International Journal of Computer Applications (0975 – 8887)
Volume 102 – No.10, September 2014

Artificial Neural Network Learning Enhancement using Bacterial Foraging Optimization Algorithm

Raminder Kaur
M.Tech Research Scholar
Department of CSE
Chandigarh Engineering College, Mohali, Punjab

Bikrampal Kaur, PhD
Professor
Department of CSE
Chandigarh Engineering College, Mohali, Punjab

ABSTRACT
The artificial neural network (ANN) is a mathematical model capable of representing any non-linear relationship between input and output data. ANN is an abstract representation of the biological nervous system which has the ability to solve many complex problems. It has been successfully applied to a wide variety of classification and function approximation problems. The information processing capability of artificial neural networks (ANNs) is related to its architecture and weights. To have a high efficiency in ANN, selection of an appropriate architecture and learning algorithm is very important. In this study, the adaption of neural network connection weights using Bacterial Foraging Optimization Algorithm (BFO) is proposed as a mechanism to improve the performance of Artificial Neural Network in classification of Software Defect Dataset. The problem concerns the classification of software as defective or non defective on the basis of software metrics data. The results show that BFO-ANNs have better accuracy than traditional ANNs. The experimental results showed that BFOA-ANN has an improvement of 2.55 % in software defect prediction accuracy than the original feed forward artificial neural network and 2.80 % in case of cascade forward neural network.

General Terms
Learning enhancement, Optimization

Keywords
Artificial Neural Network, Bacterial Foraging Optimization algorithm, swarm intelligence

1. INTRODUCTION
An artificial neural network is an adaptive system that has interesting features like the ability to adapt, learn and generalize. An ANN is also highly accurate in classification and prediction of output because of its parallel processing, self organization, fault tolerance and adaptive capability which enables it to solve many complex problems. Its ability to solve different problems is achieved by changing its network structure during the learning process.

There are many calculations, which are very complex, non-linear and parallel that could be solved by ANN. The main purpose of ANN is the capacity of the network on learning from its surroundings and improves the performance of the model during the process of learning.

The performance of ANN is highly dependent upon its architecture and connection weights connecting the nodes [1]. The determination of architecture of ANN (number of hidden layers and number of neurons in the hidden layer) and connection weight values is not a straightforward process. It is a continuous optimization problem. As a result, several algorithms have been proposed and applied as training algorithms for ANNs and these include Particle Swarm Optimization (PSO) [7,8,9,10], Bird Mating Optimizer (BMO) [13], Artificial Bee Colony [14], Ant colony Optimization [15,16], Cat Swarm Optimization [17], Genetic Algorithm [18, 19], Artificial Fish Swarm Optimization [20], Cuckoo Search Optimization [21]. In this paper, the Bacterial Foraging Optimization (BFO) algorithm is proposed to be used as the training algorithm to train the ANNs.

1.1 Artificial Neural Network
Among the various ways to perform an automatic classification, one of the most popular is the neural network. Because of the self learning and self organizing ability to adapt, artificial neural network (ANN) has the characteristics of can be trained.

The ANN consists of a large number of processing elements (artificial neurons) and weighted connections between these elements. The weights of connections encode the knowledge embedded in the network. The intelligence of a neural network emerges from the collective behavior of neurons, each of which performs very limited operation. Each individual neuron finds a solution by working in parallel.

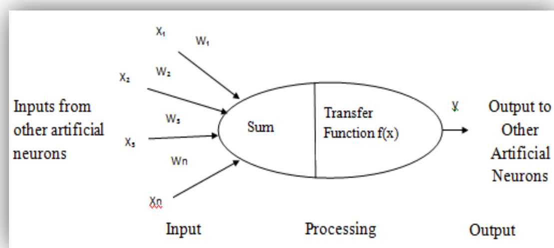

Fig 1: An Artificial Neuron

In Fig. 1, x_1, x_2 are the inputs to the neuron, w_1, w_2 are the connection weights between the neurons and the respective inputs. The Sum ($\sum$) is the weighted sum of the input data and y is the output of the neuron and f(x) is the activation function which is usually a non linear function.

An artificial neuron is a small processing unit and performs a simple computation that is fundamental to the operation of the neural network. An artificial neuron has many inputs and one output. Neural networks are the simple clustering of primitive artificial neurons. This clustering occurs by creating layers which are then connected to one another. Layers are made up of number of interconnected nodes which contain an activation function.

Input is presented to the network via the input layer which communicates to one or more hidden layers where the actual processing is done via a system of weighted connections. The hidden layers then link to an output layer.

Artificial Neural Networks consists of three types of layers: input layer, hidden layer and output layer.

- Input Layer: This layer comprises of input units which symbolizes the unrefined information provided for the networks
- Hidden Layer: This layer is represented by hidden units which are influenced by the behaviour of the input units and the weight that connect these input and the hidden units.
- Output Layer: The output unit's behaviour is dependent on the specificity of the hidden units and the weights connecting the hidden and output units.

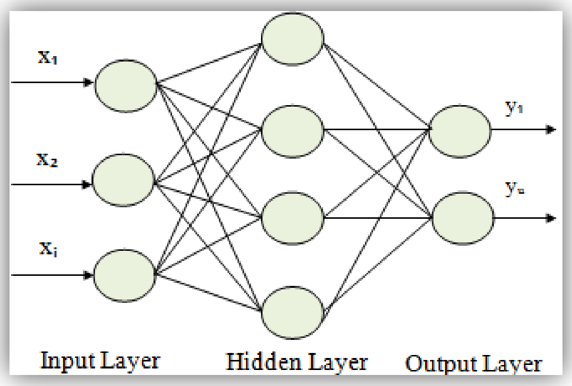

Fig 2: Layered Architecture of Neural Network

In this research, feed forward and cascade forward artificial neural networks models are employed.

Feed Forward Neural Networks: It is the most commonly used form of ANN. Feed Forward ANNs tend to be straight forward networks that associate inputs with outputs. Feed forward networks can be used for any kind of input to output mapping. Feed-Forward networks consist of a number of layers. The first layer is the input layer to which input patterns are provided. Each subsequent layer has a connection from the previous layer. The final layer is the output layer which produces the network's output.

Cascade Forward Neural Networks: A variation of the feed-forward network is the cascade forward network which has additional connections from the input to every layer and from each layer to all the following layers. Cascaded-forward neural networks are similar to feed-forward networks, but include a weight connection from the input to each layer and from each layer to the successive layer. The additional connections might improve the speed at which the network learns the desired relationship.

1.2 Bacterial Foraging Optimization Algorithm

BFOA is a new comer to the family of nature inspired optimization algorithm motivated by the behavior of bacteria Escherichia Coli (E.Coli) in its search of food. BFOA is a swarm intelligence technique that models the food seeking and reproductive strategy of common bacteria in order to find approximate solutions to extremely difficult or impossible numeric maximization and minimization problems and solve numeric optimization problems where there is no effective deterministic approach. It tends to eliminate the animals with poor foraging strategies and favor those having better and successful foraging strategies [2].

During the process of foraging, E.Coli bacteria undergo four stages i.e. Chemotaxis, Swarming, Reproduction and Elimination & Dispersal.

Chemotaxis: The characteristic movement of bacteria at the time of foraging can be defined in two ways i.e. swimming and tumbling together known as chemotaxis. During tumbling, bacterium moves chaotically in disordered manner. The tumble action modifies the orientation of the bacterium. Bacteria tend to tumble when they are in a gradient that is not improving. A bacterium is said to be swimming if it moves in predefined direction in a directed manner. Bacteria tend to swim when they are improving in some sense, finding an increasing nutrient gradient.

Mathematically,

$$\theta_i(j+1,k,l) = \theta_i(j.k.l) + C(i)\emptyset(i) \quad (1)$$

Here $\theta_i(j.k.l)$ is ith bacterium position at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event. $C(i)$>0 , i=1,2,...,S denote a basic chemotactic step size taken in unit random direction. S is the total number of bacteria.

If the health of bacteria improves after the tumble, the bacteria will continue to swim in the same direction for the specified number of steps (swim counter) or until the health degrades.

Swarming: The bacterium that has discovered the optimum path for the food tries to attract other bacteria. They exhibits swarm behavior i.e. healthy bacteria try to attract other bacteria so that they can together reach the desired optimal location (solution point) more quickly. The effect of swarming is to make the bacteria congregate into groups and move as concentric patterns with high bacterial density.

Reproduction: Health status (fitness) of each bacterium is calculated after each complete chemotaxis process. It is the overall sum of cost function.

$$J_i(health) = \sum_{j=1}^{Nc} J(i,j,k,l) \quad (2)$$

Where Nc is total number of steps in a complete chemotaxis process. All bacteria are ordered descending based on health status. Locations of healthier bacteria represent better sets of optimization parameters. Then, to further speed up and refine the search, more number of bacteria is required to be placed at these locations in the optimization domain. Here, the best set of bacteria gets divided into two groups. The healthier half of bacteria (with minimum value of cost function) replaces the other half of bacteria, which gets eliminated, owing to their poor foraging abilities. In this way population of bacteria remains constant in the process.

Elimination& Dispersal: The chemotaxis process performs the local search and reproduction process speeds up the convergence of search parameters. While to a large extent, chemotaxis and reproduction may not be enough to reach the global minimum point because it is possible that bacteria sticks in the first place and unable to navigate the entire search space. The bacteria may also get trapped in local minima. To avoid this to happen, elimination and dispersal event is performed. Then, some bacteria are eliminated from the cycle of searching according to a preset probability Ped (probability of elimination and dispersion) or moved to another location within the environment.

The Procedure for implementing the BFOA is given by the following steps [3,4,5].

Step 1: Initialize Parameters

a) P: Dimension of search space i.e. Number of parameters to be optimised.
b) S: Total number of bacteria in the population to be used for searching the region.
c) Ns: Maximum Swimming length after which tumbling of bacteria will take place in a chemotaxis step.
d) Nc: Maximum number if iterations in a chemotaxis loop.
e) Nre: Maximum number of reproduction steps.
f) Ned: Maximum number of elimination & dispersal events imposed on bacteria
g) Ped: Elimination-Disperal probability
h) C(i): The size of the step taken in the random direction specified by the tumble

Step 2: Elimination-Dispersal Loop: Increment $l = l + 1$

Step 3: Reproduction Loop: Increment $k = k + 1$

Step 4: Chemotaxis loop: Increment $j = j + 1$

For i=1,2,...,S take a chemotactic step for bacterium i as follows:

a) Compute cost function value $J_i(j.k.l)$
b) Let $Jlast = J_i(j.k.l)$ to save this value since we may find a better cost via a run.

c) If $j = 1$, tumble. Let the bacteria take the step of height $C(i)$ along a randomly generated tumble vector $\emptyset(i)$.

$$\theta_i(j+1,k,l) = \theta_i(j.k.l) + C(i)\emptyset(i)$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i.

d) Compute the cost function value $J_i(j+1.k.l)$
e) Swim: Let $m = 0$ (counter for swim length)
While $m < Ns$

If $J_i(j+1,k,l) > J_i(j.k.l)$
Increment $m = m + 1$

Set $Jlast = J_i(j+1.k.l)$

Else tumble & reset $m = 0$

f) The next bacterium $(i + 1)$ is taken for swimming and tumbling process till $i = S$.

Step 5: If $j < Nc$ go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

Step 6: Reproduction

a) For the given k and l and for each i=1,2,...,S, let

$$J_i(health) = \sum_{j=1}^{Nc} J(i,j,k,l)$$

be the health of the bacterium i. Sort the calculated value in ascending order of cost health. Higher value of cost of any bacteria means poor health or less fitness.

b) Out of the total S bacteria, half of the bacteria with the highest J(health) values will die and the better halves having lower J(health) values will sustain in the evolution process and split (this process is performed by the copies that are made are placed at the same location as their parent).

Step 7: If $k < Nre$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

Step 8: Elimination-Dispersal

According to elimination-dispersal probability (Ped), existing bacteria gets eliminated & dispersed in a new random position. To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain.

Step 9: If l<Ned, go to step 2, Otherwise end.

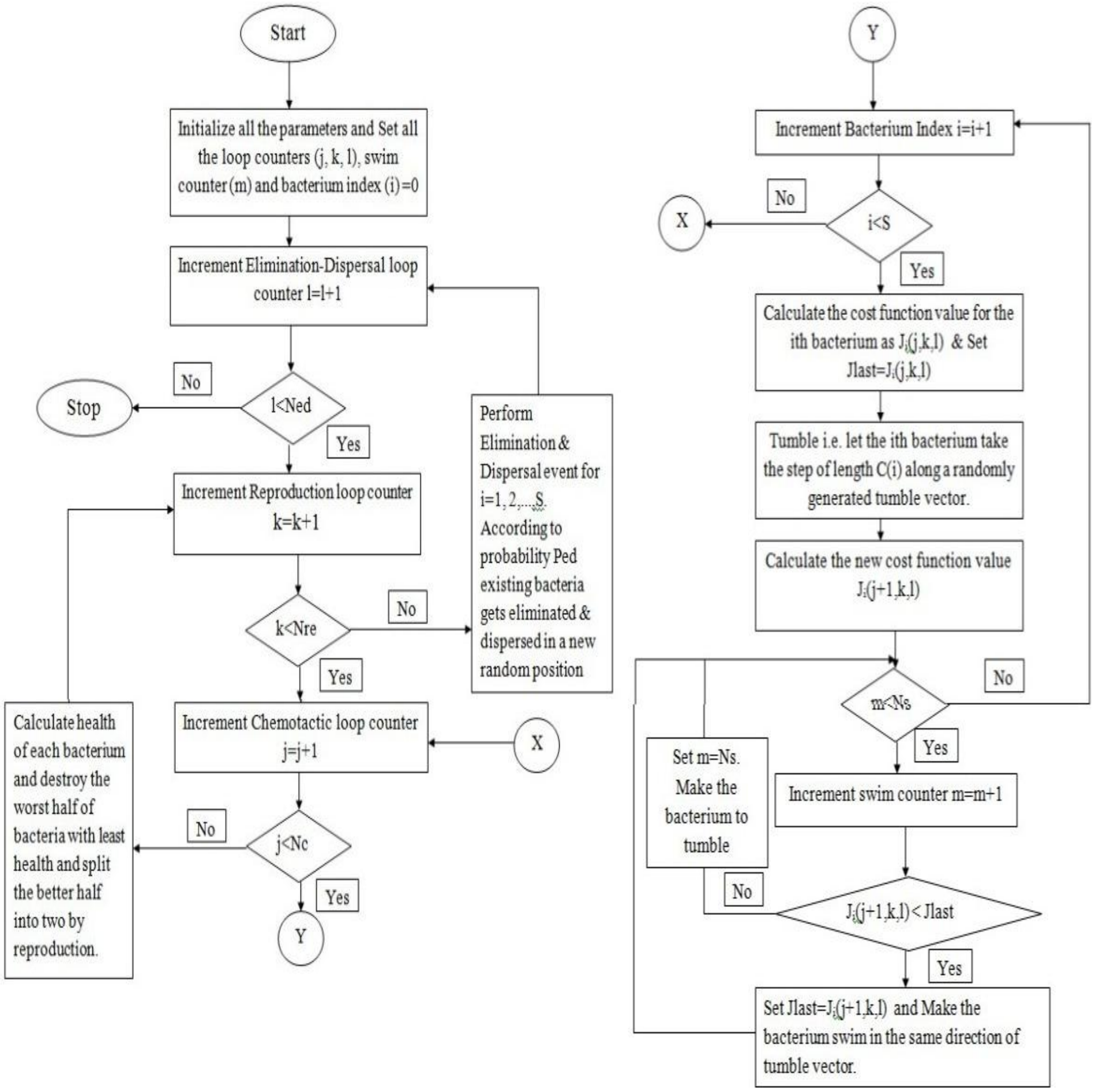

Fig 3: Flow chart of BFOA

2. RELATED WORKS ON OPTIMIZATION OF ANN

Since the determination of various parameters of artificial neural network is not a straightforward process, various researches have been conducted with the purpose of finding the optimal configuration of ANNs. Researchers have shown the interest in the study of social insect's behavior in neural network area for solving different prediction problems. Several swarm intelligence algorithms have been proposed as training algorithms for ANNs such as Particle Swarm Optimization (PSO) [7,8,9,10], Bird Mating Optimizer (BMO) [13], Artificial Bee Colony [14], Ant colony Optimization [15,16], Cat Swarm Optimization [17], Genetic Algorithm [18, 19], Artificial Fish Swarm Optimization [20], Cuckoo Search Optimization [21] Bacterial Foraging Optimization (BFO) [11,12]. In the last decade, evolutionary algorithms were proposed to fitness based methods as they are not sensitive to initial values and able to jump out of local minimal point.

Author in paper [7] proposes an artificial neural network prediction model that incorporates the constructive cost model (COCOMO) which is improved by applying particle swarm optimization (PSO), to provide a method which can estimate the software develop effort accurately. The modified model increases the convergence speed of artificial neural network and solves the problem of artificial neural network learning ability that has a high dependency of the network initial weights. This PSO based model shows an improvement of 3.27% in software effort estimation accuracy than the original artificial neural network constructive cost model.

Prediction models are often used to find the non linear relationship between metric data and quality factors. This research study [8] predicted the relationship between metric data and quality factors with historical data by using the optimized BP network based on PSO. Experiments show that the algorithm has a better performance than the BP network algorithm and perfectly solve the problem of slow convergence and easily getting into local minimum.

In the paper [12], Bacterial foraging Optimization algorithm is applied in feed-forward neural network to enhance the learning process in terms of classification accuracy and convergence rate**.** The results show that Bacterial Foraging optimization Algorithm gave a better performance in terms of convergence rate and classification accuracy compared to particle swarm optimization Feed forward Neural Network

Bird Mating Optimizer (BMO) is applied for training feed-forward ANNs which is a population based search method inspired by the mating strategies of bird species to breed broods with superior genes for designing optimum searching techniques. BMO proves to be an efficient algorithm for weight training of neural networks [13].

In the paper [14], authors investigate the use of artificial bee colony (ABC) algorithm that stimulates the foraging behaviour of a honey bee swarm. The performance of ABC is compared with the traditional BP algorithm. The simulation results show that the proposed ABC algorithm can successfully train software defect data for prediction purpose. The problems related to back propagation of yielding the networks with sub optimal weights because of many local optima in the solution space were overcome because of the powerful ability of ABC of searching global optimal solutions.

The paper [16] introduces an optimized neural network for predicting permeability based on swarm intelligent approach, named ant colony optimization. ACO is a computational intelligent approach which is inspired by the observation of ants. The number of neurons in hidden layer, weights and bias are optimized in the proposed NN using the ACO. The result demonstrates that the optimized NN performed more accurate compared to the simple NN model.

.The aim of the research [17] was to use the Cat Swarm Optimization algorithm with Optimal Brain damage pruning techniques to simultaneously optimize the connection weights and structure of artificial neural networks. Six datasets were used in the experiment and the performance of neural network was measured by how effective it is in minimizing the mean-squared error or the misclassification rate. It was found from the results that CSONN-OBD algorithm, a CSO- based ANN optimizer with OBD pruning algorithm was able to generate artificial neural networks.

In the study [19], Elman Neural Network and Feed forward neural network are used and then their accuracy is improved using Genetic Algorithm. The results obtained showed that the accuracy of the neural network got improved significantly when Genetic Algorithm is used to update the weight values and train the neural network

In this research study [20], Artificial Fish Swarm Algorithm (AFSA) is used to analyze its effectiveness in enhancing the Multilayer Perceptron (MLP) learning. Also, AFSA is compared to particle swarm optimization and Differential Evolution (DE). The experimental results have demonstrated the effectiveness of AFSA in solving MLP neural network weight optimization problem. The comparative results also showed that AFSA converges faster in less iteration and with better correct classification compared to PSO and DE in enhancing MLP training.

In the paper [21], the Cuckoo Search (CS) algorithm is implemented in training a feed forward multilayer perceptron network. Cuckoo Search is a type of swarm intelligence that is based on the brooding behavior of some kinds of cuckoo birds. It is inspired by the behavior of some species of cuckoo birds combined with the flight routines of many species of birds and flies, called levy flights. The trained MLP's accuracy is evaluated by applying four benchmark classification problems. The results obtained were also compared to those attained using Particle Swarm Optimization and Guaranteed Convergence Particle Swarm Optimization which is a PSO variant. CS was proved to be superior to PSO and GCPSO in all benchmark problems. CS yields lower average quadratic error and lower standard deviation.

3. METHODOLOGY OF BFOA-ANN

The process of optimizing the connection weights is known as training or learning. The network learns a function by adapting the strength of its connection weights in response to the training examples presented to it in accordance with a predefined learning law. The classification of fixed input data patterns to certain outputs is the main objective of training method. ANNs are trained by applying an optimizing algorithm, which attempts to reduce the error in the network output by adjusting the matrix of network weights. Recently development shows that Bacterial Foraging Optimization Algorithm is utilized to solve optimization related problems [12]. In this study, the adaption of neural networks weights using Bacterial Foraging Optimization (BFO) is proposed as a mechanism to improve the performance of Artificial Neural Network in classification of Software Defect Dataset.

There are two main issues about the performance of the neural network. First of all, it is important to determine its structure and secondly, it is also vital to specify the weights of the neural networks that help to minimize the total errors. There are some problems in ANN learning such as the difference between the target output and the actual output. Many optimization algorithms are used to optimize ANN for enhancing the error convergence and obtaining the good accuracy of ANN. In this research, BFOA is used to enhance ANN learning ability to obtain better performance.

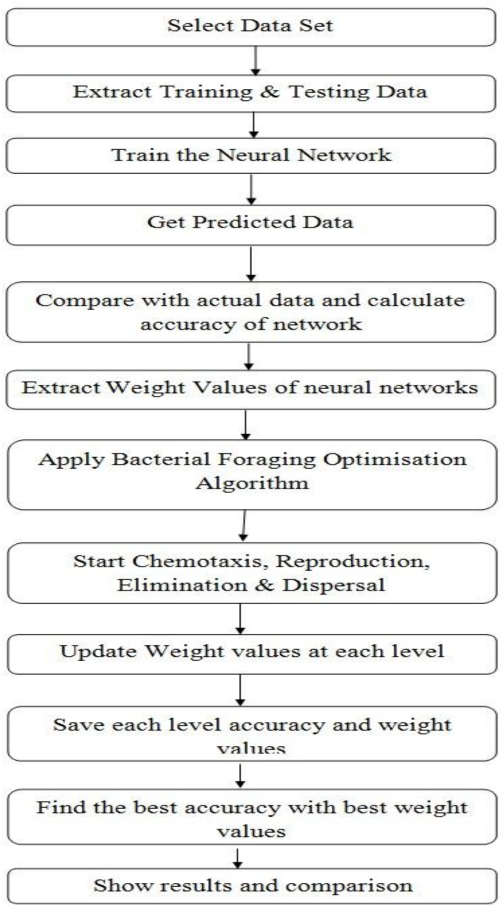

Fig 4: Overview of Proposed BFOA-ANN Framework

4. EXPERIMENT AND ANALYSIS

4.1 Data Set Description

The experiments carried out here involve dataset ant-1.7 regarding defect problems coming from the Promise repository of empirical software engineering data [23]. This dataset consisted of twenty object oriented metric values and their corresponding bug data. These data sets were pre-processed into a format that is appropriate for training, validation and testing of neural networks.

4.2 Evaluation Method

Accuracy is a widely preferred statistic to determine the performance of a classifier. Accuracy is the true classification rate. Accuracy is used as the performance criteria to evaluate the model. Accuracy is the degree of matching between the predictions and the actual data.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

In the above equation, TP (True Positive) refer to the positive tuples that were correctly labelled by the classifier. TN (True Negative) refers to the negative tuples that were correctly labelled by the classifier. FP (False Positive) refers to the negative tuples that were incorrectly labelled as positive. FN (False Negative) refers to the positive tuples that were mislabelled as negative [22].
It is calculated as:

$$ACCURACY = 100\text{-}100*\left(\frac{\sum |X-Y|}{N}\right) \quad (4)$$

Where, X is Actual Data Values, Y is Predicted Data Values, |X-Y| is Absolute value of X-Y, N is number of data points in X.

4.3 Performance Comparison

The performance comparison of the ANN and BFO-ANN model is shown in Table 1.

Table 1 Comparison of Results obtained from ANN and BFO-ANN Model

Accuracy	Prediction Model			
	Feed Forward Neural Network	**BFO Trained Feed-Forward NN**	**Cascaded-Forward Neural Network**	**BFO Trained Cascaded-Forward Neural Network**
1st Iteration	76.5854	79.0244	77.5610	79.0244
2nd Iteration	76.0976	80.0000	78.0488	81.4634
3rd Iteration	78.0488	80.4878	78.5366	80.9756
4th Iteration	77.0732	79.5122	75.1220	79.0244
Mean	76.95125	79.5061	77.3171	80.12195

Table 1 shows the result after applying data sets on Feed Forward ANN model, Cascaded ANN model and BFO trained Feed Forward ANN model, BFO trained Cascaded ANN model.

4.4. Result Analysis

Table 2: Improvement in the Prediction Accuracy of the ANN Model and BFO-ANN Model

Prediction Model	Evaluation	
Feed Forward ANN vs. BFO-Feed Forward ANN		**Accuracy**
	Feed Forward-ANN Model	76.95125
	BFO-FF-ANN Model	79.5061
	Improvement %	2.55485
Cascaded-Forward ANN vs. BFO Cascaded-Forward ANN	Cascaded ANN Model	77.3171
	BFO Cascaded ANN	80.12195
	Improvement %	2.80485

This table shows the comparison of accuracy between the traditional Feed Forward and Cascade Forward Artificial Neural Networks (ANN) and neural networks optimised using Bacterial foraging optimisation algorithm (BFO-ANN).

BFO-ANN model gives an improvement of 2.55 % in accuracy in feed forward ANN and 2.8 % in cascade forward ANN.

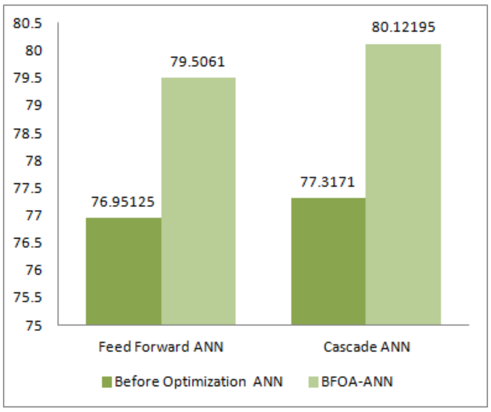

Fig 5: Graphical Representation of Experimental Results: Accuracy comparison.

5. CONCLUSION

Neural Networks are well researched and established method that have been used over decades and are very successful in predicting from the past datasets. In this study, BFOA-ANN has been used for software defect dataset. From the analysis, it has been found that the accuracy of the BFO-ANN model is 2.55 % more than the accuracy of normal feed forward ANN model and 2.80 % more in case of cascade forward ANN model. Furthermore, BFO algorithm has the powerful ability of searching global optimal solution which enhances ANN learning and improve the prediction accuracy of the trained ANNs.

6. REFERENCES

[1] Zhang Chunkai, Huihe Shao, Yu Li, "Particle swarm optimisation for evolving artificial neural network", IEEE International Conference on Systems, Man and Cybernetics, Volume 4, 2000.

[2] R.Vijay, "Intelligent Bacterial Foraging Optimization Technique to Economic Load Dispatch Problem", International Journal of Soft Computing and Engineering, Volume 2, Issue 2, May 2012.

[3] Hanning Chen, Yunlong Zhu, Kunyuan Hu, "Adaptive Bacterial Foraging Optimization", Hindawi Publishing Corporation , Abstract and Applied Sciences, Volume 2011, Article ID 108269, 27 Pages

[4] Swagatam Dass, Arijit Biswas, Sambarta Dasgupta, Ajith Abraham, "Bacterial Foraging Optimisation Algorithm: Theoretical Foundations, Analysis and Applications", Foundation of Computer Intelligence, Vol. 3 SCI 203, pp. 23-55

[5] Vipul Sharma, SS. Pattnaik, Tanuj Garg, "A Review of Bacterial Foraging Optimization And its Application", National Conference on Future Aspects of Artificial Intelligence in Industrial Automation (NCFAAIIA 2012), Proceedings Published by International Journal of Computer Applications

[6] Riya Mary Thomas, "Survay of Bacterial Foraging Optimization Algorithm", International Journal of Science and Modern Engineering, Volume 1 Issue 4, March 2013

[7] Zhang Dan, "Improving the accuracy in software effort estimation using artificial neural network model based on particle swarm optimization", IEEE International conference on Service Operations and Logistics and Informatics, 2013.

[8] Kewin Li, Jisong Kou, Lina Gong, "Predicting Software Quality by Optimised BP Network Based on PSO", Journal of computer Science, Vol. 6, No. 1, January 2011.

[9] Abdul Hamed, Haza Nuzly, "Particle Swarm Optimization for neural network learning enhancment", Masters Thesis,University Technology Malaysia , 2006.

[10] Kayarvizhy N, Kanmani S, Rhymend Uthariaraj V, "Improving Fault prediction using ANN-PSO in object oriented systems", International Journal of computer Application (0975-8887) Volume 73, No. 3, July 2013

[11] Al-Qasem Al-Hadi, Ismail Ahmed, "Bacterial foraging optimization Algorithm for neural network learning enhancement", Master's Thesis, University Technology Malaysia., 2011

[12] Ismail Ahmed Al-Qasem Al-Hadi, Siti Zaiton Mohamed Hashim, "Bacterial Foraging Optimization Algorithm for Neural Network Learning Enhancement", International Journal of Innovative Computing,Vol. 1, No. 1, 2012.

[13] Alireza Askarzadeh, "Artificial Neural network training using a new efficient optimization algorithm", Applied Soft Computing, Volume 13, Issue 2, February 2013, Pages 1206-1213.

[14] Solmaz Farshidpur, Farshid Keynia, "Using artificial bee colony Algorithm for MLP Training on software Defect Prediction", Oriental Journal of Computer science & Technology, Vol. 5 No.(2) Pgs. 231-239 December 2012

[15] Yan-Peng Liu, Ming-Guang Wu, Ji-Xin Qian, "Evolving Neural Networks Using the Hybrid of Ant Colony Optimization and BP Algorithms" ISNN 2006, LNCS 3971, pp.714-722, 2006.

[16] Amir Hatampur, Rasul Razmi,Mohammad Hossein Sedaghat, "Improving Performance of a neural network model by Artificial Ant colony Optimization for Predicting Permeability of Petroleum Reservoir Rocks" Middle-East Journal of Scientific Research 13(9): 1217-1223, 2013.

[17] John Paul T. Yusiong, "Optimizing Artificial Neural Networks using Cat Swarm Optimization Algorithm" I.J. Intelligent Systems and Applications, 2013, 01, 69-80.

[18] Jasmina Arifovic, Ramazan Gencay, "Using Genetic Algorithm to select architecture of a feedforward artificial neural network", Physica A 289 (2001) 574-594

[19] Sachin Sharma, Dr. Savita Shiwani, "Data Mining Based Accuracy Enhancement Of ANN Using Swarm Intelligence", International Journal of Communication and Computer Technologies, Volume 2, No. 9 Issue: 05 June 2014.

[20] Shafaatunnur Hasan, Tan Swee Quo, Siti Mariyam Shamsuddin, Roselina Sallehuddin, "Artificial Neural Network Learning Enhancement Using Artificial Fish Swarm Algorithm" Proceedings of the 3rd International Conference on computing and Informatics, ICOCI 2011, 8-9 June, 2011 Bandung, Indonesia.

[21] Ahmad AL Kawam, Nashat Mansour, "Metaheuritic Optimization Algorithms for Training Artificial Neural Networks", International Journal of Computer and Information Technology Volume 01-Issue 02, November 2012.

[22] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining Concepts and Techniques"

[23] http://promisedata.googlecode.com/svn/trunk/defect/ant-1.7/ant-1.7.csv