise Optimum Value				
NNIL	**IT**	**Banking**	**Consumer Goods**	**Industrial Goods**	**Basic Material**
10	1.443	1.443	1.443	1.11	2.331
50	1.332	1.332	1.554	1.21	2.22
90	1.11	1.11	1.221	1.21	2.10
130	1.11	1.32	1.0	1.11	2.442
170	1.11	1.11	1.332	1.11	2.22
210	1.11	1.11	1.332	1.0	1.776
250	1.11	1.11	1.11	1.0	1.665

Table IX Optimum value corresponding to training algorithms for all sectors

	Sector Wise Optimum Value				
Training Algorithm	**IT**	**Banking**	**Consumer Goods**	**Industrial Goods**	**Basic Material**
GDA	1.0	1.0	1.0	1.0	1.92
BFG	1.586	1.518	1.705	1.189	2.51
RP	1.046	1.13	1.141	1.14	1.995

Table X Optimum value corresponding to transfer functions for all sectors

	Sector Wise Optimum Value				
Transfer Functions	**IT**	**Banking**	**Consumer Goods**	**Industrial Goods**	**Basic Material**
TF1	1.568	1.61	1.8	1.332	1.33
TF2	1.046	1.04	1.046	1.0	1.0
TF3	1.0	1.0	1.0	1.0	1.0

Result of Tables VIII, IX and X were shown in Fig. 2, 3 and 4 respectively.

© 2015, IJARCSSE All Rights Reserved

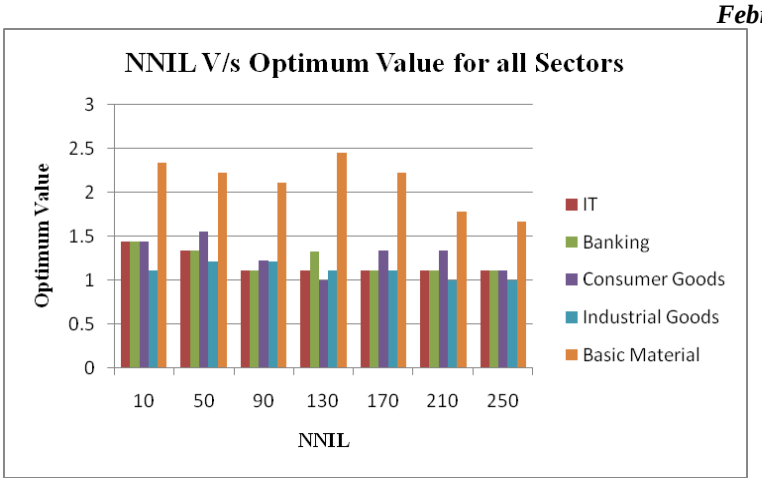

Fig. 2 Optimum Value for all Sectors corresponding to NNIL

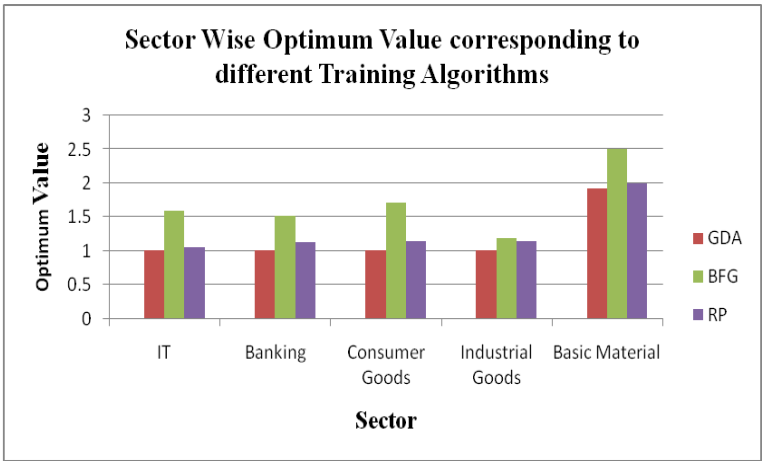

Fig. 3 Optimum Value for all Sectors corresponding to different Training Algorithms

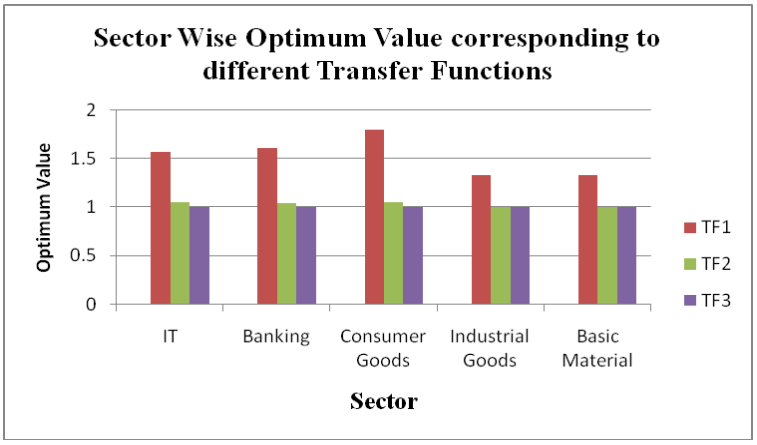

Fig. 4 Optimum Value for all Sectors corresponding to different Transfer Functions

Following conclusions can be drawn from the results shown in Fig. 2, 3 and 4:

1. Results shown in Fig. 2, indicates that for IT and Banking sector optimum value of NNIL is 10, for Consumer Goods and Industrial Goods sector NNIL is 50 and for Basic Material sector NNIL is 130.
2. Results shown in Fig. 3, indicates that training algorithm BFG found to be best for all sectors.
3. According to results shown in Fig. 4, it can observed that highest optimum value is obtained for Transfer Function TF1 .

Finally, it can be concluded that strong correlation is achieved between actual and predicted time series on following network parameters:

Training Algorithm: BFG
Transfer Function: TF1
NNIL: 10,50 and 130 depending upon sector.

© 2015, IJARCSSE All Rights Reserved

REFERENCES

[1] Ngoc Nam Nguyen and Chai Quek, "Stock price prediction using Generic Self-Evolving Takagi-Sugenokang (GSETSK) Fuzzy Neural Network," IEEE 2010.

[2] S.N Sivanandam, S Sumathi, S N Deepa, Introduction to neural networks using MATLAB 6.0. Tata McGraw Hill.

[3] http://www.yahoofinance.com

[4] Neural network toolbox of Matlab 7.10 (R2010a).

[5] http://www.google.com

[6] Suchira Chaigusin, Chaiyaporn Chirathamjaree and Judy Clayden, "The use of neural networks in the prediction of the Stock Exchange of Thailand (SET) Index," CIMCA 2008, IAWTIC 2008 and ISE 2008, IEEE Computer Society.

[7] Suk Jun Lee, Jae Joon Ahn and Kyong Joo Oh, "Using rough set to support investment strategies of real time trading in future market," Appl Intell, Vol. 32, pp.364-377, Springer 2010.

[8] Mr. Pritam R. Charka, "Stock price prediction and trend prediction using neural networks," First International Conference on Emerging Trends in Engineering and Technology, pp.592-594, IEEE 2008.

[9] Nicole Powell, Simon Y. Foo and Mark Weatherspoon, "Supervised and unsupervised methods for stock trend prediction," 40th Southeastern Symposium on System Theory, USE, pp. 203-205, IEEE 2008.

[10] Jagdish C.Patra , Nguyen C. Thanh and Pramod K. Meher, "Computationally efficient FLANN based intelligent stock price prediction system," Proceedings of International Joint Conference on Neural Networks, pp. 2431-2437, IEEE 2009.

© 2015, IJARCSSE All Rights Reserved