Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 229-235, 2016
ISSN 1990-9233;

DOI: 10.5829/idosi.mejsr.2016.24.TAET23531

Structure Optimization Using Bee and Fish School Algorithm for Mobility Prediction

[1]J. Ananthi, [2]Dr. V. Ranganathan and [3]Dr. B. Sowmya

[1]Associate Professor/ECE Department, Dhanalakshmi College of Engineering,
Chennai, Research Scholar, Anna University, Chennai, Tamil Nadu, India
[2]Professor and Head/ Electronics and Communication Engineering Department,
Vignan University, Guntur andhra Pradesh, India
[3]Professor, Centre for Research, Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India

Abstract: Neural Networks are widely used for mobility prediction in wireless networks. The back-propagation algorithm is employed effectively to train neural networks; it is widely recognized for applications to layered feed-forward networks, or multi-layer perceptron's. In this study, Multi-Layer Perceptron's Neural Network - Back Propagation (MLPNN-BP), Multi-Layer Perceptron's Neural Network – Bee Algorithm (MLPNN-BA) and Multi-Layer Perceptron's Neural Network – Fish School Search (MLPNN-FSS) are used for experiments and obtained better true positive rate, F measure, positive predictive value and reduced misclassification rate.

Key words: Multi-Layer Perceptron's Neural Network · Back Propagation (MLPNN-BP) · Multi-Layer Perceptron's Neural Network · Bee Algorithm (MLPNN-BA) and Multi-Layer Perceptron's Neural Network · Fish School Search (MLPNN-FSS)

INTRODUCTION

Wireless networks provides a more flexible communication model than traditional wire line models as users are not limited to fixed geographical locations. Mobile ad hoc networks which are different from cellular wireless networks, mobile ad hoc networks [1] do not depend on any fixed wired communication infrastructure. Ad hoc networks are deployed in applications such as disaster recovery and distributed collaborative computing, where routes are mostly multi-hop and communication between network hosts occur via radio packets. Movements of each consecutive host is arbitrary and routes are subject to frequent disconnections. Mobility presents a challenging issue for protocol design since the protocol must adapt to frequent changing network topologies in a way that is transparent to the end user.

Locating invdividuals by predicting their future geographical locations is essentia for a huge array of mobile applications such as location-based service, controlling accessibility of mobiles access, Multimedia QoS management provisions [2], as well as resource management for mobile computation and storage. Prediction abilities are useful for a wide range of mobile services. The two groups of mobile services which have significantly improved include: Personalizations - the service and applications are customized automatically and according to a user's future locations and Resource Management - services and applications allocates the resource according to user's future locations.

Learning through BPNN method is an upward slope techniquelearning which indicates the risk of becoming trapped in local minima in which any minute shift in synaptic weight can significantly heighten cost functions. At times the network is trapped between another set of synaptic weight in which cost functions are fairly less than the local minimum in a weight space. This process makes BPNN techniques undesirable and can lead to the termination of learning processes at a local minimum.

The training Mean Square Error (MSE) is the most widely used objective function for optimizing the connection weights of the MLPNN and it is usually minimized using the Back-Propagation (BP) algorithm. However, this conventional method suffers from the problems of local minima and over fitting. The limitations

Corresponding Author: J. Ananthi, Associate Professor/ECE Department, Dhanalakshmi College of Engineering, Chennai, Research Scholar, Anna University, Chennai, Tamil Nadu, India.

are addressed by invoking more sophisticated optimization technique such as PSO, ACO, BA, FSO, FSS etc.,

Related Works: Er & Liu [3] proposed a hybrid learning algorithm for a Multilayer Perceptron's (MLP) neural network using Genetic Algorithms (GA). There are dual steps involved in this hybrid learning algorithm: The first step is where all the parameters (weights and biases) of the initial neural network were encoded to form a long chromosome and tuned by the GA. In the next step, a quasi-Newton method called Broyden-Fletcher-Goldfarb-Shannon (BFGS) method was applied to train neural networks. Function approximation simulated studies are a nonlinear dynamic system of identification which is presented in a way to illustrate the performance of the proposed learning algorithm.

Park & Woo [4] proposed a prediction of a network traffic using Dynamic-Bilinear Recurrent Neural Network (D-BLRNN). D-BLRNN techniques is enhancec to develop prediction capabilities of BLRNN further by introducing dynamic learning control and optimization layer by layer procedure. Experiments were conducted on a real-world Ethernet network traffic data set. Results show that the dynamic BLRNN-based prediction scheme outperforms the conventional Multi-Layer Perceptron Type Neural Network (MLPNN) in terms of Normalized Mean Square Error (NMSE).

Mishra & Patra [5] proposed a compact MLPNN mechanism constructed through genetic and particle swarm optimization methods. This GA training enhances the chance for better accuracy than mere BP training sessions, in which it has longer time periods. It is to be noted that PSO training processes have a swift convergence rate than when compared to both the BP and GA, only with a minor accuracy compromise. The above method is suitable for implementations to be made in real-time.

Wu *et al* [6] proposed a backup multi-path protocol based on kalman filter in mobility prediction model is proposed. The multi-path routing algorithm finds one backup route and the mobility prediction algorithm based on kalman filter is developed to predict the link quality of the nodes in the next time, then a more stable route can be selected. At last, the new protocol with OLSR was simulated in QualNet, simulation comparison illustrates, the new protocol increases the average delivery rate, decreases the time that the packets stayed in the queue and decreases the average end-to-end delay.

Chen *et al*., [7] proposed a novel Bacterial Colony Foraging Optimization (BCFO) algorithm for complex optimization problems. The main idea of BCFO was to be developed an adaptive and cooperative life-cycle model by combining bacterial chemo taxis, cell-to-cell communication and self-adaptive searching strategies. The proposed BCFO is a better bacterially-realistic model due to its advantages against bacterial splits and complex deaths which occur throughout the entire foraging process and this can alter population size based on the kind of alorithm present. Cell-to-cell interactions enable each bacterium to proceed to better directions within chemotactic steps to speed up convergence rates. Self-adaptive search strategy, ensure that each bacterium maintains required balance for exploration and exploitation. There are seven versions of BCFO which are combined under different model strategies proposed and tested at a static and dynamic level. Then the proposed algorithm was created to be applied on real-world problems through its applications in dynamic RFID network optimization zones. The acquired statistical analysis of the above tests highlight significant improvement in performance because of a beneficial combination and it shows how the proposed algorithm can outperform reference algorithms.

Haklı & Uğuz [8] combined PSO with Levy flight. Levy flight was a random walk determining step size using Levy distribution. Being used Levy flight, a more efficient search takes place in the search space thanks to the long jumps to be made by the particles. In the proposed method, a limit value was defined for each particle and if the particles could not be improved self-solutions at the end of current iteration, this limit was increased. If the limit value determined was exceeded by a particle, the particle was redistributed in the search space with Levy flight method. To get rid of local minima and improved global search capability are ensured via this distribution in the basic PSO. The performance and accuracy of the proposed method called as Levy Flight PSO (LFPSO) are examined on well-known unimodal and multimodal benchmark functions. Experimental results show that the LFPSO was clearly seen to be more successful than one of the state-of-the-art PSO (SPSO) and the other PSO variants in terms of solution quality and robustness. The results are also statistically compared and a significant difference was observed between the SPSO and the LFPSO methods. Furthermore, the results of proposed method are also compared with the results of well-known and recent population-based optimization methods.

Zarei *et al*., [9] BA was applied to select best descriptors for the project. Some of the descriptors were chosen and utilized for initial inputs in Adaptive Neuro-Fuzzy Inference Systems (ANFIS). After which the model was modified to configure shaky compounds (i.e. the compounds which are ionized in aqueous solutions or can easily metabolize under specific conditions). The final step requires squared correlated coefficients to be acquired as the values 0.8769, 0.8649 and 0.8301 to be used as testing, training and validation sets, respectively. The results acquired show bee–ANFIS can be used as a powerful model for prediction of toxicity of substituted benzenes to T. pyriformis.

Methodology

Multi-Layer Perceptron Neural Network (MLPNN): MLPNNs [10] are one of the most important classes of neural networks and have many application areas, which can vary from engineering to finance. Most networks consist of input, a few hidden and output. The input layer is only responsible for feeding the input data to the neurons of the second layer, which is the first hidden layer. The output of the second layer is an input to the third layer and so on, for the rest of the net- work. The computation only takes place at the hidden and output neuron layer. The connections between all the elements of the networks are allocated synaptic weights, which are adjusted with the back propagation algorithm to provide nonlinear mapping.

This learning approach is more complex than that for a perceptron network and it is of the supervised type [11]. General MLP learning algorithm methods are is described below.

- Initialise network, with weights set to random numbers between -1 and +1.
- Present first training pattern and obtain output.
- Compare network output with target output.
- Propagate error backwards.

Hidden layer outputs of jth neuron is portrayed by the equation (1):

$$y_i = f\left(a = \sum_{1}^{N} w_{ji} \times x_i + b_j \right) \tag{1}$$

where xi is the input vector, is the synaptic weight between the input i and the neuron is the output of the jth neuron and bj is known as bias. Here, f(a) represents the activation function. Non linear differential activation functions are a part of every neuron networks. In the back propagation algorithm, the sigmoid function is one of the commonly used activation functions. The network can be trained through presentating sets of input and desired output data and to acquire a set of weightings which will provide the minimum error between the desired and the actual output of the network. The error signal for an output neuron is given by equation (2):

$$e(n)=d(n)-y(n) \tag{2}$$

where d is the desired output, y is the actual output and e is the error for the neuron.

To maximize the correct classification rate (accuracy) of all classes measured by the entropy and the correct classification rate of individual class is proposed. MLPNN, only minimizes the classification error, is sensitive to small change of the inputs and may not generalize well. The major reason is that the samples located near the decision boundary are sensitive to the changes of input. Therefore, these samples can be misclassified. In contrast, an MLPNN trained via a minimization of the MSE will minimize the difference between the real-valued output and the class value of either {0, 1} or {1, -1}, which will create a margin or a buffer between two classes in the output space.

Back Propagation Neural Network (BPNN): BP, one of the most popular techniques in the field of NN, is a kind of supervised learning neural network, the principle behind which involves using the steepest gradient descent method to reach any small approximations.

There are three layers contained in BP: input layer, hidden layer and output layer. Two nodes of each adjacent layer are directly connected, which is called a link. Each link has a weighted value, which presents the relational degree between two nodes [12]. By assuming there are n input neurons, m hidden neurons and 1 output neuron, from which we can infer the training process described by the following equations to update these weighted values, which can be divided into two steps:

Hidden layer stage: In this layer, the outputs of all neurons are calculated by following equation (3):

$$net_j = \sum_{i=0}^{n} v_{ij} x_i \text{ j=1,2,...,m}$$
$$y_j = fH(net_j) \text{ j=1,2,...,m} \tag{3}$$

Here is the activation value of the jth node, is the output of the hidden layer and fH is called the activation function of a node, usually a sigmoid function as in equation (4):

$$fH(x) = \frac{1}{1+\exp(-x)} \tag{4}$$

Output Stage: The outputs of all neurons in the output layer are given as in equation (5):

$$O = f_0\left(\sum_{j=0}^{m} \omega_{jk} y_i\right) \tag{5}$$

Here is the activation function, usually a line function. All weights are assigned with random values initially and are modified by the delta rule according to the learning samples traditionally.

Bee Algorithm (BA): Bee algorithm, an optimization algorithm originates from honeybee foraging behaviors and is used to search for optimal solution [13 - 15]. Bee colonies contain three types of bees namely, employed, on-looker and scout bees. The first group, employed bees transder place and quantity information on nectar from a specific source of food. They then shift information to another group, namely the on-looker bees which keep dancing inside the hive. The estimated time period of the dance can predict the quantity of nectar from the source. On-looker bees then select food sources on the basis of estimated nectar amounts from a food source. A desirable food source can attracts multiple on-looker bees. Then it is the duty of Scout bees to seek out the neighboring search spaces for potential sources. Scout bees on the other hand control exploration processes, while employees and on-looker bees have exploitative roles. The above algorithm considers food sources which are closest to the solutions.

The food source is a D-dimensional vector, where D is the number of optimization variables. The amount of nectar in a food source determines the value of fitness. The basic flow chart of BA is shown in Figure 1.

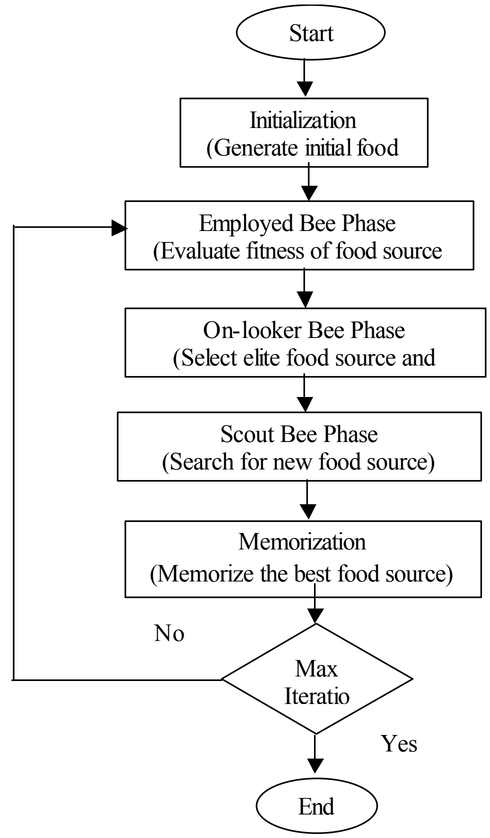

Fig. 1: Flow chart for Bee Algorithm

Step 1: Generation of random initial food sources where the initial number of sources is half the size of the bee colony.

Step 2: Deploying employee bees to food sources in order to determine and calculate the amount and quality of nectar. Each source has only one employee bee and this is relational to the total number of food sources. Additionally, employeecbees can modify solutions which are retained in memory by its ability to search neighborhoods for food sources. Employed bees can save new solution if they find that the quality is better than older ones. After the entire processes the bees travel back to their hives to share solutions with on- looker bees.

Step 3: On-looker bees decide on best sources of food using selective probability-based processes. Higher nectar quanitities attract multiple numbers of on-looker bees. These bees travel to selected food sources to investigate, improve and calculate its fitness. Quite Similar to employed bees, on-looker bees select new solutions based on better fitness solutions.

Step 4: Sources which cannot be improved through numerous iterations are eventually abandoned. So, employed bee sent to find new food sources become scouting bee and through their venturing abandoned sources are replaced.

Step 5: Memorization of best food sources and maximum number of iterations are set as a criteria for termination which is often checked after the end of each iterations. If this is not met then the algorithm reverts back to step 2 for the possible next iteration.

Fish School Search (FSS): In FSS [16-18], searching is referred to as swimming. This action is achieved via three effectors; individual movement, collective-influence movement and school performance movement. The initial individual position is random. Thereafter the movement is given by equation (6):

$$x_i = x_i(t-1) + rand(-1,1)step_{ind}(t-1) \quad (6)$$

In the above equation, rand is taken to be a random number which is uniformly generated in each interval [-1,1] and are assumed to be an single step size where each step size consequetively decreases linearly in search processes to exploit future positions. Fish evaluate food sources before moving to particular point and access whether food in that particular direction is better placed than its current position. The fish will then feeds on these selected sources. Individuals in FSS similarly ‘store’ their earlier work information via their weight patterns represented as. A fish and relation to successfully acquiring food is directly proportional to its value and its weight constraints can vary between 1 and. Every fish are initialized by the weight of its half. And the weight of every individual in a population can be calculated using the given formula:

$$w_i(t) = w_i(t-1) + \frac{f[x_i(t) - f[x_i(t-1)]}{\max\{|f[x_i(t) - f[x_i(t-1)]|\}} \quad (7)$$

In which f is taken as a finite element model and calculates each fitness value of fish in every schools. Weighted averages of every fish is calculated after each individual movements. The above biases of the future fish movement towards other successfully moving fish. Thus movements of resultant fish are calculated by the following equation (8):

$$x_i(t) = x_i(t-1) + \frac{\sum_{i=1}^{N} \Delta x_{ind}\{f[x_i(t) - f[x_i(t-1)]\}}{\sum_{i=1}^{N}\{f[x_i(t) - f[x_i(t-1)]\}} \quad (8)$$

where is taken as displaced individual fishes and N is assumed as the total number of fishes present in a school. This is also known as the collective – influencer due to its effects. It is seen that fish are able to complete swim movements with adjustable collection of volatile step. This step takes into account how the whole school is performing so far. This step requires the calculation of the average weighted school position called the barycenter at time t:

$$B(t) = \frac{\sum_{i=1}^{N} x_i(t) w_i(t)}{\sum_{i=1}^{N} x_i(t)} \quad (9)$$

Final positions of fishes are determined on the increase or decrease of the entire school weight and this can be denoted through the following equations 8 and 9. Whereas Equation (9) shall be evaluated in case weights have been increased. However, the Equation (10) will be used when the weight decreases:

$$\begin{aligned} x_i(t) &= x_i(t-1) - \mathrm{step}_{vol}\ \mathrm{rand}\frac{\{x_i(t-1) - \mathrm{B(t-1)}\}}{dist(\{x_i(t-1), \mathrm{B(t-1)}\})} \\ x_i(t) &= x_i(t-1) + \mathrm{step}_{vol}\ \mathrm{rand}\frac{\{x_i(t-1) - \mathrm{B(t-1)}\}}{dist(\{x_i(t-1), \mathrm{B(t-1)}\})} \end{aligned} \quad (10)$$

The parameter rand is a random number uniformly generated in the interval [0, 1]. The is used to control the displacement from or to the barycenter and it is decreased linearly as the search proceeds. Theis the Euclidian distance between x and y.

This final positioning based on the average school weight increasing or decreasing effectively implements the space exploitation or exploration concept respectively. The fish exploit a particular area of space if the school collectively gains weight (i.e. school radius reduce) or otherwise explores other areas by expanding away from the barycenter if the school looses weight.

RESULT

Table 1 and 2 shows the results of misclassification rate, true positive rate, positive predictive value and F Measure. Figure 2 to 5 shows the same.

Table 1: Summary of Results

	True Positive Rate	Positive Predictive Value	F Measure
MLP NN - BP			
Library	0.745	0.8247	0.7828
Academic	0.865	0.8251	0.8446
residential	0.81	0.8648	0.8365
social	0.8417	0.7842	0.8119
Admin	0.8	0.7705	0.785
MLP NN - BA (parameter optimization of BP)			
Library	0.8283	0.8599	0.8438
Academic	0.8867	0.8567	0.8714
residential	0.8483	0.8791	0.8634
social	0.8467	0.8167	0.8314
Admin	0.8267	0.8267	0.8267
MLP NN - FSS			
Library	0.8683	0.9108	0.889
Academic	0.9167	0.9031	0.9098
residential	0.885	0.8835	0.8842
social	0.8933	0.8673	0.8801
Admin	0.8633	0.8633	0.8633

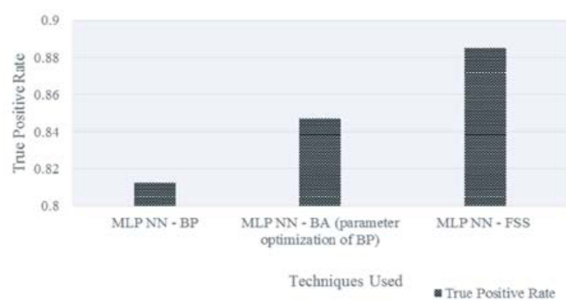

Fig. 2: True Positive Rate

Observations taken from Table 1 and Figure 2 determine the true positive rates of MLPNN-FSS performs better by increasing in the rate of 8.59% than MLPNN-BP and by 4.34% than MLPNN-BA (Parameter optimization of BP).

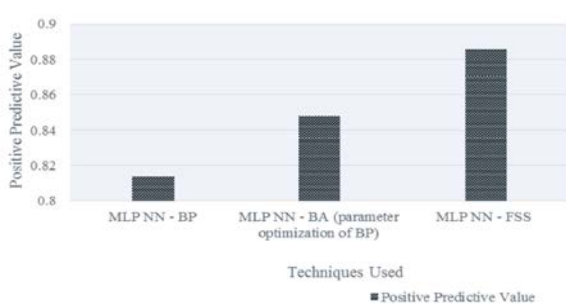

Fig. 3: Positive Predictive Value

The available observations from Table 1 and Figure 3 determine that positive predictive values of MLPNN-FSS performs better by increasing in the value of 8.44% than MLPNN-BP and by 4.4% than MLPNN-BA (Parameter optimization of BP).

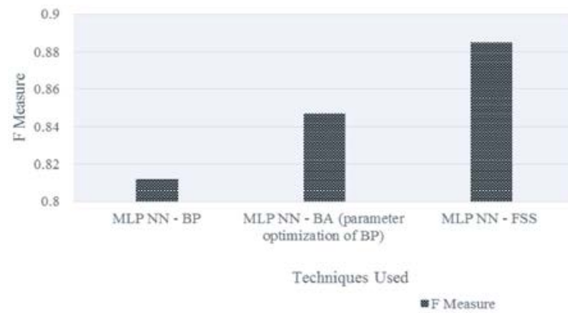

Fig. 4: F Measure

Observations from Table 1 and Figure 4 determine that F can ensure that of MLPNN-FSS performs better by increasing in the measure of 8.62% than MLPNN-BP and by 4.38% than MLPNN-BA (Parameter optimization of BP)

Table 2: Misclassification Rate

	Misclassification rate
MLP NN – BP	0.1877
MLP NN - BA (parameter optimization of BP)	0.1527
MLP NN – FSS	0.1147

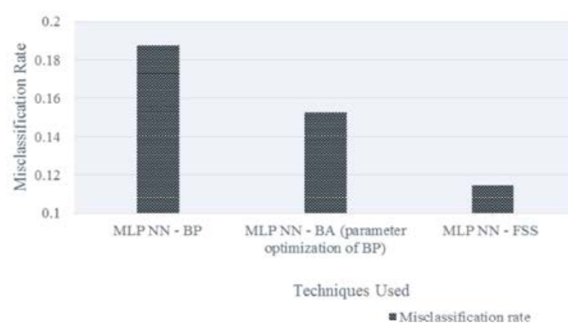

Fig. 5: Misclassification Rate

Observations from Table 2 and Figure 5 determine that misclassification rates of MLPNN-FSS performs better by lowering the misclassification rate by 48.3% than MLPNN-BP and by 28.4% than MLPNN-BA (Parameter Optimization of BP).

CONCLUSIONS

MLPNN-BA and MLPNN-FSS were applied. Results show that the true positive rate of MLPNN-FSS performs better by increasing in the rate of 8.6% than MLPNN-BP and by 4.4% than MLPNN-BA (Parameter optimization of BP). Also the positive predictive value, F measure performs better by increasing its value for proposed method. Finally the reduced misclassification rate is obtained.

REFERENCES

1. Su, W., S.J. Lee and M. Gerla, 2000. Mobility prediction in wireless networks. In MILCOM 2000. 21st Century Military Communications Conference Proceedings, 1: 491-495. IEEE.
2. Xiong, H., D. Zhang, D. Zhang, V. Gauthier, K. Yang and M. Becker, 2014. MPaaS: Mobility prediction as a service in telecom cloud. Information Systems Frontiers, 16(1): 59-75.
3. Er, M.J. and F. Liu, (2009 June). Genetic algorithms for MLP neural network parameters optimization. In Control and Decision Conference, 2009. CCDC'09. Chinese pp: 3653-3658. IEEE.
4. Park, D.C. and D.M. Woo, (2009, August). Prediction of network traffic using dynamic bilinear recurrent neural network. In Natural Computation, 2009. ICNC'09. Fifth International Conference on, 2: 419-423. IEEE.
5. Mishra, S. and S.K. Patra, (2008, July). Short term load forecasting using neural network trained with genetic algorithm & particle swarm optimization. In Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on (pp: 606-611). IEEE.
6. Wu, C., X. Di, H. Liang and X. Shuang, (2011, July). An optimization multi-path Ad Hoc network routing protocol based on mobility prediction. In Control Conference (CCC), 2011 30th Chinese, pp: 4421-4425. IEEE.
7. Chen, H., Y. Zhu, K. Hu and L. Ma, 2014. Bacterial colony foraging algorithm: combining chemotaxis, cell-to-cell communication and self-adaptive strategy. Information Sciences, 273: 73-100.
8. Haklı, H. and H. Uğuz, 2014. A novel particle swarm optimization algorithm with Levy flight. Applied Soft Computing, 23: 333-345.
9. Zarei, K., M. Atabati and K. Kor, 2014. Bee Algorithm and Adaptive Neuro-Fuzzy Inference System as Tools for QSAR Study Toxicity of Substituted Benzenes to Tetrahymena pyriformis. Bulletin of Environmental Contamination and Toxicology, 92(6): 642-649.
10. Çaylak, Ç. and İ. Kaftan, 2014. Determination of near-surface structures from multi-channel surface wave data using multi-layer perceptron neural network (MLPNN) algorithm. Acta Geophysica, 62(6): 1310-1327.
11. Sridhar, G.V. and P.M. Rao, 2015. An Improved Multilayer Perceptron for BCI with Evolutionary Optimization. International Journal for Innovative Research in Science and Technology, 2(1): 328-332.
12. Zhang, Y. and L. Wu, 2008. Weights optimization of neural network via improved BCO approach. Progress In Electromagnetics Research, 83: 185-198.
13. Pham, D.T., S. Otri, A. Afify, M. Mahmuddin and H. Al-Jabbouli, (2007, May). Data clustering using the bees algorithm. In Proceedings of 40th CIRP international manufacturing systems seminar.
14. Özbakir, L., A. Baykasoğlu and P. Tapkan, 2010. Bees algorithm for generalized assignment problem. Applied Mathematics and Computation, 215(11): 3782-3795.
15. Kavousi, A., B. Vahidi, R. Salehi, M.K. Bakhshizadeh, N. Farokhnia and S.H. Fathi, 2012. Application of the bee algorithm for selective harmonic elimination strategy in multilevel inverters. Power Electronics, IEEE Transactions on, 27(4): 1689-1696
16. Boulkabeit, I., L. Mthembu, T. Marwala and F. Buarque De Lima Neto, (2013, September). Finite Element Model Updating Using Fish School Search Optimization Method. In Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on, pp: 447-452. IEEE.
17. Jena, S., 2013. Fish school search: an interval representation (Doctoral dissertation, National Institute of Technology, Rourkela).
18. da CC Lins, A.J., F.B. Lima-Neto, F. Fages and C.J. Bastos-Filho, 2012. A comparative analysis of FSS with CMA-ES and S-PSO in ill-conditioned problems. In Intelligent Data Engineering and Automated Learning-IDEAL 2012 (pp: 416-422). Springer Berlin Heidelberg