Sentiment analysis via multi-layer perceptron trained by meta-heuristic optimisation

Alboaneen, Dabiah Ahmed; Tianfield, Huaglory; Zhang, Yan

Sentiment Analysis via Multi-Layer Perceptron Trained by Meta-Heuristic Optimisation

Dabiah Ahmed Alboaneen*†, Huaglory Tianfield*, and Yan Zhang*

*Department of Computer, Communications and Interactive Systems, Glasgow Caledonian University, Glasgow, United Kingdom
†Department of Computer, University of Imam Abdulrahman Bin Faisal, Jubail, Kingdom of Saudi Arabia
Email: {dabiah.alboaneen; h.tianfield; yan.zhang}@gcu.ac.uk

Abstract—In this paper, a new tweet analysing approach is proposed, which is composed of two main phases; feature selection and tweets classification. In the first phase, mutual information (MI) is used to select the best set of features to reduce the feature dimensions. In the second phase, a meta-heuristic algorithm is used to optimise weights and biases of multi-layer perceptrons (MLPs) network and then implemented to classify twitter sentiments. Experimental results on existing twitter dataset show better performance of the glowworm swarm optimisation (GSO) based MLP over genetic algorithm (GA)and biogeography-based optimisation (BBO) algorithms.

Index Terms—Sentiment analysis; Twitter; multi-layer perceptrons; glowworm swarm optimisation; genetic algorithm; biogeography-based optimisation

I. INTRODUCTION

Sentiment analysis is a process of classifying opinions, emotions or views from texts, speeches or tweets into categories such as positive, negative or neutral. Sentiment Analysis in twitter is quite difficult due to the characteristics of tweets. Tweets are limited by length and usually has emoticons, slang words and misspellings. These challenges force to have a pre-processing step before feature extraction.

There are two main approaches for sentiment analysis. (i) unsupervised learning approaches that based on sentiment lexicons and (ii) supervised learning approaches that based on classification.

Multi-layer perceptron (MLP) is a type of feed-forward artificial neural networks (ANNs) [1]. The purpose of training MLP networks is to find the best set of connection weights and biases to minimise the prediction (classification or approximation) error. Gradient-based algorithms such as back-propagation (BP) are considered to be a conventional choice for MLP training process [2]. However, for complex problems, gradient-based algorithms suffer from high dependency on the initial solution, high probability of local optima stagnation [3] [4], and slow convergence [5].

Previous researches showed that the performance of sentiment classifiers are topic-dependent [6] [7]. Hence, there is no one classifier is the best for all topics or one classifier consistently outperforms other classifiers. The main contribution of this paper is to investigate the application of meta-heuristic algorithms, which are glowworm swarm optimisation (GSO), biogeography-based optimisation (BBO) and genetic algorithm (GA), in optimising MLP for sentiment analysis.

The remainder of this paper is arranged as follows. Section II presents literature review on Twitter sentiment analysis using meta-heuristics algorithms. Section III puts forward the proposed algorithms for analysing sentiment. Experimental evaluations are discussed in section IV. Finally, Section V draws conclusions and sets future work.

II. LITERATURE REVIEW

Existing studies of using meta-heuristic algorithms for Twitter sentiment analysis can be divided into two categories: (i) to select optimal subset features of extracted features and (ii) to optimise the classifier.

A. Meta-Heuristic Algorithms for Feature Selection

Authors in [8] used GA to select the best features from the extracted features. Then, conditional random forest (CRF) was used to classify sentiments into positive or negative sentiments. Experimental results showed that the proposed algorithm works reasonable in the real calculation.

In [9], a feature selection technique based on binary shuffled frog algorithm (BSFA) was embedded with machine learning algorithm to classify Twitter sentiment. The stop words and stemming were used in pre-processing phase. Features were extracted using term frequency-inverse document frequency (TF-IDF) and optimal features selected using the proposed algorithm. The selected features are used to classify tweets as positive or negative through K nearest neighbour (KNN), naive bayes (NB), logistic model tree (LMT) and radial basis function (RBF) networks classifiers. Twitter corpus from Stanford University which has 1200 tweets, 325 positive and 875 negative tweets, was used to evaluate the proposed algorithm. It was observed that RBF performed better than KNN, NB, and LMT. BSFA- RBF performs better accuracy by 1.57% than BSFA-KNN, by 0.68% than BSFA-NB, and by 0.39% than BSFA-LMT.

B. Meta-Heuristic Algorithms for Optimising the Classifier

In [10], the authors proposed a hybrid method based on support vector machine (SVM) and particle swarm optimisation (PSO) to classify movie reviews from Twitter data into watchable and non-watchable. PSO was used to optimise SVM parameters. Features were selected using term frequency (TF)

and TF-IDF. The results showed the improvement of classi-fication accuracy from 71.87% to 77% compared to SVM. In addition, the results revealed that unigram outperforms bigram and trigram as feature extraction methods while TF outperformed TF-IDF as feature selection methods. However, multi-class sentiment classification that include nurtural class need to be considered.

In [11], K-means and cuckoo search (CS) was used to analyse Twitter sentiment. The proposed method modifies the random initialisation process of CS by the solutions obtained from K-means which enhances its performance. Four Twitter datasets have been tested and compared with other meta-heuristic algorithms and the results demonstrated the efficiency of the proposed method. However, no feature selection meth-ods were used to enhance the classification accuracy results.

In this paper, we used mutual information (MI) in feature selection. After that, meta-heuristic algorithms (GSO, BBO, GA) were applied to optimise weights and biases of a MLP classifier to classify tweets into positive, negative, nurtural or irrelevant.

III. PROPOSED APPROACH FOR SENTIMENT ANALYSIS

The proposed approach is composed of two major stages. The first is feature selection while the second is classification model development. The goal of the feature selection process is to find the set of features that enables the maximum classification performance of the classifier in the second stage. MI technique is used to select the possible subsets of features on the training dataset. In the second stage, meta-heuristic algorithm (GSO, BBO, GA) based MLP is trained with the features subset selected in the first stage.

In specific, the proposed approach works in four phases; (i) phase 1: pre-processing of tweets, (ii) phase 2: feature extraction, (iii) phase 3: feature selection and (v) phase 4: the hybrid meta-heuristic algorithm with MLP algorithm to classify the tweets. Flowchart of the proposed approach has been shown in Fig. 1.

A. Pre-processing

The tweets in the dataset have unwanted words, URLs, stop words etc. These tweets need to be checked before extracting features. The pre-processing of the data is a very important step as it decides the efficiency of the following steps. This step aims to make the data more machine readable and reduce ambiguity in feature extraction. In this paper, noise data was eliminated from the dataset using the following steps:

Cleaning:
 (i) Remove all the URLs
 (ii) Remove only #, from the hash-tag and keep the word as it is. For example, #Twitter is replaced with Twitter. Hash-tags provide some useful information.
 (iii) Remove parenthesis, forward slash (/), backward slash (\), and dash from tweets.
 (iv) Remove punctuation marks and digits.
 (v) Replace multiple white spaces with single white space.
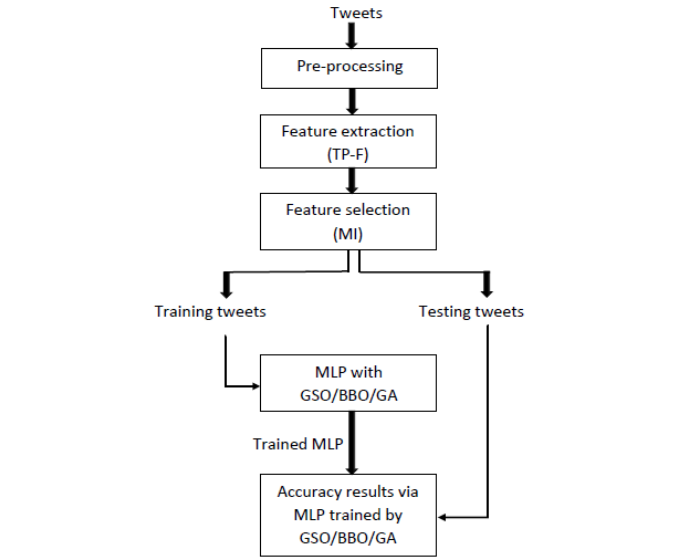 (vi) Convert all the words into lower case.

Fig. 1: Proposed approach for twitter sentiment analysis.

(vii) Remove all the stop words such as, "a", "is", "the", etc. since these words are commonly used in tweets and hold no additional information.

Tokenisation: It is the process of breaking a text into n-grams. Tokens can be separated by white space characters. The n-grams can be uni-grams which divides the tweet into single words, two words (called bi-grams), or three words (called tri-grams).

Stemming: This is the process of replacing the word with its root or stem. Advantage of stemming is that it makes comparison between words simpler, as we do not need to deal with complex grammatical transformations of the word. There are two main stemmers: Porter stemmer and Krovetz stemmer. Porter stemmer is the most widely used stemming algorithm.

B. Feature Extraction

A feature is any variable which can help classifiers in differentiating between different classes. Feature extraction step consists in transforming texts into numerical features usable for machine learning. In this paper, after applying the pre-processing, tweets are converted into the matrix where rows represent the tweets and columns represent the features. Features are extracted using terms presence and frequency (TP-F), these features are individual words or n-grams words with their frequency counts. The occurrences of words in the dataset are counted. Each individual word occurrence frequency is treated as a feature.

C. Feature Selection

In feature extraction step, a very large number of features are often generated, specially in the case of tweet sentiment classification. Moreover, training classifiers on a large dataset is computationally expensive. Therefore, feature selection techniques can select an optimal subset of features, which

reduces the dimensionality of the dataset, helps to reduce computational costs and possibly improves classification performance. In this paper, we used one of filter feature selection techniques which is called MI. In this type of technique, a set of features is selected based on a specific mathematical equation and used with a classifier.

D. Classification

This section puts forward the process of using GSO, BBO and GA algorithms as a trainer for MLP network with one hidden layer. The MLP with initial settings is employed first to obtain the initial solution and then meta-heuristic algorithm optimises the weights and biases to minimise the classification error rate of the MLP.

1) Multi-layer Perceptron (MLP): MLP is a network of neurons called perceptrons. Neurons are hierarchically arranged in multiple connected layers. The MLP network starts with an input layer followed by hidden layers and ends with an output layer. Hidden layers provide the computational processing in the network to produce the network outputs. Fig. 2 illustrates an example of MLP with one hidden layer. The connections between the layers are called weights W, which are normally defined between 0 and 1. The output value of each neuron in each layer is calculated in two subsequent stages as below:

In the first stage, the weighted summation of the input values is calculated using the following equation:

$$\forall l \in \{1, 2, ..., j\}, h_l = \sum_{i=1}^{m} W_{il}^H I_i + \beta_l^H \qquad (1)$$

where I_i is the input variable i, W_{il}^H is the connection weight between i input neuron and the hidden neuron l, m is the total number of inputs and β_l^H is the bias of the lth hidden neuron.

In the second stage, the output value of each neuron in the hidden layer is calculated based on a weighted summation using an activation function. In MLP, the sigmoid activation function is commonly used to map the hidden layer with output values. The sigmoid function can be calculated as follows:

$$\forall l \in \{1, 2, ..., j\}, H_l = sigmoid(h_l) = \frac{1}{1 + e^{-h_l}} \qquad (2)$$

The final output of the network is calculated as below:

$$\forall k \in \{1, 2, ..., n\}, o_k = \sum_{l=1}^{j} W_{lk}^O H_l + \beta_k^O \qquad (3)$$

$$\forall k \in \{1, 2, ..., n\}, O_k = sigmoid(o_k) = \frac{1}{1 + e^{-o_k}} \qquad (4)$$

where W_{lk}^O is the connection weight between the lth hidden neuron and the kth output neuron. β_k^O is the bias of the kth output neuron.

Two aspects have been taken into account when designing the proposed approach: (i) the encoding scheme of the search
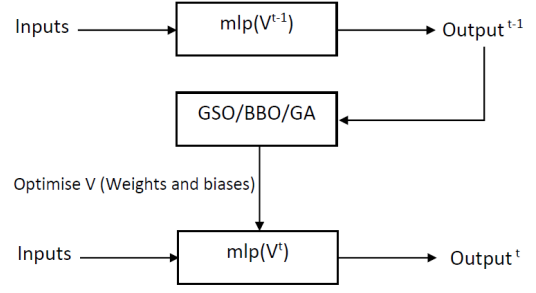

Fig. 2: MLP trained by meta-heuristic optimisers.

agents in the meta-heuristic algorithms and (ii) the fitness function.

(i) **Encoding Scheme:** Each individual in GSO, BBO or GA is encoded as a vector of real numbers in the range $[0, 1]$ to represent a candidate MLP network. Vectors include three parts: the connection weights between the input layer and the hidden layer, the connection weights between the hidden layer and the output layer and the biases. The dimension D of the problem (the length of each vector equals the total number of weights and biases in the network) can be calculated as shown below.

$$D = (m * j) + j + (j * n) + n \qquad (5)$$

where m represents the number of input variables (features) in the dataset, j is the number of neurons in the hidden layer and n is the number of output variables.

(ii) **Fitness Function:** Each individual is evaluated according to its fitness. This evaluation is done by passing the vector of weights and biases to MLP; then the MSE criterion is calculated based on the difference between the actual and predicted values by the generated agents (MLPs) for all training instances. After the maximum number of iterations is met, the optimal solution is finally achieved, which is regarded as the weights and biases of a MLP network. The aim is to minimise the value of MSE below.

$$MSE = \sum_{t=1}^{T} \frac{\sum_{k=1}^{n} (o_k^t - d_k^t)^2}{T} \qquad (6)$$

where T is the total number of instances in the training dataset, n is the total number of outputs, o_k^t is the actual output of the kth input when the tth training instance is used and d_k^t is the desired output of the kth input when the tth training instance is used.

IV. PERFORMANCE EVALUATION

A. Meta-heuristic Algorithms

Glowworm Swarm Optimisation: GSO is based on the behaviour of glowworms. A glowworm that produces more light (high luciferin) means that it is closer to an actual position and has a high objective function value. A GSO

algorithm comprises four phases, i.e., initialisation, luciferin updating, moving and local radial range updating. The GSO algorithm can be formulated as in Algorithm 1. [12] [13] GSO algorithm starts by positioning glowworms randomly in the search space and all the glowworms contain an equal quantity of luciferin. Each glowworm y converts the objective function value $f(x_y(t+1))$ at its current location $x_y(t)$ to a luciferin value $\ell_y(t+1)$ by using the formula below.

$$\ell_y(t+1) = (1-p)l_y(t) + \gamma f(x_y(t+1)) \quad (7)$$

where $\ell_y(t)$ is the luciferin value of glowworm y at time t, p is the luciferin decay coefficient ($0 < p < 1$) and γ is the luciferin enhancement coefficient.

Then, each glowworm chooses to move toward one of its neighbours z, using probability, that has a higher luciferin value within the local radial range γ_d.

$$Z_y(t) = \{z : ||x_z(t) - x_y(t)|| \le \gamma_d^y(t); \ell_y < \ell_z(t)\} \quad (8)$$

where $Z_y(t)$ is the neighbour set, z is the index of glowworm close to y, $x_z(t)$ and $x_y(t)$ are locations of glowworm z and glowworm y, respectively, $\ell_y(t)$ and $\ell_z(t)$ are luciferin values for glowworm y and glowworm z, respectively. $||x||$ is the Euclidean norm of x, and $\gamma_d^y(t)$ represents the local radial range.

$$p_{yz}(t) = \frac{\ell_z(t) - \ell_y(t)}{\sum_{s \in Z_y(t)} \ell_w(t) - \ell_y(t)} \quad (9)$$

where $p_{yz}(t)$ is the probability of glowworm y moving to glowworm z.

$$x_y(t+1) = x_y(t) + s\left(\frac{x_z(t) - x_y(t)}{||x_z(t) - x_y(t)||}\right) \quad (10)$$

where $x_y(t+1)$ and $x_y(t)$ are the new and current locations of glowworm, respectively and s is the size of moving step.

Finally, the local radial range γ_d^y is updated as below in order to formulate the neighbour set.

$$\gamma_d^y(t+1) = \min\{\gamma_s, \max\{0, \gamma_d^y(t) + \beta(z_t - |Z_y(t)|)\}\} \quad (11)$$

where β is the change rate of the neighbourhood range.

Biogeography-Based Optimisation: BBO was proposed by Simon [14] in 2008. It studies the geographical distribution of species. The habitability (suitability for biological residence) of an island is indicated by its habitat suitability index (HSI), which is determined by a number of independent variables called suitability index variables (SIVs). The higher the HSI an island is, the more the species on the island, the lower its immigration rate, and the higher its emigration rate. Species may migrate from high HSI islands to low HSI islands. The arrival of new species may increase the HSI of an island by increasing the diversity of species on the island. If the HSI of an island is too low, existing species on the island may become extinct. There are two key operators in BBO algorithm: migration and mutation. The migration is designed to probabilistically

---

Algorithm 1: GSO: <u>G</u>lowworm <u>S</u>warm <u>O</u>ptimisation

1 Initialise parameters β, p, s, z_t
2 $\forall y$, set $\ell_y(0) = \ell_0$
3 $\forall y$, set $\gamma_d^y(0) = \gamma_0$
4 **while** *termination condition not met* **do**
5   **for** $y \in m$ **do**
6 $\ell_y(t+1) = (1-p)\ell_y(t) + \gamma f(x_y(t+1))$
7 $Z_y(t) = \{z : ||x_z(t) - x_y(t)|| \le \gamma_d^y(t); \ell_y(t) \le \ell_z(t)\}$
8   **for** *each* $z \in Z_y(t)$ **do**
9 $p_{yz}(t) = \frac{\ell_z(t) - \ell_y(t)}{\sum_{w \in Z_y(t)} \ell_w(t) - \ell_y(t)}$
10 $x_y(t+1) = x_y(t) + s\left(\frac{x_z(t) - x_y(t)}{||x_z(t) - x_y(t)||}\right)$
11 $\gamma_d^y(t+1) = \min\{\gamma_s, \max\{0, \gamma_d^y(t) + \beta(z_t - |Z_y(t)|)\}\}$
12 $t = t+1$
13 **return** Optimal Solution

Algorithm 2: BBO: <u>B</u>iogeography <u>B</u>ased <u>O</u>ptimisation

1 Initialise a population of solutions x_k
2 Set emigration probability μ_k
3 Set immigration probability λ_k
4 **while** *termination condition not met* **do**
5   **for** *each solution* x_k **do**
6   Set z_k is a temporary population $z_k \leftarrow x_k$
7   **for** *each individual* z_k **do**
8   **for** *each independent variable index* s **do**
9   Use λ_k to probabilistically decide whether to immigrate to z_k
10   **if** *immigrating* **then**
11   Use μ_i to probabilistically select the emigrating individual x_j
12 $z_k(s) \leftarrow x_j(s)$
13   Probabilistically mutate z_k
14 $x_k \leftarrow z_k$

---

share SIVs between solutions, thus increasing the quality of low HSI solutions. The mutation is used to probabilistically replace SIVs in a solution by randomly generating new SIVs. The BBO pseudocode is illustrated in Algorithm 2.

Genetic Algorithm: GA was first proposed by Holland in 1975 [15] and belongs to the larger class of evolutionary algorithms (EA), which generate solutions using techniques inspired by natural evolution, such as mutation, selection, and crossover. With these three simple operators a GA works as illustrated in Algorithm 3.

B. Experimental Setup

Sanders Analytics have collected dataset for Apple Corporation on four different topics namely; Apple, Google,

---
Algorithm 3: GA: Genetic Algorithm
---

1 Generate a population.

2 Evaluate population using fitness function.

3 **while** *termination condition not met* **do**

4   Select the chromosomes using selection operator for reproduction.

5   Apply the crossover operation on the pair of chromosomes obtained in step 4.

6   Apply the mutation operation on the chromosome.

7   Evaluate the fitness value of new generated chromosomes "offsprings".

8   Update the population by replacing bad solutions with better chromosomes from offsprings.

9 **return** Best chromosome as the final solution.

TABLE I: Parameters settings

Algorithm	Parameter	Value
GSO	Luciferin decay coefficient	0.4
	Luciferin enhancement coefficient	0.6
	Rate of the neighbourhood range	0.08
	No. of neighbours	5
	Step size of moving	0.03
	Initial luciferin	0.05
BBO	Mutation probability	0.05
	Number of elites	2
GA	Crossover probability	1
	Mutation probability	0.01
-	Number pf population	50
-	Number of iterations	20

Microsoft and Twitter. Each tweet was manually annotated to either positive, negative, neutral or irrelevant by Niek Sanders [16]. Any tweet which contains positive indicator or topic is considered as positive tweets. Those tweets which neither have positive nor negative indicators, or have mixed positive and negative indicators, or have simple factual statements, or have questions with no strong emotions are considered as neutral tweets. Tweets with negative indicator or topic are classified as negative tweets. We extracted 1050 tweets from the Sanders corpus, which contains 270 positive tweets, 260 negative tweets, 260 neutral tweets and 259 irrelevant tweets. We used 80% of the dataset as training dataset while the rest 20% as the test dataset.

In the second stage of the proposed approach, meta-heuristic algorithm to optimise MLP classifier is applied on the testing data using the subset of features found by the MI. For final evaluation, the accuracy rate and fitness function are calculated. The parameter settings for all algorithms have been presented in Table I.

C. Experimental Results

To measure the performance of the proposed method, two parameters have been considered namely; classification accuracy rate and fitness function value. For comparison, each algorithm has been executed 10 times. Table II and III represent the average (AVG), standard deviation (STD) and

TABLE II: Comparison of all algorithms in terms of classification accuracy rate

Dataset		GSO-MLP	BBO-MLP	GA-MLP
Twitter sentiment corpus	AVG	**54.00%**	48.86%	51.90%
	STD	0.0072	0.0110	0
	Best	**54.76%**	50.48%	51.90%

TABLE III: Comparison of all algorithms in terms of fitness value

Dataset		GSO-MLP	BBO-MLP	GA-MLP
Twitter sentiment corpus	AVG	0.5197	0.4958	**0.4939**
	STD	0.0070	0.0066	0.0071
	Best	0.5137	0.4875	**0.4839**

best (Best) values of classification accuracy rate and fitness function values, respectively.

Table II represents AVG, STD and Best values of classification accuracy rates of all three algorithms for Twitter dataset. From this table, it can be seen that the GSO-MLP gives the best accuracy (54%) in analysing tweets among the three algorithms while BBO has the lowest accuracy of 48.86%.

Table III represents AVG, STD and Best values of MSE of all three algorithms for Twitter dataset. Although the GSO-MLP has the highest MSE value, this result is close to the other algorithms.

To show the convergence behaviour of all the considered methods and proposed method, convergence curves have also been plotted in Fig. 3. In the convergence plot, the x axis represents the number of iterations and y axis represents the average MSE values over 10 runs. From the convergence plots, it is observed that the proposed method converges quickly as compared to all the considered methods and gives better results.

Figures 4 and 5 depict the box-plots employed using GSO, BBO and GA algorithms. The box-plots in Fig. 4 are used to analyse the variability in getting MSE values for 10 MSEs obtained by each trainer in the last iteration while the box-plots in Fig. 5 are used to analyse the variability in getting classification accuracy rates for 10 runs obtained by each trainer. In these plots, the box relates to the interquartile range, the whiskers represent the farthest values and the bar in the box represents the median value. The box-plots show that
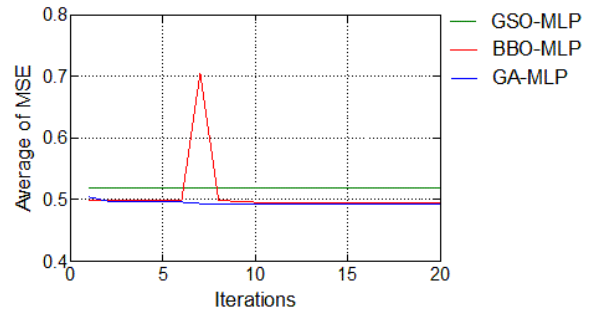

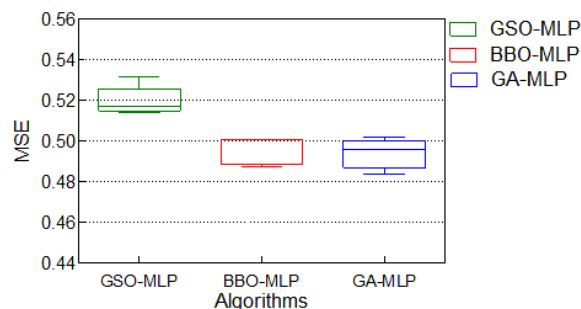Fig. 3: Convergence curves of all algorithms
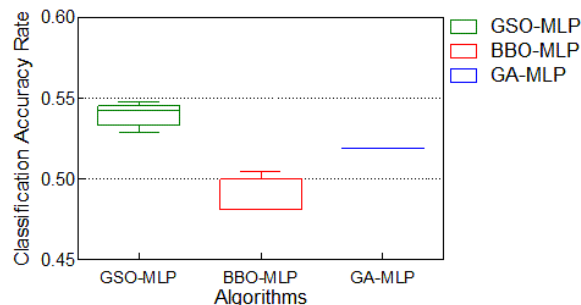
Fig. 4: Boxplot charts of all algorithms

Fig. 5: Boxplot charts of all algorithms

GSO algorithm performed well for training MLP networks for Twitter dataset.

V. CONCLUSION AND FUTURE WORK

In this paper, a hybrid meta-heuristic approach was proposed for analysing Twitter sentiment. The proposed approach is applied in two stages. In the first stage, MI is performed as a feature selection technique while in the second stage, hybrid meta-heuristic and MLP is applied for classifying tweets. Evaluation results show that GSO-MLP outperforms BBO-MLP and GA-MLP for classifying tweets. As to future work, more twitter datasets need to be evaluated with more feature extraction and selection techniques to speed the process of classifier and improve the classification accuracy rates.

REFERENCES

[1] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.

[2] D. R. Hush and B. G. Horne, "Progress in supervised neural networks," *IEEE signal processing magazine*, vol. 10, no. 1, pp. 8–39, 1993.

[3] M. Gori and A. Tesi, "On the problem of local minima in backpropagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 76–86, 1992.

[4] Y. Lee, S.-H. Oh, and M. W. Kim, "An analysis of premature saturation in back propagation learning," *Neural networks*, vol. 6, no. 5, pp. 719–728, 1993.

[5] S. E. Fahlman, "An empirical study of learning speed in back-propagation networks," 1988.

[6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[7] Y. Choi and C. Cardie, "Learning with compositional semantics as structural inference for subsentential sentiment analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 793–801, Association for Computational Linguistics, 2008.

[8] J. Zhu, H. Wang, and J. Mao, "Sentiment classification using genetic algorithm and conditional random fields," in *Information Management and Engineering, The 2nd IEEE Int. Conf. on*, pp. 193–196, IEEE, 2010.

[9] N. Yuvaraj and A. Sabari, "Twitter sentiment classification using binary shuffled frog algorithm," *Intelligent Automation & Soft Computing*, vol. 23, no. 2, pp. 373–381, 2017.

[10] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Engineering*, vol. 53, pp. 453–462, 2013.

[11] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, "Twitter sentiment analysis using hybrid cuckoo search method," *Information Processing & Management*, vol. 53, no. 4, pp. 764–779, 2017.

[12] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Glowworm swarm optimisation algorithm for virtual machine placement in cloud computing," in *Ubiquitous Intelligence & Comp., Advanced and Trusted Comp., Scalable Comp. and Communications, Cloud and Big Data Comp., Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Intl IEEE Conf.*, pp. 808–814, IEEE, 2016.

[13] D. Alboaneen, H. Tianfield, and Y. Zhang, "Glowworm swarm optimisation based task scheduling for cloud computing," in *Proc. of the Int. Conf. on Internet of Things and Cloud Comp.,(Cambridge, 22-23 Mar)*, pp. 1–7, ACM, 2017.

[14] D. Simon, "Biogeography-based optimization," *Evolutionary Computation, IEEE Trans. on*, vol. 12, no. 6, pp. 702–713, 2008.

[15] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Comp.*, vol. 2, no. 2, pp. 88–105, 1973.

[16] N. Sanders, "Sanders Analytics." http://www.sananalytics.com/lab/twitter-sentiment/, 2011. [Online; accessed 19-August-2017].