

Hybrid Ant Bee Colony Algorithm for Volcano Temperature Prediction

Habib Shah, Rozaida Ghazali, and Nazri Mohd Nawi

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia (UTHM)
Parit Raja, 86400 Batu Pahat, Johor, Malaysia
Habibshah.uthm@gmail.com,
{rozaida,nazri}@uthm.edu.my

Abstract. A social insect's techniques become more focus by researchers because of its nature behavior processing and by training neural networks through agents. Chief among them are Swarm Intelligence (SI), Ant Colony Optimization (ACO), and recently Artificial Bee Colony algorithm, which produced easy way for solving combinatorial problems and for training NNs. These social based techniques mostly used for finding optimal weight values in NNs learning. Usually, NNs trained by a standard and well known algorithm called Backpropagation (BP) have difficulties such as trapping in local minima, slow convergence or might fail sometimes. For recovering the above cracks the population or social insects based algorithms used for training NNs for minimizing network output error. Here, the hybrid of nature behavior agents' ant and bees combine's techniques used for training ANNs. The simulation result of a hybrid algorithm compared with, ABC and BP training algorithms. From the experimental results, the proposed Hybrid Ant Bee Colony (HABC) algorithm did improve the classification accuracy for prediction of a volcano time-series data.

Keywords: Swarm Intelligence, Ant Colony Optimization, Artificial Bee Colony, Hybrid Ant Bee Colony Algorithm, Back propagation.

1 Introduction

Neural Networks (NNs) a model of the biological neuron is the most important and suitable mathematical and scientific tool for solving a different combinatorial difficulty such as linear and non linear modeling, prediction, forecasting and classification [1-3]. It has powerful, flexible applications that have been successfully used in various applications such as statistical, biological, medical, industrial, mathematical, and software engineering. [4-6]. ANN learned their training techniques by parallel processing. NNs are capable of achieving many scientific research applications by providing optimal network architecture, activation function, input pre processing and optimal weight values.

Recently three NNs models, an ANN, radial-basis function, and recurrent Neural Networks (RNN) are used for the categorical prediction of seismic events occurring [7].

The probabilistic neural network can be used for prediction of earthquake magnitude but it does not yield a good result for prediction of magnitude greater than 7.5 [8]. The RNN technique can be used for prediction of the location and time of occurrence, but it is limited from 6.0 to M 7.5 for southern California earthquake [10].

ANN has applications in large range areas of human interests such as function approximation, process optimization, system identification, image processing, time series prediction, and intelligent control [9-14]. It can be trained by different training algorithms such as: BP, Improved BP algorithm, Evolutionary Algorithms (EA), Swarm Intelligence (SI), Differential Evolution (DE), and Hybrid Bee Ant Colony [15-23]. However, a BP learning algorithm has some difficulties; especially, it's getting trapped in local minima, where it can affect the NNs performance [24].

In order to overcome the drawbacks of standard BP, many algorithms used, which are based on mathematical approach, local and global optimization population techniques have been proposed for training the ANN, such as: Particle Swarm Optimization (PSO), ACO, (ABC-LM), ABC-MLP, HABC, Hybrid Bee Ant Colony (HBAC), Simulated Annealing (SA) and Tabu Search (TS)[14-24]. Recently population-based and Evolutionary algorithms are having reliable performance on training NNs [18].

In this study, the hybrid of two populations based algorithms; ABC and ACO HABC are proposed for training NNs for recovering the BP gap. These algorithms have recently been successfully applied in optimization problems and for training NNs [25]. The experimentation test is done by a volcano time-series data while the result compared with different approaches such as ABC and BP algorithms.

The rest paper is organized as follows: Related works given in Section 2. A brief review on ANN and BP training is given in Section 3. Section 4 contains on swarm intelligence briefing with subsection ABC and ACO algorithms. The proposed HABC algorithm is detailed in Section 5. Section 6 contains the simulation result with discussion. Finally, the paper concludes in Section 7.

2 Related Works

Volcanoes are the most impressive natural phenomena among seismic events, and people are captivated by their nature beauty as well as by their forceful eruptions. They exhibit a wide variety of eruption styles ranging from effusive eruptions typically resulting in lava flows or lava fountains, over medium sized explosive eruptions, to large eruptions with eruption columns of several tens of kilometers in height. Besides earthquakes, floods and storms, the volcanic eruptions present the largest natural hazards, and compared to earthquakes, floods and storms, they can even influence the earth's climate [26].

In this respect, NNs provides a quick and flexible approach for data integration and model development. To date, there have been a number of research advancements taken place in the area of NN's applications. Among them are automatic classifications of seismic signals at Mt. Vesuvius volcano, Italy and at Soufriere Hills volcano Montserrat [10, 28]. NNs are information processing paradigms that are

inspired by the way in which human brain processes information. NNs are very useful when the underlying problem is not clearly understood. Their applications do not require a priori knowledge about the system being modeled. Furthermore, they save data-storage requirements, since it is not required to keep all past data in the memory.

In this research, volcano time-series data are used for prediction task using the HABC training algorithm. The most important parameter of volcano eruption is the temperature is used here for simulation experiment. The data were obtained from the <http://www.climatedata.info/Forcing/Forcing/volcanoes.html>.

3 Artificial Neural Networks

ANNs are the most interested and understandable used in many tasks such as mathematical problems, statistical modeling by using different background of various data types. The accuracy makes this particular use of NNs as attractive to scientist analysts in various areas a different task as prediction, image processing, and other desired combinatorial task.

MLP is a universal approximate and mathematical model that contains a set of processing elements known as artificial neurons [11]. The network which is also known as feed forward neural network was introduced in 1957 to solve a non-linear XOR, and was then successfully applied to different combinatorial problems [16]. The basic building of MLP is constructed by neurons, which have some categories as input, hidden and output layers, as shown in Figure 1. The weight values between input and hidden nodes and between hidden and output nodes are randomly initialized. The network is highly used and tested on a different job. Figure 1 shows the basic architecture of Multilayer Perceptron [3].

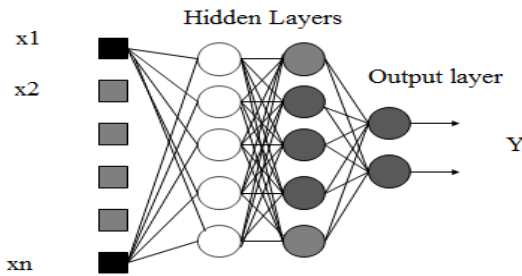


Fig. 1. Multilayer Perceptron Model

The output value of the MLP can be obtained by the following formula:

$$Y = f_i \left(\sum_{j=1}^n w_{ij} \cdot x_j + b_i \right) \tag{1}$$

Where Y is the output of the node x is the j_{th} input to the node, w is the connection weight between the input node and output node, b_i is the threshold (or bias) of the

node, and f_j is the node transfer function. Usually, the node transfer function is a non-linear function such as: a sigmoid function, a Gaussian functions. Network error function E will be minimized as

$$E(w(t)) = \frac{1}{n} \sum_{j=1}^n \sum_{k=1}^k (d_k - o_t) \quad (2)$$

Where $E(w(t))$ is the error at the t_{th} iteration; $w(t)$ is the weight in the connections at the t_{th} iteration; j shows training set, d_k is the desired output; o_t is the actual value of the k_{th} output node; k is the number of output nodes; and n is the no of patterns.

3.1 Backpropagation (BP) Learning Algorithms

BP is currently the most widely and well known used algorithm for training MLP. It was developed by Rumelhart [3]. The BP is a gradient descent method in which gradient of the error is calculated with respect to the weight's values for a given input by propagating the error backwards from output layer to hidden layer and further to input layer continuously. This step by step mathematical procedure adjusts the weights according to the error function. So, the adjustment of weights which decrease the error function is considered to be the optimal solution of the problem. In the input layer only inputs propagate through weights and passing through hidden layers and get output by some local information. For the BP error, each hidden unit is responsible for some part of the error.

Although the BP algorithm is a powerful technique applied to classification, combinatorial problems and for training MLP. However, as the problem complexity increases, the performance of BP falls off rapidly because gradient search techniques tend to get trapped at local minima. When the nearly global minima are well hidden among the local minima, BP can end up bouncing between local minima, especially for those non-linearly separable pattern classification problems or complex function approximation problem [24]. A second shortcoming is that the convergence of the algorithm is very sensitive to the initial value. So, it often converges to an inferior solution and gets trapped in a long training time.

4 Swarm Intelligence (SI)

Since the last two decades, SI has been the focus of many researches because of its unique behaviour inherent from the swarm insects [23, 28]. Bonabeau has defined the SI as "any attempt to design algorithm or distributed problem-solving devices inspired by the collective behaviour of nature insect colonies and other animal societies". He mainly focused on the behaviour of social insects alone such as termites, bees, wasps, and different ant species. However, swarms can be considered as any collection of interacting agents or individuals. Ants are individual agents of ACO [25].

4.1 Ant Colony Optimization (ACO)

ACO is a meta-heuristic procedure for the solution of a combinatorial optimization and discrete problems that has been inspired by the social insect's foraging behaviour

of real ant decision developed in 1990s [20]. Real ants are capable of finding Food Source (FS) by a short way through exploiting pheromone information, because ants leave pheromone on the ground, and have a probabilistic preference for trajectory with larger quantity of pheromone. Ants appear at a critical point in which they have to choose to get food, whether to turn right or left. Initially, they have no information about which is the best way for getting the FS.

Ants moves from the nest to the FS blindly for discovering the shortest path. The above behaviour of real ants has inspired ACO, an algorithm in which a set of artificial ants cooperate to the solution of a problem by sharing information. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds FS, it evaluates the quantity and the quality of the food and carries some of it back to the nest. The following is the ACO pseudo code.

```

Initialize Trail
Do While (Stopping Criteria Not Satisfied) - Cycle Loop
Do Until (Each Ant Completes a Tour) - Tour Loop
Local Trail Update
End Do
Analyze Tours
Global Trail Update
End Do
    
```

The parameters considered here are those that affect directly or indirectly the computation of the probability in formula:

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)][\eta(r,s)]^\beta}{\sum_{u \in j_k(r)} [\tau(r,u)][\eta(r,u)]^\beta} & \text{if } s \in j_k(r) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Where $s \in j_k(r)$ is the feasible Neighborhood of node r .

The following formula used for global updating rule: Once all ants have built their complete tours, pheromone is updated on all edges as follows:

$$\tau(r,s) \leftarrow (1 - \rho) \cdot \tau(r,s) + \sum_{k=1}^m \Delta \tau_k(r,s) \tag{4}$$

$\Delta \tau_k(r,s)$ is the amount of pheromone ant k puts on the coupling (i,j)

$$\Delta \tau_k(r,s) = \begin{cases} \frac{1}{L_k} & \text{if } (r,s) \in \text{tour done by ant } k. \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

Where m shows the number of ants, Q : a constant related to the quantity of a trail laid by ants as trail evaporation.

4.2 Artificial Bee Colony Algorithm (ABC)

ABC was proposed for optimization, classification, and NNs problem solution based on the intelligent foraging behaviour of honey bee [23]. The three bees determine the objects of problems by sharing information to other's bees. The employed bees use multidirectional search space for FS with initialization of the area. They get news and all possibilities to find FS and solution space. Sharing of information with onlooker bees is performed by employed bees. Onlooker bees: Onlooker bees evaluate the nectar quantity obtained by employed bees and choose a FS depending on the probability values calculated using the fitness values. Onlooker bees watch the dance of hive bees and select the best FS according to the probability proportional to the quality of that FS. Scout bees: Scout bees select the FS randomly without experience. If the nectar quantity of a FS is higher than that of the old source in their memory, they memorise the new position and forget the previous position. Whenever employed bees get a FS and use the FS very well again, they become scout bees to find a new FS by memorizing the best path. The detailed of ABC algorithm is shown as:

1. Initialize the population of solution's X_i where $i=1 \dots SN$
2. Evaluate the population
3. Cycle=1
4. Repeat from step 2 to step 13
5. Produce new solution v_{ij} in the neighbourhood of x_{ij} for the employed bees as

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \tag{6}$$

Where k is a solution in the neighbourhood of i , Φ is a random value between $[-1, 1]$.

6. Apply the Greedy Selection process for the onlookers between x_i and v_i .
7. Calculate the p_i for the solutions x_i by means of their fitness values by using eq (7).

$$P_i = \frac{fit_i}{\sum_{k=1}^{SN} fit_n} \tag{7}$$

Calculate of fitness values

$$fit_i = \begin{cases} \frac{1}{1 + f_i} & f_i \geq 0 \\ 1 + abs(f_i) & f_i < 0 \end{cases} \tag{8}$$

8. Normalize p_i values into $[0, 1]$
9. Produce the new solutions v_i for the onlookers from the solutions x_i , selected depending on P_i , and evaluate them.
Apply the Greedy Selection process for the onlookers between x_i and v_i
10. Determine the abandoned solution (source), if exists, replace it with a new randomly produced solution x_i for the scout using the following equation.

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \quad (9)$$

11. Memories the best FS position (solution) achieved so far.
12. Cycle=cycle+1
13. Until the cycle= Maximum Cycle Number (MCN)

5 Hybrid Ant Bee Colony (HABC) Algorithm

ABC and ACO were proposed for optimization, classification, and ANN problem solution based on the intelligent foraging behavior of honey bee and ant swarms [21, 23-25]. The ABC algorithm has a strong ability to find the global optimistic results optimal weight's values by bee's agents. It is successfully trained NNs for classification and prediction task. HABC combines the ACO properties in the ABC algorithm which may accelerate the evolving speed of ANNs and improve the classification precision of the well-trained networks. The hybrid algorithm is easily understandable, using an ABC algorithm to search the optimal combination of all the network parameters, and ACO used for selection best FS to find the accurate value of each parameter. HABC algorithm provides a solution in an organized form by dividing the agents into different tasks such as, employed bee, onlooker ant and scout bees. The detailed pseudo code of HABC algorithm is shown as follows:

1. Load colony size and food Number

$$FN = \frac{SN}{2} \quad (10)$$

King Bee {

If

$$FN = SN \% 2 = 0 \quad (11)$$

Then

$$FN = \frac{SN + 1}{2} \quad (12)$$

2. Initialize of solutions x_i
3. Evaluate the fitness of the population
4. Cycle =1
5. Repeat
6. Produce a new solution v_i by using equation

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \quad (13)$$

Calculate τ_i

$$\tau_{(i)} = \begin{cases} \tau \geq 0 & \text{for } \frac{Q}{1 + f_i} \\ \tau < 0 & 1 + \text{abs}(f_i) \end{cases} \quad (14)$$

$$Q = \frac{1 + FN}{\sum_{i=1}^{SN} (E + O)} \quad (15)$$

Apply greedy selection process

7. Calculate the probability values $P_{(i)}$ for the solution x_i by

$$P(i) = \frac{(0.09) * \tau(i)}{\sum_{1 \leq j \leq m} \tau(j)} \quad (16)$$

8. FOR each onlooker ant {

Select a solution x_i depending on P_i

Produce new solution v_i

9. Calculate the value τ_i by eq (14)

10. Apply greedy selection process}

11. Continue ABC from step 9 to 13.

Where Q , E and O shows numbers of Queen, Employed and Onlookers ant respectively, x_i represents a solution is the fitness solution o v_i indicates a neighbour solution of x_i , and P_i value of x_i . Also τ represents the fitness solution of trial i which is not improved and j represents the improved solution. In the algorithm, first half of the colony consists of employed ant, and the second half constitutes the onlooker ant. The Scout bee will be deciding the best values between onlookers and employed ant. In HABC algorithm, the position of a FS represents a possible solution to the optimization problem, and the nectar total of a FS corresponds to the fitness solution of the associated solution by King Bee. The King Bee initialized the colony size for employed and onlooker's ant. After initialization, the population of the positions (solutions) is subjected to repeated cycles, $C = 1, 2, \dots$ Maximum Cycle Number (MCN), of the search processes of the employed, onlooker and Scout Bee. An employed ant produces a modification on the position in her memory depending on the local information (visual information) and tests the nectar total (accurate solution) of the new source. The King Bee will gather employed and onlooker ant for decision of fitness solution. After all employed ant around complete the search process; they combine the nectar information of the FS and their position information with the onlooker ant on the food's area. An onlooker ant evaluates the nectar amount taken from all employed ant and chooses a FS with a probability related to its nectar quantity. The onlooker ant chooses a FS depending on the probability value associated with that FS, p_i , calculated by the eq (16). The King Bee initialized the colony size for bee. The FS will be divided in same quantity. King's bee will update

the FS for equal division on employed and onlookers' bee. The number of FS equals the half of the colony size and after division on employed and onlooker ant the will start searching for finding optimal FS. The information sharing with onlooker ant is performed by employed. An employed ant produces a modification on the source position in her memory and discovers a new FS position. Provided that the nectar quantity of the new source is higher than that of the previous source, the employed ant memorizes the new source position and forgets the old one. The onlooker ants evaluate the nectar quantity obtained by employed ants and choose a FS depending on the probability values. For this purpose, a fitness-based selection technique can be used. Onlooker ants watch the dance of hive ants and select the best FS according to the probability proportional to the quality of that FS.

6 Simulation Results and Discussion

In order to evaluate the performance of the HABC using volcano time-series data scheme for prediction, simulation experiments performed on a 1.66 GHz Core 2 Duo Intel Workstation with 2 GB RAM using Matlab 2010a software. The volcano temperature is used for the simulation. The comparison of standard BP-MLP, HABC and ABC algorithms are discussed based on the simulation results. The set of data contains about 1000 records are divided into two data sets 70% for training, 30% for testing. The learning rate (LR) and momentum (MOM) are set to 0.6 and 0.5 respectively. It should have to be noted that the ranges of weights are different for both experimentation as [1,-1], [10,-10] for BP, ABC and HABC. Furthermore, the minimum value of average of MSE selected for testing. The stopping criteria minimum error is set to 0.0001 BP-MLP while ABC and HABC stopped on MCN.

During the experimentation, five trials were performed for training. Each case, run was started with a same number of input parameters and with random foods and Area. The sigmoid function is used as activation function for network output. During the simulation, when the number of inputs, hidden and output nodes of the NNs and running time varies, the performance of training algorithms were stable, which is important for the designation of NNs in the current state. There is no specific convention for the decision of the number of hidden nodes. Here, we collect 2-2-1 network structure for BP, ABC and HABC.

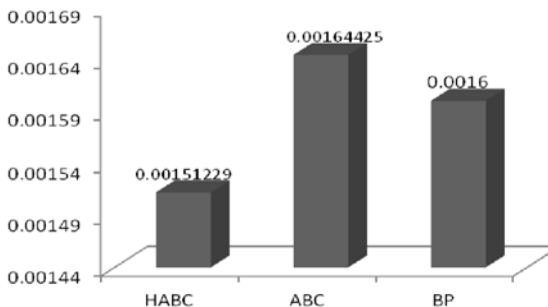


Fig. 2. Average MSE of Training HABC, ABC and BP

Finally, average of mean square errors (MSE), normalized MSE, Signal Noise Ratio, and CPU Time, are calculated for the above algorithms. Simulation results show the effectiveness and the efficiency of HABC algorithm. The comparison simulation of different network structure is presented in following figures.

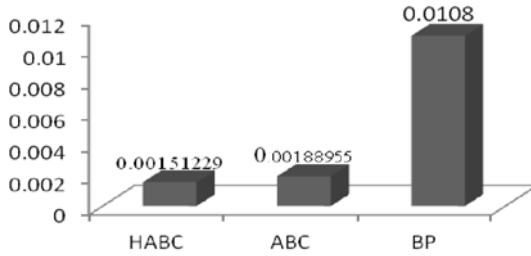


Fig. 3. Average MSE of Testing HABC, ABC and BP

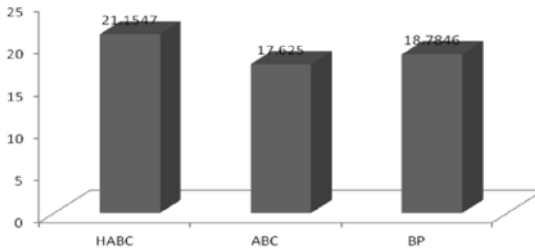


Fig. 4. Max SNR of HABC, ABC and BP algorithms

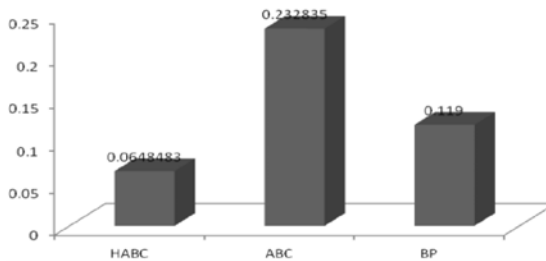


Fig. 5. Average of Normalized MSE for HABC, ABC and BP

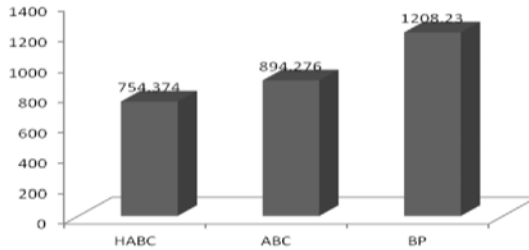


Fig. 6. CPU time for simulation of HABC, ABC and BP

In Fig 2 the average of five runs Mean Square Error is presented of HABC, BP and ABC training algorithms. The MSE of HABC is less as compared to ABC and BP for MCN 500 using volcano time series data. The testing results of the proposed HABC with BP and ABC are given in Fig 3, where the BP and ABC algorithm average MSE is greater than HABC. The SNR is shown in Fig 4, where the HABC algorithm shows higher than BP and ABC. The average of five NMSE of above algorithms is given in Fig 5. The CPU time of these algorithms with 500 MCN and five times running time are shown in Fig 6. The efficiency of HABC on the bases of MSE, NMSE, and CPU time shows outstanding performance than standard BP and ABC algorithms, especially in MSE for training and MSE for testing shows the best results for proposed HABC algorithm. Furthermore the SNR result shows the quality of the proposed approach.

7 Conclusion

The HABC algorithm merges the properties of the two nature swarm's ACO and ABC algorithm successfully, which are exploration and exploitation. The best possible weight values prove the high performance of the training MLP for time-series data prediction tasks using HABC algorithm. HABC has the great skill of searching global optimum solution. The optimum weights set of the algorithm may speed up the training process and improve the prediction accuracy. The above simulation results show that the proposed HABC algorithm can outstandingly learn the time-series data, which further extends the quality of the given approach with lower prediction error.

Acknowledgment. The authors would like to thank University Tun Hussein Onn Malaysia (UTHM) and Ministry of High Education (MOHE) for supporting this research under the FRGS.

References

1. Romano, Michele, A.U., et al.: Artificial neural network for tsunami forecasting. *Journal of Asian Earth Sciences* 36, 29–37 (2009)
2. Ghazali, R., Hussain, A.J., Liatsis, P.: Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals. *Expert Systems with Applications* 38(4), 3765–3776 (2011)

3. Rumelhart, D.E., McClelland, J.L.: And the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, 2. MIT Press, Cambridge (1986)
4. Osamu, F.: Statistical estimation of the number of hidden units for feedforward neural networks. *Neural Networks* 11(5), 851–859 (1998)
5. Shokri, A., Hatami, T., et al.: Near critical carbon dioxide extraction of Anise (*Pimpinella Anisum* L.) seed: Mathematical and artificial neural network modeling. *The Journal of Supercritical Fluids* 58(1), 49–57 (2011)
6. Thwin, M.M.T., Quah, T.-S.: Application of neural networks for software quality prediction using object-oriented metrics. *Journal of Systems and Software* 76(2), 147–156 (2005)
7. Taylor, S.R., Denny, M.D., Vergino, E.S., Glaser, R.E.: Regional discrimination between NTS explosions and earthquakes. *B. Seismol. Soc. Am.* 79, 1142–1176 (1989)
8. Adeli, H., Panakkat, A.: A probabilistic neural network for earthquake magnitude prediction. *Earthquake Engineering* 22, 1018–1024 (2009) ISBN- 0893-6080
9. Connor, J., Atlas, L.: Recurrent Neural Networks and Time Series Prediction. In: IEEE International Joint Conference on Neural Networks, New York, USA, pp. I 301–I 306.
10. Scarpetta, S., Giudicepietro, F., Ezin, E. C., Petrosino, S., Del Pezzo, E., Martini, M., and Marinaro, M.: Automatic Classification of seismic signals at Mt. Vesuvius Volcano, Italy using Neural Networks. *B. Seismol. Soc. Am.*, 95, 185–196, (2005)
11. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5), 359–366 (1989)
12. Cass, R., Radl, B.: Adaptive process optimization using functional-link networks and evolutionary algorithm. *Control Engineering Practice* 4(11), 1579–1584 (1996)
13. Gail, A. C.: Network models for pattern recognition and associative memory. *Neural Networks* 2(4), 243–257 (1989)
14. Tyan, C.-Y., Wang, P.P.: An application on intelligent control using neural network and fuzzy logic. *Neurocomputing* 12(4), 345–363 (1996)
15. Shah, H., Ghazali, R., Nawi, N.M.: Using Artificial Bee Colony Algorithm for MLP Training on Earthquake Time Series Data Prediction. *Journal of Computing* 3(6), 135–142 (2011)
16. Rosenblatt, F.: A Probabilistic Model for Information Storage and Organization in the Brain, vol. 65, pp. 386–408. Cornell Aeronautical Laboratory (1958)
17. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, NY (1999)
18. Yao, X.: Evolutionary artificial neural networks. *International Journal of Neural Systems* 4(3), 203–222 (1993)
19. Ilonen, J., Kamarainen, J.-K., et al.: Evolution Training Algorithm for Feed-Forward Neural Networks. *Neural Processing Letters* 17(1), 93–105 (2003)
20. Mendes, R., Cortez, P., Rocha, M., Neves, J.: Particle swarm for feedforward neural network training. In: Proceedings of the International Joint Conference on Neural Networks, vol. 2, pp. 1895–1899 (2002)
21. Dorigo, M., Di Caro, G., Corne, D., Dorigo, M., Glover, F.: The ant colony optimization meta-heuristic. *New Ideas in Optimization*, 11–32 (1999)
22. Ozturk, C., Karaboga, D.: Hybrid Artificial Bee Colony algorithm for neural network training. In: 2011 IEEE Congress on Evolutionary Computation, CEC 2011 (2011)
23. Karaboga, D., Akay, B., Ozturk, C.: Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) *MDAI 2007. LNCS (LNAI)*, vol. 4617, pp. 318–329. Springer, Heidelberg (2007)

24. Gori, M., Tesi, A.: On the problem of local minima in back-propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* 14(1), 76–86 (1992)
25. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press (2004)
26. Schneider, S.H., Easterling, W.E., Mearns, L.O.: Adaptation: Sensitivity to natural variability, agent assumptions and dynamic climate changes. *Climatic Change* 45, 203–221 (2000)
27. Tiira, T.: Detecting tele-seismic events using artificial neural networks. *Comput. Geosci.* 25, 929–939 (1999)
28. Chau, K.W.: Particle swarm optimization training algorithm for A NNs in stage prediction of Shing Mun River. *Journal of Hydrology*, 363–367 (2006)