BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE)
ISSN (P): 2348-0513, ISSN (E): 2454-471X,
Vol. 4, Issue 7, Jul 2016, 77-88

DEMAND FORECASTING USING ARTIFICIAL NEURAL NETWORKS OPTIMIZED BY ARTIFICIAL BEE COLONY

JALAL A. SULTAN & RAGHAD M. JASIM

Department of Mathematics, College of Education Pure Sciences,

University of Mosul and Kirkuk, Iraq

ABSTRACT

Bee colony algorithms are new swarm intelligence techniques inspired from the smart behaviors of real honeybees in their foraging behavior**.** This paper examines the use of Artificial Bee Colony (ABC) to train a multi-layer feed forward neural network for demand forecasting. We use in this paper two data sets represent the weekly demand data for cement and towels, which have been outfitted by the Sorthern General Company for Cement and General Company of prepared clothes respectively. The results showed superiority of trained neural networks using ABC on neural networks trained using error back propagation because their ability to escape from local optima.

KEYWORDS: Artificial Bee Colony, Artificial Neural Network, Demand Forecasting, Evolutionary Algorithms, Weight Optimization

INTRODUCTION

Knowing future better has attracted many people for thousands of years. The forecasting methods vary greatly and will depend on the data availability, the quality of models available, and the kinds of assumptions made, amongst other things. Generally speaking, forecasting is not an easy task and therefore it has attracted many researchers to explore it.

Artificial neural network (ANN) has found increasing consideration in forecasting theory, leading to successful applications in various forecasting domains including economic, business financial [1] and many more. ANN can learn from examples, recognize hidden pattern in historical observations and use them to forecast future values. In addition to that, they are able to deal with incomplete information or noisy data and can be very effective especially in situations where it is not possible to define the rules or steps that lead to the solution of a problem.

Back Propagation (BP) algorithm is one of the most effective methods to train ANN. Any continuous function in a closed interval can be approximated by using a BP ANN with one hidden layer. For any complicated system, if its samples of input and output are enough, a BP ANN model that reflects the relationships between the input and output variants can be constructed after repeated learning and training. Therefore, BP ANN has very strong capabilities of nonlinear modeling and analysis for huge and complex system [2].

However, since the initial interconnecting weights of ANN are often stochastically given, the learning times and final interconnecting weights of the network are therefore changed for different times of training. That is to say, the trained network is not unique and sometimes the network possibly plunges into partial minimum. In addition, the blindness of the determination of initial interconnecting weights always results in too many training times and slow convergence. These shortages of ANN seriously impact its precision of modeling and effects of application. It is quite necessary to optimize and improve the weights of ANN [3].

In this paper, an artificial bee colony (ABC) has been used to optimize the weights of ANN for demand forecasting, and presents an integrated model that combined advantages of both ANN and ABC. As a case study, this model is applied to predict forecast weekly demand from packed cement and towels, which has been outfitted by the Northern General Company for Cement and General Company of prepared clothes respectively.

The paper is organized as follows: Section 2 briefly introduces the basic foundations of ANN, and the artificial bee colony for training ANN has been proffered in Section 3. In Section 4, Experiments and discussion have been created for an industrial case study. The Comparison between ABC and BP is given on Section 5. The conclusion of this study has been presented in Section 6.

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are one of the most dynamic research areas during the contemporary period, which has attracted the attention of many experts of various scientific fields. Recent research activities regarding to the artificial neural networks indicated that this method is a very powerful tool to solve complicated problems under consideration in all engineering fields [4]. Neural network, similar to real human brain, has the required ability for learning and are able to utilize the acquired new experiences from new and similar affairs. Although ANNs are not comparable to the real human brain system, these networks equipped special features which make them privileged in some applications-abilities such as separation of patterns and its amenability to learn the networks by linear and nonlinear mapping wherever the learning is required. Among the artificial neural networks' features and abilities, it can be referred to some cases including its amenability to learning and its adaptability (very well to changing or new levels of information) to available information, the capacities of generalization, massively parallel processing the network inputs, consequently to accelerate processing time, high fault (noise) tolerance and so on [5].

The configuration of the whole network changes frequently and adapts very well to changing or new levels of information. Moreover, this network is massively parallel, robust, fault tolerant and amenable to learning.

Artificial neural networks are a system equipped with parallel processor of accumulation-mass-massive information and they function similar to a real human brain. The following principles represent the ANNs basics:

- Data are processed in the singular units named "node".
- The signals are transferred between nodes through connection lines.
- The weight attributed to each connection lines indicated the communication capacity of that line.
- Each node typically has activation and transfer functions, in order to specify output signals from input data of the network [6].

The structure of artificial neural networks is introduced by connection patterns between nodes, a method determining the connection weights and activation function. The typical structure of artificial neural network has been usually formed by input layers, middle (hidden) layers and output layers (Figure 1). Input layer is a transfer layer, a means for providing data. The last layer (output layer) includes the predicted values by the network, whereby it represents the output of the model. Middle (hidden) layers consist of some processing nodes, where data is processed [5].

In this study, we have used neural network with one hidden layer. In time series analysis, the determination of number of input nodes, which are lagged observations of the same variable, plays a crucial role as it helps in modeling the autocorrelation structure of the data. The determination of number of output nodes is relatively easy. In this paper, one output node has been used and multi-step ahead forecasting has been done using the iterative procedure as used in Box-Jenkins method. This involves use of forecast value as an input for forecasting the future value. It is always better to select the model with small number of nodes at hidden layer as it improves the out of the sample forecasting performance and also avoids the problem of over fitting [7].

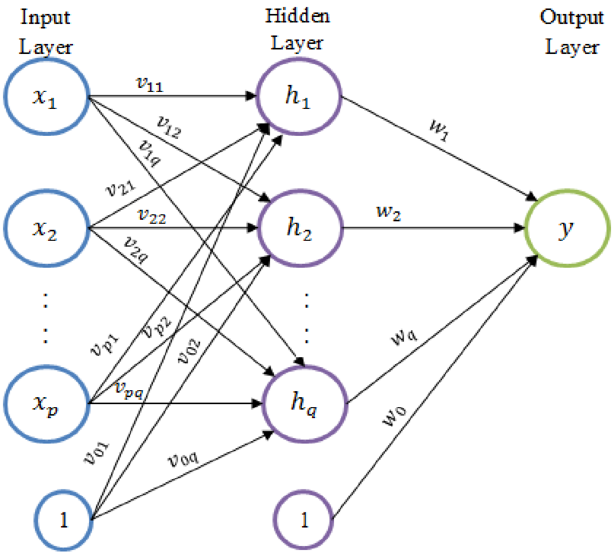

Figure 1: Structure of Neural Network

The general expression for the final output value yin a multi -layer feed forward neural network is given by

$$y = g\left(\sum_{j=0}^{q} w_j f\left(\sum_{i=0}^{p} v_{ij} x_i\right)\right) \quad (1)$$

where f and g denote the activation function at hidden and output layer respectively, p is number of input nodes, q is the number of hidden nodes, ($v_{ij}$is the weight attached to the connection between ith input node to the jth node of hidden layer, w_j is the weight attached to the connection from jth hidden node to the output node and Xi is the ith input of the model. Each node of the hidden layer receives the weighted sum of all inputs including a bias term for which the value of input variable will always take a value one. This weighted sum of input variables is then transformed by each hidden node using the activation function *f* which is usually nonlinear sigmoid function. In a similar fashion, the output node also receives the weighted sum of the output of all hidden nodes and produces an output by transforming the weighted sum using its activation function *g*. In time series analysis, *f* is often chosen as logistic function and *g*as an identity function. For *p* input nodes, *q* hidden nodes, one output node and biases at both hidden and output layer, the total number of parameters (weights) in a three layer feed forward neural network is $q(p + 2) + 1$ [3].

For a univariate time series forecasting problem, past observations of a given variable serves as the input variables. The neural network model attempt to map the following function

$$y_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-p+1}, w) + \varepsilon_{t+1} \quad (2)$$

Where y_{t+1} pertains to the observation at time t +1, p is the number of lagged observation, w is the vector of network weights, and ε_{t+1} is the error term at time t +1. Hence, the neural network acts like non-linear autoregressive model [7].

ARTIFICIAL BEE COLONY

By simulating the forging behavior of bee colonies, artificial bee colony (ABC) algorithm, which is a swarm intelligence-based optimization algorithm, was proposed by Karaboga in (2007) for numerical function optimization [14]. The main steps of ABC algorithm can be described as follows.

Initialization

Repeat

Employed bee stage: Place the employed bees on the food sources in the memory.

Onlooker bee stage: Place the onlooker bees on the food sources in the memory.

Scout bee stage: Send the scout bees to the search area for discovering new food sources.

Until (conditions are satisfied)

In ABC algorithm, the colony consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony is employed bees, and the other half is onlooker bees. The employed bees explore the food source and send the information of the food source to the onlooker bees. The onlooker bees choose a food source to exploit based on the information shared by the employed bees. The scout bee, which is one of the employed bees whose food source is abandoned, finds a new food source randomly. The position of a food source is a possible solution to the optimization problem. Denote the food source number as SN, the position of the ith food source as xi (i =1… SN), which is a D dimensional vector [12, 13].

In ABC algorithm, the ith fitness value i fit for a minimization problem is defined as [13]:

$$\mathrm{fit_i} = \begin{cases} 1/(1+\mathrm{f_i}) & \mathrm{if} \; \mathrm{f_i} \geq 0 \\ 1+\mathrm{abs}(\mathrm{f_i}) & \mathrm{if} \; \mathrm{f_i} < 0 \end{cases} \tag{3}$$

Where $\mathrm{f_i}$ is the cost value of the ith solution? The probability that food source being selected by an onlooker bee is given by:

$$\mathrm{p_i} = \frac{\mathrm{fit_i}}{\sum_{i=1}^{SN} \mathrm{fit_i}} \tag{4}$$

A candidate solution from the old one can be generated as:

$$\mathrm{v_{ij}} = \mathrm{x_{ij}} + \phi_{ij}(\mathrm{x_{ij}} - \mathrm{x_{kj}}) \tag{5}$$

Where $k \in \{1,2,\dots,SN\}$, $k \neq i$ and $j \in \{1,2,\dots,D\}$ are randomly selected indices, $\phi_{ij} \in [-1, 1]$ is a uniformly distributed random number. The candidate solution is compared with the old one, and the better one should be remained [14].

If the abandoned food source is xi, the scout bee exploits a new food source according to:

$$\mathrm{x_{ij}} = \mathrm{x_{min,j}} + \mathrm{rand}(0,1)\left(\mathrm{x_{max,j}} - \mathrm{x_{min,j}}\right) \quad (6)$$

Where $x_{max,\mathrm{j}}$ and $x_{min,\mathrm{j}}$ are the upper and lower bounds of the jth dimension of the problem's search space [12].

USING ABC TO OPTIMIZE ANN

The key problem to do the time series forecasting using Artificial neural network is restructure the measured data and establish the function relationship between preliminary observation data and subsequent data.

Training process begins by entering data into the network, these data composed of two parts: the first part represents the independent set (Input variables) and the second part represents the dependent set (target variables), these two sets together are modeled by a matrix, which represents the inputs data of the network. The matrix form can be written as follows:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_p & T_{p+1} \\ x_2 & x_3 & \cdots & x_{p+1} & T_{p+2} \\ \vdots & \vdots & \ddots & \vdots & \\ x_{n-p} & x_{n-p+1} & \cdots & x_{n-1} & T_n \end{bmatrix} \quad (3)$$

It should be noted that all the input training data should be normalized first, suitable embedding dimension should be calculated; and then training and testing samples can be established and normalized. The normalized method is shown in eqn. (4).

$$\hat{\mathrm{x}} = 2 * \left(\frac{\mathrm{x} - \mathrm{x_{min}}}{\mathrm{x_{max}} - \mathrm{x_{min}}}\right) - 1 \quad (4)$$

Where x is measured data and y is normalized data, x max and x min are the maximum and minimum values of the measured data. In many cases, the prepared data for the training contains high values. So, it is always advisable to normalize the inputs and outputs of the network to be within the range [-1, 1], and this normalization has many advantages such as:

- All inputs become convergent values and therefore do not dominate the entrance to another.
- The normalization values make training faster [7]. The following steps can describe the general steps of training ANN using ABC.

Create Initial Food Sources

In order to optimize the initial weights of the neural network, all layers of feed forward neural network weights and bias should be encoded as food source structure (Figure 2.). Due to the neural network weights and bias parameters are a complexity continuous optimization problem, it is popular to use traditional real encoding method, and this also will affect the accuracy and computational efficiency of evolution algorithm, because this encoding method suits for large span and high precision artificial bee colony. All layers of neural network weights are encoded to ABC food source in accordance with order. Each food source consists of a vector of N weights were generated randomly from the uniform distribution in the range [-1, 1]. The total number of weights of *(p, q, 1)* ANN model is $q(p+2)+1$.

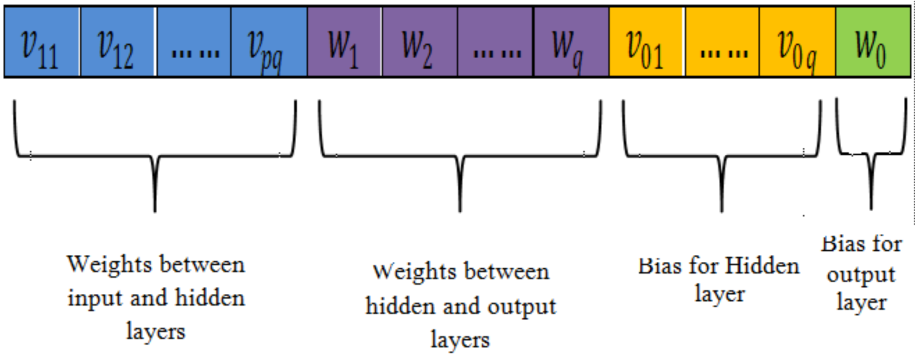

Figure 2: The Food Source Structure

Fitness Evaluation

The fitness function value is the reciprocal value of error sum of squares. The fitness function is considered as formula (5).

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{5}$$

Search Mechanism

It is well known that both the exploration and exploitation abilities are necessary for the population based algorithms. How to balance these two abilities to achieve good optimization performance is very important.

In ABC algorithm, the employed bee stage represents the exploration ability of the algorithm, and the onlooker bee stage represents the exploitation ability of the algorithm. The search equation proposed in ABC algorithm is good at exploration but poor at exploitation, so that it will affect the convergence speed of the algorithm.

Inspired of PSO [5], in order to improve the exploitation ability of ABC algorithm, take the advantages of the search equation in PSO, the global best solution will be considered in the new search equation in the onlooker bee stage. The modified search equation in onlooker bee stage is described as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \vartheta_{ij}(y_j - x_{ij}) \tag{6}$$

Where k $\in$ {1, 2... SN} is a random selected index which is different from i,j $\in \{1,2,\dots,D\}$ is a random selected index, yj is the jth element of the global best solution, $\phi_{ij} = (-1,1)$, $\vartheta_{ij} \in (0,1.5)$, are both uniformly distributed random numbers.

Differential evolution (DE) [4] is a population based algorithm to function optimization, whose main strategy is to generate a new position for an individual by calculating vector differences between other randomly selected members in the population. “DE/current-to-rand/1” is a variant DE mutation strategy, which can effectively maintain population diversity according to randomness of the search equation. Motivated by “DE/current-to-rand/1”mutation strategy and based on the property of ABC algorithm, a new search equation in employed bee stage is proposed as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + c_{ij}(x_{r1j} - x_{r2,j}) \tag{7}$$

Where$\phi_{ij}, c_{ij} \in (-1,1)$andi $\in \{1,2,\dots,SN\}$, j $\in \{1,2,\dots,D\}$, r1, r2 $\in \{1,2,\dots,SN\}$andr1, r2 $\neq$ i, $\phi_{ij}, c_{ij}$are both negativeor both positive, which can keep the search direction the same.

In general, inspired by DE and PSO, the new search equation and search mechanism are proposed to balance the exploration ability and exploitation ability in ABC algorithm. In the employed bee stage, search equation (7) is used to keep the exploration ability of ABC algorithm; while, in the onlooker bee stage, search equation (6) is employed to increase the exploitation ability of the algorithm. Figure 3 shows how to train ANN using ANN.

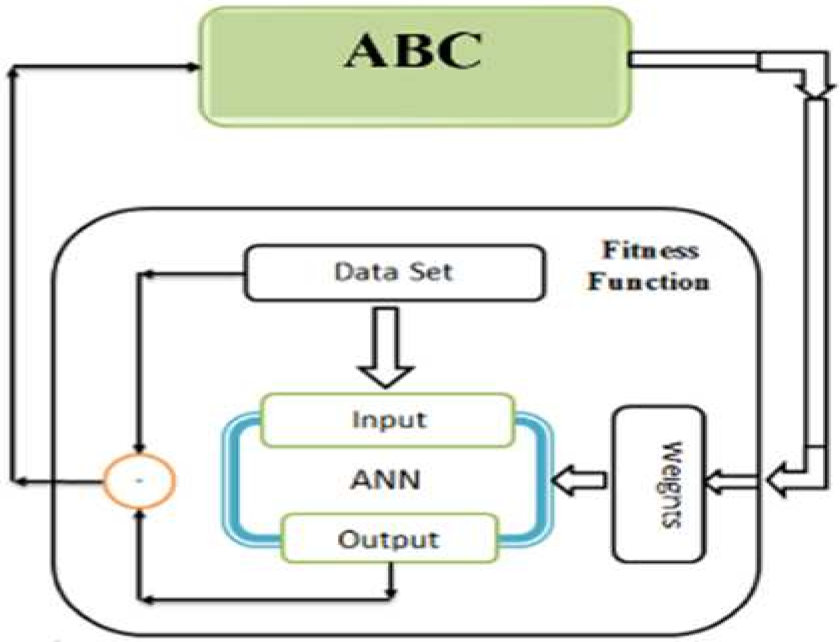

Figure 3: How to Train ANN Using ABC

Stopping Condition

The purpose of training artificial neural networks using ABC is to get balance between the correct responses to training samples as well as good responses for new entry samples (i.e. the balance between remembering and forecasting). This training process is not supposed to continue until we get to the smallest value of MSE, but it can determine by a stopping condition.

So, the data were randomly split into two groups during the training process (training group and testing group), these two groups are independent and there is no any correlation between them.

The algorithm will continue in training and improving the weights, it depends choose weights on the output of the use of the test set error, in each iteration, the error value of the square of the group training and a test account and as long as the error to set the test decreases the training process will continue, and when it begins to increase the network remembering the training samples and after so it will lose its ability to predict, and at this point ends the training process. Figure 4.showsthe best time to stop training process.

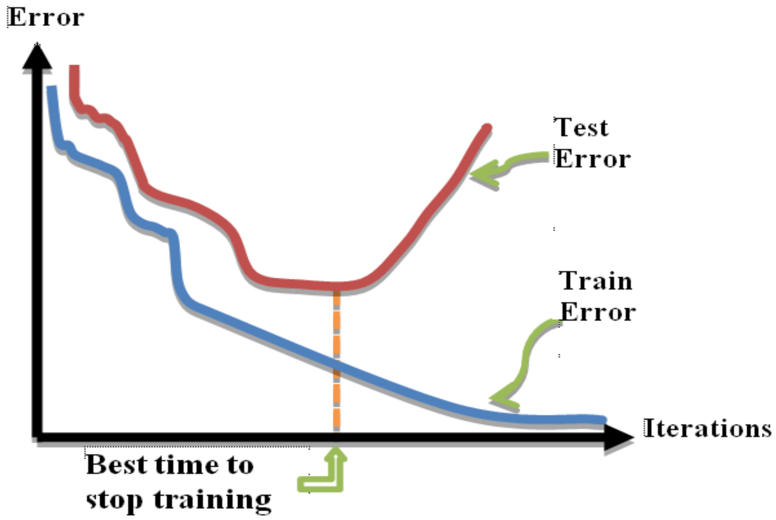

Figure 4: The Best Time to Stop Training

In this paper, we stopped training process at the end of the evolutionary algorithm iterations and then choose the specific weights group that gave less square error for the group to be a test weights

EXPERIMENTS AND DISCUSSIONS

Data

Two data sets—the packed cement data and the towels data—are used in this study to demonstrate the effectiveness of the ANN optimized by ABC to forecast the demand.

The packed cement series we consider contains the weekly demand data for packed cement from 02.01.2007 to 05.01.2013, giving a total of 315 observations. The data are plotted in Figure 5.

The towels series contains the weekly demand data for hand towels, which have been produced by the General Company of prepared clothes. The data are plotted in Figure 6. The data set has 104 observations, corresponding to the period of 12.01.2011 to 09.01.2013.

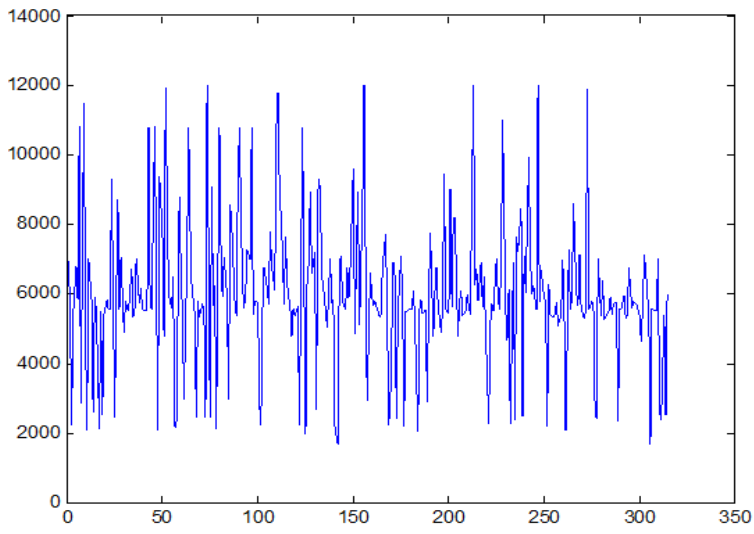

Figure 5: Time Series plot for Packed Cement

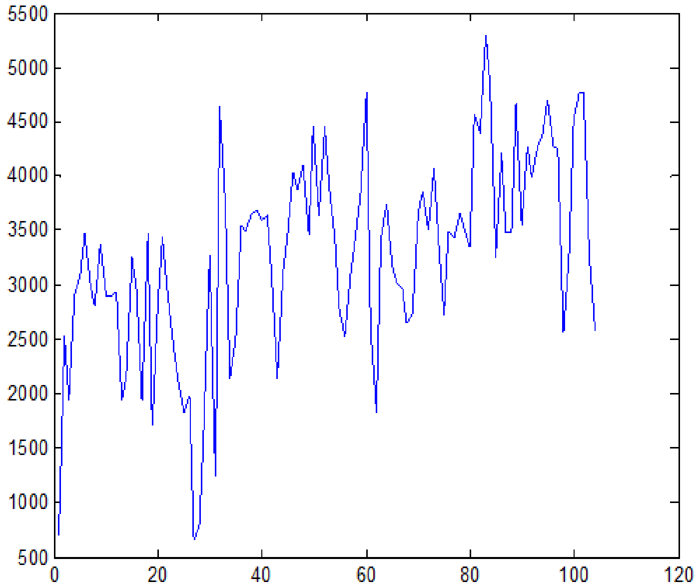

Figure 6: Time Series Plot for Towels

To assess the forecasting performance of different models, each data set is divided into two samples of training and testing. The training data set is used exclusively for model development and then the test sample is used to evaluate the established model. The data compositions for the two data sets are given in Table 1.

Table 1: Sample Size Sample Compositions in Two Data Sets

Series	Sample Size	Training Set Size	Test Set Size
Cement	315	240	75
Towels	104	78	26

The Network Architecture

As indicated earlier, a three layer feed forward neural network model has been used for this paper, also the logistic and identity function have been used as activation function for the hidden nodes and output node respectively. In designing ANN for forecast, one must determine the following variables:

- number of input nodesp
- number of hidden nodesq

The selection of these parameters is basically problem dependent. Although there exists many different approaches such as the pruning algorithm, the polynomial algorithm and genetic algorithm for finding the optimal architecture of an ANN, these methods are usually quite complex in nature and are difficult to implement. In this study, genetic algorithm is used to optimize the structure each network [10]. The maximum number of the nodes in the input layer will be 10 and the maximum number of the nodes in the hidden layer will be 15. To determine the optimal structure for each series in terms of the number of the input nodes and the number of the nodes in the hidden layer, it were used the learning algorithm of the network and transfer function, the capacities of the optimizing genetic algorithm available in Neuro Solutions Software. The best structure among many runs for packed cement data was [7, 9, 1] and for towels was [10, 5, 1].

Training and Forecasting

Artificial bee colony is a population based search technique that finds the best solution in the search space. The error surfaces for neural networks are much irregular and contain several irregular geometries. Hence, for this study (NColony=100, Ngen=1000, Limit=30) have been used for training the neural network. As indicated earlier, each food source is associated with two parameters: position and velocity. Initially, we randomly distribute food sources in several nodes of the search space. Each data point is associated with a location and a velocity. All these nodes in the domain represent a neural network model. The optimization experiment was run 50 times with different initial values for position and velocity in order to record average and the best value for performance measures.

The forecasting ability has been assessed with respect to two common performance measures: the mean squared error MSE and the mean absolute deviation MAE. The mean square error measures the overall performance of namely 6 weeks ahead and 10 weeks ahead respectively. Each table from (2-3) indicates the best MSE and MAE based on 50 runs for different training algorithms relating to a forecasting data series, which are. The *MSE* and *MAE* were observed to assess the consistency of the performance of different algorithms.

Table 2: Forecasting Comparison for Packed Cement Data

Method	6 Points Ahead		10 Points Ahead	
	MSE	***MAE***	***MSE***	***MAE***
ABC	$7.992*10^5$	764.64	$1.248*10^6$	965.5
Leven berg-Marquardt	$9.382*10^5$	854.44	$1.566*10^6$	1068.5

Table 3: Forecasting Comparison for Towels Data

Method	6 Points Ahead		10 Points Ahead	
	MSE	***MAE***	***MSE***	***MAE***
ABC	1868.27	108.31	2801.59	127.80
Leven berg-Marquardt	2169.65	113.19	3060.82	130.33

DISCUSSIONS

Results clearly show that both ANN based training algorithm provide better forecasting ability with respect to the best MSE and MAE across all the two forecasting horizons. In packed cement forecasting, the best MSE value of 7.992*105was obtained in case of ABC as compared to the best MSE value of 9.382*105 for Leven berg-Marquardt algorithm for 6 weeks period. With the ABC trained algorithm, we obtained 17.39 %, and 25.27 % decrease in terms of MSE over the standard back-propagation trained neural network for 6 weeks and 10 weeks ahead forecasts respectively. In terms of MAE, ABC provided 10.5098% and 9.63 % decrease as compared to usual back propagation algorithm for 6 weeks and 10 weeks respectively.

In towels forecasting, the best MSE value of 1868.27was obtained in case of ABC as compared to the best MSE value of 2169.65 for Leven berg-Marquardt algorithm for 6 weeks period. With the ABC trained algorithm, we obtained 16.13% and 9.25% decrease in terms of MSE over the standard back-propagation trained neural network for 6 weeks and 10 weeks ahead forecasts respectively. In terms of MAE, ABC provided 4.50% and 1.941% decrease as compared to usual back propagation algorithm for 6 weeks and 10 weeks respectively.

Further, neural network model based on ABC uniformly provided better forecasting accuracy than Leven berg-

Marquardt with respect to mean and the best value of performance measures across all the two forecasting horizons (As seen in Figure 7). It has been observed that the mean values of MSE and MAE were always high for ABC based training algorithm as compared to standard back propagation training algorithm. The prediction ability of all the training algorithms deteriorated with the increase in forecasting periods which is obvious. Because of this reason neural network based models are preferred for short term forecasting.

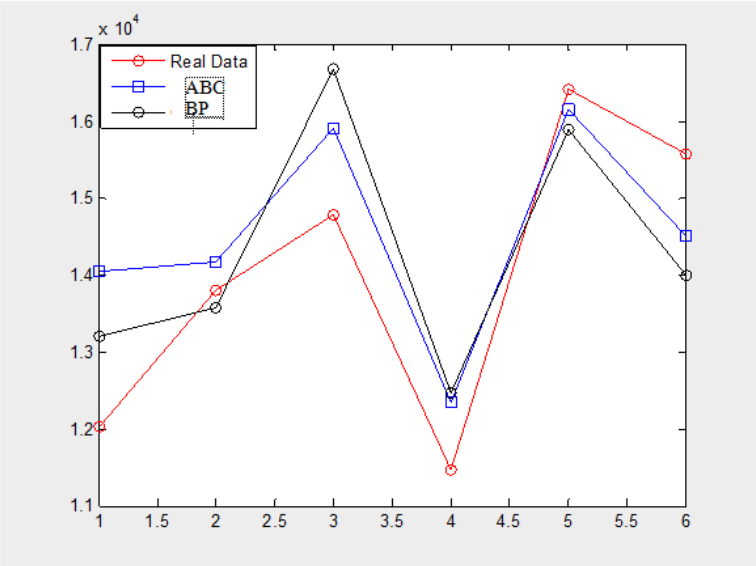

Figure 7: ABC and BP Forecasting of Packed Cement for 6 Weeks

CONCLUSIONS

Time series analysis and demand forecasting are active research areas over the last few decades. The demand forecasting is fundamental to many decision processes and hence the research for improving the effectiveness of forecasting models and algorithms has never stopped. In this study, an effort was made to assess the forecasting ability of ABC to train a multi-layer feed forward neural network with a real demand time series data. ABC along with Leven berg-Marquardt algorithm was used to train the neural network. Results showed that ABC based training algorithm provided better forecasting ability with respect to the best MSE and *MAE* across all the two forecasting horizons and for each data series.

REFERENCES

1. C.H. Chen, "Neural networks for financial market prediction", IEEE World Congress Computational Intelligence,1994, 27 (2) 1199–1202.
2. S. Sarle, and S. Warren, Neural Networks and Statitical Models. Proceeding of The Nineteenth Annual SAS Vsers Group Internaternational Conference, SAS Institute Inc., Cary, NC, USA, 1994.
3. K. Girish, T. Parimala, and K. Ruppa, "PSO Based Neural Network for Time Series Forecasting",Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, 2009, pp. 1422–1427.
4. K.L. Klaussen and J.W. Uhrig, "Cash soybean price prediction with neural networks", NCR-Z34 Conf on Applied Commodity Analysis, Price Forecasting, and Market Risk Management Proc., Chicago, 1994 56-65.

5. M. Cottrell, B. Girard, Y. Girard, M. Mangeas, and C. Muller, "Neural modeling for time series: a statistical stepwise method for weight elimination", IEEE Trans. Neural Networks, 6, 1995, 1355–1364.

6. R.R. Trippi and E. Turban," eds. Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real- World Performance", (Probus, Chicago, 1993).

7. G.Zhang, B.Patuwo, and M. Hue, "Forecasting with artificial neural networks: The state of the art", International Journal of Forecasting, vol.14, 1998.

8. J. Kennedy, and R.C. Eberhart, "Particle Swarm Optimization", Proceedings of IEEE, 1995, pp 1942–1948.

9. J. Kennedy, R. Eberhart, and Y. Shi, Swarm Intelligence. New York: Morgan Kaufmann, 2001.

10. D. Goldberg, Genetic Algorithms in Search Optimization & Machine Learning. Addison-Wesley. Reading, 1989.

11. D . Fogel, Evolutionary Computation: The Fossil Record. Piscataway NJ: IEEE Press, 1998.

12. D. Karaboga, B. Akay, (2010)."Amodified Artifical Bee colony algorithm for real-Parameter optimization" Information Sciences and Elservier,doi:10.1016/j.ins.2010.07.015.

13. D. Karaboga, B. Basturk, (2008). "On the performance of artificial bee colony (ABC) algorithm", Applied Soft Computing, Vol. 8, pp. 687-697.

14. D. Karaboga, B. Basturk, (2007), "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", J. Glob. Optim. Vol. 39, pp.459-471.

15. D. Srinivasan, and T.H.Seow, "Evolutionary Computation", In: CEC 2003.December 8-12. vol. 4. pp. 2292–2297. Canberra. Australia, 2003.

16. F. Lin, X . Yu, S. Gregor, Shirely, and R. Irons, "Time Series Forecasting with Neural Networks", Complexity International, Volume 02, ISSN 1320-0682, Australia,1995.