ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2018.0247

OPTIMUM PARAMETERS SELECTION USING BACTERIAL FORAGING OPTIMIZATION FOR WEIGHTED EXTREME LEARNING MACHINE

S. Priya and R. Manavalan
Department of Computer Applications, Arignar Anna Government Arts College, India

Abstract

Extreme Learning Machine (ELM) is a Single Layer Feed Forward Network (SLFN) model with extremely learning capacity and good generalization capabilities. Generally, the performance of ELM for classification task highly based on three factors such as the input weight matrix, the value of bias and the number of hidden neurons presented. ELM randomly chooses the input weights and biases and determines analytically the weights as output. The random selection of biases and the input weight produce an unforeseen result which causes training error and also produces lesser prediction accuracy. Bacterial Foraging Optimization algorithm (BFOA) was used to find the optimum input weight and hidden bias values for ELM. With the unequal distribution of classes in imbalanced data sets, ELM algorithms tussle to find good accuracy. So, ELM algorithm doesn't get the necessary information about the minority class to make an accurate classification. To deal the issues associated with ELM, in this paper the hybrid algorithms Weighted ELM and Weighted ELM with BFO are proposed. Weighted ELM is proposed to handle the classification data that has imbalanced nature of class distribution. The main objective of weighted ELM is that the related weight value is computed and assigned for each training sample to increase the classification rate. Bacterial Foraging Optimization method is also integrated with the weighted ELM to find the optimum input weight and bias to maximize the classification accuracy. The comparative analysis has been performed over Hepatitis dataset. Further, the experimental results clearly revealed that one of the proposed methods Weighted ELM with BFO performs quite well when compared to others.

Keywords:

ELM, Weighted ELM, Bacterial Foraging Optimization, Initial Weight, Bias

1. INTRODUCTION

Neural networks find significant patterns of data; it evolves from a biological brain nerve cell. Applications of neural networks includes segmentation, classification and predictions [1]. In traditional ANN, Feed-Forward Neural Network (FFNN), the information moves in only forward direction without any cycles or loops. Training of Artificial Neural Network is fully based on past experience. When applied an unspecified input to network which yields output through past experiences. [2]. Backpropagation is one among the supervised learning technique established on the theory of gradient descent. In back-propagation, training process is achieved by tuning all parameters and therefore many iterative learning steps may be required for locating the optimum learning performance. This tuning process causes a decrease in the learning process and also an increase in the possibility of converging to local minima [3]. To achieve the learning process in multi-layer neural networks, gradient-based learning methods have been widely applied. The most common issues behind gradient based methods were training errors occurs due to local minima, slow convergence, hard to setting of learning parameters and output weight determination. To solve the problems, the ELM algorithm was originally proposed by Huang et al., on 2006 [4]. It was proven by Huang et al., the activation functions applied in the hidden layer are infinitely differentiable, so the input weight and biases of a single hidden layer feed forward networks (SLFN) can be arbitrarily assigned. Extreme Learning Machine (ELM) has an especially quick train stage with a high generalization performance. High train speed is relying on discovering the input weights and biases arbitrarily and the output weights analytically can be determined by using Moore–Penrose generalized inverse [5]. The main difficulties encountered with ELM and traditional gradient based learning algorithms for feed-forward neural networks are [6] [7]:

- The ELM has a very quick learning capacity.
- The ELM tends to accomplish the solutions forthright while not trivial problems occurred in the gradient based learning algorithm such as momentum rate, learning rate, local minima and over-fitting.
- It accustomed to train the network with many activation functions that was non-differentiable in nature.

Bacterial foraging Algorithm is simulating of biological bacterial food-searching behavior of E.Coli that has been pertinent in optimization field. It was proven by the author Passino is that the bacterial foraging algorithm created on the basis of bacterial chemotaxis behavior, reproduction steps and elimination dispersal events [8][9]. BFOA is adopted to compute the optimal input weight and the value of hidden biases. The bacterium attempts to compute optimal input weight and hidden biases of ELM after randomly assigning the initial position of bacteria [10].

The motivation of this research is to increase the classification accuracy of ELM and Weighted ELM. ELM classifier struggle to find better accuracy rates in case of imbalanced dataset. WELM handles the imbalance dataset efficiently, but produce unforeseen prediction accuracies due to random selection of initial weight and bias. So the BFOA is integrated with WELM for better performance. In this research, weighted ELM and weighted ELM with Bacterial Foraging optimization are proposed for the classification of Hepatitis Dataset. The proposed model of the research work is exposed in Fig.1.

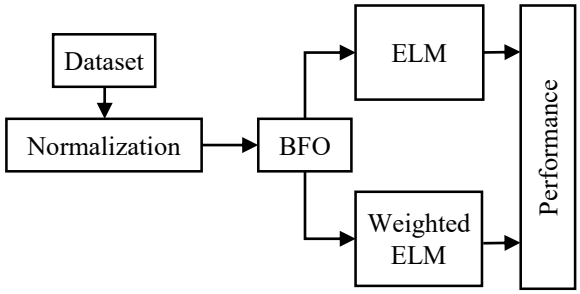

Fig.1. Proposed Model

This research work has been structured into five sections. Section 2, deals with the materials and methods used for classification of Hepatitis Dataset. Section 3, describes the proposed models weighted ELM and weighted ELM with BFO for the classification of Hepatitis dataset with explanations. Experimental results of proposed methods and their detailed explanations are shown in section 4. Finally, in section 5, the conclusions with further research scope are described.

2. MATERIALS AND METHODS USED

The design of classification model for diagnosing Hepatitis is crucial task, since data in dataset is imbalance nature. The detailed description about the methods Extreme Learning Machine and Bacterial Foraging Optimization (BFO) are given in the section 2.1, 2.2 respectively.

2.1 EXTREME LEARNING MACHINE

Extreme Learning Machine (ELM) is a modification of single hidden layer feed forward neural networks, introduced by Huang [4]. ELM overcomes the issues behind the conventional neural network learning algorithms such as over-fitting and slow training speed [11]. ELM provide solutions with minimum training error [12]. To learn through a single iteration, empirical risk minimization theory concept is used in ELM. It evades the issues associated with local minimization and the multiple iterations steps. A typical architecture of ELM is revealed in Fig.2. In the ELM Structure, ω represents the weights associated between the input layer and hidden layer, b indicates the threshold value of the hidden nodes, and β denotes the output weights. Activation function in the hidden layer performs the nonlinear transformation. A typical ELM has L hidden layer nodes, N instances, and an activation function $G(\chi)$ can be exhibited as:

$$\sum_{i=1}^{N} \beta_i G\left(\omega_{ij} \cdot x_j + b_i\right) = o_j, j = 1,2,...,N \quad (1)$$

The output weight β can be solved by

$$\beta = H^{\dagger} + T \quad (2)$$

where $H^{\dagger}$ is the value of the Moore-Penrose generalized inverse of hidden layer H

In ELM, the weights ω are generated randomly within the range [-1, 1] and the thresholds of hidden layer nodes b, are randomly chosen within the range of [0, 1] using a uniform sampling distribution. The value of β is decided by the Moore-Penrose generalized inverse as shown in Eq.(4). Sine function, Sigmoid function, Hardlim function, Triangular basis function, and Radial basis function are most commonly used activation functions ($G(\chi)$). In this research, the sigmoid function is used as an activation function. The Fig.3 shows computational steps of ELM algorithms.

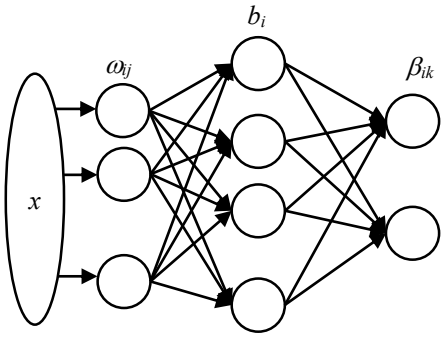

Fig.2. Structure of ELM

ELM Algorithm for Hepatitis Dataset

Input: The Training Dataset is represented as N. Here $N = \{(X_i, t_i) | \in R, t_i \in R, i = 1, 2,..., n\}$, where X_i represents input attributes and t_i represents target class, activation function represented as g and the amount of hidden neurons needed.

Output: Classification of Hepatitis dataset.

Step 1: Initially ELM arbitrarily assigns input weight W_i and the threshold value of hidden nodes b_i ($i = 1, 2, ..., n$).

Step 2: Output matrix H is calculated, where H represents Hidden layer.

Step 3: Output weight β is calculated through Moore-Penrose generalized inverse equation. Here $\beta = H^{\dagger}T$, $H^{\dagger}$ is the inverse of H.

Fig.3. ELM Algorithm for Hepatitis Dataset

Bacterial Foraging Optimization Algorithm (BFOA) is used to find the optimum initial weight and bias for ELM for classifying data in Hepatitis Dataset. The detailed explanation of BFOA is given in subsection 2.2

2.2 BACTERIAL FORAGING OPTIMIZATION ALGORITHM (BFOA)

Bacterial Foraging Optimization Algorithm (BFOA) is one amongst the swarm intelligence techniques that is modeled as the food seeking and reproductive strategy of common bacteria Escherichia Coli (E.Coli).

BFOA for finding Optimum Weight and Bias

Input: The Training Dataset is represented as N, where $N = \{(X_i, t_i) | \in R, t_i \in R, i = 1,2,\ldots,n\}$, Where X_i represents input attributes and t_i represents target class, activation function represented as g and the number of hidden nodes.

Output: Optimum Input Weight and Bias for ELM.

Step 1: The necessary Parameters required for the algorithm are initialized, initial parameters are represented in Table.11.

Step 2: In the Elimination-Dispersal event, increment $e = e + 1$

Step 3: Reproduction Stage occurs until specified condition met, Increment $r = r + 1$

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

For $i = 1,2,...,S$ take a chemotactic loop for bacterium and compute the Cost Function MSE(j, k, l) and save $J_{last..}$ Determine the movement of the bacterium either

Tumbling or swimming and compute the Cost Function. Repeat the process until $i = S$.

Step 5: Check the condition if $j < N_c$ go to step 4, continue the chemotaxis process because the bacteria life time is not over.

Step 6: In the Reproduction step of BFOA, the fitness value of each bacterium is computed to check for the health status of each bacterium. From the entire set of S bacteria, the half of the bacteria with the highest fitness values (J (health)) were eliminated and therefore the other halves having lower fitness values can stay within the evolution process.

Step 7: If $r < N_r$ then move to step 3. During this case, the bacteria not endure the required number of steps, therefore the chemotactic loop is processed yet again with the succeeding generation of bacteria.

Step 8: In Elimination-Dispersal event, existing bacteria gets eliminated and dispersed in a new random position based on the value of Elimination-Dispersal Probability (P_e).

Step 9: If $e < N_e$, move to step 2, Otherwise end.

Fig.4. Computational Step of BFOA

BFOA is efficiently applied to solve numeric optimization problems [13]. The implementation steps of BFOA comprises of four steps such as Chemotaxis, Swarming, Reproduction and Elimination and Dispersal [14] [15] [16]. The implementation step of BFOA is given in Fig.4. The initial bacterium structure has 1-by-400 matrix, it consists of 1×400 random values as initial position. In this research, initial Parameters for Bacterial Foraging Optimization algorithm are given in Table.1. Bacterial Foraging Optimization algorithm starts its tasks with initial parameters. Then after specified number of iterations BFO yield optimum input weight and bias values. The obtained optimum input weight and bias values given as input to ELM for further processes.

Table.1. Parameter Initialization of BFOA

Parameter	Value
N contains the total number of bacteria available population	2
N_c represents the Chemotactic steps	2
N_p specifies the number of reproduction steps	2
N_e indicates the number of elimination-dispersal events	2
N_s represents Swarming steps	2
P_e specifies the probability of Elimination-dispersal event	0.5
C represents the size of the step taken in the random direction specified by the tumble (C)	0.01
Hidden Neuron	20

3. PROPOSED METHOD

The methods ELM and ELM with BFOA not achieved absolute and concrete results in terms of accuracy. Even though Bacterial Foraging Optimization (BFO) algorithm was integrated with ELM used to find the optimum input weights and hidden biases, still, the ELM algorithms faces struggle to find good classification results in terms of accuracy for imbalanced data sets. To address the above issues, in this paper, Weighted ELM and Weighted ELM with BFOA are proposed for the classification of Hepatitis dataset. The detailed description of Weighted ELM and Weighted ELM with BFOA are given in the section 3.1 and 3.2 respectively.

3.1 WEIGHTED ELM (WELM)

The weight are computed for each instances in the imbalance two classes of dataset and the same is used in WELM [16] for the prediction of new instance. Here the samples of different classes will be automatically assigned different weights. So the formula of the WELM is,

$$\text{Minimize: } L = 0.5\|\beta\|^2 + 0.5CW\sum_{i=1}^{N}\|\varsigma_i\|^2 \tag{6}$$

$$\text{s.t. } h(x_i)\beta = t_i^T - \varsigma_i^T, i = 1,2,...,N$$

In the diagonal matrix W, elements in w_{ii} represents the principal diagonal vector corresponds to a sample x_i. Regarding Karush–Kuhn–Tucker (KKT) theorem [15], the Lagrange function is applied to find the solution for the dual optimization problem with respect to Eq.(6) is

$$\text{Minimize: } L_{WELM} = 0.5\|\beta\|^2 + 0.5CW\sum_{i=1}^{N}\|\varsigma_i\|^2 - \sum_{i=1}^{N}\alpha_i\begin{pmatrix} h(x_i)\beta \\ -t_i^T + \varsigma_i \end{pmatrix} \tag{7}$$

Here the constant α_i represents the Lagrange multiplier of instance x_i in the linear arrangement to model the final decision function. The KKT optimality conditions are acquired after constructing the partial derivatives with respect to variables (β, ς_i, α_i) that are assigned to zero.

$$\frac{\partial L_{DELM}}{\partial \beta} = 0 \rightarrow \sum_{i=1}^{N}\alpha_i h(x_i)^T = H^T\alpha \tag{8}$$

$$\frac{\partial L_{DELM}}{\partial \varsigma_i} = 0 \rightarrow \alpha_i = CW\varsigma_i \text{ where } i = 1,2,...,N \tag{9}$$

From the above derivation β is represented as,

$$\beta = H^T\left(WHH^T + \frac{1}{c}\right)^{-1}WT \tag{10}$$

For binary classification problems, the decision function needed for ELM is based on only one output node.

$$f(x) = sign\left(\left(h(x)H^T * WHH^T + \frac{1}{c}\right)^{-1}WT\right) \tag{11}$$

The formula of decision function $f(x)$ based on the kernel function is represented as

$$f(x)_{\text{kernel}} = signh(x)H^T\left(\frac{1}{c} + WHH^T\right)^{-1}WT \tag{12}$$

From the class information two weighting schemes are generated, they specify the two important cases of the cost sensitive learning [15]:

$$W_1 : W_{ii} = \frac{1}{\#(t_i)}$$

$$W_2 : \begin{cases} W_{ii} = \dfrac{0.618}{t_i} & if\ t_i > avg(t_i) \\ W_{ii} = \dfrac{1.000}{t_i} & f\ t_i \le avg(t_i) \end{cases}$$

Here $\#(t_i)$ is the no. of instances in class t_i, $i = 1, 2,\ldots, m$. The computational steps of weighted ELM for hepatitis classification are shown in Fig.5. A major problem with an ELM classification with the imbalanced data set can be the unfair distribution of majority and minority classes, since high level classification accuracy is achieved on the majority class by giving up the accuracy of minority class. Weighted ELM (WELM) is one of the cost-proportionate weighted sampling methods that efficiently handles the classification difficulties faced by ELM on imbalanced data sets [18]. For binary classification, ELM assigns the fixed misclassification cost value to all instances when the quantity of negative samples is far larger than that of the quantity of positive samples or contrariwise. It is one of the flaws of traditional ELM. WELM is proposed to overcome the above issues of ELM. Nevertheless, WELM also yields poor accuracy, since, the input weight and bias are selected randomly. To address issues in weighted ELM, the BFOA is integrated with WELM to find optimum weight and bias in order to increase the classification accuracy. The detailed explanation of WELM with BFOA algorithms is given in section 3.2.

Weighted ELM Algorithm for Hepatitis Classification

Input: Training dataset $N = \{(X_i,t_i)| \in R_d\ t_i \in R,\ i = 1,2,...,n\}$, activation function and the number of hidden node

Output: Classification of Hepatitis dataset.

Step 1: Calculate weight Matrix W

Step 2: Initially ELM arbitrarily assigns input weight W_i and the threshold value of hidden nodes b_i ($i = 1, 2,..., n$).

Step 3: Output matrix H is calculated, where H represents Hidden layer.

Step 4: Calculate the output weight matrix based on Moore-Penrose generalized inverse function.

Fig.5. Weighted ELM Algorithm for Hepatitis Classification

3.2 WEIGHTED ELM WITH BACTERIAL FORAGING OPTIMIZATION METHOD

The weight for each attribute is calculated as discussed in previous section. ELM parameters such as initial weights and bias values are obtained using BFOA. Based on these optimized parameters WELM algorithm classify the individual object. The algorithm of weighted ELM with BFO is shown in Fig.6.

Weighted ELM with Bacterial Foraging Optimization Algorithm

Input: The training dataset is represented as N. Here $N = \{(X_i,t_i)|\in R,\ t_i \in R,\ i = 1,2,\ldots,n\}$ Where X_i represents input attributes and t_i represents target class, Activation function represented as g and the number of Hidden Neurons.

Output: Optimum input weight and the threshold of hidden neuron such as bias of imbalanced training data.

Fitness: Set Fitness function as Mean Squared Error of Training data MSE_{trn}

Step 1: In initialization phase of Weighted ELM, Construct a diagonal matrix W and assign different weights based to the number of the two class instances percentage. In the diagonal matrix W, elements in w_{ii} represents the principal diagonal vector corresponds to a sample x_i and assign number of hidden neurons.

Step 2: Sigmoid activation function is established for WELM.

Step 3: Initialize the parameters of BFOA as mentioned in Table.1. The size of each bacterium is 1×400. Initial position of Bacteria is assigned with a Random vector of size 1×400

Step 4: Calculate the Fitness value of Each Bacterium. Mean Square Error (MSE_{trn}) of the training samples is calculated and it was used as the fitness value for each bacterium.

Step 5: BFOA algorithm is optimized for input weight and bias for WELM. Each Bacterium enter into four stages such as Chemotaxis, Swarming, Reproduction and Elimination Dispersal event. Initially Bacteria Enter into Chemotaxis stage, each bacterium calculates Cost function and stores the better cost in J_{last}. Similarly, better cost value is updated in each stages of bacterium movement. Finally, best fitness value is obtained and get the best position of bacteria in J_{last}. Through the best fitness value, the corresponding bacterium position such as input weight and bias for training data are obtained.

Step 6: By assigning the obtained optimum input weight and bias of Weighted ELM into Eq.(10) to calculate the hidden layer output weight matrix.

Step 7: Implemented the WELM model with optimum input weight and bias to classify Hepatitis dataset.

Fig.6. Weighted ELM with Bacteria Foraging Optimization

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The studied algorithms are implemented by using the software Matlab R2015(a). The detailed explanation about dataset used in the research, result and their discussions are given in the section 4.1 and 4.2 respectively.

4.1 DATASET

Experiment is conducted over the Hepatitis Dataset [19] that comprises the report about the patients affected by the Hepatitis disease. The objective of the research is to diagnostically predict if these patients will die (Class Label: 1) or survive (Class Label: 2). The dataset contains 155 instances, all instances have 19 input attributes (Conditional Attributes) and one output attribute is also called decision attribute. From the dataset, 70% and 30% of data are taken for training and testing.

4.2 PERFORMANCE METRICS

The performances of the proposed methods are analyzed by using statistical parameters such as sensitivity, specificity, precision, F1 score and Accuracy. The detailed descriptions of these Statistical parameters are given below. Confusion matrix is one of the way to express the result of a classifier. It is also referred as contingency table. It consists of 2 rows and 2 columns for binary classification. The Table.1 shows the general format of a confusion matrix. Across the top of Table consists of observed class labels and down side represents predicted class labels. Each cell holds the number of predictions made by the classifier that fall into the matching cell. Sensitivity deals with the proportion of actual positive rates which are correctly recognized by the classifier. Specificity processes the proportion of negative instances are identified correctly. The balance between majority and minority class performance are measured through Geometric Mean or G-mean. It is a one of the measure of the ability of a classifier to balance sensitivity rate and specificity rate. Accuracy deals with how well a binary classification correctly recognizes or rejects a condition. Accuracy is a balanced measure of a classifier whereas sensitivity measures only positive cases and specificity identifies only negative cases. *TP* is number of true positives, *FP* is number of false positives, *TN* is number of true negatives and *FN* is number of false negatives. The Table.3 shows the performance metrics and their formula.

Table.2. Confusion Matrix

Actual Predicted	Positive	Negative
Positive	True Positive (*TP*)	False Positive (*FP*)
Negative	False Negative (*FN*)	True Negative (*TN*)

Table.3. Performance Metrics and Formulae

Metrics	Formula
Sensitivity	$TP / (TP + FN)$
Specificity	$TN / (FP + TN)$
G-mean	$(\text{Sensitivity} \times \text{Specificity})^{0.5}$
Accuracy	$(TP + TN)/(TP + FP + FN + TN)$

where,

TP - Predicts the cases with Die as Die,

FP - Predicts the cases with Die as Survive,

TN - Predicts the cases with Survive as Survive,

FN - Predicts the cases with Survive as Die.

5. RESULT AND DISCUSSION

Sensitivity, Specificity, G-Mean and Accuracy achieved by the above algorithms are presented in Table.5 and their performance analysis charts are also exposed in Fig.7.

Table.4. Hepatitis Classification Performance of Proposed Algorithms

Metrics	Training or Testing	ELM	ELM with BFO	Weighted ELM	Weighted ELM with BFO
Sensitivity	Training	0.4964	0.4688	0.4848	0.4545
	Testing	0.3625	0.6208	0.4730	0.6667
Specificity	Training	0.9085	0.9839	0.9821	0.9878
	Testing	0.8333	0.6208	0.8049	0.8049
G-Mean	Training	0.6437	0.6788	0.6876	0.6701
	Testing	0.5412	0.5178	0.6036	0.7325
Accuracy	Training	0.8985	0.9083	0.9233	0.9247
	Testing	0.6696	0.6739	0.6925	0.7581

From Table.5, it was observed that ELM method yielded an accuracy of 90%, 67% training and testing respectively, 50% Sensitivity and 91% specificity and 65% G-Mean are achieved for training and 36% sensitivity, 83% specificity and 54% G-Mean are obtained for testing. The combination of ELM and Bacteria Foraging optimization achieved an accuracy of 90% and 67% for training and testing respectively, 47% Sensitivity and 98% specificity and 68% G-Mean are achieved for training and 62% sensitivity, 92% specificity and 52% G-Mean are obtained for testing. Weighted ELM method achieved 92% accuracy, 48% sensitivity and 98% specificity and 69% of G-Mean are obtained for training samples and 69% accuracy, 47% sensitivity and 80% specificity and 60% of G-Mean are achieved for testing samples. The validation accuracy of 76% is arrived by Weighted ELM with BFO, which is 9% higher than the ELM and 7% higher than Weighted ELM. Similarly, Weighted ELM with BFO achieved 45% sensitivity and 98% specificity levels for training and for testing 67% sensitivity and 80% specificity are achieved. The experiment results clearly reveal that there is considerable performance variability among the various proposed methods. The computational results clearly reveal that the weighted ELM with BFO performing better than other algorithms.

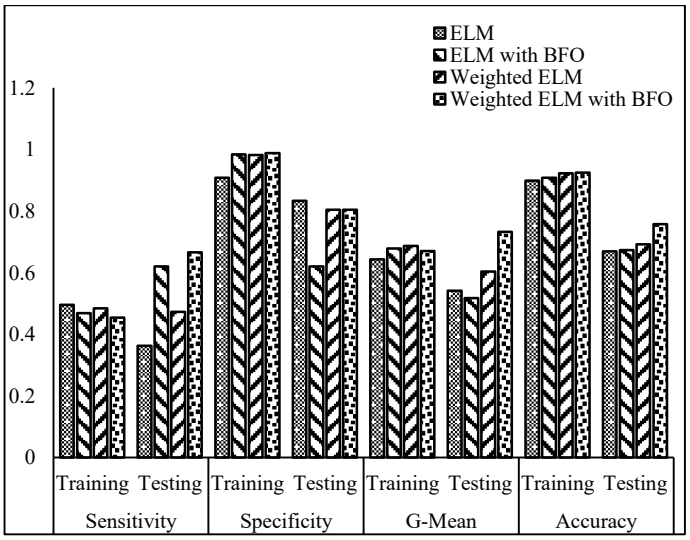

Fig.7. Performance Measures

6. CONCLUSION

In this paper, the hybrid algorithms Weighted ELM and Weighted ELM with BFO are proposed for the efficient classification of hepatitis disease. One of the demerits of ELM algorithm is choosing random weights and bias. To overcome this issue BFOA technique was proposed to get optimum weights and bias. The weight based ELM algorithm is introduced to deal the data with imbalanced class distribution. Bacterial Foraging Optimization method is also integrated with the weighted ELM to find the optimum input weight and bias to maximize the classification accuracy. One of the proposed algorithms WELM with BFOA shows better classification accuracy compared to others. The validation accuracy of 76% is arrived by Weighted ELM with BFO, which is 9% higher than the ELM and 7% higher than Weighted ELM. The experimental results proved that Hybrid Algorithms Weighted ELM with BFOA to be more effective than the other models. The performance of ELM differs based on the number of hidden neurons, so it is necessary to fix the number of hidden neurons. In further research WELM with BFOA can be extended to implement multiclass imbalance classification learning.

REFERENCES

[1] Richard O. Duda, Peter E. Hart and David G. Stork, "*Pattern Classification*", 2nd Edition, Wiley-Interscience, 2001.

[2] C. Bishop, "*Neural Networks for Pattern Recognition*", Oxford: University Press, 1995.

[3] L. Fausett, "*Fundamentals of Neural Networks*", Prentice Hall, 1994.

[4] G. Huang, Q. Zhu and C. Siew, "Extreme Learning Machine: Theory and Applications", *Neurocomputing*, Vol. 70, No. 1-3, pp. 489-501, 2006.

[5] G. Huang, Q. Zhu and C. Siew, "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks", *Neural Networks*, Vol. 2, pp. 985-990, 2004.

[6] Omer Faruk Ertugrul and Yılmaz Kaya, "A Detailed Analysis on Extreme Learning Machine and Novel Approaches Based on ELM", *American Journal of Computer Science*, Vol. 1, No. 5, pp. 43-50, 2014.

[7] G. Huang, Q. Zhu and C. Siew, "Extreme Learning Machine: Theory and Applications", *Neurocomputing*, Vol. 70, No. 1-3, pp. 489-501, 2006.

[8] H.J. Bremermann, "Chemotaxis and Optimization", *Journal of the Franklin Institute*, Vol. 297, pp. 397-404, 1974.

[9] K.M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control", *IEEE Control Systems*, Vol. 22, No. 3, pp. 52-67, 2001.

[10] Jae Hoon Cho and Dae-Jong Lee, "Parameter Optimization of Extreme Learning Machine using Bacterial Foraging Algorithm", *Journal of Korean Institute of Intelligent Systems*, Vol. 17, No. 2, pp. 807-812, 2007.

[11] Sunday Olusanya Olatunji, "Sensitivity-Based Linear Learning Method And Extreme Learning Machines Compared For Software Maintainability Prediction of Object-Oriented Software Systems", *ICTACT Journal on Soft Computing*, Vol. 3, No. 3, pp. 514-523, 2013.

[12] Sunday Olusanya Olatunji and Hossain Arif, "Identification of Skin Diseases using Extreme Learning Machine and Artificial Neural Network", *ICTACT Journal on Soft Computing*, Vol. 4, No. 1, pp. 627-632, 2013.

[13] R. Vijay, "Intelligent Bacterial Foraging Optimization Technique to Economic Load Dispatch Problem", *International Journal of Soft Computing and Engineering*, Vol. 2, No. 2, pp. 55-59 , 2012.

[14] Hanning Chen, Yunlong Zhu and Kunyuan Hu, "Adaptive Bacterial Foraging Optimization", *Abstract and Applied Sciences*, Vol. 2011, pp. 1-27, 2011.

[15] Swagatam Dass, Arijit Biswas et al., "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis and Applications", *Foundation of Computer Intelligence*, Vol. 3, pp. 23-55, 2009.

[16] Vipul Sharma, S.S. Pattnaik and T. Garg, "A Review of Bacterial Foraging Optimization and its Application", *Proceedings of National Conference on Future Aspects of Artificial intelligence in Industrial Automation*, pp. 9-12, 2012.

[17] W. Zong, G.B. Huang and Y. Chen, "Weighted Extreme Learning Machine for Imbalance Learning", *Neurocomputing*, Vol. 101, pp. 229-242, 2013.

[18] Kai Hu et al., "An Optimization Strategy for Weighted Extreme Learning Machine based on PSO", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 31, No. 1, pp. 1-16, 2017.

[19] Hepatitis, Available at https://archive.ics.uci.edu/ml/datasets/hepatitis