InternationalJournal of ResearchStudies in ComputerScienceandEngineering (IJRSCSE)
Volume 4, Issue 4, 2017, PP 90-97
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: http://dx.doi.org/10.20431/2349-4859.0404011
www.arcjournals.org

The Effect of Bat Algorithm and Genetic Algortihm on the Training Performance of Artificial Neural Networks

Mehmet Hacibeyoglu[1], Mohammed H. Ibrahim[1], Kemal Alaykiran[2]

[1]*Department of Computer Science, Necmettin Erbakan University, Konya, Turkey*

[2]*Department of Industrial Science, Necmettin Erbakan University, Konya, Turkey*

Corresponding Author:** ***Mehmet Hacibeyoglu, *Department of Computer Science, Necmettin Erbakan University, Konya, Turkey*

Abstract:*Meta-heuristic algorithms have been successfully used in hard continuous optimization problems. The training of Artificial Neural Networks (ANNs) is one of these hard continuous optimization problems which has been solved in the literature using different optimization algorithms. In this study, a Bat algorithm (BA) and a Genetic Algorithm (GA) is proposed for the training of the ANNs. The performance of the algorithms has been tested with well-known seven datasets from UCI (University of California, Irvine) machine learning repository. The obtained results are compared with Back-propagation (BP) learning algorithm. It is figured out that the ANNs trained with BA ensures better performance than GA.*

Keywords:*Bat Algorithm, Genetic Algorithm, Artificial Neural Netwroks, Optimization*

1. Introduction

Artificial Neural Networks (*ANNs*) which is a branch of artificial intelligence is one of the frequently used classification algorithm to solve data mining problems in real applications [1]. Training process has great importance for the success of *ANNs* and several learning algorithms are proposed by researchers in the literature. The Back-propagation learning algorithm is generally used for the training process of *ANNs* in the literature. But *BP* learning algorithm may fall into local minimum and converges slowly which is caused by the neuron saturation in the hidden layer [2]. Recently, optimization algorithms have been widely used for the learning process of *ANNs* instead of *BP* learning algorithm by researchers. Various optimization algorithms with a global search feature are used to improve the performance of *ANNs* from being stuck to local minima problem. In [3], tabu search algorithm is examined as an alternative to problematic backpropagation algorithm and when the results derived from seven test functions are investigated, it is found out that tabu search algorithm yielded significantly better results than backpropagation solutions. [4] investigated the performance of a variation of hill climbing algorithm on artificial neural network training and compared the results to the performance of simulated annealing and standard hill climbing algorithms. *PSO* algorithm that is adopted by [5] is one of the most important algorithms that have the interesting performance for training ANNs. Another study has been done by for training *ANNs* using Convexity Based Algorithm (*CBA*). The results of the study prove that the CBA fill in a critical gap in utilizing of the classification algorithms [6]. In [7], researchers adopted a hybrid approach that combine *PSO* algorithm with gravitational search algorithm in order to solve the *ANNs* training problem. The new proposed training method reduced the problem of falling in local minima and slow convergence. The adopted method was compared to the other approaches. The results showed that the proposed method outperformed the others [7]. [8] considered taking advantage of both local search and global search algorithms. They merged PSO with the back-propagation algorithm. The statistical results proved *PSO* as a robust algorithm for training *ANNs.* The authors compared another method using modified differential evolution algorithm in order to train *ANNs*. In [9], the researchers used an adaptive differential evolution algorithm to train *ANNs*. The aim of the calculation is to find optimal weights. Adopted algorithm compared with the other method for different classification problems [9]. (Das et al., 2014) [10] used PSO algorithm for training *ANNs* for solving the problem of equalization channel. The result showed that the proposed equalizer achieved better results than the fuzzy equalizers in all conditions [10]. [11] used both PSO algorithm and another important nature inspired algorithm, ANN-Bee colony algorithm, for training *ANNs*. ANN-Bee colony algorithm was used for solving the problem predicting future hydropower generation values in Turkey. [12] adopted a hybrid approach including the idea of physics inspired gravitational search algorithm and biology inspired flower pollination algorithm. They combined these two algorithms to modify the velocity. The results for this adopted method presented to enhance the performance. Swarm intelligence algorithms which simulate behaviors of animals include many studies such as, [13] where, ant colony algorithm is used for learning *ANNs*. They proposed a developed ant colony algorithm in order to training *FFNN*. In [14], the researcher used a developed bat algorithm for training *ANNs*. Bat algorithm depends on the optimal solution in the velocity adjustment. This method is also used in a real life problem and showed good results. The Krill Herd algorithm is utilized for learning process of *ANNs* in [15]. In this work, the authors translated the positon of all weights and biases from *ANNs* into the vector. Mirjalili et al. utilized a Multiverse method that simulates nature-animals algorithm to global optimization [16]. [17] proposed an algorithm using the Gaussian method and the fuzzy method to improve the synaptic weight to FFNN and enhance FFNN layers. Using the Gaussian method enhanced the *PSO* convergence. After that, fuzzy reasoning rule used to delete unnecessary weights in *ANN*'s structure. The Iris data set is used to investigate the performance of the proposed method. Other PSO versions also developed to solve various problems; [18] adopted *PSO* classifier for detect possible damage in *RC* building. They adopted *PSO* method for detecting weights for the neural network model.

In this study, pattern classification problem of *ANNs* is considered providing two approaches *GA* and *BA*. The performance of the algorithms is tested on benchmark classification data sets and the test results of the two algorithms are compared. In the first part of this article, *ANNs* classification algorithm is explained in a demonstrated way. In the second part, the *ANNs* is explained. In the third part, the *BA* is demonstrated. In the fourth part, the *GA* is displayed. The training of *ANNs* using *BA* and *GA* is explained in the fifth part. In the sixth part, experimental results are discussed. The paper is finalized with conclusions.

2. Artificial Neural Netwroks (ANNs)

Artificial intelligence is defined as the ability of a machine to carry out tasks related to higher cognitive processes like reasoning, meaning extraction, generalization and learning from past experiences which are generally supposed to be human-specific characteristics. Generally, *ANNs* is used on classification problems of data mining [1]. *ANNs* based on neurons which are the structure of biological nerve cells in our brain, they can learn and make decisions according to learned information mechanisms. *ANNs* is composed of artificial neuron cells connected to each other hierarchically as shown in Figure 1.

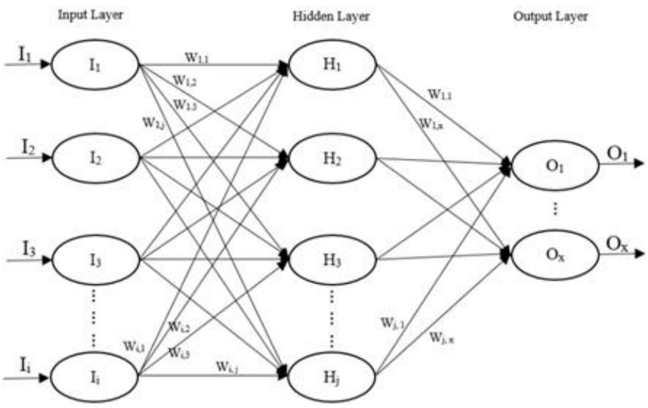

Fig1.*Structure of ANNs*

At Figure 1, $I_1, I_2, \ldots, I_i$ are known as inputs of *ANNs*, $H_1, H_2, H_3, \ldots, H_j$ represent the nodes in the hidden layer, $O_1, O_2, \ldots, O_x$ are known as outputs of *ANNs* and $W_{1,1}, W_{1,2}, \ldots, W_{1,j}$ are defined as weights. Each node output is calculated according to the following Equation (1) and the function of each output is calculated according to Equation (2) as shown below [20]:

$$MSE = \frac{1}{n}\sum_{x=1}^{n}(O_x - T_x)^2 \qquad (1)$$

where, *n*represents number of instances in the training data set, O_x is the calculated output of the x_{th} instance and T_x is the target output of the x_{th} instance. After the training process is over, the trained network can predict the target value of any unseen instance according to the last values of the weights.

3. BAT ALGORITH

The Bat Algorithm is a swarm intelligence optimization algorithm which was proposed by Yang in 2010, it is inspired by the behavior of the bats [22]. This algorithm utilizes the echo of the sound of the wounds which is called echolocation. Bats spread signals at a certain frequency and use echolocation for determining their location, communicate with each other, detect all kinds of objects even in completely dark environments, move without hitting them and distinguish any insects in the motion. They can identify the prey and barriers around their echolocation system. These values change the position of the bat in the frame again to get closer to the target. According to Yang, a bat algorithm takes place within the framework of the following rules [23].

1. All wounds detect the location of the prey by echolocation.
2. The velocity (v), position (x), frequency (f), wavelength (r) and sound output (A) are the values of each wound.
3. They can adjust the wave length and sound output with using the frequency, velocity and position which are calculated according to the following equations (4),(5) and (6), respectively.

$$f_i = f_{min} + (f_{max} - f_{min})\alpha \quad (4)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i \quad (5)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (6)$$

where, α is a random number in the range [0, 1], $f_i$is the frequency value of i_{th} bat, f_{min} and f_{max} are the minimum and maximum frequency values, respectively, v_i^t is the new velocity of i_{th} bat and x^* is the best solution in the population, x_i^t is the new position i_{th} bat. After selecting the best solution from the available solutions, a new solution is generated using local random walk by Equation (7).

$$X_{new} = X_{old} + \varepsilon L^t \quad (7)$$

where, ε represents a random number in the range [1,-1] and L^t represents the average loudness of all bats in the time *t*. While the bats are finding the prey, the signal propagation rate (*r*) increases and the loudness (*L*) decreases.

$$L_i^{t+1} = \beta L_i^t,\ r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \quad (8)$$

In Equation 8, β is a constant number in the range [0,1] and γ is a constant positive number. When $t \to \infty$, the loudness is $L_i^t \to 0$ and$r_i^t \to r_i^0$. The pseudo code of the bat algorithm is given in Figure 2 below [23],[24].

$Objectivefunction f(x), \quad x = (x_1, x_2, \ldots, x_d)^T$
$Initializethebatpopulation x_i and v_i$
$Definepulsefrequency f_i at x_i$
$Initializepulserate r_i andtheloudness A_i$
$\textbf{while}\ (t < \max numberofiterations)$
$Generatenewsolutionsbyadjustingfrequency,$
$updatest\ e\ velocitiesandt\ e\ positionsby\ (5)\ to\ (8)$
$\textbf{if}\ (rand > r_i)$
$Selectasolutionamongthebestsolutions$
$Generatealocalsolutionaroundtheselectedbestsolution$
$\textbf{endif}$
$Generateanewsolutionbyflyingrandomly$
$\textbf{if}(rand < A_i \& f(x_i) < f(x_*))$
$Acceptthenewsolutions$
$Increase r_i andreduce A_i$
$\textbf{endif}$
$Rankthebatsandfindthecurrentbest x_*$
$\textbf{endwhile}$

Fig2.*The pseudocoe of BA*

4. Genetic Algortihm

The *GA* is developed by John in 1975 and influenced from the evolution and change in living beings and carried out this genetic evolution process to the computer environment [25]. The *GA* uses a search technique to find the optimal solutions and GA is classified under global search meta-heuristics, search for the best solution in a multi-dimensional search space. As the population passes from generation to generation, bad solutions disappear and good solutions are used to produce the best solutions. In each population, the smallest genetic unit carrying genetic information is known as the gene. When one or more genes come together the chromosomes occur [26]. This algorithm uses the three of the basic operations used in genetics: Chromosome Selection operations, Crossover operation and Mutation operation [27]. In chromosome selection operation, the best chromosomes are selected from individual ones to achieve a stronger population [28]. The roulette wheel selection, Boltzman selection, tournament selection and rank selection are the chromosome selection methods. The crossover and mutation operations are actually basic methods however they play an important role in a genetic change to maintain diversity in the population. In crossover operation, the structure of two selected chromosome is varied from one generation to the next. The single point crossover, two point crossover, uniform crossover and arithmetic crossover can be used for the crossover operation. Insert mutation, inversion mutation, scramble mutation, swap mutation, flip mutation, interchanging mutation, reversing mutation, uniform mutation and creep mutation are the mutation methods [28]. One of them can be used according to the encoding of chromosome of the type of the problem. The pseudo code of the *GA* is given below in Figure 3.

```
Generate initial population
Evaluate the fitness of all individuals
Repeat
  Select the best chromosomes
  Crossover operaiton
  Mutation operation
  Evaluate the fitness of new population
Until the stop criteria is met
```

Fig3. *The pseudocoe of GA*

5. Training of ANNs by BA and GA

In general, training of ANNs is carried out with the BP learning algorithm. In this work, the BA and GA optimization algorithms are used with global search capability to train ANNs by finding the optimum weights. The pseudo code of training ANNs with BA and GA optimization algorithms is given below in Figure 4.

```
Determine the parameters of BA and GA
Initialize BA and GA with random populations
Calculate the fitness function values with MSE
Determine the best solution from population
Repeat
   Run BA and GA for updating the populationCalculate the fitness function with MSE
   Update best solution
Until stopping criteria is met
```

Fig4. *The pseudocoe of training ANNs with BA and GA*

It is seen in Figure 4 that the *BA* and *GA* are initialized with random populations. The fitness values of the individual solutions are calculated with MSE which is the fitness function of the optimization algorithms and the best solution is selected among these individual solutions which may be the result of the algorithm. Until the stopping criteria is met both algorithms are executed according to their structure which are explained in Section 3 and Section 4, respectively.

6. THE EXPERIMENTAL RESULTS

GA and *BA* algorithms are implemented using Visual Studio *C#.NET* 2015 version for training *ANNs*. The structure of the *ANNs* represented by *I-H-O*, where *I*is the number of input nodes at the same time it represents the number of attributes of the data set, *H* is the number of hidden nodes and *O* is the number of output nodes also it represents the number of classes of the data set. The number of weights in an *ANNs*structure represents the number of problem dimension calculated by$(I * H) + (H * O) + H + O$ [29]. For both *BA*and*GA* algorithms the initial weights are assigned as a random number in range [-10,10]. The number of population and the number of iterations are set to 30 and 5000, respectively. In addition to these common parameters, each algorithm has its own parameters. For *BA*, C_1 and C_2 are set to 1.48 [30]. For *GA*, mutation rate is set to 0.20, crossover rate is set 0.40 and roulette wheel selection is used for chromosome selection. All parameters of *GA* are determined empirically. In the experiments, 7 dataset are used from UCI machine learning repository whose properties are given in Table 1 below.

Table1.*Properties of datasets*

Dataset Name	Number of				ANNs Structure
	DA	CA	C	S	
Iris	0	4	3	150	4-6-3
Glass	0	9	6	214	9-12-6
New Thyroid	0	5	3	215	5-3-3
Ionosphere	0	33	2	351	33-5-2
Balance-Scale	4	0	3	625	4-4-3
Breast cancer	0	9	2	699	9-8-2
Diabetes	0	8	2	768	8-8-2

DA = Discrete Attributes, CA = Continuous Attributes, C = Classes, S = Samples

For determining the performance of the proposed algorithms for training *ANNs*, all experimental results are compared according to the criterion accuracy, sensitivity, specificity, precision and F-measure which are calculated according to the confusion matrix and are shown in Equations (9), (10), (11), (12) and (13) below.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{9}$$

$$Sensitivity = TP / (TP + FN) \tag{10}$$

$$Specificity = TN / (TN + FP) \tag{11}$$

$$Precision = TP / (TP + FP) \tag{12}$$

$$F - Measure = 2 * \frac{(Sensitivity * Precision)}{(Sensitivity + Precision)} \tag{13}$$

where, TP is true positive, TN is true negative, FN is false negative and FP is false positive.

10-Fold Cross validation is used for classification. The experimental results of *BA* and *GA* are compared to *BP* learning algorithm which is given in Table 2 below:

Table2. *The Exoerımental Results of GA and BA*

Data set	Method	Accuracy	Sensitivity	Specificity	Precision	F-Measure
Iris	BP	97.8	96.7	98.3	96.7	96.7
	GA	96.4	94.7	97.3	94.7	94.7
	BA	**98.2**	**97.3**	**98.7**	**97.3**	**97.3**
Glass	BP	**91.5**	**70.2**	**95.0**	**70.2**	**70.2**
	GA	85.2	55.6	91.1	55.6	55.6
	BA	88.0	64.0	92.8	64.0	64.0
New Thyroid	BP	**97.5**	**96.3**	**98.1**	**96.3**	**96.3**
	GA	95.3	93.0	96.5	93.0	93.0
	BA	97.2	95.8	97.9	95.8	95.8
Ionosphere	BP	**90.6**	**90.6**	**90.6**	**90.6**	**90.6**
	GA	87.2	87.2	87.2	87.2	87.2
	BA	89.7	89.7	89.7	89.7	89.7

Balance-Scale	BP	**95.3**	**93.0**	**96.5**	**93.0**	**93.0**
	GA	93.4	90.1	95.0	90.1	90.1
	BA	94.1	91.2	95.6	91.2	91.2
Breast cancer	BP	95.4	95.4	95.4	95.4	95.4
	GA	96.7	96.7	96.7	96.7	96.7
	BA	**97.1**	**97.1**	**97.1**	**97.1**	**97.1**
Diabetes	BP	72.9	72.9	72.9	72.9	72.9
	GA	76.6	76.6	76.6	76.6	76.6
	BA	**76.7**	**76.7**	**76.7**	**76.7**	**76.7**

When we analyze the results given at Table 2 the *BP* and *BA* algorithms are showed a competitive performance. For datasets *Glass*, *New Thyroid*, *Ionosphere* and *Balance Scale* the *BP* algorithm yielded the best results for all criteria and showed the second best performance only for *Iris* dataset. If the results obtained by the *BA* algorithm are considered, the *BA* algorithm performed the best results 3 out of 7 datasets which are Iris, Breast Cancer and Diabetes. Besides, the *BA* algorithm yielded the second best results for the rest of the datasets. The results of *GA* showed that this algorithm obtained promising results for all of the datasets while *GA* did not produce the best results for any of the datasets. However for datasets Breast Cancer and Diabetes *GA* performed as the second algorithm.

7. Conclusions

In the literature, metaheuristic optimization algorithms and classical solution methods are available to train *ANNs* which is a hard optimization problems. The main problem with classical solution methods like *BP* learning algorithm is sticking into local minima. In order to overcome this problem, metaheuristic optimization algorithms are proposed. In this study, we proposed *BA* and *GA* to train ANNs. The experimental results showed that the *BA* yielded *GA* for all datasets. Moreover, the results of *BA* and *BP* are competitive. For future work, meta-heuristic methods may be used in hybrid for improving the accuracy of the classification.

References

[1] Han, J., Pei, J. and Kamber, M. (2011). Data mining: concepts and techniques.

[2] Kapanova, K. G., Dimov, I. and Sellier, J. M. A genetic approach to automatic neural network architecture optimization. Neural Computing and Applications, 1-12.

[3] Sexton, R. S., Alidaee, B., Dorsey, R. E. and Johnson, J. D. (1998). Global optimization for artificial neural networks: A tabu search application. European Journal of Operational Research, 106(2-3), 570-584.

[4] Chalup, S., & Maire, F. (1999). A study on hill climbing algorithms for neural network training. In Evolutionary Computation, CEC 99. 3, 2014-2021.

[5] Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning, 760-766.

[6] Pham, H. N. A. and Triantaphyllou, E. (2011). A meta-heuristic approach for improving the accuracy in some classification algorithms. Computers & Operations Research, 38(1), 174-189.

[7] Mirjalili, S., Hashim, S. Z. M. and Sardroudi, H. M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Applied Mathematics and Computation, 218(22), 11125-11137.

[8] Yaghini, M., Khoshraftar, M. M. and Fallahi, M. (2013). A hybrid algorithm for artificial neural network training. Engineering Applications of Artificial Intelligence, 26(1), 293-301.

[9] Ahadzadeh, B. and Menhaj, M. B. (2014). Training neural networks using modified differential evolution algorithm for classification problems. In Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on (pp. 598-603).

[10] Das, G., Pattnaik, P. K. and Padhy, S. K. (2014). Artificial neural network trained by particle swarm optimization for non-linear channel equalization. Expert Systems with Applications, 41(7), 3491-3496.

[11] Uzlu, E., Akpınar, A., Özturk, H. T., Nacar, S and Kankal, M. (2014). Estimates of hydroelectric generation using neural networks with the artificial bee colony algorithm for Turkey. Energy, 69, 638-647.

[12] Chakraborty, D., Saha, S. and Maity, S. (2015, February). Training feedforward neural networks using hybrid flower pollination-gravitational search algorithm. In Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on (pp. 261-266).

[13] Salama, K. M. and Abdelbar, A. M. (2015). Learning neural network structures with ant colony algorithms. Swarm Intelligence, 9(4), 229-265.

[14] Jaddi, N. S., Abdullah, S. and Hamdan, A. R. (2015). Optimization of neural network model using modified bat-inspired algorithm. Applied Soft Computing, 37, 71-86.

[15] Kowalski, P. A., & Łukasik, S. (2016). Training neural networks with krill herd algorithm. Neural Processing Letters, 44(1), 5-17.

[16] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27(2), 495-513.

[17] Melo, H. and Watada, J. (2016). Gaussian-PSO with fuzzy reasoning based on structural learning for training a Neural Network. Neurocomputing, 172, 405-412.

[18] Chatterjee, S., Sarkar, S., Hore, S., Dey, N., Ashour, A. S. and Balas, V. E. (2017). Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings. Neural Computing and Applications, 28(8), 2005-2016.

[19] Hagan, M. T., Demuth, H. B., Beale, M. H. and De Jesús, O. (1996). Neural network design, 20.

[20] Ismailov, V. E. (2014). On the approximation by neural networks with bounded number of neurons in hidden layers. Journal of Mathematical Analysis and Applications, 417(2), 963-969.

[21] Haykin, S. and Network, N. (2004). A comprehensive foundation. Neural Networks, 2.

[22] Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010), 65-74.

[23] Yılmaz, S., & Küçüksille, E. U. (2015). A new modification approach on bat algorithm for solving optimization problems. Applied Soft Computing, 28, 259-275.

[24] Nawi, N. M., Rehman, M. Z. and Khan, A. (2014). A new bat based back-propagation (BAT-BP) algorithm. In Advances in Systems Science, 395-404

[25] Holland, J. H. (1975). Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press.

[26] Man, K. F., Tang, K. S. and Kwong, S. (1996). Genetic algorithms: concepts and applications [in engineering design]. IEEE transactions on Industrial Electronics, 43(5), 519-534.

[27] Gen, M. and Cheng, R. (2000). Genetic algorithms and engineering optimization, 7 (Vol. 7). John Wiley & Sons.

[28] Soni, N. and Kumar, T. (2014). Study of various mutation operators in genetic algorithms. International Journal of Computer Science and Information Technologies, 5(3), 4519-4521.

[29] IBRAHIM, M. H., JIHAD, K. H., & Kamal, L. L. (2017). Determining Optimum Structure for Artificial Neural Network and Comparison between Back-Propagation and Levenberg-Marquardt Training Algorithms. International Journal of Engineering Science, 14887.

[30] Li-Ping, Z., Huan-Jun, Y., & Shang-Xu, H. (2005). Optimal choice of parameters for particle swarm optimization. Journal of Zhejiang University-Science A, 6(6), 528-534.

AUTHORS' BIOGRAPHY

Mehmet Hacibeyoglu, is currently Assistant Professor at the Computer Engineering Department of the Konya Necmettin Erbakan University (Konya, Turkey). He received the B.Sc., M.Sc., and Ph.D. Degrees in Computer Engineering from the Selcuk University (Konya, Turkey) in 2003, 2006, and 2012, respectively. His research interests are system administration in Unix and Linux, machine learning, feature selection, logic circuits, meta heuristic algorithms, information security, and data mining.

Mohammed Hussein, IBRAHIM is a doctoral student and same time is a lecturer in the department of computer engineering, Necmetin Erbakan University, Konya, Turkey. He received his bachelor of engineering BS degree in software engineering in 2003 from Kirkuk University, Kirkuk, Iraq. Master of technology degree in computer engineering in 2011 from Selcuk University, Konya, Turkey. His research interests include artificial intelligence technique, Optimization algorithms, fuzzy logic, artificial neural network and computer architecture.

Kemal Alaykiran, is currently working as an Assistant Professor at the Industrial Engineering Department of Necmettin Erbakan University (Konya, Turkey). He received the B.Sc., M.Sc., degrees in Industrial Engineering at Selcuk University and Ph.D. Degree in Industrial Engineering at Gazi University (Ankara, Turkey). His research interests are mathematical modeling, optimization and simulation of systems.

Citation: *Mehmet Hacibeyoglu et al. (2017). The Effect of Bat Algorithm and Genetic Algortihm on the Training Performance of Artificial Neural Networks, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 4(4), pp.90-97, DOI: http://dx.doi.org/10.20431/2349-4859.0404011*