2015 IEEE Power, Communication and Information Technology Conference (PCITC)
Siksha 'O' Anusandhan University, Bhubaneswar, India.

A Comparative Study of Radial Basis Function Network with Different Basis Functions for Stock Trend Prediction

Rajashree Dash
Siksha 'O' Anusandhan University,
Khandagiri Square, Bhubaneswar, OR, India
rajashree_dash@yahoo.co.in

P. K. Dash
Siksha 'O' Anusandhan University,
Khandagiri Square, Bhubaneswar, OR, India
pkdash.india@gmail.com

Abstract— **This paper proposes a radial basis function (RBF) network trained using ridge extreme learning machine to predict the future trend from the past stock index values. Here the task of predicting future stock trend i.e. the up and down movements of stock price index values is cast as a classification problem. Recently extreme learning machine (ELM) is used as an efficient learning algorithm for single hidden layer feed forward neural networks (SLFNs). ELM has shown good generalization performances for many real applications with an extremely fast learning speed. To achieve better performance, an improved ELM with ridge regression called ridge ELM (RELM) is proposed in the study. Gaussian function is the most popular basis function used for RBFN in many applications. But the basis function may not be appropriate for all the applications. Hence the effect of the RBF network with seven different basis functions is compared for addressing the classification task. Again the performance of the RBF network is also compared with back propagation and ELM based learning over two benchmark financial data sets. Experimental results show that evaluating all recognized basis functions suitable for RBF networks is advantageous.**

Keywords- ELM; RBF; Radial Basis functions ; Stock Price index; Tchnical Indicators.

I. INTRODUCTION

The successful prediction of future price of stock index and its direction of movement may not only helpful for the investors to make effective trading strategies, but also for policy maker to monitor stock market. Keeping track of upswings and downswings over the history of individual stocks will reduce the uncertainty associated to investment decision making. Investors can choose the best times to buy and sell the stock through proper analysis of the stock trends. Hence developing more realistic models to predict the rise and fall situations in the stock price index movement is a big challenge for most of the investors and professional analysts. In literature a number of models combining technical analysis with computational intelligent techniques are available for prediction of stock price index movements [1-4].

Due to the inherent capabilities to identify complex nonlinear relationship present in the time series data based on historical data and to approximate any nonlinear function to a high degree of accuracy, the application of ANN in modeling economic conditions is expanding rapidly. A survey of literature indicates that among different types of ANNs, i.e. Multi Layer Perception Network (MLP) [5, 6], Radial Basis Function Neural Network (RBF) [7-14] and Functional Link Artificial Neural Network (FLANN) are the most popular ANN tool used for financial time series analysis. In this study a classifier using RBF network is developed for predicting the future stock trends. Because of the simple three layer architecture, RBFN has advantage of easy design, learning, strong tolerance to input noise and good generalization ability. Though the traditional back propagation algorithm with gradient descent method is the commonly used learning technique, but it suffers from the issues of imprecise learning rate, local minimal and slow rate of convergence. To avoid the common drawbacks of back propagation algorithm and to increase the accuracy some scholars proposed several improved measures, including additional momentum method, self-adaptive learning rate adjustment method, Recursive Least square method and various evolutionary algorithms like GA, PSO, DE algorithms. Recently a new batch learning algorithm called Extreme Learning Machine (ELM) has been proposed in [15] for training of single hidden layer feed forward neural network. Mainly the ELM algorithm randomly initializes parameters of hidden nodes and analytically determines output weights of SLFN. ELM has shown good generalization performances for many real applications with an extremely fast learning speed [16-18].

In this study the problem of stock market trend prediction is cast as a classification problem with two class values: one for the upward movement in stock index value and another for the downward movement in stock index value. Instead of predicting the actual stock index values, a classification model using the radial basis function network (RBFN) is proposed for predicting the future up and downward direction of stock price index movements. The predicted direction of future change can then be useful for investors to make valuable decision on when to buy and sell stocks. Instead of training the RBFN using traditional back propagation algorithm, the ELM learning is proposed for the network. Again to achieve better performance, an improved ELM with ridge regression called ridge ELM (RELM) is proposed in the study. Gaussian function is the most popular basis function used for RBFN in many applications. But the basis function may not be appropriate for all the applications. Optimal choice of basis function for a RBFN is problem dependent. Hence the effect of the RBF network with seven different basis functions is compared for addressing the classification task. Six technical indicators calculated from the historical stock index price values are selected as inputs for the proposed model. The proposed model is validated over 5 years of historical data from 2010 to 2014 of two stock indices BSE SENSEX and S&P 500. Classification accuracy and F-measure are used to evaluate the performance of the classification models.

The remainder of the paper is organized in to following sections; Section II provides a brief overview of RBF network with the details of different basis functions suitable for the network. The details of ELM Learning and the ridge extreme learning machine are presented in section III. Section IV describes the detailed steps of stock trend prediction using RBFN with RELM. Section V shows experimental results obtained from the comparative analysis. Finally section VI contains the concluding remarks.

II. RADIAL BASIS FUNCTION NETWORK

Being a universal approximator, Radial basis function network (RBFN) has gained its popularity in different applications like function approximation, curve fitting, time series prediction and classification problems [7-14]. The structure of a RBF network is same as a three layered feed forward network as shown in Fig. 1. The only difference is that there are no connection weights between the input and hidden layer. Again in hidden layer a set of radial basis functions are used for calculating the output of hidden layer nodes. The input nodes simply pass the input values to each of the nodes in the hidden layer. Each node in the hidden layer then produces an activation based on the associated radial basis function. Finally, each node in the output layer computes a weighted linear combination of the activations of the hidden nodes. Normally the hidden layer of RBFN is nonlinear where as the output layer is linear. With a d dimensional input vector and M number of hidden layer nodes, the output of hidden layer node H_i and output of output layer node y_j is calculated as follows:

$$H_i(x) = \phi(x)$$
$$y_j = \sum_{i=1}^{M} w_{ij} H_i(x) + w_{0j} \tag{1}$$

Where w_{ij} is the connection weights between ith hidden layer and jth output layer node. Then the error obtained by comparing the output with desired output is used to update the weights and parameters of the radial basis functions using learning algorithm.

The performance of the network is greatly dependent on the choice of radial basis function. Gaussian function is the commonly used radial basis function. But the basis function may not be appropriate for all the applications. Optimal choice of basis function for a RBFN is normally problem dependent. The radial basis functions having useful properties for the RBF network are specified as follows:

a. Gaussian

$$\phi(x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2}$$

b. Multiquadratic

$$\phi(x) = \left(x^2 + \sigma^2\right)^{\frac{1}{2}} \tag{3}$$

c. Inverse Multiquadratic

$$\phi(x) = \frac{1}{\left(x^2 + \sigma^2\right)} \tag{4}$$

d. Thin Plate Spline

$$\phi(x) = \left(\frac{x}{\sigma}\right)^2 \ln\left(\frac{x}{\sigma}\right) \tag{5}$$

e. Logistic

$$\phi(x) = \frac{1}{\left(1 + \exp\left(\frac{x}{\sigma}\right)\right)} \tag{6}$$

f. Cubic

$$\phi(x) = x^3 \tag{7}$$

g. Linear

$$\phi(x) = x \tag{8}$$

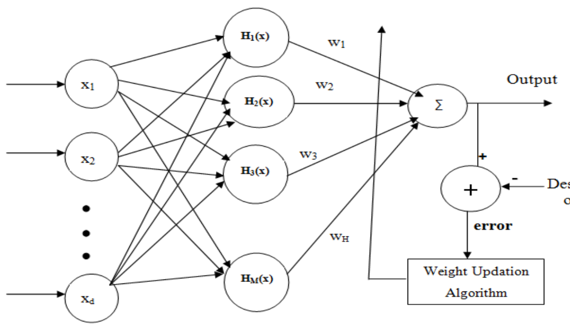

Fig. 1 Architecture of Radial Basis Function Network

The training of an RBF network is done by finding the parameters of radial basis functions and the weights connecting the hidden layer neurons to the output layer neurons. The K-means clustering algorithm is commonly used to train the centers of the basis functions. The width of a basis function is usually set to the distance between its center and the center of its nearest neighbor. The training of the weights between the hidden layer and the output layer is usually done by the gradient descent algorithm. Because there is only one layer to train, the network converges much faster than a MLP network.

III. EXTREME LEARNING MACHINE

Extreme learning machine is a recently introduced learning algorithm for single-hidden layer feed-forward neural networks (SLFNs) which randomly chooses the weights of connections between the input variables and neurons in the hidden layer and the bias of neurons in the hidden layer and analytically determines the output weights instead of iterative tuning [15]. ELM not only has the capability of extremely fast learning and testing speed but also tends to achieve better generalization performance. The main advantage of ELM is that the hidden layer of SLFNs need not be tuned and it can work with a wide range of activation functions including piecewise continuous functions [16-18]. With a given a set of N training dataset $D = (x_i , y_i)$, $i=1$ to N where each x_i is a d dimensional input pattern and y_i is the desired output, activation function for hidden layer nodes, M number of hidden layer nodes and a linear activation function in the output neuron, the output function of ELM for SLFN can be represented as:

$$y_j = \sum_{i=1}^{M} w_{ij} H_i(x) + w_{0j} \tag{9}$$

Where $H_i(x)$ is the output of hidden layer node i and W is the weight vector connecting the hidden layer neurons to output layer neuron. Equation (9) can be written as

$$Y=HW \tag{10}$$

Where H is a $N \times (M+1)$ hidden layer feature mapping matrix in which ith row specifies the hidden layer's output vector for an instance x_i. Equation (10) being a linear system can be solved by

$$W = H^{\Psi} Y = (H^T H)^{-1} H^T Y \tag{11}$$

Where H^{Ψ} is the Moore-Penrose generalized inverse of matrix H .

According to the ridge regression theory, the resultant solution can be more stable by adding a regularization parameter value to the robust least square solution [17]. The use of least squares method with regularization parameter can enhance the generalization performance of ELM in presence of noisy data. Hence the weights of the network can be found by:

$$W = H^{\Psi} Y = (H^T H + \alpha I)^{-1} H^T Y \tag{12}$$

$\alpha > 0$ is the regularization parameter that adds extra cost to the squared norm of the output weights and I is the $H \times H$ identity matrix.

Such improved ELM with regularization parameter is known as Ridge ELM (RELM). RELM actually tries to minimize the loss function $\|HW - Y\|^2 + \alpha\|W\|^2$.

IV. DETAILED STEPS OF STOCK TREND PREDICTION USING RBFN WITH RELM

Step 1: Data collection and preparation

Five years of historical stock index price values of two stock indices (BSE SENSEX and S&P 500) are used in this study. The detail of the data set is given in Table 1. Number of increase and decrease cases in each year in the entire dataset of BSE SENSEX and S&P 500 data set is shown in Table 2. Both the data sets are divided into training and testing sets. For BSE data set the training set consists of 650 patterns and 569 patterns are used for testing and for S&P dataset the training set consists of 650patterns leaving the 583 patterns for testing. The training set comprises nearly 50% samples from each year of the data set.

Table 1 Data set Description

Data Set	Period
BSE SENSEX	4- Jan- 2010 to 31-Dec-2014
S&P 500	4- Jan- 2010 to 31-Dec-2014

Table 2 Number of increase and decrease cases in each year in the entire dataset of S&P 500, BSE SENSEX

Data set	Year	Increase	Decrease	Total
BSE SENSEX	2010	127	100	227
	2011	104	143	247
	2012	140	111	251
	2013	130	120	250
	2014	141	103	244
	Total	642	577	1219
S&P 500	2010	130	97	227
	2011	138	114	252
	2012	132	118	250
	2013	147	105	252
	2014	144	108	252
	Total	691	542	1233

In literature, researchers have used different types of technical indicators for analysis of future movement of stock index values. In this study, total 6 popular technical indicators are chosen as input to the proposed model. The calculation of technical indicators from historical prices is given in table 3.

The direction of daily change in stock price index is categorized as "0" or "1". The direction value 1 represents that index value at time t is greater than time t-1indicating an upward movement and direction value 0 represents that index value at time t is smaller than time t-1indicating a downward movement. These 0 and 1 values are used as class values in the classification model.

Step 2: Data normalization

Originally the six technical indicator values represent continuous values in different ranges. So the input data is scaled in the range 0 to 1 using the min max normalization as follows:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{13}$$

Where y = normalized value, x = value to be normalized
x_{min} = minimum value of the series to be normalized
x_{max} = maximum value of the series to be normalized

Scaling the input data ensures that larger value input attributes does not overwhelm smaller value inputs.

Step 3: Creating Network Structure

RBFN is a single hidden layer neural network with only an output layer. The output layer contains a single neuron to provide the corresponding class value of the given input sample. The normalized values of the six chosen technical indicator values are given as input to the network. So the input layer contains six neurons. There is no weight between the input and hidden layer. The network performance varies based on the number of neurons chosen in hidden layer, radial basis function chosen in the hidden layer and learning technique used. So simulation is done with different number of hidden layer neurons and with different basis functions.

Step 4: Training and testing the Network

The training of an RBF network is done by finding the parameters of the radial basis functions and the weights connecting the hidden layer neurons to the output layer neurons. In this study the model is trained using a ridge extreme learning machine, in which the weights between the hidden and output layer are obtained analytically using a robust least squares solution including a regularization parameter. The regularization parameter value is set through a parameter selection process. But the parameters of the radial basis functions are set before training only once. The K-means clustering algorithm is used to set the centers of the radial basis functions and the width of the function is set to the distance between its center and the center of its nearest neighbor. Then the trained network with derived parameters is applied on the test data. Finally the class values of the training and testing samples are obtained by comparing the weighted sum of the expanded inputs with a specified threshold value.

Step 5: Performance Evaluation

The performance of the model is evaluated based on classification accuracy and F-measure value. Computation of these evaluation measures are done by estimating Precision and Recall values. In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class. Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (14)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (15)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \quad (16)$$

$$F-measure = \frac{2 \times Precision \times Recal}{Precision + Recal} \quad (17)$$

V. EXPERIMENTAL RESULT ANALYSIS

To measure the generalization ability of the network initially the dataset is divided in to a single train and test set. Then simulation is done by passing each set of training and testing data to the individual models for 20 times. The average performance out of these 20 runs has been reported for both the dataset. The performance of the proposed model depends on different factors like the number of hidden layer neurons, radial basis function chosen in hidden layer, value of regularization parameter, input space size and so on.

Initially the number of neurons of the network is set to 10 and the network is built with Gaussian basis functions. Through a number of simulations value of the regularization parameter is set to 0.06 for BSE SENSEX dataset and 0.04 for S&P500 dataset. The network is trained independently 20 times using back propagation, ELM and RELM approach. The mean accuracy and F-measure obtained, out of the 20 independent runs are reported in Table 4 for both the data set. The RBFN trained using RELM approach shows clearly the superior performance compared to the other two learning strategies. Then the performance of the network is observed with seven different radial basis functions as shown in table 5. From the analysis of the results, it is found that the network provides better results with Gaussian and inverse multiqudratic radial basis functions. Fig. 2,3,4,5 show the testing accuracy and f measure value of the network with different set of hidden layer neurons and with Gaussian and inverse multiqudratic radial basis functions. The testing accuracy of the network with different hidden layer nodes is given in table 6. Analyzing the performance of the proposed model with different hidden layer neurons ranging from 10 to 90, it is observed that, the model provides better result with 80 neurons for both the dataset.

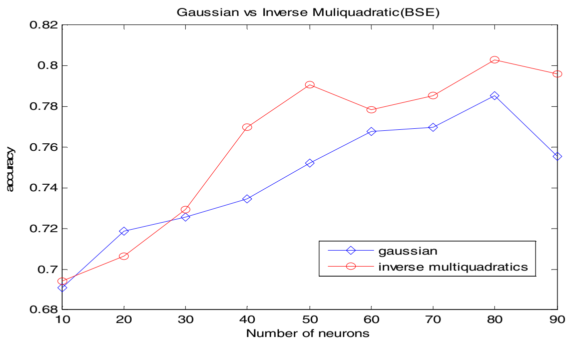

Fig. 2 Testing accuracy of RBFN with different number of hidden layer neurons and Gaussian, inverse multiquadratic basis function for BSE SENSEX dataset

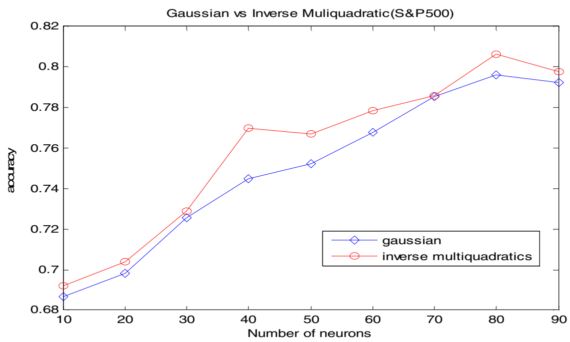

Fig. 3 Testing accuracy of RBFN with different number of hidden layer neurons and Gaussian, inverse multiquadratic basis function for S&P500 dataset

Table 3 Technical indicators

Technical Indicator	Equation
Simple Moving Average (MA)	$MA_n = \frac{1}{n}\sum_{i=1}^{n} cp(i)$
Moving Average Convergence and Divergence (MACD)	$MACD = EMA_{12} - EMA_{26}$ $EMA(i) = (CP(i) - EMA(i-1)) \times Multiplier + EMA(i-1)$ $where \quad Multiplier = 2/(no\ of\ days\ to\ be\ considered + 1)$
Stochastic KD	$K\%(i) = \frac{cp(i) - L_n}{H_n - L_n} \times 100$ $D\%(i) = (K\%(i-2) + K\%(i-1) + K\%(i))/3$
Relative Strength Index (RSI)	$RSI = 100 - \frac{100}{1 + RS} \quad where \quad RS = \frac{Average\ of\ n\ day's\ up\ closes}{Average\ of\ n\ day's\ down\ closes}$
Larry William's R%	$R\%(i) = \frac{H_n - cp(i)}{H_n - L_n} \times 100$
Where cp(i) is the closing price, L_n is the lowest price of last n days, H_n is the highest price of last n days.	

Table 4 Performance comparison of RBFN with different learning algorithms

Data set	Learning Algorithm	Training		Testing	
		Accuracy	F- measure	Accuracy	F- measure
BSE SENSEX	RELM	0.7354	0.7651	0.6872	0.7063
	ELM	0.7185	0.7529	0.6721	0.6915
	Back Propagation	0.7211	0.7358	0.6689	0.6728
S&P 500	RELM	0.6862	0.7444	0.7170	0.7410
	ELM	0.6777	0.7399	0.6762	0.7132
	Back Propagation	0.6571	0.7187	0.6592	0.7010

Table 5 Performance comparison of RBFN with different radial basis functions

Dataset	No. of neurons	Radial Basis Function	Training		Testing	
			Accuracy	F- measure	Accuracy	F- measure
BSE SENSEX	10	**Gaussian**	0.7354	0.7651	0.6872	0.7063
		Multi quadrics	0.6092	0.6532	0.6239	0.6503
		Inverse multi quadrics	**0.7369**	**0.7622**	**0.6942**	**0.7061**
		Logistic	0.5431	0.7039	0.5079	0.6737
		Thin plate spline	0.5462	0.7053	0.5114	0.6752
		Cubic	0.5431	0.7039	0.5079	0.6737
		Linear	0.6123	0.7169	0.5641	0.6876
S&P 500	10	**Gaussian**	0.6862	0.7444	0.7170	0.7410
		Multi quadrics	0.6385	0.7131	0.6278	0.6860
		Inverse multi quadrics	**0.6969**	**0.7497**	**0.7204**	**0.7457**
		Logistic	0.5708	0.7267	0.5489	0.7087
		Thin plate spline	0.5769	0.7291	0.5592	0.7128
		Cubic	0.5708	0.7267	0.5489	0.7087
		Linear	0.6108	0.7356	0.5849	0.7071

Table 6 Testing accuracy of RBFN with different hidden layer neurons

Data Set	Basis functions	10	20	30	40	50	60	70	80	90
BSE SENSEX	**Gaussian**	0.6872	0.7188	0.7258	0.7346	0.7522	0.7680	0.7698	**0.7856**	0.7557
	Inverse multi quadrics	0.6942	0.7065	0.7293	0.7698	0.7909	0.7786	0.7856	**0.8032**	0.7961
S&P 500	**Gaussian**	0.7170	0.7203	0.7258	0.7448	0.7552	0.7680	0.7856	**0.7979**	0.7925
	Inverse multi quadrics	0.7204	0.7256	0.7290	0.7698	0.7669	0.7786	0.7858	**0.8062**	0.7976

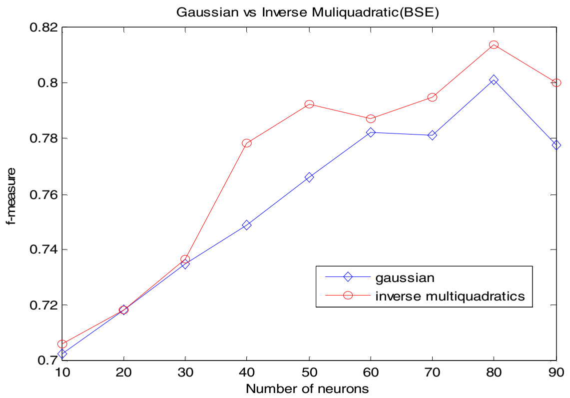

Fig. 4 F-measure of RBFN with different number of hidden layer neurons and Gaussian, inverse multiquadratic basis function for BSE SENSEX dataset

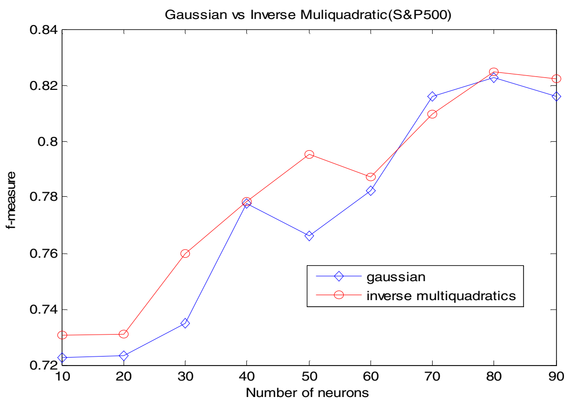

Fig. 5 F-measure of RBFN with different number of hidden layer neurons and Gaussian, inverse multiquadratic basis function for S&P500 dataset

VI. CONCLUSION

This paper presents a comparative study of effect of the different basis functions over a radial basis function network (RBFN) for predicting the future trend of financial time series data. The RBFN is trained by a ridge extreme learning machine to capture the relationship between the future trend and the past values of the technical indices. The model is also compared with back propagation learning and simple extreme learning algorithm. Instead of using only Gaussian basis function, six other basis functions are implemented for comparing the performance of the network model. The two benchmark financial data sets like BSE SENSEX and S&P500 stock index dataset are considered for validating the performance of the model. Summarizing the results, the following inference is drawn:

- A comparison of back propagation, ELM and RELM learning over RBFN model shows clearly the superior performance of RBFN trained using RELM approach.
- Again from the experimental result analysis it is clearly apparent that the proposed model provides superior classification accuracy and F- measure value with the inverse multi quadratics basis functions compared to other radial basis functions.
- Choosing basis function for any network is application dependent. Hence not only using one basis function, evaluating all recognized basis functions suitable for RBFN is advantageous for any application.

REFERENCES

[1] Y. Kara, M. A. Boyacioglu, O. K. Baykan, "Predicting Direction of Stock Price Index Movement using Artificial Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange", Expert Systems with Applications, 38, pp. 5311-5319, 2011.

[2] J. Patel, S. Shah, P. Thakkar, K. Kotecha, "Predicting Stock and Stock Price Index Movement using Trend Deterministic Data Preparation and Machine Learning Techniques", Expert Systems with Applications, 42, pp. 259-268, 2015.

[3] A. Wysocki, M. Lawrynczuk, "An Investment Strategy for the Stock Exchange using Neural Networks", Proceedings of Federated Conference on Computer Science and Information Systems, pp. 183-190, 2013.

[4] H. Tahersima, M. H. Tahersima, M. Fesharaki, "Forecasting Stock Exchange Movements Using Neural Networks: A Case Study", International Conference on Future Computer Sciences and Application, pp.123-126, 2011.

[5] F. A. Oliveira, C. N. Nobre, "The Use of Artificial Neural Networks in the Analysis and Prediction of Stock Prices", IEEE International Conference on Systems, Man and Cybernetics (SMC), pp.2151-2155, 2011.

[6] M. P. Naeini, H. Taremian, H. B. Hashemi, "Stock Market Value Prediction Using Neural Networks", International Conference on Computer Information Systems and Industrial Management Applications (CISIM), pp.132-136, 2010.

[7] W. Shen, X. G. C. Wub, D. Wu, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm", Knowledge Based Systems, 24, pp.378-385, 2011.

[8] H. Feng , H. Chou, "Evolutional RBFNs prediction systems generation in the applications of financial time series data", Expert Systems with Applications, 38, pp. 8285-8292, 2011.

[9] S.Oh, W. Kim, W. Pedrycz, S. Joo, "Design of K-means clustering-based polynomial radial basis function neural networks (PRBF NNs) realized with the aid of particle swarm optimization and differential evolution", Neuro computing, 78, pp.121-132, 2012.

[10] H. Sarimveis, P. Doganis, A. Alexandridis, "A classification technique based on radial basis function neural networks", Advances in Engineering Software, 37, pp. 218-221, 2006.

[11] M. Korurek, B. Dogan, "ECG beat classification using particle swarm optimization and radial basis function neural network", Expert Systems with Applications, 37,pp. 7563–7569, 2010.

[12] C. Harpham, C. W. Dawson, "The effect of different basis functions on a radial basis function network for time series prediction: A comparative study", Neurocomputing, 69, pp. 2161–2170, 2006.

[13] R. Dash, P. K. Dash, R. Bisoi, "A Self Adaptive Differential Harmony Search based Optimized Extreme Learning Machine for Financial Time Series Prediction", Swarm and Evolutionary Computation,19, pp. 25-42, 2014..

[14] G. Huang, Q. Zhu, C. Siew, "Extreme learning machine: Theory and applications", Neuro computing, 70, pp. 489-501, 2006.

[15] G. B. Huang, D. H. Wang, Y. Lan, "Extreme Learning machines: A survey", Int. J. Mach. Learn. & Cyber, 2, pp. 107-122, 2011.

[16] G. B. Huang, H. Zhou, X. Ding, R. Zhang, "Extreme Learning Machine for Regression and Muliclass Classification", IEEE Transactions on Systems, Man, and Cybernatics, 42(2), pp.513-529, 2012.

[17] A. B. Crespo, P. J. G. Laencina, J. L. S. Gomezc, "Neural architecture design based on extreme learning machine", Neural Network, 48, pp.19-24, 2013.

[18] M. Luo, K. Zhang, "A hybrid approach combining extreme learning machine and sparse representation for image classification", Engineering Applications of Artificial Intelligence, 27, pp.228-235, 2014.