ISSN:2320-0790

An International Journal of Advanced Computer Technology

A FLANN and RBF with PSO Viewpoint to Identify a Model for Competent Forecasting Bombay Stock Exchange

Dr. Asif Perwej [1], Dr. Yusuf Perwej [2], Nikhat Akhtar [3], and Firoj Parwej [4]

[1]Assistant Professor, Department of Management Studies and senior member of IACSIT, Singapore

[2] Assistant Professor, Department of Computer Science & Engg., Al Baha University, Al Baha, Kingdom of Saudi Arabia (KSA)

[3] Research Scholar Ph.D (Computer Science & Engg.), Department of Computer Science & Engg., BBD University, Lucknow, India

[4] Research Scholar, Department of Computer Science & Engg., Singhania University, Jhunjhunu, Rajasthan, India

Abstract: Forecasting is the process of computation in unknown situations from the historical data. Financial forecasting and planning is usually an essential part of the business plan, and would be done as part of setting up the organization and receive funds. Financial forecasting and planning is also an essential activity to confirm a good management, keeping the organization financially sound is a key objective. The prediction of stock market has been a long time tempting topic for researchers from different fields. Stock analysts use various forecasting methods to determine how a stock's price will move in the ensuing day. The purpose of this paper is to explore the radial basis function (RBF) and function linked artificial neural network (FLANN) algorithms for forecasting of financial data. We have based our models on data taken and compared those using historical data from the Bombay Stock Exchange (BSE). The RBF and FLANN parameters updated by Particle swarm optimization (PSO). In this paper, we have examined this algorithm on a number of various parameters including error convergence and the Mean Average Percentage Error (MAPE) and comparative assessment of the RBF and FLANN algorithms is done. The proposed method indeed can help investors consistently receive gains. Finally, a simple merchandise model is established to study the accomplishment of the proposed prediction algorithm against other criterion.

Keywords: Function Linked Artificial Neural Network (FLANN), Multi Branch Neural Network (MBNN), Forecasting, Particle Swarm optimization (PSO), Bombay Stock Exchange (BSE), Radial Basis Function (RBF).

1. INTRODUCTION

A share market is a place of high interest to the investors as it presents them with an opportunity to benefit financially by investing their resources in shares and derivatives of various companies. It is a chaos system; meaning the behavioral traits of share prices are unpredictable and uncertain. To make some sort of sense of this chaotic behavior, researchers were forced to find a technique which can estimate the effect of this uncertainty to the flow of share prices. Prediction of stock trend has long been an intriguing topic and is extensively studied by researchers from different fields [1]. Stock price prediction is one of the most important topics in finance and business. However, the stock market domain is dynamic and unpredictable. Several research efforts have been carried out to predict the market in order to make profit using different techniques ranging from statistical analysis, technical analysis, to fundamental analysis among others, with different results [2]. These techniques cannot provide deeper analysis that is required and therefore not effective in predicting stock market prices. The primary area of concern is to determine the appropriate time to buy, hold or sell. Financial time-series has high volatility and the time-series changes with time. In addition, stock market's movements are affected by many macro-economical [3] factors such as political events, firms' policies, general economic conditions, investors' expectations, institutional investors' choices, movement of other stock market, psychology of investors, etc. In particular, we are interested in the correlation between the closing prices of the markets that stop trading right before or at the beginning of Indian markets. The recent advancement in soft computing has given new dimension to the field of financial forecasting. Tools based on Artificial Neural Network have increasingly gained popularity due to their inherent potency to approximate any nonlinear function to a high degree of accuracy [4]. The banks and Financial Institutions are investing heavily in the development of neural network models and have started to deploy it in the financial trading arena. Radial Basis Function (RBF), Recurrent Neural Network (RNN) and Back propagation in Multilayer Perceptron (MLP) [4] and function linked artificial neural network (FLANN) is the most popular tool for the task [5]. The optimization techniques when combined with neural networks make the computation derivative free thereby reducing the computational complexity to a greater extent. In mathematics as well as in computer science an optimization problem attempts to find the best possible solution among all the feasible solutions. The Particle swarm optimization is the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge.

2. LITERATURE SURVEY

An abundance of research has gone into the development of models based on a range of intelligent soft computing techniques over the last two decades. Early models employed the Multi Layer Perceptron (MLP) architecture using Backpropagation algorithm, while an abundance of recent work is based on evolutionary optimization techniques such as Genetic Algorithms (GA). A study was done on the effect of change of network parameters of an Artificial Neural Network Backpropagation model [6] on the stock price prediction problem. In addition to Artificial Neural Network using Backpropagation, the Probabilistic Neural Network (PNN) has also been employed to stock prediction [7]. In their work, the model is used to draw up a conservative thirty day stock price prediction of a specific stock. Another architecture introduced to the prediction problem is the Multi Branch Neural Network (MBNN) proposed by [8] (Yamshita, Hirasawa & Hu, 2005) and applied to the TOPIX (Tokyo Stock Exchange). The simulations show that MBNN, based on the concept of Universal Learning Networks (ULN), have higher accuracy of prediction than a conventional Neural Network.

There are instances of application of fuzzy logic based models to the stock market prediction as well. Hiemstra proposes a fuzzy logic forecast support system to predict the stock prices using parameters such as inflation, GNP growth, interest rate trends and market valuations. According to the paper, the potential benefits of a fuzzy logic forecast support are better decision making due to the model-based approach, knowledge management and knowledge accumulation [9]. A hybrid model proposed by (Kuo, Chen & Hwang, 2001) integrates GA based fuzzy logic and Artificial Neural Network. The model involves both quantitative factors (technical parameters) and qualitative factors such as political and psychological factors [10]. Evaluation results indicate that the neural network considering both the quantitative and qualitative factors excels the neural network considering only the quantitative factors, both in the clarity of buying-selling points and buying and selling performance. Another hybrid model involving GA proposed by (Hassan, Nath & Kirley, 2006) utilizes the strengths of Hidden Markov Models (HMM), Artificial Neural Network and GA to forecast financial market behavior [11]. Using Artificial Neural Network, the daily stock prices is transformed to independent sets of values that become input to HMM. The job of the GA is to optimize the initial parameters of the HMM. The trained HMM is then used to identify and locate similar patterns in the historical data. Other studies and research in the field of stock market prediction using soft computing techniques include comparative investigation of both the Artificial Neural Network and the statistical ARIMA model (Schumann & Lohrbach, 1994) for the German stock index (DAX) [12].

3. THE PROPOSED SYSTEMIZATION MODEL

3.1 Radial Basis Function Network

Radial Basis Function Network, which is multilayer and feedforward, is often used for strict interpolation in multi-dimensional space. The term feedforward means that the neurons are organized in the form of layers in a layered neural network. The basic architecture of a three-layered neural network is shown in figure 1. A Radial Basis Function Network has three layers including input layer, hidden layer and output layer. The input layer is composed of input data. The hidden layer transforms the data from the input space to the hidden space using a non-linear function [13]. The output layer, which is linear, yields the response of the network. The argument of the activation function of each hidden unit in a Radial Basis Function Network computes the Euclidean distance between the input vector and the center of that unit. In the structure of Radial Basis Function Network, the input data X is an I-dimensional vector, which is transmitted to each hidden unit. The activation function of hidden units is symmetric in the input space, and the output of each hidden unit depends only on the radial distance between the input vector X and the center of the hidden unit. The output of each hidden unit, hj, j = 1, 2, . . , k is given by

$$h_j(x) = \phi(\|x - c_j\|)$$

Where|| || is the Euclidean Norm, cj is the center of the neuron in the hidden layer and Φ () is the activation function.

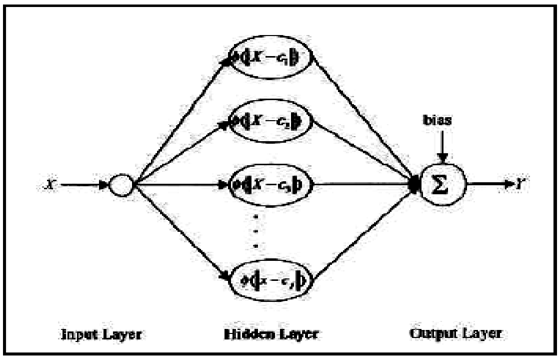

Figure 1. The Architecture of a Radial Basis Function Network

The activation function is a non-linear function and is of many types such Gaussian, multi-quadratic, thin spline and exponential functions. If the form of the basis function is selected in advance, then the trained Radial Basis Function Network will be closely related to the clustering quality of the training data towards the centers. The Gaussian activation function can be written as

$$\phi_j(x) = \exp\left[-\frac{\|x - c_j\|^2}{2\rho^2}\right]$$

Where x is the training data and ρ is the width of the Gaussian function. A center and a width are associated with each hidden unit in the network. The weights connecting the hidden and output units are estimated using a least mean square method [14]. Finally, the response of each hidden unit is scaled by its connecting weights to the output units and then summed to produce the overall network output. Therefore, the k^{th} output of the network $\hat{y}_k$ is

$$\hat{y}_k = w_0 + \sum_{j=1}^{M} w_{jk}\phi_j(x)$$

Where $\Phi j(x)$ is the response of the j^{th} hidden unit, w_{jk} is the connecting weight between the j^{th} hidden unit and the k^{th} output unit, and w_0 is the bias term [15]. The Radial Basis Function Network model, with its mathematical properties of interpolation and design matrices, is one of the promising neural models for pattern classification and has also gained popularity in forecasting of financial data.

3.2 Radial Basis Function Network Structure

A Radial Basis Function Network consists of an input layer of source nodes, a single hidden layer of nonlinear processing units, and an output layer of linear weights, as depicted in figure 2.

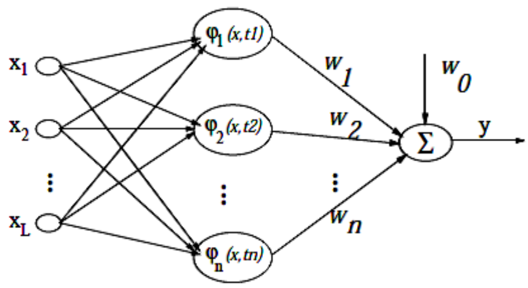

Figure 2. The Radial Basis Function Network Structure

Using the terminology of this figure, we may describe the input, output mapping performed by the Radial Basis Function Network as follows

$$y(x) = w_0 + \sum_{i=1}^{n} w_i \varphi_i(\mathbf{x}; \mathbf{t}_i) \;,$$

Where the term $\varphi_i(x;\ t_i)$ is the i^{th} radial basis function that computes the distance between the input vector x and the center t_i. Gaussian kernels are the most commonly used in practice. When the centers t_i correspond to the inputs x_i in the training data set, i = 1……..n, one gets

$$y(x) = w_0 + \sum_{i=1}^{n} w_i \exp\left(-\frac{1}{\sigma_i^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right)$$

Where σ_i is the width of the i^{th} radial basis function and is fixed by the user. The parameters w_i can be estimated by least squares [16].

3.3 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) was invented by Kennedy and Eberhart1 in the mid 1990s, while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a socio cognitive study investigating the notion of collective intelligence in biological populations [17]. In Particle Swarm Optimization, a set of randomly generated solutions (initial swarm) propagates in the design space towards the optimal solution over a number of iterations based on a large amount of information about the design space that is assimilated and shared by all members of the swarm [18]. The Particle Swarm Optimization is inspired by the ability of flocks of birds, schools of fish, and herds of animals adapt to their environment, find rich sources of food, and avoid predators by implementing an information sharing perspective consequently, developing an evolutionary advantage. The basic Particle Swarm Optimization algorithm consists of three steps, namely, generating particles' positions and velocities, velocity update, and finally, position update. Here, a particle refers to a point in the design space that changes its position from one move to another based on velocity updates [19]. The first positions, X_k^i , and velocities, V_k^i, of the initial swarm of particles are randomly generated using upper and lower bounds on the design variable values, $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$, as expressed in below equations. The positions and velocities are given in a vector format with the superscript and subscript denoting the i^{th} particle at time k. The rand is a uniformly distributed random variable that can take any value between 0 and 1. This initialization process allows the swarm particles to be randomly distributed across the design space.

$$\mathbf{x}_0^i = \mathbf{x}_{\min} + rand\left(\mathbf{x}_{\max} - \mathbf{x}_{\min}\right)$$

$$\mathbf{v}_0^i = \frac{\mathbf{x}_{\min} + rand(\mathbf{x}_{\max} - \mathbf{x}_{\min})}{\Delta t} = \frac{\text{position}}{\text{time}}$$

The second step is to update the velocities of all particles at time k +1 using the particles objective or fitness values which are functions of the particles current positions in the design space at time k. The fitness function value of a particle [20] determines which particle has the best global value in the current swarm, P_k^g , and also determines the best position of each particle over time p^i in current and all previous moves. The velocity update formula uses these two pieces of information for each particle in the swarm along with the effect of current motion, V_k^i, to provide a search direction, V_{k+1}^i, for the next iteration [21]. The velocity update formula includes some random parameters, related by the uniformly distributed variables rand to ensure good coverage of the design space and avoid entrapment in local optima. The three values that effect the new search direction, namely current motion, particle own memory and swarm influence are unified via a summation perspective with three weight factors, namely, inertia factor w self confidence factor, c_1 , and swarm confidence factor, c_2 serially.

The original Particle Swarm Optimization algorithm uses the values of 1, 2 and 2 for w, c_1, and c_2 respectively, and suggests upper and lower bounds on these values. However, the research presented in this paper found out that setting the three weight factors w , c_1 , and c_2 at 0.5, 1.5, and 1.5 serially provides the best convergence rate [22] for all test problems considered. Further combinations of values usually lead to much slower convergence or sometimes non-convergence at all. The tuning of the Particle Swarm Optimization algorithm weight factors is a topic that warrants proper investigation, but is outside the scope of this work. The weight factors use the values of 0.5, 1.5 and 1.5 for w, c_1, and c_2 respectively. Position update is the last step in each iteration. The position of each particle is updated using its velocity vector.

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \Delta t$$

The three steps of velocity update, position update, and fitness calculations are repeated until a desired convergence criterion is met. The hold-back criteria are that the maximum change in best fitness should be smaller than the specified tolerance for a specified number of moves S. The S is specified as ten moves and ε are specified as 10^{-5} for all test problems. The performance of Particle Swarm Optimization is depending upon the weight value, the larger the value of greater the global search capability [23], smaller the value of w greater the local search capability. Initially, every particle adjusts its position using certain characteristics such as the current positions, the current velocities.

$$\left| f\left(\mathbf{p}_k^g\right) - f\left(\mathbf{p}_{k-q}^g\right) \right| \le \varepsilon \qquad q = 1,2,..S$$

3.4 Function Linked Artificial Neural Network Model

The Function Linked Artificial Neural Network has been developed as an alternative architecture of the multi-layer perceptron network with application to function approximation and pattern recognition. The Function Linked Artificial Neural Network is a flat network with a single neuron that has an increased input space given by the functional expansion of its initial inputs. The main advantage of the Function Linked Artificial Neural Network is the reduced computational cost in the training stage, while maintaining a good performance of approximation [24]. The Function Linked Artificial Neural Network is a feed-forward, single layer neural network with a number of enhancement nodes referred to as functional links. These are used as supplementary inputs within the network. The different types of non-linear enhancements have been investigated. A flat network results for which only the connection weights and the bias term must be learned. Thus, the back-propagation learning algorithm, used for adapting the Function Linked Artificial Neural Network parameters, becomes very simple [25]. The structure of a Function Linked Artificial Neural Network is depicted in figure 3. The initial N inputs of the net, un, n = 1,..., N, are functionally expanded to constitute the actual inputs of the neuron, Vm, m = 1,..., N +M . In the following, the functional expansion given by a sub-set of orthogonal trigonometric functions is considered. This provides a more compact representation of the function to be approximated, in the mean-square sense, than other orthogonal basis functions. For a pre-specified order of the functional expansion, S, the actual inputs of the neuron are given by the following set

$$\{u_n, \{\cos(s\pi u_n), \sin(s\pi u_n)\}\};\ s = 1,...S;\ n = 1,..., N$$

In this way, M=2. S . N supplementary inputs of the neuron are added to the initial *N* ones. The single neuron of the network is considered to have the activation function of hyperbolic-tangent type

$$\hat{y}[k] = \frac{e^{z[k]} - e^{-z[k]}}{e^{z[k]} + e^{-z[k]}}, \qquad z[k] = x[k] + \theta^o$$

Where y is the output of the neuron, [k] represents the sampling time instant k, θ o is the bias term, and x is the sum of the original and expanded neuron. S inputs using the connection weights wm

$$x[k] = \sum_{m=1}^{N+M} w_m \cdot v_m[k]$$

In order to perform the identification of dynamic systems, the Function Linked Artificial Neural Network has to be provided with dynamic elements and appropriate learning methods. One way of doing this is by considering the Function Linked Artificial Neural Network with external dynamic elements.

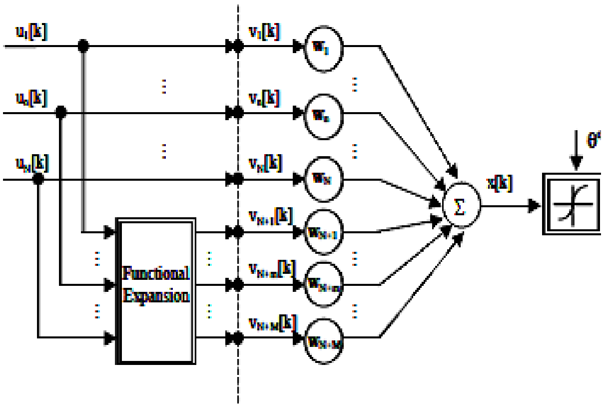

Figure 3. The Structure of a Functional Link Neural Network

The implementation of the dynamic elements as simple tapped delay units is the most applied. To model a non-linear process, the most suitable structure of the neural net is the input-output format. The considerateness of simplicity, a single input and single output dynamic system is considered. In this case, the input and output model procured using a neural network is

$$\hat{y}[k] = f(u_P[k-d],\ldots,u_P[k-d-k_u],\; y_P[k-1],\ldots,y_P[k-k_y])$$

Where u_P denotes the process input, y_p represents the process output, and y denotes the approximated output given by the trained Artificial Neural Network [26]. The maximum time delays k_u and k y are the dynamic orders of the process, and d denotes the dead time. The number of time delay units requires that the system dynamics must be known beforehand. In practice, a trial-and-error tuning of these parameters is applied [27]. The Artificial Neural Network with internal dynamic elements overcomes this drawback. The dimension of the input space of the Artificial Neural Network with external dynamics increases, depending on the number of the used past values of the input and the output of the process to be modeled.

3.5 Using Bombay Stock Exchange Network Input Selection Data

The data for the stock market prediction experiment has been collected for Bombay Stock Exchange (BSE). The time series data of all the stock indices were collected from between January 2005 to January 2011. The data collected for the stock indices consisted of the cessation price, commencement price, and undershot value on the day, exalted value in the day and the total volume of stocks traded in each day of Bombay Stock Exchange. Bear in mind that one day cessation price of the index can be slightly different from the next day commencement price, due to the introduction of after hours trading between institutions private exchanges. The proposed forecasting model is developed to forecast the cessation price of the index in each day of the forecast period. The technical pointer looks to predict the future price levels, or simply the common price flank, of a security by looking at past patterns. The many technical pointers used by traders. The indicators have been selected as input to the network, which has been used before by many researchers for stock market forecasting problems. We have been selecting the five parameters so that our analysis takes into account various factors into thought including tempo factor, volume of trade and any arbitrary spurt in prices which are normally changeable. These five parameters accompanied by the present day input form the six inputs into the system. Nevertheless, many other parameters can be derived from the input data which depict various behaviors of the data set.

4. EXPERIMENTAL RESULTS

The data samples are divesting from BSE stock exchange data. Data samples are accumulated from the historical values of Bombay Stock Exchange data. We make use of models using Radial Basis Function Network and Function Linked Artificial Neural Network structure where the parameters of each of the structure are updated using either Particle Swarm Optimization learning. In this paper we are using 1800 days data on BSE is sufficient enough to train the two models for 1 day, 30 days, and 90 days afore prediction shown in figure 4, 5 and 6. The Function Linked Artificial Neural Network is single neuron architecture. Each input is echeloned up into five branches, each being a segregate function of the primary input. Thereby efficaciously we now have five times the primary inputs we had presumed that go as inputs to the single neuron. For our experiment we have taken 6 input parameters for each pattern.

The Radial Basis Function has one layer hidden layer having 6 midpoints. The energizing function is a Gaussian one which depends on the radial distance which is distance of input sample from the referred by midpoints. Thereby the first layer is a non linear reliance on. The outcome is multiplied by a weight corresponding to each center and all of these are summed up to give a value which is called the plant output. The parameters are trained using particle swarm optimization shown in figure 7, 8 and 9. The total data set of a particular Bombay Stock Exchange stock market index is echeloned up into two, one for training of the network and the rest for testing the performance of the network after jellification the weights. The particle swarm optimization is used to train parameters of the structure using particles. We have used 22 particles by which each particle represents a solution to the problem. These parameters as a whole represent one particle and each particle discovery optimal solution to the problem. Every particle has a fitness value associated with it and it is an error in our usage. The objective is to make each particle discovery of the optimal suitable solution and in turn abbreviate error. The Mean Absolute Percentage Error (MAPE) is used to admeasure the executions of the trained prediction model for the test data.

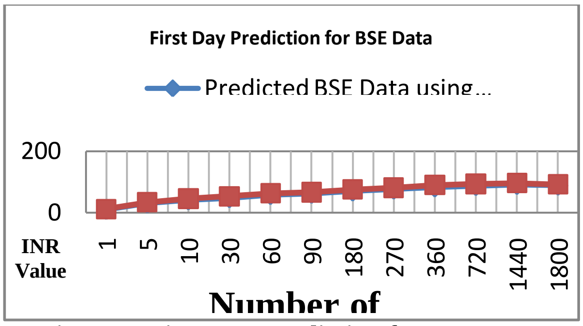

Figure 4. First Day Prediction for BSE Data using FLANN Parameters with PSO

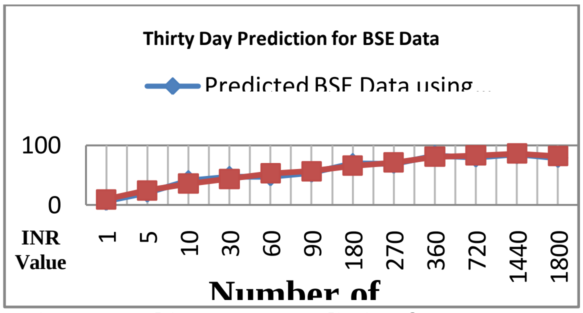

Figure 5. Thirty Days Prediction for BSE Data using FLANN Parameters with PSO

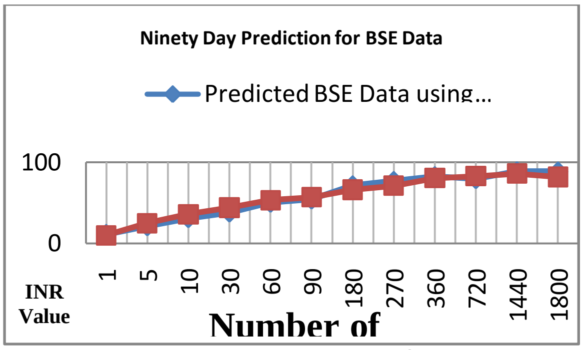

Figure 6. Ninety Days Prediction for BSE Data using FLANN Parameters with PSO

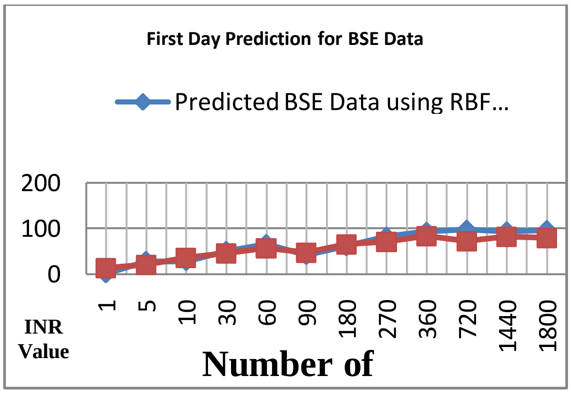

Figure 7. First Day Prediction for BSE Data using RBF Parameters with PSO

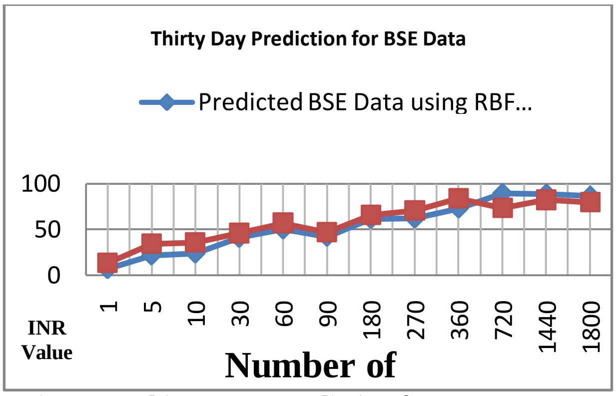

Figure 8. Thirty Days Prediction for BSE Data using RBF Parameters with PSO

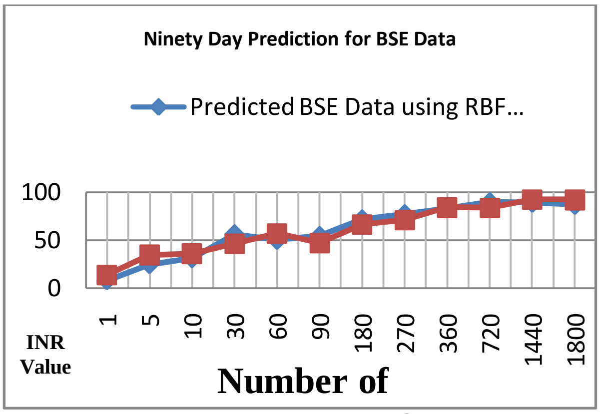

Figure 9. Ninety Days Prediction for BSE Data using RBF Parameters with PSO

Table 1: The FLANN and RBF Model Comparison through MAPE for BSE Data

The Number of Days Prediction for Further BSE Data	The Number of Days Training Done for BSE Data	The Predicted BSE Data Using FLANN Parameters with PSO MAPE (%)	The Predicted BSE Data Using RBF Parameters with PSO MAPE (%)
First Day	1800 Days	1.106	3.597
Thirty Days	1800 Days	5.781	6.089
Ninety Days	1800 Days	7.129	7.943

5. CONCLUSION

Financial Forecasting or in particular Stock Market prediction is one of the hottest fields of disquisition lately due to its commercial applications owing to the high stakes and the kinds of seductive advantage that it has to offer. The predicting the price gesticulation in stock markets has been a major defiance for common investors, businessmen, brokers and speculators. The primary area of concern is to determine the reasonable time to buy, hold or sell. In their looking-for to forecast, the investors assume that the future trends in the stock market are based at least in part on present and past events and data. The prediction of stock trend has long been an intriguing topic and is extensively studied by researchers from different fields. In this paper, we are proposing the model uses functional linked artificial neural network (FLANN) and Radial basis function (RBF) algorithm respectively, for predicting the Bombay Stock Exchange Stock Price Indices for First Day, Thirty Days, Ninety Days. In this paper Radial basis function and functional linked artificial neural network parameters updated by Particle swarm optimization (PSO). The Mean Absolute Percentage Error (MAPE) is calculated for performance assessment. In our lucid simulations, we tested our algorithms on Bombay Stock Exchange (BSE) stock exchanges. The outcome is compiled along with the response plots. The experiments show that FLANN parameters updated with PSO algorithm give the optimal outcome.

6. REFERENCES

[1]Chang Pei-Chann,Liu Chen-Hao. A TSK typ e fuzzy rule based system for stock price prediction[J].Expert Systems with Applications.1 (2008) 135-144.

[2]Cao L.J,F.E.H. Tay. Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Trans. Neural Networks.6 (2003) 1506-1518.

[3]Wang Y F. Predicting stock price using fuzzy grey prediction system [J].Expert Systems with Applications.1 (2002) 33-38.

[4] Yusuf Perwej, Asif Perwej, "Prediction of the Bombay Stock Exchange (BSE) Market Returns Using Artificial Neural Networks and Genetic Algorithms" for published in the Journal of Intelligent Learning Systems and Applications (JILSA), which is published by Scientific Research Publishing (SRP www.scirp.org), USA. Vol. 4, No. 2, May 2012, pages 108-119, ISSN Print: 2150-8402, ISSN Online: 2150-8410, DOI: 10.4236/jilsa.2012.42010

[5]E.W. Saad , D.V. Prokhorov, D.C. Wunsch (1998), " Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks", IEEE Transactions of Neural Network, 9(6), 1456-1470.

[6]. Clarence N.W. Tan and Gerhard E. Wittig, "A Study of the Parameters of a Backpropagation Stock Price Prediction Model", Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems p. 288-91,1993

[7]. Tan, H.; Prokhorov, D.V.; Wunsch, D.C., II; "Conservative thirty calendar day stock prediction using a probabilistic neural network", Computational Intelligence for Financial Engineering, 1995., Proceedings of the IEEE/IAFE 1995 9-11 April 1995 Page(s):113 – 117

[8]. Yamashita, T.; Hirasawa, K.; Jinglu Hu; "Application of multi-branch neural networks to stock market prediction", IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. VoL. 4, Aug. 2005 pp.2544 – 2548 vol. 4.

[9]. Hiemstra, Y.; "A stock market forecasting support system based on fuzzy logic", Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences, 1994. Vol. III: Information Systems: Decision Support and Knowledge-Based Systems, Volume 3, 4-7 Jan. 1994 Page(s):281 – 287.

[10]. R.J. Kuo; C.H. Chen, Y.C. Hwang; "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network", Fuzzy Sets Systems 118, 2000, pp 21-45.

[11]. Md. Rafiul Hassan , Baikunth Nath, Michael Kirley; "A fusion model of HMM, ANN and GA for stock market forecasting" , Expert Systems with Applications 2006.

[12]. Hyun-jung Kim, Kyung-shik Shin, "A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets" , Applied Soft Computing 2006, March 2006.

[13] Powell M. (1987), Radial basis functions for multivariable interpolation : A review, J.C. Mason and M.G. Cox, eds, Algorithms for Approximation, pp.143-167.

[14] S. Miyoung and P. Cheehang, "A radial basis function approach to pattern recognition and its applications", ETRI Journal, Volume 22, Number 2, June 2000.

[15] Chen, F. Cowan, and P. Grant (1991), Orthogonal least squares learning algorithm for radial basis function networks, IEEE Transactions on Neural Networks, Vol. 2, No. 2, 302-309, 1991

[16] S. Haykin. Adaptive Filter Theory. Third Edition, Prentice Hall, Inc, A Simon & Schuster Company, New Jersey, 1996.

[17] Nikhat Akhtar " Perceptual Evolution for Software Project Cost Estimation using Ant Colony System " for published in the International Journal of Computer Applications (IJCA) USA , Volume 81, No.14, Pages 23 – 30, November 2013, ISSN 0975 - 8887, http://www.ijcaonline.org., DOI : 10.5120/14185-2385

[18]J. Kennedy, and R. Eberhart, "Particle Swarm Optimization", Proceedings in IEEE International Conference Neural Networks, vol. 4, pp. 1942–1948, 1995.

[19] Dong J and Qi M, “A New Algorithm for Clustering based on Particle Swarm Optimization and K-means”, Proceedings of the International Conference on Artificial Intelligence Computational Intelligence, pp. 264-268, 2009.

[20] Van der Merwe D W and Engelbrecht A P, “Data clustering using particle swarm optimization”, Proceedings of IEEE Congress on Evolutionary Computation, Vol. 1, pp. 215- 220, 2003.

[21]Carlisle, A., and Dozier, G.. (2000). Adapting particle swarm optimization to dynamic environments. Proceedings of International Conference on Artificial Intelligence, 2000
pp. 429-434. Las Vegas, Nevada, USA.

[22]Hu, X., and Eberhart, R. C., and Shi, Y.. (2003). Particle swarm with extended memory for multiobjective optimization. Proc. of 2003 IEEE Swarm Intelligence Symposium, pages 193-197. Indianapolis, Indiana, USA, April 2003. IEEE Service Center.

[23]Parsopoulos, K.E., Vrahatis, M.N.. (2001). Particle swarm optimizer in noisy and continuously changing environments, M.H. Hamza (ed.), Artificial Intelligence and Soft Computing, pp. 289-294, IASTED/ACTA Press (Anaheim, CA, USA).

[24]Chen, C.L.P. and J.Z. Wan (1999). A Rapid Learning and Dynamic Stepwise Updating Algorithm for Flat Neural Networks and the Application to Time-Series Prediction. IEEE Trans. Systems, Man, and Cybernetics . part B, 29 (1), 62-72.

[25]Teeter, J., Mo-Yuen, C.: Application of functional link neural network to HVAC thermal dynamic system identification. IEEE Transactions on Industrial Electronics 45(1), 170–176 (1998).

[26]Emrani, S., et al.: Individual particle optimized functional link neural network for real time identification of nonlinear dynamic systems. In: 2010 The 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), (2010).

[27]Ayoubi, M. (1996). Non-linear System Identification Based on Neural Networks with Locally Distributed Dynamics and Application to Technical Processes. Fortschritt-Berichte VDI, Reihe 8, Nr.591, VDI-Verlag, Düsseldorf.