Hybrid Global Artificial Bee Colony Algorithm for Classification and Prediction Tasks

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@uthm.edu.my
nazri@uthm.edu.my

Abstract
From last ten years, computer scientists show the interest in the study of social insect’s behavior algorithms like, Artificial Bee Colony (ABC), Ant Colony Optimization. Chief among of them, Standard ABC is well-known and new swarm optimization technique used for solving different combinatorial problems; however, it is often trapped in local optima in global optimization. This study investigates the new hybrid technique called Global Artificial Bee Colony-Backpropagation (GABC-BP) algorithm. This new technique shows great advantages of convergence property and excellent solution. To hybrid technique, GABC-BP algorithm used to this work for classification and prediction task. The performance of GABC-BP is benchmarked against BP, ABC and GABC. The experimental result shows that GABC-BP performs better than that standard BP, ABC and GABC for Boolean function classification and heat wave's temperature time series prediction.

Keywords- Global Artificial Bee Colony, Backpropagation, Hybrid Artificial Bee Colony.

1. INTRODUCTION

Artificial Neural Network (ANN) is inspired by the functioning of the human biological neural networks has provided an exciting alternative method for solving a variety of problems in different areas of science and engineering technology. “A neural network is a massively parallel distributed processor who has a natural propensity for storing experimental knowledge and making it more available for use.” It resembles the brain in two respects 1) knowledge is required for network through a learning process 2) interneuron connection strength known as synaptic weights are used to store knowledge” (Haykin, 1999). Neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes”(Darpa, 1988).

ANN can often provide a suitable solution for problems that generally are characterized by nonlinear, high dimensionality, noisy, complex, imprecise, imperfect and/or error-prone sensor data, poorly understood physical and statistical models, and lack of clearly stated mathematical solution or algorithm (Zaknich, 2003). Mostly, ANN approaches are capable of scientific, electrical engineering, earth knowledge, mathematics and of course for the
The well known approaches of ANN showed the best performance in different problems such as image processing (Dunstone, 1994; Leung et al., 1990; Naka et al., 1993) pattern and speech recognition (Filippi et al., 1994; Kumar et al., 2009; Min & Jianhui, 2002; Tsenov & Mladenov, 2010; Yonghong et al., 1997), smooth function approximation (Ferrari & Stengel, 2005), weather forecasting (Niska et al., 2005), volcanoes classification (Falsaperla et al., 1996), earthquake magnitude prediction, clustering (K.-L., 2010), scheduling (Chang-Shui et al., 1991; Changshui et al., 1992; Yong et al., 2009), feature selection techniques (Hofmann et al., 2004; Kavzoglu & Mather, 2000), constrained (Youshen et al., 2002) and unconstrained problems (Seong-Whan, 1995).

Learning in the context of ANN is defined as a process by which the free parameters of ANN are adapted through a process of presenting signals from the environment in which the network is embedded. The type of learning is determined by the way the parameter changes take place (Haykin, 1999). The notion of learning in ANN, is the process of guiding the network to provide a particular output or response for a specific given input. Learning is necessary when information about the input-output relationship is unknown or incomplete a-priori.

If the ANN finds the behavior of data, optimal weight values with suitable activation function, then performance will be appropriate for the different task such as prediction, classification, pattern recognition, clustering and scheduling. With the absence of the above four categories, the ANN can be trapped in local minima; effect on convergence speed and sometimes networks might be failing. The learning object is to minimize the cost function which is the difference between desired output and neural network output. The network will train for finding optimal weights, which reduce the error until the convergence.

From the last decade, there have been several types of computational, hybrid, local search, global search, mathematical, biological-inspired meta heuristic algorithms developed for classification and time series prediction, these are, ant colony optimization (ACO) inspired by the foraging behavior of ant colonies (Dorigo & Stützle, 2003), particle swarm optimization (PSO) inspired by the social behavior of bird flocking or fish schooling (Kennedy & Eberhart, 1995), and Hybrid Ant Bee Colony Algorithm inspired by the foraging behavior of ant and bee colonies (Shah et al., 2012) Hybrid Artificial Bee Colony inspired by the foraging behavior of bees and Levenberg–Marquardt algorithm, Global Artificial Bee Colony, Gbest_Guided Artificial bee Colony, Multiple Gbest_Guided Artificial Bee Colony, Improved Artificial Bee Colony, and so many other bio inspired population based algorithms (Karaboga et al., 2007; Karaboga & Ozturk, 2011; Shah et al., 2012; Shah et al., 2012; Shah et al., 2013; Shah et al., 2012; Zhu & Kwong, 2010).

Recently, Artificial Bee Colony Algorithm has been very famous and attracted to solve many kinds of problems besides classification and prediction like clustering, numerical function optimization, scheduling, software testing, training Artificial Neural Networks and Higher Order Neural Networks for earthquake magnitude prediction, heat waves temperature, forecasting, modeling and so many other tasks (Ghazali et al., 2007). The performance of ABC shows that compared with GA, ACO, BP, and PSO, ABC algorithm can obtain better quality solutions. For getting high efficiency, researchers have used the hybridization method of standard and improved ABC algorithms with BP, PSO, ACO, Differential Evolution (DE), GA, etc, for different tasks (Ozturk & Karaboga,
Global Artificial Bee Colony (GABC), is population-based algorithms that can provide the best possible solutions for different mathematical problems by using inspiration techniques from honey bees (Shah, et al., 2013). Backpropagation (BP) is a well known supervised form learning algorithm of obtaining the many weight's values in ANN applications, devolved by (Rumelhart et al., 1986.). BP algorithm is widely used to solve many engineering modeling problems (Najafzadeh et al., 2013; Zweiri et al., 2002). The basic BP algorithm is based on minimizing the error of the network using the derivatives of the error function. The BP used to adjust the network’s weight and threshold so to minimize the error for the different task such as classification, clustering and prediction on the training set (Abunawass et al., 1998; Bin Mohd Azmi & Cob, 2010). The hybridization of BP with swarm intelligence algorithms makes it more attractive for further improvement.

In this research article, two famous learning algorithms, Global Artificial Bee Colony and Backpropagation are hybrid called HGABC algorithm. It is hoped that the proposed learning algorithm will cover the drawbacks of standard ABC and BP. The hybrid technique is proposed here for Boolean classification problems and heat waves temperature time series prediction. The performance of the algorithm is compared with standard BP, ABC, and GABC algorithms.

2. RELATED WORKS

In this research paper two solution presented by proposed and standard learning algorithms, are Boolean Function Classification and Time Series Prediction. Each task discussed in the following subsections.

a) Boolean Function Classification

There are many types of Boolean operations; however, commonly used are XOR, 3-Bit Parity and Encoder / Decoder operators. XOR is not linearly separable, consequently it cannot be implemented using single layer network; a three layered network is required to solve the problem. The 3-bit parity or even parity problem can be considered as a generalized XOR problem but it is more difficult for classification (Stork & Allen, 1992). This problem has been addressed because they are nonlinearly separable, and hence can not be solved by Single Layer Perceptron (SLP), such as the perceptron (Minsky & Papert, 1969). In other words, if the number of binary inputs is odd, the output is 1, otherwise it is 0 (V.P. Plagianakos, 1998). The third classification task is 4-Bit Encoder/Decoder, which is well-known in computer science. The network is presented with four distinct input patterns, each having only one bit turned on. A decoder is a logic circuit which accepts a set of inputs that represents a binary number and activates only the output that corresponds to the input number. 4-Bit Encoder/Decoder is quite close to the real-world pattern classification task, where small changes in the input pattern cause small changes in the output pattern (Fahlman, 1988).

b) Heat Waves Temperature Prediction

The Oklahoma City, UK, daily heat wave's temperatures for up to four months of May, June, July, August and September of the year of 2000, 2006, 2011 and 2012 will be used for prediction. The prediction is based on its pattern which is heat wave temperatures in Fahrenheit. The heat waves dataset of temperature derived from the
National Oceanic and Atmosphere Administration (NOAA) (NOAA, 2012). In ANN learning, the network has one input pattern with one output pattern is the heat wave temperature.

3. **Global Artificial Bee Colony (GABC) algorithm**

Global Artificial Bee Colony algorithm is the advance version of standard ABC, has been successfully applied for training ANN and numerical function optimization and so on (Peng et al., 2011; Shah, et al., 2013). GABC has improve the exploitation procedure of standard ABC with global best neighbour information updates rules. There are three steps involved in GABC algorithm in standard employed, onlookers and scout bees of ABC. They become global employed, onlookers and scout bees through GABC strategy. The modified steps are:

**Step 1:** It modifies the employed section as

\[
v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + y
\]

\[
y = c_{1}\text{rand}(0,1)(x_{j}^{\text{best}} - x_{ij}) + c_{2}\text{rand}(0,1)(y_{j}^{\text{best}} - x_{ij})
\]

(1) (2)

**Step 2:** Repeat the above formula with onlookers section.

Where \(y\) shows Best Food Source, \(c_{1}\) and \(c_{2}\) are two constant values, \(x_{j}^{\text{best}}\) is the \(j\)-th element of the global best solution found so far, \(y_{j}^{\text{best}}\) is the \(j\)-th element of the best solution in the current iteration, \(\phi_{ij}\) is a uniformly distributed real random number in the range \([-1, 1]\).

**Step 3:** Modified the scout section as:

\[x_{ij}^{\text{rand}} = x_{ij}^{\text{min}} + \text{rand}(0,1)(x_{ij}^{\text{max}} - x_{ij}^{\text{min}})\]

If \(\text{rand}(0,1) \leq 0.5\), then

\[x_{ij}^{\text{mutation}} = x_{ij} + \text{rand}(0,1)\left(1 - \frac{\text{iter}}{\text{iter}_{\text{max}}}\right)^{b}(x_{j}^{\text{best}} - x_{ij})\]

Else

\[x_{ij}^{\text{mutation}} = x_{ij} + \text{rand}(0,1)\left(1 - \frac{\text{iter}}{\text{iter}_{\text{max}}}\right)^{b}(y_{j}^{\text{best}} - x_{ij})\]

(3) (4) (5)

Then comparing the fitness value of random generated solution \(x_{ij}^{\text{rand}}\) and mutation solution \(x_{ij}^{\text{mutation}}\) the better one is chosen as a new food source, where \(b\) is a scaling parameter which is a positive integer within the range of \([2, 5]\).

4. **Back Propagation Learning Algorithm**

Back-Propagation is one of the most attractive and novel supervised-learning algorithm proposed by Rumelhart, Hinton, & Williams (1986) for Multilayer Perceptron training. Due to its high rate of flexibility and learning capabilities, it has been successfully used in wide range of science and engineering applications. It can be proved that BP can reach the extreme within a limited number of epochs for a given task. The merits of BP are that the
adjustment of weights is always towards the descending direction of the error function and that only some local information is needed. The forward phase where the activations propagate from the input layer to the output layer. The backward phase, where then the observed actual value and the requested nominal value in the output layer are propagated backwards so it can modify the weights and bias values. Each node is composed of two sections. The first section generates a sum of the products of the weights multipliers and input signals. The second takes the result of the first section and puts it through its activation function, with scales the its input to a value between 0 and 1. In the feed-forward phase the input signals are propagated through the input and hidden layers of processing elements, generating an output pattern in response to the input pattern presented.

![Artificial Neural Network with BP learning algorithm](image)

*Figure 1: Artificial Neural Network with BP learning algorithm*

Each node, or artificial neuron (Threshold Logic Unit), is composed of two sections. The first section generates a sum of the products of the weights multipliers and input signals. The second section takes the result of the first section and puts it through its activation function, with scales the its input to a value between 0 and 1. Signal $e$ is the output of the First section, and $y = f(e)$ is the output of the second section. Signal $Y$ is also the output signal of an artificial neuron. There are several types of activation function, the most common activation function of a neuron $f(x)$ is a sigmoid function (Wang et al., 2004) as shown below:

$$f(\text{net}_j) = \frac{1}{1 + e^{\text{net}_j}}$$

where:

$\text{net}_j = \sum w_{ij} a_i$,

$a_i$ is the input activation from unit $i$, and

$w_{ij}$ is the weight connecting unit $i$ to unit $j$.

In the next algorithm step, the output signal of the network $y$ is compared with the desired output value (the target). The difference is called error signal of output layer neuron, which is calculated as:
\[ E = \frac{1}{2} \sum_{i=1}^{n} (t_k - y_k)^2 \quad (7) \]

\( E \) = error vector, \( t_k \) is the actual output and \( y_k \) is the network value. In order to derive the BP learning rule, chain rule use to rewrite the error gradient for each pattern as the product of partial derivatives. Thus, the error gradient becomes:

\[
\frac{\partial E}{\partial w} = \frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n} \quad (8)
\]

The partial derivative reflects the change in error as a function of the net input; the second partial derivative reflects the effect of a weight change on a change in the net input. By using the chain rule to with respect to weight and biases, in the BP algorithm determined as follows:

\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial S_i} \frac{\partial S_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} \quad (9)
\]

where \( w_{ij} \) is the weight from neuron \( j \) to \( i \), \( S_i \) represent the output of neuron, and \( net_i \) is the weighted sum of the inputs of neuron \( i \). The weight will update with the gradient rules, learning rate with derivative to minimize the error function as:

\[
W_{ij}(t+1) = W_{ij}(t) - \eta \frac{\partial E}{\partial W_{ij}}(t) \quad (10)
\]

where, \( W_{ij}(t+1) \) shows the new weight value, \( W_{ij} \) represent old weight values and \( \eta \) represents learning rate, which can control the learning and has important effect on convergence time. The learning rate is a constant used in error BP learning that affects the speed of learning. The smaller the learning rate, the more steps it takes to get to the stopping criterion. A too large or too small \( \eta \) will cause negative inferences to converge (Von et al., 1988). If it is too small, the learning process can be very slow (Knight, 1990). The different combinations of the learning rate and momentum are introduced to try to find the right combination that will allow the solution to escape local minima but not skip over the global solution. To make the learning process more stable, the momentum term is used to the weight changes as:

\[
\Delta W_{ij}(t) = -\eta \frac{\partial E}{\partial W_{ij}}(t) + \mu \Delta W_{ij}(t-1) \quad (11)
\]

where, \( \Delta W_{ij}(t) = W_{ij}(t) - W_{ij}(t-1) \) and the momentum term represents by \( \mu \), where the momentum factor \( 0 < \mu < 1 \), usually set to around 0.9 (Wasserman, 1989). Using high learning rate, momentum term can avoid the oscillation.
The Global Artificial Bee Colony (GABC) algorithm has a global ability to find global optimistic result, and the BP algorithm is a standard method, relatively with simple implementation and work very well (Shah, et al., 2012; Shah et al., 2012). Combining the step of GABC with BP, a new hybrid (HGABC) algorithm is proposed in this article for training MLP. The key point of this hybrid GABC-BP algorithm is that the GABC is used at the initial stage of searching for the optimum using global best methods, Then, the training process is continued with the BP learning algorithm (Peng, et al., 2011). The flow-diagram of the ABC-BP model is shown in Figure 2. In the initial stage, the ABC algorithm finishes its training procedures, then, the BP algorithm starts training with the optimal weights of GABC algorithm and then BP trains the network for 100 epochs more. The Figure 2, shows the flow chart for proposed HGABC algorithm as:
8. SIMULATION RESULTS AND DISCUSSION

In this work, swarms intelligent-based combine technique based on HGABC algorithm is used to train feed-forward artificial neural networks. In order to calculate the performance of the HGABC with ABC and BP algorithms in terms of Mean Square Error (MSE), Normalized Mean Square Error, Signal to Noise Ratio and success rate using the boolean function for classification, where simulation experiments performed by Matlab 2010a software. The stopping criteria minimum error is set to 0.0001 BP while ABC, GABC and HGABC stopped on MCN. During the experimentation, 10 trials were performed for training. 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. The value of $c_1$ and $c_2$ were selected 2.5. From the simulation experiment, the HGABC performance can be affected by $c_1$ and $c_2$. So the best values selected for these two constant values. Parameters setting is given in Table 1 and 2, respectively.

**Table 1:** Parameters of the problems for Boolean Function Classification

<table>
<thead>
<tr>
<th>Problem</th>
<th>Colony Range</th>
<th>NN structure</th>
<th>D</th>
<th>MCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>XOR6</td>
<td>[100, -100]</td>
<td>2-2-1 (No Bias)</td>
<td>6</td>
<td>7500</td>
</tr>
<tr>
<td>XOR9</td>
<td>[10, -10]</td>
<td>2-2-1+ Bias(3)</td>
<td>9</td>
<td>100</td>
</tr>
<tr>
<td>XOR13</td>
<td>[-10, 10]</td>
<td>2-2-1+ Bias(4)</td>
<td>13</td>
<td>75</td>
</tr>
<tr>
<td>3-Bit</td>
<td>[-10, 10]</td>
<td>3-3-1+ bias(4)</td>
<td>16</td>
<td>1000</td>
</tr>
<tr>
<td>Enc/Dec</td>
<td>[10, -10]</td>
<td>4-2-4+Bias(6)</td>
<td>22</td>
<td>1000</td>
</tr>
</tbody>
</table>

**Table 2:** Parameters of the problems for Heat Waves Temperature

<table>
<thead>
<tr>
<th>Problem</th>
<th>Colony Range</th>
<th>Hidden Nodes</th>
<th>Dimension (D)</th>
<th>MCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>[10, -10]</td>
<td>[2, 9]</td>
<td>[9, 49]</td>
<td>2000</td>
</tr>
<tr>
<td>GABC</td>
<td>[10, -10]</td>
<td>[2, 9]</td>
<td>[9, 49]</td>
<td>2000</td>
</tr>
<tr>
<td>BP</td>
<td>[10, -10]</td>
<td>[2, 9]</td>
<td>[9, 49]</td>
<td>2000</td>
</tr>
<tr>
<td>HGABC</td>
<td>[10, -10]</td>
<td>[2, 9]</td>
<td>[9, 49]</td>
<td>1900+100</td>
</tr>
</tbody>
</table>

The stopping criterion for ABC and GABC, GGABC and 3G ABC is 1000 Maximum Cycle Number (MCN) for boolean function classification and 2000 MCN for heat waves temperature time series prediction. During the experimentation, 10 trials performed in training. The values of $c_1$ and $c_2$ was selected 2.5. The sigmoid function used as activation function for network production for classification and prediction. During the simulation, when the number of input signals, hidden nodes, output node and running time varies, performing training algorithms were stable, which is important for delegation NNs in the current state. The average results for boolean function classification using standard and proposed learning algorithm given in Table 3. The success rate given in table 5.
Table 3: Average of Mean Square Error using ABC, BP, GABC, HGABC algorithms.

<table>
<thead>
<tr>
<th>Problems/Methods</th>
<th>ABC</th>
<th>BP</th>
<th>GABC</th>
<th>HGABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>XOR6</td>
<td>0.007051</td>
<td>0.1107</td>
<td>0.12076</td>
<td>0.000541</td>
</tr>
<tr>
<td>XOR9</td>
<td>0.006956</td>
<td>0.0491</td>
<td>0.000106</td>
<td>1.12E-12</td>
</tr>
<tr>
<td>XOR13</td>
<td>0.006079</td>
<td>0.0078</td>
<td>0.000692</td>
<td>2.21E-11</td>
</tr>
<tr>
<td>3-Bit Parity</td>
<td>0.006679</td>
<td>0.0209</td>
<td>0.00062</td>
<td>5.12E-09</td>
</tr>
<tr>
<td>4-bit End/Dec</td>
<td>0.008191</td>
<td>0.0243</td>
<td>0.000294</td>
<td>1.10E-09</td>
</tr>
</tbody>
</table>

Table 4: Success rate of ABC, BP, GABC and HGABC algorithm

<table>
<thead>
<tr>
<th>Problems/Methods</th>
<th>ABC</th>
<th>BP</th>
<th>GABC</th>
<th>HGABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>XOR6</td>
<td>100</td>
<td>6</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>XOR9</td>
<td>100</td>
<td>66.66</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>XOR13</td>
<td>100</td>
<td>96.66</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3-Bit Parity</td>
<td>100</td>
<td>73.33</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>4-bit End/Dec</td>
<td>100</td>
<td>73.33</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

The testing Mean Square Error for XOR6, XOR9, XOR13, 3-bit Parity and 4-bit encoder/decoder are 0.000221, 1.02E-09, 2.11E-09, 4.12E-05 and 1.10E-07 respectively using HGABC algorithm. The simulation results of Heat Waves Temperature for Prediction are given in table 5 and 6.

Table 5: MSE Best Average results by for Heat Waves Temperature Prediction

<table>
<thead>
<tr>
<th>NN Structure</th>
<th>ABC</th>
<th>BP</th>
<th>GABC</th>
<th>HGABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-2-1</td>
<td>0.009323</td>
<td>0.007185</td>
<td>0.009984</td>
<td>0.000122</td>
</tr>
<tr>
<td>2-3-1</td>
<td>0.002486</td>
<td>0.006887</td>
<td>0.000311</td>
<td>0.004810</td>
</tr>
<tr>
<td>3-3-1</td>
<td>0.002221</td>
<td>0.007283</td>
<td>0.000128</td>
<td>0.001023</td>
</tr>
<tr>
<td>3-5-1</td>
<td>0.002797</td>
<td>0.005643</td>
<td>0.001001</td>
<td>0.001081</td>
</tr>
<tr>
<td>4-4-1</td>
<td>0.008318</td>
<td>0.003872</td>
<td>0.000391</td>
<td>0.000110</td>
</tr>
<tr>
<td>4-5-1</td>
<td>0.005065</td>
<td>0.005652</td>
<td>0.000669</td>
<td>0.000368</td>
</tr>
<tr>
<td>5-6-1</td>
<td>0.001062</td>
<td>0.006121</td>
<td>0.001031</td>
<td>0.000711</td>
</tr>
<tr>
<td>5-9-1</td>
<td>0.000417</td>
<td>0.006291</td>
<td>0.000287</td>
<td>0.000118</td>
</tr>
</tbody>
</table>
Table 6: Best Simulation Results for Heat Waves temperature Prediction

<table>
<thead>
<tr>
<th>Time Series</th>
<th>MSE</th>
<th>NMSE</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>0.000352</td>
<td>0.295393</td>
<td>34.4177</td>
</tr>
<tr>
<td>BP</td>
<td>0.002853</td>
<td>0.372521</td>
<td>25.0961</td>
</tr>
<tr>
<td>GABC</td>
<td>0.000479</td>
<td>0.345013</td>
<td>35.0848</td>
</tr>
<tr>
<td>HGABC</td>
<td><strong>0.000115</strong></td>
<td><strong>0.273215</strong></td>
<td><strong>36.3023</strong></td>
</tr>
</tbody>
</table>

Figure 3: Heat Waves Time Series Prediction by Proposed HGABC during Training

Figure 4: Heat Waves Time Series Prediction by Proposed HGABC during Testing
The Proposed HGABC success in getting maximum SNR for heat wave's temperature time-series data as given in Table 5. The SNR reached to 36.3023 by HGABC, while the ABC arises to 34.8159 and GABC-MLP 36.8972. The NMSE of the proposed HGABC learning algorithm is given in table 5, shows the minimum NMSE compared to other's algorithms. For evaluating the performance of proposed learning technique, the optimal weight values test for a prediction job with the 25% data of heat wave's temperature time series. The testing MSE and NMSE of HGABC give very less prediction error from standard ABC, BP and GABC, given in Figure 3 and 4. The learning The BP, ABC and GABC training algorithm did not converge faster than other proposed HGABC for heat wave's temperature time-series data. Comparative simulation studies have been presented to show the effectiveness of our new algorithms. The simulation results in Figure 5, demonstrate the effectiveness of the new HGABC learning algorithms.

9. CONCLUSION

The HGABC algorithm collects the exploration and exploitation processes successfully, which proves the high performance of training MLP. It has the powerful ability of searching global optimal solution. So, the proper weights of the MLP may speed up the initialization and improve the classification and prediction accuracy of the trained NNs. The simulation results show that the proposed HGABC algorithm can successfully train boolean data for classification purpose and heat wave's time series for prediction, which further extends the quality of the given approach. The performance of HGABC is compared with the traditional BP, GABC and ABC algorithms. HGABC shows significantly higher results than BP, GABC and ABC algorithms during experiment.

ACKNOWLEDGEMENT

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


