I J C T A, 10(8), 2017, pp. 329-341 ISSN: 0974-5572

Spam Identification through Modified Teaching Learning Based Optimization Method

[1]Amaresh Sahu, [2]Sabyasachi Pattnaik and [3]Debahuti Mishra

ABSTRACT

Electronic Mail (Email) is a fast and effective method to share and exchange information all over the world. Email spam is one of the key problem in electronic mail communication. In general, Spam is a unwanted email contains commercial content is sent to the large number of recipients in huge quantities. It consumes large amount of storage of mail servers, waste of time and consumes more network bandwidth. Therefore identification of spam Email is necessary. In this proposed work a Multilayer Perceptron (MLP) training method is introduced, known as Modified Teaching Learning Based Optimization (MTLBO). The proposed metaheuristic algorithm MTLBO has utilized some of the major concepts of particle swarm optimization algorithm (PSO) and motivated from the non-parameterized approach of nonlinear problems solving nature of Teaching Learning Based optimization algorithm (TLBO). The MTLBO algorithm is separated into two key steps, In the first step learners learn from teacher to enhance their knowledge and in second step learners learn by cooperating with each others as well as with the best learner or team leader among them. In this literature various experiments carried out on benchmark functions and Spam Assassin dataset to calculate the performance of proposed MTLBO algorithm. The evaluated results show that the MTLBO has got more generalization, optimization and convergence capabilities compared to other metaheuristics optimization methods.

Index Terms: Multilayer Perceptron, Feature selection, Teaching Learning based Optimization, Particle Swarm Optimization, Modified Teaching Learning Based Optimization, Functions Optimization and Classification.

1. INTRODUCTION

The past 11 years have witnessed significant increase in usage of internet and the change trend appears to be continuously growing in future. Now a day internet has been proved as a vital part of our daily routine life. Email is one of the major services of internet. It is fast, economical and secure mode of message exchange between wide range of users. In contrast there are several issues that become barrier in professional procedure of email. On tops of the list is email related spam. Email spam is also named as unsolicited bulk Email (UBE), junk mail, or unsolicited commercial email (UCE) is generally sent by spammer. These spam emails are aimed at and passed randomly to a huge number of email users. These emails can contain harmful malwares or phishing links, making email users vulnerable to several security violating attacks and can even crash the email servers. Spam messages increase rapidly in spite of substantial development of anti-spam services. Spam could be controlled by blocking emails that come from definite addresses or by ûltering out the messages that contain definite subject lines. To conquer this difficulty many email spammers started to denote sender addresses randomly and they tried to attach randomly generated characters to the message at the end of the subject. In general there are two major approaches are used for legitimate mail identification first is knowledge engineering (KE) and second is machine learning (ML). First one is knowledge engineering approach in which a set of rules are created priory for message identification. These spam filtering rules are created by the

[1,3] Computer Science Engineering Department, SOA University Bhubaneswar, Odisha 751030, India, *Emails: amaresh_sahu@yahoo.com,mishradebahuti@gmail.com*

[2] Information & Communication Technology Department, Fakir Mohan University Balasore, Odisha 756019, India, *Email: spattnaik40@yahoo.co.in*

user or by other authority like software company. The main weakness of this approach is that the rules must be constantly updated and also rules maintenance approach may be unsuitable for the users. Contrast to that in the second approach specifying any rules explicitly is not required [1]. In this approach a set of pre-classified training samples/documents are required for training. Classification rules are extracted from samples by using the learning algorithm. Now, the machine learning concepts have been widely used and many suitable algorithms have been proposed for solving this type of classification problem. In this article several well known machine learning methods have been considered and applied for identification of spam from SpamAssassin dataset.

2. RELATED WORK

Spam or unwanted mail communications, usually sent in bulky volume of mails that affects server storage space, networks bandwidth, work productivity and user time [2]-[5]. Usually, spam in email service is used for advertising about various products and services related to the entertainment of adult, earning money quickly scheme and to other gorgeous type of products [6]. In the viewpoint of the internet users, spammers are exploiting various internet applications like email service organizations, social media network platforms, various web related blogs, various web related forums and sometime internet search based engines [7]. Spammer can generate revenue more than one million dollars approximately in a month in a single affiliated program [8]. Spam detection accuracy is influenced by many factors like the spam subjective nature, processing time overhead, different language issues, delay in message communication, and the unbalanced cost associated with errors classification [9]. By using machine learning based filter spam can be detected, in this approach training is required to extract the necessary knowledge for spam detection. Many filters use machine learning algorithms are Naïve Bayes classifier [11], Support Vector Machine (SVM) classifier [10] and Artificial Neural Networks (ANN) classifier [2] so on for spam classification. Also, by hybridizing various ML learning methods together more robust and accurate spam detection methods can be generated [12]-[14]. ANN is very popular and commonly used technique for spam classification which generally gives very accurate results [15] [16]. MLP is a kind of feed forward ANN model that needs a substantial time for its training and its proper parameters selection, which encourages many researchers to optimize it by using several ways [17]. Gradient based techniques are very popular techniques for MLP optimization. Back-propagation (BP) algorithm is one of such gradient based optimization technique. Although gradient based techniques are very popular but, they suffer from some major limitations like slow convergence rate, selection of proper parameters for training and higher probability of being trapped in local minima solution [18]. To overcome from those limitations many researchers have proposed various stochastic based training methods for MLP model. Now a day, many researchers are using nature-inspired meta-heuristic algorithms. These algorithms are most popular and attractive stochastic type algorithms. Genetic Algorithm (GA) [15], Particle Swarm Optimization (PSO) [14], Differential Evolution (DE) [19], Ant Colony Optimization (ACO) [20] and Teaching learning based optimization (TLBO) [21][22] are examples of such type of stochastic algorithms. In this article, we have proposed a learning algorithm for MLP model named as the Modified Teaching Learning based Optimization (MTLBO) for identifying email spam[22][24]. In Contrast to other nature-inspired meta-heuristic algorithms and gradient based algorithms, proposed MTLBO algorithm is a parameters free algorithm that reduces the need of proper selection of some important parameters usually related to stochastic algorithms. By using feature extraction module spam based text and the non-spam based text are extracted, then feature dictionary is produced consisting of feature vectors and these are used as input to chosen algorithm for classification task.

3. DATASETS

To evaluate the effectiveness of proposed MTLBO training method on MLP network in classifying e-mail spam, we have implemented the proposed method on SpamAssassin public mail corpus1 dataset. This dataset consists of 9346 records and having 90 features in it. Every example record within the dataset is labeled as Ham (legitimate) or Spam mails. The dataset consist of roughly 6951 numbers of ham emails and 2395 numbers of spam emails. So there are approximately 25.6 percentage of spam emails in the dataset. The

dataset is imbalanced hence more difficult and challenging for classification. The details of the features available in the dataset can be viewed in literature [23]. In this application, we have followed the two fold approach where the dataset is evenly divided into two parts. One part is used for training and another part is used for testing. Each experiment has been repeated for 10 times in order to acquire statistically considerable results.

3.1. Extracting Features

The aim of the proposed method is to classify the text messages, i.e. strings. Unfortunately, strings are not very convenient objects to grip. Most of the machine learning algorithms are based on classifying the numerical objects (i.e. real numbers or vectors) or otherwise require some measure of similarity exists between the objects (i.e a distance metric or scalar product). In the ûrst all messages are converted to vectors of numbers known as feature vectors and then classify these vectors. For example, it is very normal to take the vector of numbers of occurences of certain words in a message as the feature vector. When we extract features we usually lose information and it is clear that the way we represent our feature-extractor is crucial for the performance of the spam filter. If the features are chosen so that there may exist a spam message and a legitimate mail with the same feature vector values, then no matter how good our machine learning algorithm is, it will make mistakes in filtering process. On the other hand, a wise correct choice of features will make classiûcation task much easier. In this paper for the training part, this module accounts the words frequency present in the text of email, here we have taken words having the time of appearance is more than three times as the feature word of this class. Here, every email in training is denoted as a feature vector [25].

4. CONCEPTS AND METHODOLOGIES

4.1. Multilayer Perceptron Network

The human brain has multitasking capability through which it controls various activities in our body simultaneously such as controlling temperature of body, pressure of blood, pulse rate and respiratory system. It also enable us to do some perception activities that let human beings to see, feel, hear, test and smell. The human brain holds over 20 billion number of neurons and a neuron is connected with maximum of 10,000 numbers of synaptic connections [26]. Nerve signals are transferred through neurons to and from the human brain. Various computer models are designed to simulate the activities of human brain. ANN is one of such computer model that are trained by using various learning algorithms. Learning algorithms have the task of ANN training by modifying the associated weights or knowledge in order to achieve the objective of the problem. The neuron of a ANN is consisting of some major components like, A synapse link that is characterized by its own weight, An adder that is used for adding up the input signals flow to the neuron, lastly, An activation function is used to evaluate the output of the neuron. Feedforward Neural Network (FFNN) model is also known as Multilayer Perceptron (MLP).

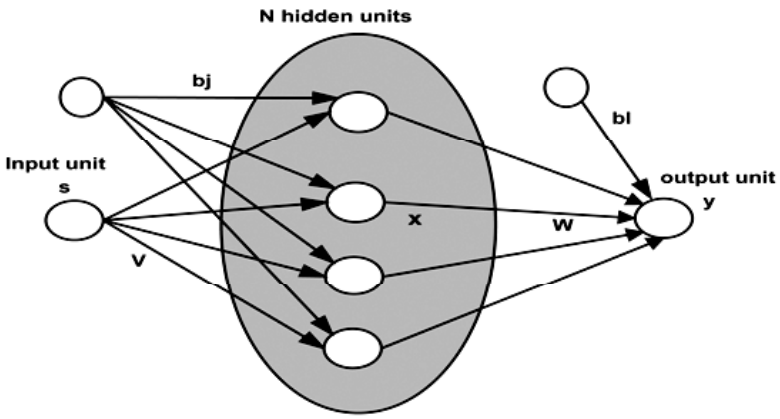

Figure 1: Example of a Three Layers MLP

From the Figure.1. it can be observed that a MLP is taken with I units of input , N units of hidden and J units of output. Here Let, $S = (s_1, s_2,\ldots s_I)^T$ be the I inputs to the input nodes, $X = (x_1, x_2,\ldots x_N)^T$ be the N outputs of the hidden nodes and $Y = (y_1, y_2, \ldots y_J)^T$ be the outputs generates from the output nodes respectively. b_j represents the bias associated to input layer and b_l is the bias associated to the output layers.

The activation from input layer is transmitted to the output layer by using forward pass. The weighted inputs of all input nodes combine with bias b_j generate the activations of the hidden nodes. The j^{th} hidden node activation is indicated as Net_j, and is calculated by using equation (1) as follows:

$$Net_j = \sum_{i=1}^{I} V_{ji} S_i + b_j \tag{1}$$

The j^{th} node output in the hidden layer is calculated by using a sigmoid function by using equation(2) as follows:

$$x_j = \frac{1}{1 = e^{net_j}}, \tag{2}$$

The outputs generate from the hidden layer denoted as $x_1, x_2,\ldots x_N$ are utilized as inputs to the output layer. The weighted inputs of all hidden nodes combine with bias b_l generate the activations of the output nodes denoted as $y_1, y_2, \ldots y_J$. The weight connection from the j^{th} hidden node x_j to the l^{th} output node y_l is represented as W_{lj}. This is represented in equation(3) as follows:

$$y_l = \sum_{j=1}^{N} W_{lj} x_j + b_l, \tag{3}$$

The backpropagation starts by propagating error backward which is calculated by the difference between the output generated by network denoted as y_l and the actual output denoted as $y_l^{'}$. The error value is used to update weight and biase values associated with network. The Error (E) of MLP network needs to be minimized by using BP algorithm repeatedly to optimize the weights and biases [27]. In each iteration the error is calculated as follows by using equation (4):

$$E = \sum_{e=1}^{p} \sum_{l=1}^{J} \left| y_l - y_{l'} \right|^2, \tag{4}$$

Where, P represents the number of patterns associated with training dataset. Initially random weight and bias values are allotted to the MLP network. The goal is to evaluate optimize weights and biases by training the MLP network to match the output values of network with teacher values or original value by minimizing the error.

4.2. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a stochastic algorithm inspired by the nature. This algorithm is originated by Kennedy and Eberhart [28]. PSO algorithm has been inspired from natural activities like flocking of bird and schooling of fish etc.

A set of initial solutions generated randomly in PSO which transmitted to a multidimensional design space to find the optimal solution in number of iterations. The multidimensional design space information is shared by the elements or particles present in the swarm. The complete development history of PSO heuristic algorithm has been elaborated by Kennedy and Eberhart [28] [29].

The PSO algorithm computes various values like positions of new particles by updating old particles, new velocities by updating the old velocities. A particle $\vec{s}_k$ in design space of dimension d is denoted as $\vec{s}_k$ = ($s_{k1}, s_{k2}, s_{k3}, \ldots, s_{kd}$), where k varies from 1, 2,. . ., n. n is used to represent the total quantity of particles.

Particle in the swarm retains its associated velocity and has memory of its associated personal best position denoted by, $\vec{b}_k$ = (b_{k1} , b_{k2} , b_{k3} ,. . ., b_{kd}). Let us assume the position $\vec{b}_g$ = (b_{g1} , b_{g2} , b_{g3} ,. . ., b_{gd}) be the best position discovered by the member particles of the surrounding that has provided the best possible solution . The particles change their own position in each loop based on the velocity updation operation. In standard PSO algorithm the new velocity and new position of particle is calculated as follows by using equations (5) and (6) at iteration t.

$$\vec{v}_k(t+1) = w \circledast \vec{v}_k(t) + \vec{c}_1 \circledast \vec{r}_1(t) \times (\vec{b}_k(t) - \vec{s}_k(t)) + \vec{c}_2 \circledast \vec{r}_2(t) \circledast (\vec{b}_g(t) - \vec{s}_k(t)) \quad (5)$$

$$\vec{s}_k(t+1) = \vec{s}_k(t) + \vec{v}_k(t+1) \quad (6)$$

In the mentioned equations (5) sign represents point by point vector multiplication. The weight (w) is the inertia momentum factor lies within the range 0 to 1 (0< w ≤1), c_1 is treated as self-confidence factor and c_2 is treated as swarm confidence factor. c_1 and c_2 are non-negative real constants. $\vec{r}_1$ and $\vec{r}_2$ are the random vectors values lies within the range 0 to 1. They are typically selected as a uniform random numbers within the range 0 to 1 [30].

The steps of velocity update, position update, and fitness calculations are carried out repeatedly until a desired termination convergence criterion is achieved. One of the termination criterion is represented in equation (7) as follows.

$$\left| f\left(\vec{b}_{g(t)}\right) - f\left(\vec{b}_{g(t+1)}\right) \right| \leq \in \; t = 2, 3, ..., It \quad (7)$$

Many researchers have proved that standard PSO algorithm works properly in many complex non linear optimization problem solving., In order to represent this Clerc and Kennedy [31] have proposed a generalize PSO model.

3.3. Teaching Learning Based Optimization

In general, the TLBO method is sub divided into two major steps. The first step of the method is treated as the ‘Teacher Phase’ and the second step of the method is treated as the ‘Learner Phase’. The ‘Teacher Phase’ means learners can be trained by teacher having more knowledge than them and the ‘Learner Phase’ means learners can enhance themselves by sharing knowledge with each others.

3.3.1. Teacher Phase

The learner having the best knowledge is considered as the teacher in the society. The knowledge is distributed by the teacher among the learners to improve the knowledge of the learners. The more is the talent of the teacher the more is the mean MA and approaching towards mean MB more quickly. A teacher is evaluated by his/her capability of enhancing his/her learners knowledge equal to his stage. But, practically this is not possible. The knowledge of learners and mean of the class increases up to certain degree by a teacher depends on the learners performances. This leads to a arbitrary process depends on many factors. Let us consider at iteration i, MEi be the mean of learners and TEi be the teacher. Teacher TEi always aim to upgrade mean MEi of learners towards his/her own knowledge level. Therefore the mean produced newly would be assigned as mean MEnew. The gap between the current mean and the new mean is used to update solution and is represented by the equation,

$$\text{Difference\_Mean}E_i = r_i \; (\text{MEnew}\_ \text{TF ME}_i) \quad (8)$$

Here, TF is used to represent teaching factor which is used to take decision on the change of mean value and random number ri lies within the range of 0 to 1. The TF value is randomly taken as 1 or 2, with giving same probability to both the values. This leads to a heuristic approach and calculated as,

$$TF = \text{round}[1 + (\text{rand}(0, 1))(2 - 1)] \quad (9)$$

The present solution is enhanced by the difference mean is given as follows,

$$x_{new,i} = x_{old,i} + Difference\_Mean\ E_i \quad (10)$$

4.3.2 Learner Phase

The knowledge of learners is influenced by using two different means, one mean can be used to enhance learners input knowledge through the teacher and the other mean can be used to enhance learner input knowledge through interaction of knowledge between learners with each others. By using the help of various group discussions, presentations and communications, etc, a learner interacts randomly with other learners to enhance their knowledge. A learner is able to learn something more if the other learner has more knowledge or capability than him/her. For population size Pn learner phase is expressed as,

For varying loop i from 1 to population size Pn. Two learners Xi and Xj are randomly chosen where i < > j, means they should not be same learners.

If fitness function values, f (Xi) < f (Xj) then Xnew,i = Xold,i + ri (Xi – Xj) (11)

If fitness function values, f (Xi) ≥ f (Xj) then Xnew,i = Xold,i + ri (Xj – Xi) (12)

Accept the Xnew if it provides a better fitness function value.

4.3.3. Proposed MTLBO

The proposed MTLBO method has the teacher phase is similar to teacher phase of used in TLBO algorithm. The learner phase is modified by expanding the level of learner phase. According to learner phase concept a learner communicates with other existing learners randomly to improve his standard. A learner can gain knowledge and learn more if other learner has better knowledge than him/her. Additionally, in this method the learner has to follow best learner among them usually treated him as a team leader. The PSO updates the particles positions by following their own personal best position as well as global best position among all particles present in the swarm in a single phase. In this proposed method MTLBO is similar to the concepts of PSO, where personal/previous best learner knowledge and global best learner knowledge of all learners are applied in the single learner phase. In MTLBO a learner can learn something new if the other learner has more knowledge than him quite similar as updating using personal best position in PSO method. Additionally, the learners can also learn from the best learner or team leader among themselves similar as updating using global best position in PSO method. The learner knowledge modification in the Learner phase is evaluated as,

For varying loop i from 1 to population size Pn. Two learners Xi and Xj are randomly chosen where i < > j, means they should not be same learners.

If fitness function values, f (Xi) < f (Xj) then Xnew,i = Xold,i + ri (Xi – Xj) + ri (Xg – Xi) (13)

If fitness function values, f (Xi) ≥ f (Xj) then Xnew,i = Xold,i + ri (Xj – Xi) + ri (Xg – Xi) (14)

Here, Xg represents the Knowledge of best learner who is acting as team leader. Xnew is accepted if it provides a better fitness value for used fitness function.

The flow of the MTLBO algorithm is represented in Figure1A.

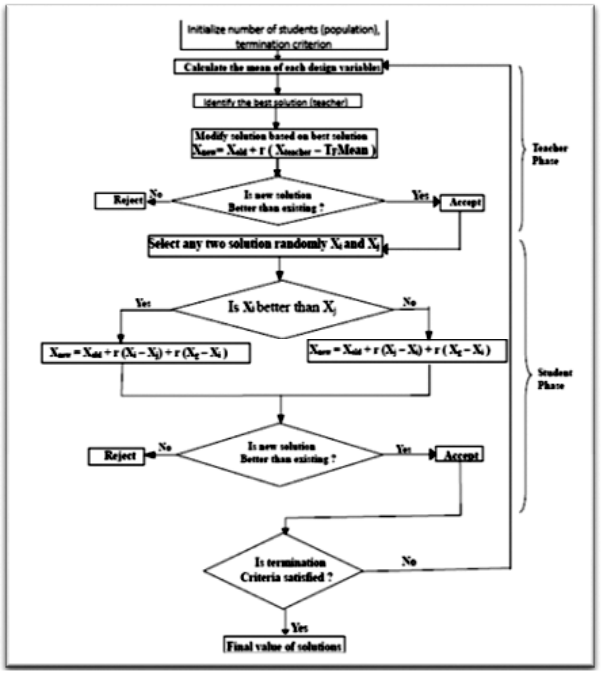

Figure 1A: FLOWCHART MTLBO

5. EXPERIMENTS AND RESULTS

5.1. Experiment 1

5.1.1. Proposed MTLBO for Function Optimization

In this experimental study we have tried to optimize benchmark functions. Functions have different characteristics like separability, multimodality and regularity. A multimodal function has two or more local optimal solutions. A separable function can be represented by separate sum of functions of variable. A function is said to be regular if it is differentiable at each point. It is quite hard to optimize functions of non-separable with added multimodal properties. Also, functions difficulty increases with the distribution of local optima solutions randomly in solutions space. Also, problem difficulty enhances with the increase of dimensions of the function. The results of application of proposed method and other well known methods are compared with each other by applying different benchmark functions. Details of parameters and properties of various benchmark functions are mentioned in the following Table.1.

Table 1
Properties of various functions

Name Offunction	*Range*	*Min-value*	*Multimodality Y/N ?*	*Separablity Y/N ?*	*Regularity Y/N ?*
Sphere(FUN1)	-100 to 100	0	N	Y	Y
Step(FUN2)	-100 to 100	0	N	Y	N
Schwefel(FUN3)	-500 to 500	0	Y	Y	N
Rosenbrock(FUN4)	-100 to 100	0	N	N	Y
Rastrigin(FUN5)	-5.12 to 5.12	0	Y	Y	Y
Griewank(FUN6)	-600 to 600	0	Y	N	Y
Ackley(FUN7)	32 to 32	0	Y	N	Y
Penalized(FUN8)	-50 to 50	0	Y	N	Y

In this experiment, eight different type of benchmark functions with different properties have been taken from Akay and Karaboga [33] and Rao, Savsani, Vakharia [21]. The results of MTLBO method

compared with other popular meta-heuristic methods such as PSO, DE, ACO and GA. The mean results have been taken for each algorithm for comparison with different runs results. This experiment has been performed on above mentioned functions by taking high dimension of 500 for evaluation. The results of experiment are calculated for 30 independent runs. Here the functions are evaluated for maximum of 100,000. Just like TLBO, in MTLBO also maximum number of functions evaluations is set as 2000, that is merely equal to one fifth of others algorithms. The population size is taken as 10 as mentioned by Akay and Karaboga. population size is 10 for Penalized and Schwefel function and population size is 50 for Rosenbrock function. Table 2. shows MTLBO results have outperformed all other meta-heuristic algorithm results within 2000 function evaluations.

Table 2
Results obtained by PSO, DE, ACO, GA and MTLBO

FunctionUSED	*PSO ALGORITHM*	*DE ALGORITHM*	*ACO ALGORITHM*	*GA ALGORITHM*	*MTLBO ALGORITHM*
FUN1	180.16	21.33	9.67	231.02	0
FUN2	1621	1998.03	2005.03	19520	0
FUN3	-98168.1	-138152.03	-19090.65	-18429.58	-180206.28
FUN4	1.08E+06	7.62E+10	1110.65	1497.80	386.50
FUN5	998.97	473.58	1187.73	1888	0
FUN6	02.12	0.534	0.044	12.20	0
FUN7	2.98	12.0	0.147	4.26	0
FUN8	6.17	1.37E+10	2.35E-01	6.9981	1.2E-02

Also, the results show that MTLBO is very effective at very high dimensions complex functions.

5.2. Experiment 2

5.2.1. Proposed MTLBO Algorithm for MLP Training

The MTLBO-MLP classifier is used to evaluate the SpamAssasin dataset and compared the results with some of the most popular algorithms like GA, PSO, DE, ACO. and BP algorithms. The algorithm to train MTLBO is free from algorithmic parameters is described as:

Algorithm for training MLP using MTLBO:

1. Input Data Set
2. Divide the Data into Training data and Testing Data
3. Initialize the weight and bias by taking random values
4. Train the MLP by taking weight, bias and training dataset
5. Calculate the mean square error
6. Repeat
7. Update the weight and bias by applying MTLBO
8. Train the MLP
9. Calculate the error
10. Until permissible error/Number of steps
11. Apply test set to find the classification percentage accuracy.

The parameters setting is one of the important task for training MLP using different algorithms. The parameters that are used in the comparison of different training algorithms are elaborated in the following Table.3.

In proposed method, we tried hidden layer with dissimilar numbers of active neurons present in the hidden layer of the MLPs. Here we have taken five, ten and fifteen number neurons in MLPs respectively. Dataset we have used here is equally divided into two different parts, one part of dataset is used for training of the MLP and the other part of dataset is applied for testing the MLP. Each experiment we have repeated 10 times for better analysis.

Table 3
Various Parameters and their values used in algorithms.

Algorithm used	*Parameters for algorithm*	*Considered Value*
Ant Colony Optimization	• Initial value of pheromone	1e–06
	• Update constant of pheromone	20
	• constant used for pheromone	1
	• Rate of decay of global pheromone	0.9
	• Rate of decay of local pheromone	0.5
	• Value of sensitivity of pheromone	1
	• Visibility sensitivity value	5
Differential Evolution	• probability of crossover	0.9
	• Value of Differential weight	0.5
Particle Swarm Optimization	• constants for acceleration	[2.1,2.1]
	• range of Inertia weights	[0.9,0.6]
Genetic Algorithm	• probability of crossover	0.9
	• probability of mutation	0.1
	• Sample Selection mechanism	Stochastic universal
Modified Teaching Learning Based Optimization	• Population	30

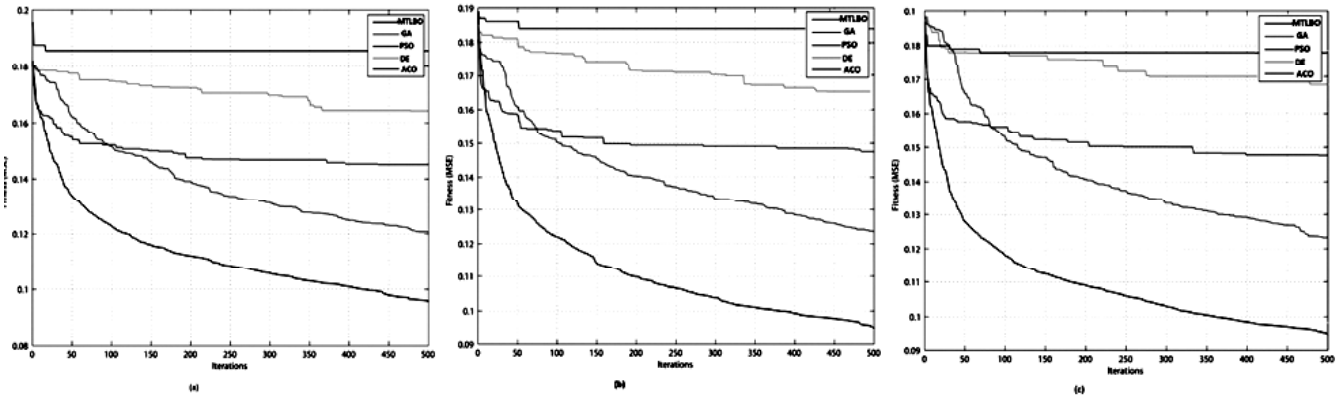

Figure 2: Shows the various Convergence Curves of methods like MTLBO, GA, PSO, DE and ACO training methods for SpamAssassin dataset where MLP neural networks contain (a) 5 Neurons, (b) 10 Neurons and (c) 15 Neurons in the hidden layer.

Figure 2. shows, the performance of different meta-heuristic algorithms applied on the SpamAssassin dataset with different number of nodes or neurons in the hidden layer. From the analysis of figures we can conclude by applying the MTLBO training method we could attain the fastest and better convergence curves. GA and PSO methods came in second place and ACO method provided lowest performance results than others.

All the MLP training algorithms like GA, PSO, DE, ACO, BP, and MTLBO work on MLP model are evaluated based on the accuracy rate. Where, Accuracy rate is calculated as the quantity of instances correctly classified divided by the total quantity of instances Table 4. shows the results of average, standard deviation and best values obtained by each method. We have used these results for comparison. We have observed from the results that the MTLBO classifier has outperformed the results of other well known meta-heuristics and gradient based classifiers. MTLBO has provided the best averages and accuracy results than other spamAssasin dataset classifiers. Also we have noticed there is significant change when we are using fifteen nodes in hidden layer. Therefore, more number of nodes in hidden layer provides better results on proposed dataset clasification.

Table 4
Comparision Results of MTlBO with GA,PSO,DE,ACO and BP.

		Quantity of neurons present in the one hidden layer		
Algorithm	*values*	*Five*	*Ten*	*Fifteen*
MTLBO	Average	0.896	0.898	0.911
	Stdv	0.008	0.007	0.006
	Best	0.880	0.897	0.925
GA	Average	0.831	0.833	0.843
	Stdv	0.019	0.019	0.014
	Best	0.868	0.867	0.888
PSO	Average	0.809	0.799	0.798
	Stdv	0.006	0.011	0.010
	Best	0.818	0.820	0.824
DE	Average	0.776	0.774	0.774
	Stdv	0.008	0.013	0.011
	Best	0.783	0.793	0.785
ACO	Average	0.752	0.749	0.757
	Stdv	0.008	0.014	0.012
	Best	0.765	0.767	0.773
BP	Average	0.776	0.771	0.787
	Stdv	0.022	0.016	0.025
	Best	0.808	0.793	0.825

5.3. Experiment 3

Here, in this particular experiment we have analyzed the results like records of the dataset are correctly classified or not, accuracy of result and confusion matrix. We have followed various types of parameters in this experiment:

Correctly classified Instance: It is used to define ability of algorithm, how much classify the instances correctly.

Correctly classified instances (CCI) = *TP + TN*

Incorrectly Classified Instances: It is used to define the instances which are incorrectly classified.

Incorrectly classified (ICI) = *FP + FN*

Kappa statistic (KS) : The kappa instance or value is a metric which is used to compare the observed accuracy with an expected accuracy.

Kappa Instance (KI) = ((*observed accuracy-expected accuracy*)/(1–*expected accuracy*))

Mean absolute error (MAE) = (number of instances *in correctly Classified/Total Number of Instances*)

Root mean squared error (RMS) : square root of *mean absolute error*

Relative absolute error (RAE) : (*Absolute err/true value*) * 100

Root relative squared error (RRSE) : square root of *relative absolute error*

True Positive (TP): It is used to represent the instances that are truly classified.

False Positive (FP): It is used to represent the instances that are falsely classified.

True Positive Rate (TPR) = TP/(TP+FN)

False Positive Rate (FPR) = TN/(FP+TN)

Precision (P): It is used to define the fraction of retrieved instances in the dataset that are relevant. *P = TP/(TP+FP)*

Recall (R): It is used to evaluate the fraction relevant instances that are retrieved. $R = TP/(TP+FN)$

Accuracy (ACC): It is represented by ACC = $(TP+TN)/(TP+TN+FP+FN)$

Here, TN is the true negative and FN is the false negative. TPR is also described as a sensitivity which is used to evaluate recall. FPR is also known as fall-out.

Table 5
Classification Analysis

CCI	8598
ICI	948
KS	0.904
MAE	0.1014
RMSE	0.3184
RAE	8.48%
RRSE	29.12

Table.5. shows the classification of instances based on different parameters using MTLBO.

Table 6
Confusion Matrix Analysis

Parameters	*Class 0*	*Class 1*	*Aggregate*
TPR	0.927	0.923	0.925
FPR	0.07	0.03	0.05
Precision (P)	0.925	0.925	0.925
Recall	0.927	0.923	0.925
F-Measure	0.926	0.924	0.925

Table 6. shows the aggregate of different parameters on the basics of class 0 and class 1. Class 0 and class 1 signifies the e-mails as a non-spam and spam. These parameters are utilized to analysis the confusion matrix. Confusion matrix are useful for evaluating the performance of classifier tool, as they provide an specific table layout that is used to represent the distribution of correct and incorrect classified instances.

6. CONCLUSION

The proposed work, a meta-heuristic algorithm inspired from nature named as MTLBO developed from concepts applied in PSO and TLBO algorithms. In experiment .1. the convergence and accuracy of developed approach is evaluated on various benchmark functions and compared with results of other meta-heuristic algorithms like GA, PSO, DE and ACO. In the experiment.2.MTLBO has been used for spam dataset classification and has shown better results than gradient decent based BP and other meta-heuristic approaches. Finally, in experiment .3.various parameters and confusion matrix are employed to evaluate the performance of classifier based on MTLBO. The results of the experiments concluded that MTLBO is more effective and efficient in solving nonlinear complex problems and successful in avoiding local minima problem, also it has fast rate of convergence to solution than other popular methods.

7. FUTURE SCOPE

The proposed optimization MTLBO algorithm can be applied to other benchmark problems where, PSO is sometime incapable due to improper parameters selection. MTLBO needs to be applied to other engineering problems related to clustering and classification. Also, we have to apply MTLBO for more complex datasets. Hopefully the proposed MTLBO algorithm can be applied to more nonlinear problems.

REFERENCES

[1] M. N. Marsono, M. W. El-Kharashi, and F. Gebali, (2008) "Binary LNS-based naïve Bayes inference engine for spam control: Noise analysis and FPGA synthesis", IET Computers & Digital Techniques.

[2] Guzella, T.S. and Caminhas, W.M. (2009) A Review of Machine Learning Approaches to Spam Filtering. Expert Systems with Applications, 36, 10206-10222. http://dx.doi.org/10.1016/j.eswa.2009.02.037

[3] Rao, J.M. and Reiley, D.H. (2012) The Economics of Spam. The Journal of Economic Perspectives, 26, 87-110. http://dx.doi.org/10.1257/jep.26.3.87

[4] Stern, H., et al. (2008) A Survey of Modern Spam Tools. CiteSeer.

[5] Su, M.-C., Lo, H.-H. and Hsu, F.-H. (2010) A Neural Tree and Its Application to Spam E-Mail Detection. Expert Systems with Applications, 37, 7976-7985. http://dx.doi.org/10.1016/j.eswa.2010.04.038

[6] Cranor, L.F. and LaMacchia, B.A. (1998) Spam! Communications of the ACM, 41, 74-83. http://dx.doi.org/10.1145/280324.280336

[7] Kanich, C., Weaver, N., McCoy, D. Halvorson, T., Kreibich, C., Levchenko, K., Paxson, V., Voelker, G.M. and Sa- vage, S. (2011) Show Me the Money: Characterizing Spam-Advertised Revenue. USENIX Security Symposium, 15.

[8] Stone-Gross, B., Holz, T., Stringhini, G. and Vigna, G. (2011) The Underground Economy of Spam: A Botmaster's Perspective of Coordinating Large-Scale Spam Campaigns. USENIX Workshop on Large-Scale Exploits and Emer- gent threats (LEET), Vol. 29.

[9] Pérez-Díaz, N., Ruano-Ordás, D., Fdez-Riverola, F. and Méndez, J.R. (2012) Sdai: An Integral Evaluation Methodol ogy for Content-Based Spam Filtering Models. Expert Systems with Applications, 39, 12487-12500. http://dx.doi.org/10.1016/j.eswa.2012.04.064

[10] Amayri, O. and Bouguila, N. (2010) A Study of Spam Filtering Using Support Vector Machines. Artificial Intelligence Review, 34, 73-108.

[11] Song, Y., Kocz, A. and Giles, C.L. (2009) Better Naive Bayes Classification for High-Precision Spam Detection. Software: Practice and Experience, 39, 1003-1024.

[12] Herrero, A., Snasel, V., Abraham, A., Zelinka, I., Baruque, B., Quintian, H., Calvo, J., Sedano, J. and Corchado, E. (2013) Combined Classifiers with Neural Fuser for Spam Detection. [Advances in Intelligent Systems and Computing] International Joint Conference CISIS12-ICEUTE12-SOCO12 Special Sessions Volume 189.

[13] Manjusha, R. and Kumar, K. (2010) Spam Mail Classification Using Combined Approach of Bayesian and Neural Network. IEEE 2010 International Conference on Computational Intelligence and Communication Networks (CICN 2010), Bhopal.

[14] Idris, I., Selamat, A., Nguyen, N.T., Omatu, S., Krejcar, O., Kuca, K. and Penhaker, M. (2015) A Combined Negative Selection Algorithm-Particle Swarm Optimization for an Email Spam Detection System. Engineering Applications of Artificial Intelligence, 39, 33-44. http://dx.doi.org/10.1016/j.engappai.2014.11.001

[15] Arram, A., Mousa, H. and Zainal, A. (2013) Spam Detection Using Hybrid Artificial Neural Network and Genetic Algorithm. 2013 13th International Conference on Intelligent Systems Design and Applications (ISDA), IEEE, 336-340. http://dx.doi.org/10.1109/isda.2013.6920760

[16] Xu, H. and Yu, B. (2010) Automatic Thesaurus Construction for Spam Filtering Using Revised Back Propagation Neural Network. Expert Systems with Applications, 37, 18-23. http://dx.doi.org/10.1016/j.eswa.2009.02.059

[17] Yu, B. and Xu, Z.-B. (2008) A Comparative Study for Content-Based Dynamic Spam Classification Using Four Machine Learning Algorithms. Knowledge-Based Systems, 21, 355-362. http://dx.doi.org/10.1016/j.knosys.2008.01.001

[18] Mirjalili, S. (2015) How Effective Is the Grey Wolf Optimizer in Training Multi-Layer Perceptrons. Applied Intelligence, 43, 150-161. http://dx.doi.org/10.1007/s10489-014-0645-7

[19] Idris, I., Selamat, A. and Omatu, S. (2014) Hybrid Email Spam Detection Model with Negative Selection Algorithm and Differential Evolution. Engineering Applications of Artificial Intelligence, 28, 97-110.

[20] El-Alfy, E.-S.M. (2009) Discovering Classification Rules for Email Spam Filtering with an Ant Colony Optimization Algorithm. IEEE Congress on Evolutionary Computation, Trondheim, 18-21 May 2009, 1778-1783. http://dx.doi.org/10.1109/cec.2009.4983156

[21] Rao R.V., Savsani V.J., Vakharia D.P.,(2012) Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems, Information Sciences, 183 (2012) 1–15.

[22] sahu A., Panigrahi S., pattnaik S., (2013) An Empirical Study on Classification Using Modified Teaching Learning Based Optimization, IJCSN International Journal of Computer Science and Network, Vol 2, Issue 2.

[23] Alqatawna, J., Faris, H., Jaradat, K., Al-Zewairi, M. and Adwan, O. (2015) Improving Knowledge Based Spam Detection

Methods: The Effect of Malicious Related Features in Imbalance Data Distribution. International Journal of Communications, Network and System Sciences, 8, 118. http://dx.doi.org/10.4236/ijcns.2015.85014

[24] sahu A., pattnaik S.,"Evolving Neuro Structure Using Adaptive PSO and Modified TLBO for Classification", Procedia Computer Science 92 (2016), 450 – 454. http://creativecommons.org/licenses/by-nc-nd/4.0/

[25] El-Sayed, M., El-Alfy, Radwan, E., Abdel-Aal(2011) Using GMDH-based networks for improved spam detection and email feature analysis.Applied Soft Computing, Volume 11, Issue 1

[26] Haykin, S. (1999) Neural Networks: A Comprehensive Foundation. 2nd Edition, Prentice Hall, Upper Saddle River.

[27] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning Internal Representations by Error Propagation. In: Rumelhart, D.E., McClelland, J.L. and the PDP Research Group, Eds., Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, MIT Press, Cambridge, MA, 318-362.

[28] Kennedy, J., Eberhart, R.C., (1995) Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948.

[29] Kennedy, J., Eberhart, R.,(2001) Swarm Intelligence Morgan Kaufmann, 3rd edition. Academic Press, New Delhi, India.

[30] Jiang, M., Luo, Y.P., Yang, S.Y., (2007) Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Information Processing Letters 102 (1), 8–16.

[31] Clerc, M., Kennedy, J., (2002) The particle swarm—explosion, stability, and conver-gence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6 (1), 58–73.

[32] Alqatawna, J., Faris, H., Jaradat, K., Al-Zewairi, M. and Adwan, O. (2015) Improving Knowledge Based Spam Detection Methods: The Effect of Malicious Related Features in Imbalance Data Distribution. International Journal of Communications, Network and System Sciences, 8, 118. http://dx.doi.org/10.4236/ijcns.2015.85014

[33] Akay B., Karaboga D., "Artificial bee colony algorithm for large-scale problems and engineering design optimization", Journal of Intelligent Manufacturing (2010), doi:10.1007/s10845-010-0393-4.