An Efficient Hybrid Multilayer Perceptron Neural Network with Grasshopper Optimization

Article *in* Soft Computing · July 2018

DOI: 10.1007/s00500-018-3424-2

4 authors:

Ali Asghar Heidari
National University of Singapore

Hossam Faris
University of Jordan

Ibrahim Aljarah
University of Jordan

Seyedali Mirjalili
Griffith University

Soft Computing
https://doi.org/10.1007/s00500-018-3424-2

METHODOLOGIES AND APPLICATION

An efficient hybrid multilayer perceptron neural network with grasshopper optimization

Ali Asghar Heidari[1] · Hossam Faris[2] · Ibrahim Aljarah[2] · Seyedali Mirjalili[3]

Abstract

This paper proposes a new hybrid stochastic training algorithm using the recently proposed grasshopper optimization algorithm (GOA) for multilayer perceptrons (MLPs) neural networks. The GOA algorithm is an emerging technique with a high potential in tackling optimization problems based on its flexible and adaptive searching mechanisms. It can demonstrate a satisfactory performance by escaping from local optima and balancing the exploration and exploitation trends. The proposed GOAMLP model is then applied to five important datasets: breast cancer, parkinson, diabetes, coronary heart disease, and orthopedic patients. The results are deeply validated in comparison with eight recent and well-regarded algorithms qualitatively and quantitatively. It is shown and proved that the proposed stochastic training algorithm GOAMLP is substantially beneficial in improving the classification rate of MLPs.

Keywords Optimization · Classification · Grasshopper Optimization Algorithm · Multilayer perceptron · Medical diagnosis

1 Introduction

Artificial neural networks (ANNs) (McCulloch and Pitts 1943) can learn, recognize, and tackle many intricate and complex tasks in engineering and science (Heidari and Abbaspour 2018; Aljarah et al. 2016; Faris et al. 2016b; Esteva et al. 2017; Trujillo et al. 2017). ANNs can be seen as influential learning models that can disclose desirable results in dealing with countless supervised/unsupervised machine learning challenges (Krogh 2008). ANNs are suited generally well to deal with machine perception tasks, where the available primary features cannot be individually interpreted (Esteva et al. 2017). Hence, they have been extensively scrutinized and utilized to realize the pattern recognition, clustering, classification, and prediction tasks (Aljarah et al. 2016; Almonacid et al. 2017; Ata 2015; Ding et al. 2013; Lee and Choeh 2014; Chaudhuri and Bhattacharya 2000). For instance, in medical applications, researchers have effectively utilized various forms of neural networks to handle difficult diagnostic tasks (Esteva et al. 2017). They also employed neural networks to classify the biomedical info such as heart diseases and diabetes. The success of these methods is because of the capabilities of ANNs to process a large volume of info during the training phase and decrease the required diagnosis time (Faris et al. 2016a).

A single-layer perceptron (SLP) only has two input and output layers, and it is considered as the most abstracted form of ANNs (Yi-Chung 2014). It was proved that the SLPs cannot efficiently handle the nonlinearly separable patterns (Ojha et al. 2017). In this regard, multilayer perceptron (MLP) NNs were proposed, which do not suffer from the disadvantages of SLPs by employing one or several hidden layers in the ANNs. Therefore, the most utilized class of ANNs is the MLP version (Chen et al. 2015). Some of the core influential features of MLP are its learning capacity, non-

Communicated by V. Loia.

✉ Ibrahim Aljarah
i.aljarah@ju.edu.jo

Ali Asghar Heidari
as_heidari@ut.ac.ir

Hossam Faris
hossam.faris@ju.edu.jo

Seyedali Mirjalili
seyedali.mirjalili@griffithuni.edu.au

1 School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran

2 Business Information Technology Department, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

3 Institute of Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane QLD 4111, Australia

Published online: 30 July 2018

linearity, parallelism, fault tolerance, robustness to noise, and its great capacity to generalize as well (Faris et al. 2016a). The effectiveness of ANNs can be extremely enriched based on the learning strategy employed to train the system (Ojha et al. 2017). According to Kolmogorov's theorem, MLP with a single hidden layer is capable of approximating various continuous functions (Cybenko 1989). In this research, the weights and biases of the single hidden layer (SHL) networks are optimized.

Recently, learning models have attracted considerable attention in machine learning community. For instance, Hamidzadeh and Namaei (2018) proposed a belief-based chaos-based technique for support vector data description. Sadeghi and Hamidzadeh (2018) developed an automatic support vector data description. Hamidzadeh et al. (2017) proposed a chaotic bat algorithm to efficiently deal with the weighted support vector data description. There are also many techniques devoted to the development of related concepts and learning models (Hamidzadeh et al. 2012, 2015, 2016, 2018; Moghaddam and Hamidzadeh 2016; Hamidzadeh and Moradi 2018; Hamidzadeh et al. 2014).

For the MLP, either supervised or unsupervised training strategies can be employed to train the ANNs (Aljarah et al. 2016). The main supervised trainers can also be seen in two categories: gradient-based and stochastic-based approaches (Ojha et al. 2017). The different varieties of back-propagation techniques can be branded as one of the most widespread classes of gradient-descent-based approaches in the literature (Wang et al. 2015). These methods can be utilized as local search algorithms, because of their exploitation tendency (Zhang et al. 2007). In order to attain the global optimum, any optimizer should be capable of making a fine balance between two core capacities: exploration and exploitation. The exploration is vital to explore new and unknown regions on the fitness landscape, and the exploitation is essential to take the advantages of the previously explored positions (Heidari et al. 2017a; Faris et al. 2018a). However, the stagnation to LO, lazy convergence, and strong reliance on parameters are some of the limitations of gradient-descent strategies (Mirjalili 2015; Faris et al. 2016b). In this regard, the necessity of several stochastic optimizers for the training of MLP networks was recognized in the literature (Ojha et al. 2017). The meta-heuristic-based trainers (meta-trainers) can demonstrate a high efficiency in escaping from LO (Aljarah et al. 2016; Faris et al. 2016b).

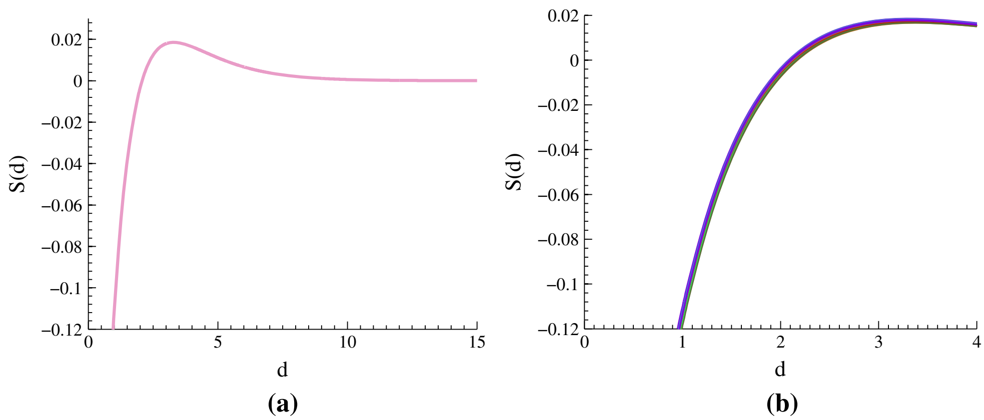

Fig. 1 Demonstration of s when $l = 1.5$ and $f = 0.5$ when d is inside [0, 15] and a window when d is inside [0, 4]. **a** When $l = 1.5$ and $f = 0.5$. **b** When d is inside [0, 4]

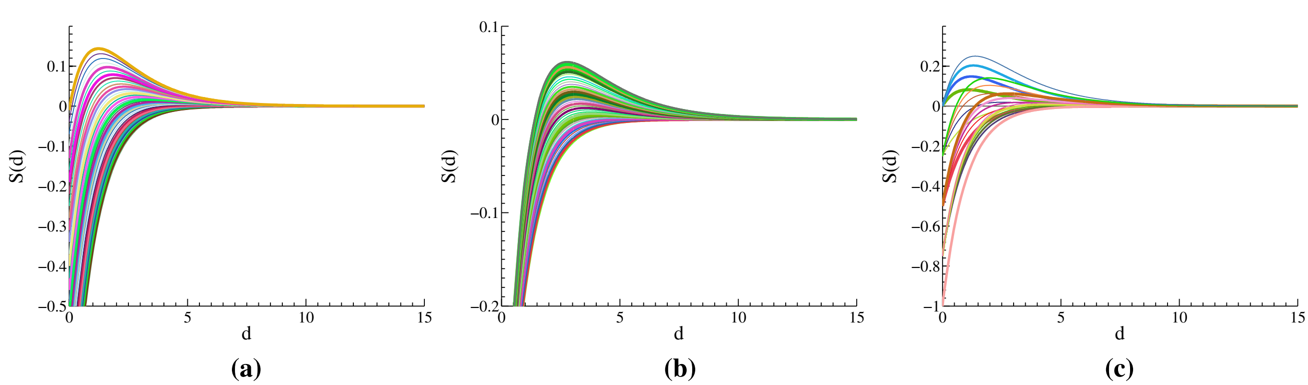

Fig. 2 Dynamics of s-function by changing l and f. **a** $l = 1.5$ and f in [0, 1]. **b** $f = 0.5$ and l in [1, 2]. **c** f in [0, 1], l in [1, 2]

When it is required to simultaneously optimize the structure and weights, the MLP trainer should tackle a large-scale problem (Ojha et al. 2017). In the literature, many researchers frequently have utilized the evolutionary and swarm-based meta-heuristic algorithms (MHAs) to handle MLP networks. The meta-trainers can be deployed in order to optimize not only the connection weights but also the parameters and structure of the MLP network. In this research, several MHAs are evaluated to optimize either weights or biases in the single hidden layer (SHL) networks. Genetic algorithm (GA) is a traditional evolutionary MHA for handling MPL networks (Heidari and Delavar 2016). In Seiffert (2001), the GA has been employed to optimize the connection weights. The obtained results demonstrate that GA can reveal better results than the back-propagation technique when the complexity of the concerned tasks is increased. Gupta and Sexton (1999) have also employed the GA, and they have compared it against the back-propagation technique for the chaos-based time-series prediction applications, and it was exposed that GA can be more efficient. Several other researchers also utilized the GA-based trainers for their developed ANNs (Whitley et al. 1990; Siddique and Tokhi 2001; Sexton and Gupta 2000; Alba and Chicano 2004; Sexton et al. 1999).

Differential evolution (DE) (Mallipeddi et al. 2011) and evolution strategy (ES) (Hansen et al. 2003) are popular, well-known evolutionary optimizers. These MHAs also been utilized by several researchers for training MLP networks (Ilonen et al. 2003; Slowik and Bialko 2008; Wienholt 1993; Wdaa et al. 2008). The obtained results revealed that these MHAs are capable of providing preferable solutions in realizing MLP networks. Swarm-based MHAs (SMHAs) are other class of MHAs. The SMHAs try to inspire the idealistic cooperative and self-organized movements of the swarm such as birds, grasshoppers, whales, and wolves in nature (Heidari et al. 2017a; Heidari and Pahlavani 2017). The particle swarm optimizer (PSO) is one of the first groundbreaking SMHAs inspired from the social life of birds (Jordehi and Jasni 2013; Heidari et al. 2017b). The artificial bee colony (ABC) (Karaboga and Basturk 2007) and ant colony optimization (ACO) (Dorigo et al. 2006) can be reported as other well-established SMHAs. The efficacy of PSO, ACO, ABC-based trainers and their modified variants for MLP networks was intensively evaluated in previous researches. Such contributions can be studied in Jianbo et al. (2008), Braik et al. (2008), Blum and Socha (2005), Socha and Blum (2007), Karaboga et al. (2007), Ojha et al. (2017). The results affirm that these trainers can show a high LO avoidance tendency in dealing with the MLP networks. Whale optimization algorithm (WOA) (Aljarah et al. 2016), gray wolf optimizer (GWO) (Mirjalili 2015), lightning search algorithm (LSA) (Faris et al. 2016a), and social spider optimizer (SSO) (Mirjalili et al. 2015) can be considered as some the newest training methods applied to this problem.

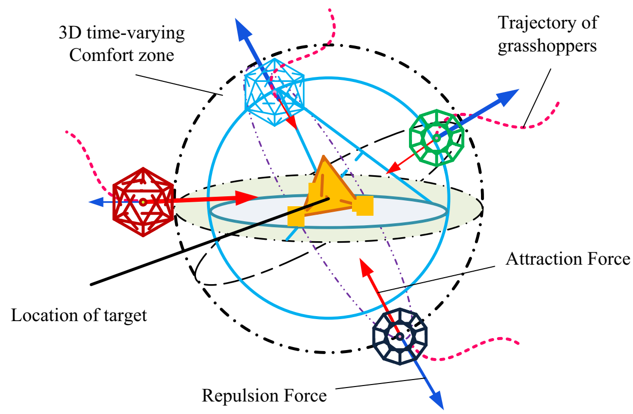

Fig. 3 Corrective patterns among grasshoppers

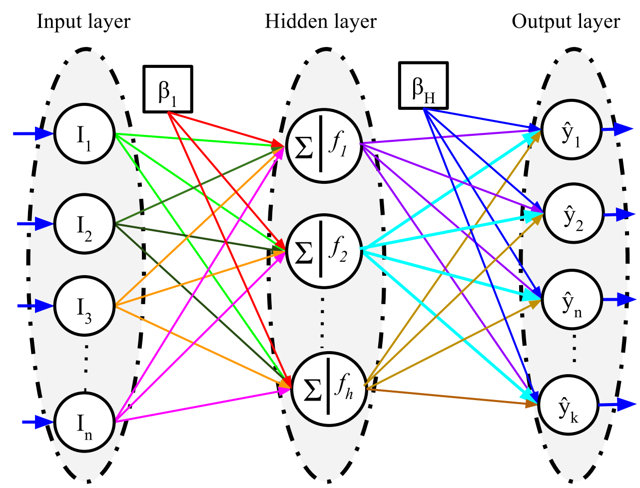

Fig. 4 MLP neural network

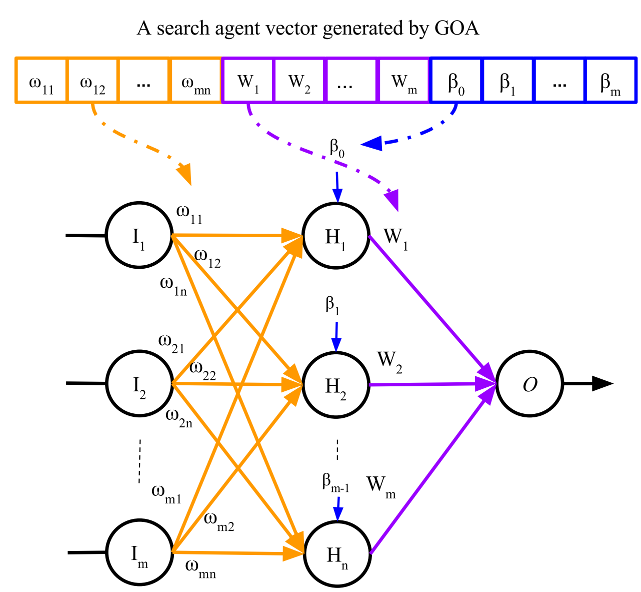

Fig. 5 Solution representation

Although several MHAs have analyzed in past works, the search for global results is still open (Faris et al. 2016a, b). No Free Lunch (NFL) theorem says that an MHA cannot outperform all other MHAs for all classes of problems (Wolpert and Macready 1997). Referring to NFL theory, a new MHA can still be developed for training MLP networks. Inspired by this reason, the main motivation of this paper is to design a new MHA-based trainer for SHL networks. One of the noticeable recent MHAs is the grasshopper optimization algorithm (GOA) (Mafarja et al. 2018). GOA is a new nature-inspired optimizer proposed recently by Saremi et al. (2017). The GOA is an efficient SMHA inspired by the team hunting behaviors of grasshoppers in nature. GOA has shown very promising results in optimizing complex problems and difficult benchmark functions (Saremi et al. 2017; Aljarah et al. 2018a). The main contributions of this research can be summarized as follows:

- A new hybrid stochastic training algorithm using the recently proposed GOA is developed for MLP neural networks.
- In this paper, the exploratory and exploitative capabilities of GOA are utilized to simultaneously realize the optimal weights and biases of the MLP for the first time.
- In this work, the proposed GOAMLP model is applied to deeply study five important medical classification problems related to breast cancer, Parkinson, diabetes, coronary heart disease, and orthopedic patients.
- To verify the effectiveness of the proposed model, this paper also develops eight other MLP-based classifiers based on different evolutionary and SMHAs. We also conduct statistical tests to validate the performance of all approaches.
- Unlike the majority of the previous works which rely only on the accuracy and MSE as evaluation measures, we assessed the performance of all models using different evaluation metrics like specificity, sensitivity, and G-mean.

The obtained results for all investigated problems show a promising performance and high stability of the GOAMLP model in solving complex medical classification problems.

This paper is structured as follows: Sect. 2 describes the GOA optimizer. Section 3 introduces the MLP. The new proposed GOAMLP model is explained in Sect. 4. Section 5 is devoted to the results and discussions. Finally, the concluding remarks and directions for future research are explained in Sect. 6.

Fig. 6 Optimizing MLP network using GOA

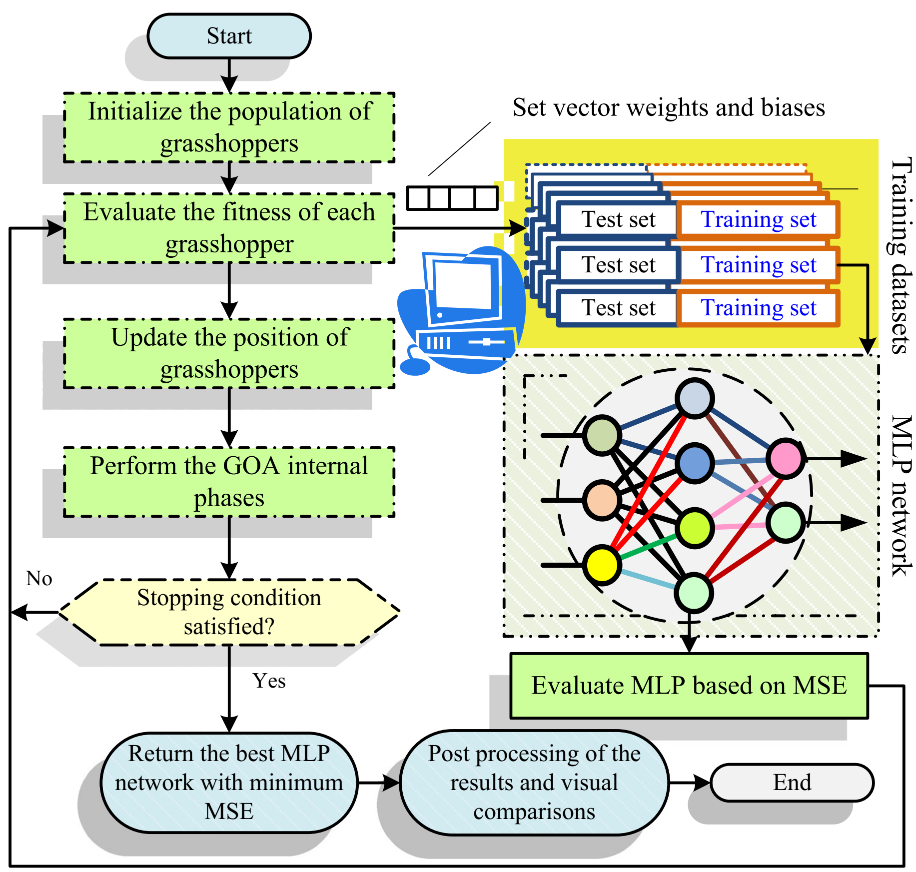

Table 1 Summary of the medial diagnosis classification datasets

No.	Dataset	No. of attributes	No. of training samples	No. of test samples
1	Breast cancer	8	461	238
2	Parkinsons	22	128	67
3	Diabetes	8	506	262
4	Saheart	9	304	158
5	Vertebral	6	204	106

2 Grasshopper optimization algorithm (GOA)

The GOA is a novel optimizer that tries to inspire the social life of grasshopper insects in nature (Saremi et al. 2017). It was verified by the discoverers of the GOA optimizer that it is capable of outperforming several well-established optimizers such as the GSA, BA, FA, GA, and PSO algorithms in dealing with artificial and realistic optimization tasks (Saremi et al. 2017). The main behaviors of the grasshoppers are foraging, target pursuing, and team behaviors in either nymph or adulthood phases. In the larval level, they often exhibit short-length jumps with slow motions. In adulthood, they do long-range and swift movements to obtain food sources from farming areas. To simulate these facts, a model was designed as (Saremi et al. 2017):

$$X_i = S_i + G_i + A_i \tag{1}$$

where X_i is the location of ith insect, S_i shows the social communications, G_i denotes the gravity strength on ith insect, and A_i is the wind advection. Note that Eq. (1) can be written as $Xi = r_1 S_i + r_2 G_i + r_3 A_i$, where r_1, r_2, r_3 are random values inside [0, 1]. The components of Eq. (1) can be attained as:

$$S_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})\hat{d_{ij}}, d_{ij} = \left|x_j - x_i\right|, \hat{d_{ij}} = (x_j - x_i)/d_{ij} \tag{2}$$

$$G_i = -g\hat{e_g} \tag{3}$$

$$A_i = -u\hat{e_w} \tag{4}$$

where d_{ij} is the distance of two grasshoppers, $\hat{d}_{ij}$ is a unit vector, g denotes the gravitational constant, $\hat{e}_g$ is the unity vector of gravity, u denotes a constant drift, and $\hat{e}_w$ is the unity vector of wind. The s function in Eq. (2) calculates either social attraction or repulsion forces and can be attained by:

$$s(r) = fe^{-r/l} - e^{-r} \tag{5}$$

where f represents the amplitude of attraction and l shows the length scale. The comfort zone is a circumstance that the trend of $s(r)$ function is neither attractive nor repulsive. Hence, f and r parameters not only can significantly affect the comfort area, but also attraction and repulsion regions. The social interaction of grasshoppers are affected by the behavior of s-function. This fact is demonstrated in Fig. 1. The grasshoppers in GOA can disclose various social activities based on the parameters l and f in s-function as demonstrated in Fig. 2. According to Saremi et al. (2017), the values of f and l parameters can be set to 0.5, and 1.5, respectively. In addition, the distances of insects are mapped in the interval of [1, 4]. Figure 3 illustrates the interactions among grasshoppers with respect to comfort zone.

		Actual class	
		Positive	Negative
Predicted class	Positive	True Positive TP	False Positive FP
	Negative	False Negative FN	True Negative TN

Fig. 7 Simple confusion matrix

According to Eqs. (2)–(4), the main rule in Eq. (1) can be reformulated as:

$$X_i = \sum_{j=1, j\neq i}^{N} s\left(\left|x_j - x_i\right|\right) \frac{(x_j - x_i)}{d_{ij}} - g\hat{e_g} - u\hat{e_w} \tag{6}$$

where N is the size of swarm. According to Eq. (6), the population cannot be converged to the specific target, because the insects will rapidly reach to the comfort region (Saremi et al. 2017). Therefore, a modified rule can be utilized as:

$$X_i^d = c\left(\sum_{j=1, j\neq i}^{N} c\frac{\mathrm{UB}_d - \mathrm{LB}_d}{2} s\left(\left|x_j^d - x_i^d\right|\right) \frac{(x_j^d - x_i^d)}{d_{ij}}\right) + \hat{T_d} \tag{7}$$

where d shows dimension, UB_d and LB_d are the upper and lower boundaries, T_d is the best solution (target) attained so far, and c is a decreasing factor. Note that the internal c assists GOA for decreasing of repulsion/attraction powers among grasshoppers, while the external c decreases the search tendency nearby the best location by more iteration. The c factor is attained as follows:

$$c = c_{\max} - l\frac{c_{\max} - c_{\min}}{L} \tag{8}$$

where $c_{\max}$ and $c_{\min}$ are, respectively, the maximum and minimum values, l is the in progress iteration, and L denotes the upper bound of iterations. The values of $c_{\max}$ and $c_{\min}$ are set to 1 and 10^{-5}, respectively.

In GOA, the new position of a grasshopper is attained according to its current location, the location of the specific target, and the situation of all population. In GOA, the best-found solution should be considered as the target to be discovered and enriched by the grasshoppers. The decreasing c factor assists GOA to gradually reduce the comfort zone. Hence, it can perform a smooth transition from exploration to exploitation of the fitness landscape. The repulsion forces can assist population for broad exploration of the fitness topogra-

Table 2 Evaluation results of training MLP networks for breast cancer dataset

Algorithms	Metric	AUC	Accuracy	Specificity	Sensitivity	*G*-Mean
GOA	AVG	0.99536	0.97115	0.95309	0.98047	0.96665
	STD	0.00105	0.00514	0.01352	0.00553	0.00675
	BEST	0.99693	0.97899	0.97531	0.98726	0.97810
	WORST	0.99229	0.95798	0.92593	0.96178	0.94991
GA	AVG	0.99549	0.96751	0.94362	0.97983	0.96151
	STD	0.00072	0.00583	0.01796	0.00294	0.00884
	BEST	0.99709	0.97479	0.97531	0.98726	0.97492
	WORST	0.99418	0.95378	0.90123	0.97452	0.94022
PSO	AVG	0.99558	0.97045	0.94774	0.98217	0.96471
	STD	0.00100	0.00752	0.02331	0.00565	0.01126
	BEST	0.99756	0.97899	0.98765	0.99363	0.98107
	WORST	0.99402	0.95378	0.90123	0.96815	0.94022
ABC	AVG	0.98544	0.96891	0.96214	0.97240	0.96713
	STD	0.01499	0.00793	0.02592	0.00890	0.01161
	BEST	0.99772	0.98319	1.00000	0.98726	0.98718
	WORST	0.92593	0.94958	0.91358	0.94904	0.94047
FPA	AVG	0.99511	0.97241	0.95432	0.98174	0.96789
	STD	0.00090	0.00621	0.01838	0.00277	0.00922
	BEST	0.99670	0.98319	0.98765	0.98726	0.98427
	WORST	0.99292	0.95378	0.90123	0.97452	0.94022
BAT	AVG	0.97079	0.96218	0.93457	0.97643	0.95494
	STD	0.02875	0.01422	0.04541	0.00805	0.02250
	BEST	0.99568	0.98319	0.98765	0.99363	0.98127
	WORST	0.89188	0.92437	0.79012	0.96178	0.88605
FF	AVG	0.99619	0.97311	0.98089	0.95802	0.96938
	STD	0.00024	0.00324	0.00000	0.00951	0.00481
	BEST	0.99678	0.97899	0.98089	0.97531	0.97810
	WORST	0.99575	0.96639	0.98089	0.93827	0.95935
MBO	AVG	0.99453	0.96695	0.94074	0.98047	0.96031
	STD	0.00245	0.00770	0.02302	0.00708	0.01115
	BEST	0.99693	0.97899	0.98765	0.99363	0.98107
	WORST	0.98616	0.94958	0.87654	0.95541	0.93026
BBO	AVG	0.99550	0.97255	0.95514	0.98153	0.96822
	STD	0.00082	0.00478	0.01316	0.00256	0.00678
	BEST	0.99670	0.98319	0.97531	0.98726	0.98127
	WORST	0.99355	0.96639	0.93827	0.97452	0.95935

phy, whereas attraction forces can stimulate grasshoppers for exploitation of the promising vicinities of the better quality solutions.

The following remarks can also assist the readers to recognize why the novel GOA can realize very promising solutions for various optimization cases:

- Grasshoppers can perform several abrupt, large-step jumps in the early phases of the search, which enable them to globally search the unexplored areas.
- Grasshoppers tend to search locally in the last phases of the optimization, which enriches their exploitation capacities.
- The decreasing comfort zone factor obliges grasshoppers to progressively make a fine balance between exploration and exploitation tendencies, which assists GOA to avoid immature convergence and discover a promising global peak.
- The GOA can improve the average fitness of all grasshoppers, which assists GOA to effectually enhance the initial randomly generated solutions.
- The fitness of target location can be enhanced over the progression of search, which reveals that the approximation of the global best can be improved after more iteration.

3 Perceptron neural networks

Feedforward neural networks (FFNNs) are the popular form of ANN models that can perceive and approximate computational models using their advanced parallel layered structure (Ojha et al. 2017). They are composed of a set of neurons, which act as processing elements distributed over a series of the fully connected stacked layers. MLP is a special class of FFNN. In MLP, neurons should be organized in a one-directional mode. Data transition in MLP occurs between three classes of parallel layers: input, hidden, and output layers. Figure 4 demonstrates an MLP network with a single hidden layer. The connections among the layers should be characterized by some weights that are located inside [− 1, 1]. Each node in the MLP can perform two functions: summation and activation. The product of inputs, weights, and bias are summed using the summation function in Eq. (9).

$$S_j = \sum_{i=1}^{n} \omega_{ij} I_i + \beta_j \tag{9}$$

where n shows the number of inputs, I_i denotes the input variable i, β_j is a bias term, and w_{ij} shows the connection weight.

Second, an activation function should be instigated using the output of Eq. (9). Several forms of activation functions can be employed in the MLP. The most applied type in past works is S-shaped curved sigmoid function (Aljarah et al. 2016). This function is described in Eq. (10).

$$f_j(x) = \frac{1}{1 + e^{-S_j}} \tag{10}$$

Consequently, the concluding output of the neuron$j$ can be obtained by Eq. (11):

$$y_i = f_j\left(\sum_{i=1}^{n} \omega_{ij} I_i + \beta_j\right) \tag{11}$$

Once the structure of ANN is designed, the learning step is performed to fine-tune and update the weights of network. These weights are rationalized to estimate the outcomes and minimize the error of the results. Learning (training) procedure of the NN is a challenging task that can represent the capability of the MLP for tackling various classes of problems.

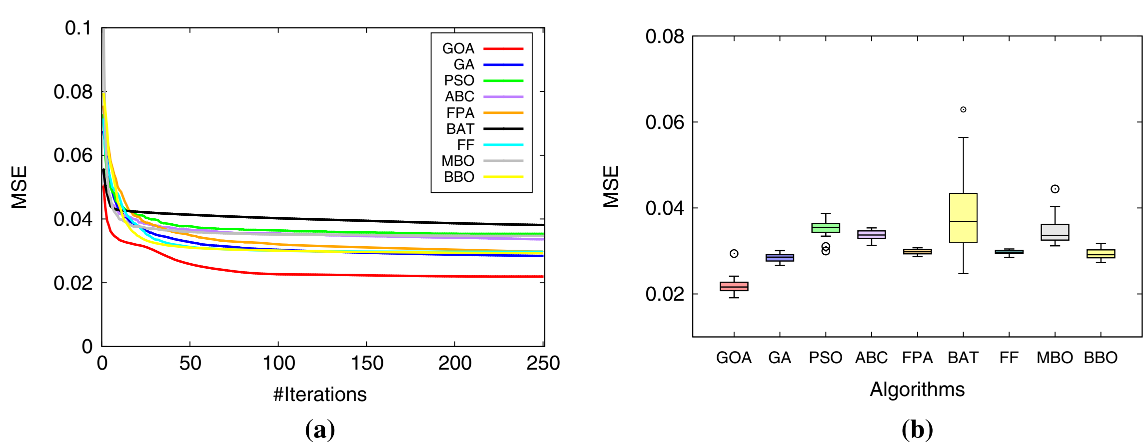

Fig. 8 Convergence curve and boxplot chart for breast cancer dataset. **a** Convergence curve. **b** Boxplot

4 GOA for training MLPs

In this section, the proposed GOA-based MLP trainer (GOAMLP) algorithm is described in detail. As mentioned before, GOA is utilized to train the SHL-MLP network. For this purpose, two crucial points have to be addressed: exactly how the grasshoppers can be encoded in the GOA optimizer to train the MLP and how the fitness function can be presented. In GOAMLP, all search agents are encoded as one-dimensional vectors of randomly created real values inside [1, 1]. Each solution represents a candidate NN. Figure 5 demonstrates an example of encoding strategy of grasshoppers in GOAMLP. As it is observed in Fig. 3, the designed encoding vector covers three key parts including two sets of connection weights among the layers and a series of bias terms. The length of these vectors can be attained with regard to the total number of weights and biases in the target network. A similar encoding strategy is utilized for GOAMLP.

Another point that should be considered here is the selection of fitness function. In order to attain the fitness of grasshoppers, they should be sent to the MLP network as the connection weights. The MLP network can evaluate those

Table 3 Evaluation results of training MLP networks for Parkinsons dataset

Algorithms	Metric	AUC	Accuracy	Specificity	Sensitivity	*G*-Mean
GOA	AVG	0.91891	0.90746	0.72500	0.96471	0.83084
	STD	0.08401	0.03601	0.15536	0.01885	0.09294
	BEST	0.99387	0.97015	1.00000	1.00000	0.98020
	WORST	0.68873	0.82090	0.43750	0.92157	0.64169
GA	AVG	0.92823	0.88109	0.60000	0.96928	0.75794
	STD	0.02968	0.02055	0.12128	0.03364	0.06631
	BEST	0.96201	0.92537	0.87500	1.00000	0.87867
	WORST	0.84069	0.83582	0.43750	0.88235	0.66144
PSO	AVG	0.85118	0.84478	0.46042	0.96536	0.66365
	STD	0.03778	0.03026	0.08605	0.03203	0.06245
	BEST	0.90319	0.89552	0.62500	1.00000	0.75000
	WORST	0.70833	0.76119	0.31250	0.88235	0.53665
ABC	AVG	0.80098	0.83831	0.43750	0.96405	0.64645
	STD	0.09376	0.01842	0.08041	0.02962	0.05034
	BEST	0.92525	0.86567	0.68750	1.00000	0.77015
	WORST	0.56985	0.79104	0.31250	0.86275	0.55351
FPA	AVG	0.89011	0.86070	0.50417	0.97255	0.69764
	STD	0.04068	0.02458	0.08675	0.02280	0.05833
	BEST	0.95343	0.91045	0.68750	1.00000	0.81274
	WORST	0.80025	0.80597	0.37500	0.92157	0.60634
BAT	AVG	0.84641	0.86269	0.55625	0.95882	0.72112
	STD	0.11313	0.04156	0.16926	0.03865	0.10801
	BEST	0.98284	0.95522	0.93750	1.00000	0.90749
	WORST	0.62500	0.76119	0.25000	0.82353	0.50000
FF	AVG	0.88039	0.87214	0.98693	0.50625	0.70564
	STD	0.01810	0.02025	0.01296	0.06218	0.04496
	BEST	0.91176	0.91045	1.00000	0.62500	0.79057
	WORST	0.83701	0.83582	0.96078	0.43750	0.64834
MBO	AVG	0.85633	0.85075	0.47083	0.96993	0.67160
	STD	0.04241	0.03208	0.10346	0.02168	0.07894
	BEST	0.91544	0.89552	0.62500	1.00000	0.77491
	WORST	0.73775	0.76119	0.25000	0.90196	0.49507
BBO	AVG	0.93656	0.88159	0.55208	0.98497	0.73480
	STD	0.02311	0.02446	0.09297	0.01517	0.06249
	BEST	0.98039	0.92537	0.75000	1.00000	0.85749
	WORST	0.89951	0.82090	0.37500	0.94118	0.61237

vectors according to a training dataset. Finally, the network will obtain the fitness values of corresponding solutions. In this article, the mean-squared error (MSE) is employed as the fitness function in the GOAMLP trainer for assessing the fineness of the MLPs. For the training samples, the MSE metric can be obtained using the variance of the actual and predicted solutions by the generated grasshoppers (MLPs). The objective is to minimize the value of the MSE as much as possible.

MSE can be attained by Eq. (12):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{12}$$

where y indicates the actual value, $\hat{y}$ shows the predicted one, and n denotes the total number of instances.

The algorithm of GOAMLP trainer can be described by the following steps:

1. *Initialization* the GOAMLP starts by creating a random set of grasshoppers.
2. *Mapping of grasshoppers* the elements of the grasshoppers are assigned to the weights and biases of a potential MLP network.
3. *Fitness evaluation* the quality of the resulted MLPs is evaluated using the MSE function for all samples in the training dataset.
4. The GOAMLP should find the MLP with the lowest MSE value. Those MLPs with lower MSEs are more preferable than those with higher MSEs.
5. *Update the positions* the grasshoppers should be updated.
6. Repeat steps 2–4 until the latest cycle.
7. *Termination and testing* finally, the process is terminated and the MLP with the minimum MSE should be tested on the test/validation instances.

The overall steps of the GOAMLP technique are demonstrated in Fig. 6. Note that all algorithms are similar in the MLP part and the computational complexity of GOA part is of $O(t \times S \times N^2)$, where t is the current iteration, S is dimension, and N represents total number of grasshoppers.

5 Results and discussions

In this section, we describe the conducted experiments in this work to evaluate the performance of the developed GOAMLP model for medical classification models and compare its performance to other well-regarded and recent models. The medical classification problems, evaluation measures, the experiments setup, and the results are described and discussed in the following subsections.

5.1 Medical problems and datasets

In this work, the proposed and developed GOAMLP model is experimented based on five medical datasets. The datasets concern five problems in medicine where each of which focuses on identification of a specific disease. Table 1 summarizes the datasets in terms of number of attributes, number of training samples, and number of testing samples.

The utilized medical datasets are described in more details as follows:

– *Breast cancer* This dataset was originally obtained from Dr. William H. Wolberg, the University of Wisconsin Hospitals, Madison. This dataset contains 699 instances where each instance represents a patient that had undergone surgery for breast cancer. Four variables are measured for each patient and labeled as benign or malignant (Wolberg and Mangasarian 1990; Mangasarian et al. 1990).

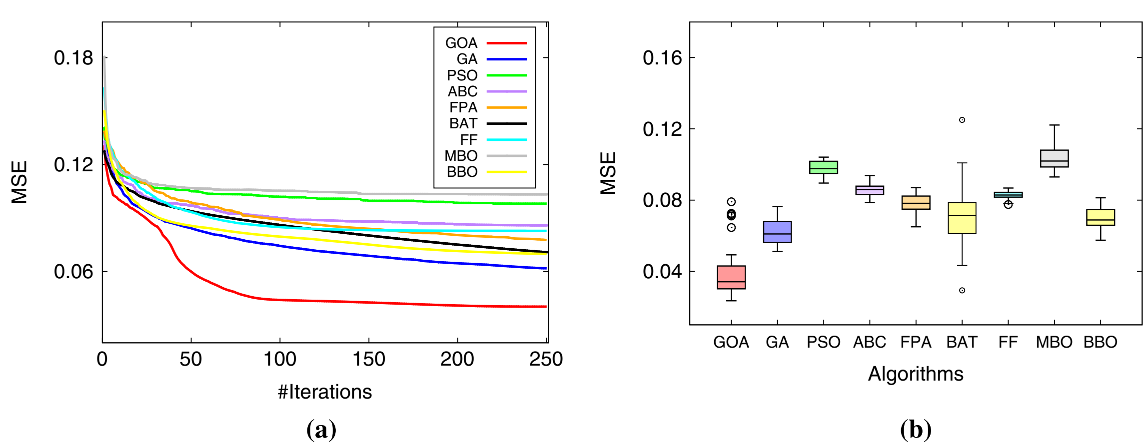

Fig. 9 Convergence curve and boxplot chart for Parkinsons dataset. **a** Convergence curve. **b** Boxplot

- *Parkinson* This dataset was collected by the University of Oxford, and the National Center for Voice and Speech, Denver, Colorado. The dataset contains information about the people with Parkinson disease. The dataset consists of 23 of biomedical voice measurements for 31 persons (almost 6 recordings for each person), 23 persons with Parkinsons disease, and 8 healthy persons. Total number of voice recordings is 195 (Little et al. 2007).
- *Diabetes* This dataset is part of larger dataset donated by the National Institute of Diabetes and Digestive and Kidney Diseases. Instances of the dataset represent patients who are are Pima-Indian women at least 21 years old and living near Phoenix, Arizona, USA. The class label of the dataset is binary, '1' for a positive test for diabetes and '0' a negative test for diabetes. There are 268 (34.9%) cases identified as positive tests and 500 (65.1%) cases as negative tests. The measured variables for each case are: (1) number of times pregnant; (2) plasma glucose concentration a 2 h in an oral glucose tolerance test; (3) diastolic blood pressure (mm Hg); (4) triceps skinfold thickness (mm); (5) 2-hour serum insulin (mu U/ml); (6)

Table 4 Evaluation results of training MLP networks for diabetes dataset

Algorithms	Metric	AUC	Accuracy	Specificity	Sensitivity	*G*-Mean
GOA	AVG	0.85528	0.76489	0.88373	0.55938	0.70303
	STD	0.00602	0.00836	0.00674	0.01645	0.01138
	BEST	0.86433	0.77863	0.89157	0.59375	0.72511
	WORST	0.84682	0.75191	0.87349	0.54167	0.68785
GA	AVG	0.83888	0.75165	0.85723	0.56910	0.69804
	STD	0.00782	0.01207	0.01653	0.03378	0.01890
	BEST	0.85950	0.78244	0.89759	0.63542	0.73016
	WORST	0.82674	0.72137	0.83133	0.47917	0.65582
PSO	AVG	0.81890	0.73104	0.86265	0.50347	0.65648
	STD	0.01773	0.02475	0.04603	0.07088	0.03879
	BEST	0.84419	0.78244	0.95181	0.62500	0.73887
	WORST	0.77987	0.67176	0.79518	0.37500	0.57216
ABC	AVG	0.82251	0.74822	0.84799	0.57569	0.69560
	STD	0.01646	0.01983	0.04606	0.08174	0.03950
	BEST	0.86509	0.80534	0.94578	0.70833	0.78115
	WORST	0.78181	0.70992	0.72892	0.37500	0.59554
FPA	AVG	0.83299	0.75242	0.86546	0.55694	0.69369
	STD	0.00894	0.01486	0.02183	0.03847	0.02173
	BEST	0.84752	0.78244	0.92169	0.62500	0.73376
	WORST	0.81859	0.72137	0.82530	0.48958	0.65620
BAT	AVG	0.84349	0.76374	0.87289	0.57500	0.70779
	STD	0.01629	0.01405	0.01792	0.04329	0.02411
	BEST	0.86383	0.79389	0.90361	0.64583	0.75367
	WORST	0.79744	0.73282	0.81325	0.48958	0.66068
FF	AVG	0.84664	0.76349	0.57153	0.87450	0.70695
	STD	0.00328	0.00465	0.00595	0.00594	0.00471
	BEST	0.85398	0.77481	0.58333	0.88554	0.71873
	WORST	0.83967	0.75191	0.56250	0.86145	0.69611
MBO	AVG	0.82938	0.74733	0.85763	0.55660	0.68939
	STD	0.01912	0.02026	0.03450	0.05903	0.03197
	BEST	0.86195	0.78626	0.95783	0.64583	0.73802
	WORST	0.77548	0.70611	0.78916	0.42708	0.60867
BBO	AVG	0.84211	0.75611	0.86325	0.57083	0.70183
	STD	0.00744	0.01062	0.01638	0.01823	0.01136
	BEST	0.85442	0.77099	0.89157	0.60417	0.72143
	WORST	0.82649	0.73282	0.83133	0.53125	0.67885

body mass index; (7) diabetes pedigree function; and (8) age (years) (Shanker 1996).

– *Saheart* This dataset represents 462 samples of males in a heart disease in high-risk region of the Western Cape, South Africa. There are roughly two controls per case of coronary heart disease. Many of the coronary heart disease-positive men have undergone blood pressure reduction treatment and other programs to reduce their risk factors after their coronary heart disease event. In some cases, the measurements were taken after these treatments. The class label indicates if the person has a coronary heart disease or not. The dataset contains information about 9 features, namely systolic blood pressure, cumulative tobacco, low-density lipoprotein cholesterol, adiposity, family history of heart disease, type A behavior, obesity, current alcohol consumption, and patient's age.
– *Vertebral* This dataset was built by Dr. Henrique da Mota. The dataset contains values for six biomechanical features used to classify orthopedic patients into two classes (normal or abnormal). There are 100 normal patients and 210 abnormal patients. Abnormal patients are disk hernia patients or spondylolisthesis patients. The six features of the datasets are derived from the shape and orientation of the pelvis and lumbar spine (in this order): pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius, and grade of spondylolisthesis. The following convention is used for the class labels: disk hernia (DH), spondylolisthesis (SL), normal (NO), and abnormal (AB).

5.2 Evaluation measures

In order to evaluate the performance of the developed classification models in these experiments, we used five different measures commonly popular in the medical field. The first four measurements are derived from the confusion matrix which is one of the basic sources for evaluating binary classifications models. An illustration of the confusion matrix is shown in Fig. 7. The evaluation measures that used in this work are discussed in following paragraphs.

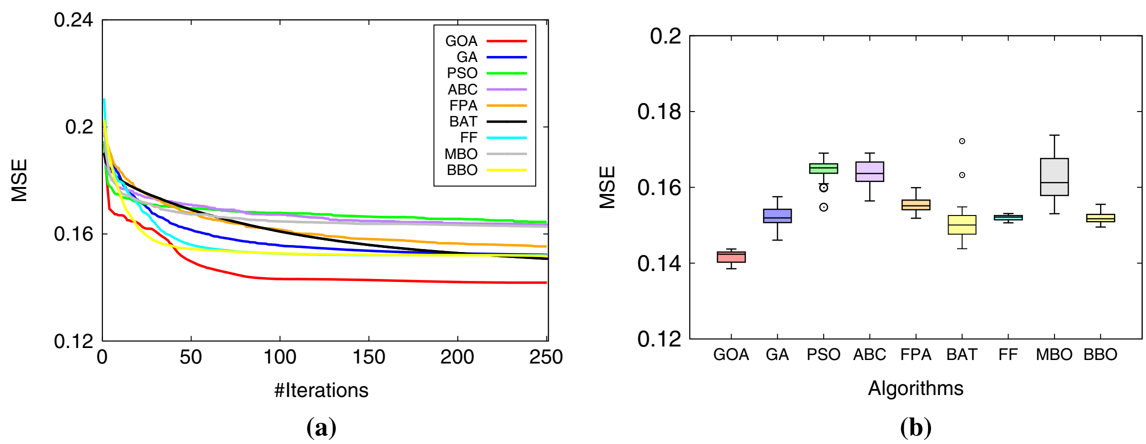

Fig. 10 Convergence curve and boxplot chart for diabetes dataset. **a** Convergence curve. **b** Boxplot

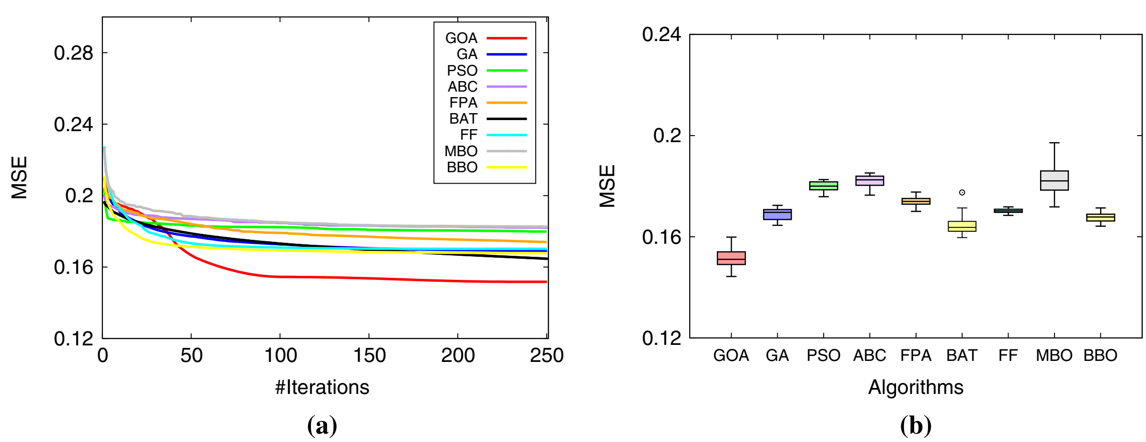

Fig. 11 Convergence curve and boxplot chart for Saheart dataset. **a** Convergence curve. **b** Boxplot

– *Accuracy or classification rate* which measures the percentage of the correctly classified samples (positive or negative) to the actual total number of samples. The accuracy is calculated according to the following equation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \quad (13)$$

– *Specificity* which measures the percentage of the correctly predicted negative samples to the actual total number of negative samples. This measure is very useful in the medical field, because the specificity can help to detect correctly the patients who do not carry disease. Specificity is calculated by using the following equation:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (14)$$

– *Sensitivity* which measures the percentage of the correctly predicted positive samples to the actual total number of the positive samples. High sensitivity values mean that the classification model can correctly predict the patients who carry disease. Sensitivity is calculated using the following equation:

Table 5 Evaluation results of training MLP networks for Saheart dataset

Algorithms	Metric	AUC	Accuracy	Specificity	Sensitivity	*G*-Mean
GOA	AVG	0.75555	0.73122	0.49383	0.85449	0.64913
	STD	0.01379	0.02378	0.03809	0.02645	0.02879
	BEST	0.78793	0.79114	0.57407	0.91346	0.71238
	WORST	0.72685	0.67722	0.42593	0.79808	0.58653
GA	AVG	0.76671	0.71814	0.82372	0.51481	0.65030
	STD	0.01242	0.01757	0.03645	0.03587	0.01702
	BEST	0.78241	0.75949	0.94231	0.55556	0.67792
	WORST	0.73326	0.68354	0.75962	0.40741	0.61048
PSO	AVG	0.76022	0.72658	0.85096	0.48704	0.64126
	STD	0.02382	0.02768	0.05845	0.06018	0.02874
	BEST	0.79897	0.77848	0.95192	0.61111	0.68990
	WORST	0.71546	0.64557	0.70192	0.38889	0.57689
ABC	AVG	0.74454	0.71160	0.82276	0.49753	0.63644
	STD	0.03388	0.02120	0.05373	0.07355	0.03546
	BEST	0.81250	0.76582	0.95192	0.61111	0.71909
	WORST	0.66560	0.67722	0.72115	0.29630	0.53109
FPA	AVG	0.76480	0.72869	0.84231	0.50988	0.65336
	STD	0.02171	0.02307	0.04802	0.05728	0.02813
	BEST	0.80520	0.76582	0.93269	0.64815	0.71050
	WORST	0.72489	0.68354	0.77885	0.38889	0.59377
BAT	AVG	0.76642	0.72405	0.83654	0.50741	0.65086
	STD	0.01025	0.01840	0.03292	0.03248	0.01780
	BEST	0.78846	0.75949	0.89423	0.59259	0.67779
	WORST	0.74252	0.68987	0.76923	0.42593	0.61715
FF	AVG	0.77276	0.71730	0.51667	0.82147	0.65137
	STD	0.00402	0.01259	0.01709	0.01650	0.01288
	BEST	0.77902	0.74051	0.55556	0.84615	0.68172
	WORST	0.76086	0.69620	0.48148	0.79808	0.62361
MBO	AVG	0.76113	0.72932	0.83782	0.52037	0.65881
	STD	0.02234	0.02126	0.04101	0.05246	0.02611
	BEST	0.79790	0.76582	0.89423	0.62963	0.70408
	WORST	0.71599	0.68354	0.75000	0.40741	0.59706
BBO	AVG	0.77369	0.72911	0.83910	0.51728	0.65776
	STD	0.00905	0.01395	0.02976	0.04508	0.02235
	BEST	0.78775	0.75316	0.91346	0.61111	0.69696
	WORST	0.75036	0.69620	0.78846	0.40741	0.60359

$$\text{Sensitivity} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{15}$$

– *G-mean* which is a geometric mean that combines between the sensitivity and specificity measures. G-mean is calculated using the following equation:

$$G\text{-mean} = \sqrt{\text{Specificity} \times \text{Sensitivity}} \tag{16}$$

– *Area under the ROC curve (AUC)* which measures the area under the curve of receiver operating characteristics (ROC) graph. AUC is very useful measure specially when the total number of positive and negative samples have unequal distribution in the dataset. AUC is shown in the following equation:

$$\text{AUC} = \int_0^1 \frac{\text{TP}}{P} \, \text{d} \, \frac{\text{FP}}{N} = \frac{1}{P.N} \int_0^1 \text{TP d FP} \tag{17}$$

where $P = \text{TP} + \text{FP}$ and $N = \text{TN} + \text{FN}$ (Faris et al. 2016a).

Table 6 Evaluation results of training MLP networks for Vertebral dataset

Algorithms	Metric	AUC	Accuracy	Specificity	Sensitivity	*G*-Mean
GOA	AVG	0.94060	0.86321	0.88444	0.81183	0.84723
	STD	0.00501	0.00885	0.00948	0.02553	0.01307
	BEST	0.95140	0.88679	0.90667	0.87097	0.87547
	WORST	0.93204	0.84906	0.86667	0.77419	0.82540
GA	AVG	0.93508	0.85786	0.82151	0.87289	0.84664
	STD	0.00465	0.01049	0.02198	0.01706	0.01026
	BEST	0.94495	0.88679	0.83871	0.92000	0.87203
	WORST	0.92602	0.83962	0.77419	0.84000	0.82540
PSO	AVG	0.91911	0.85031	0.78925	0.87556	0.82942
	STD	0.01703	0.03349	0.07023	0.05628	0.03254
	BEST	0.94108	0.90566	0.90323	0.97333	0.87203
	WORST	0.87570	0.77358	0.61290	0.76000	0.73441
ABC	AVG	0.92189	0.84371	0.81828	0.85422	0.83483
	STD	0.01995	0.03265	0.05522	0.05216	0.02786
	BEST	0.95441	0.90566	0.93548	0.94667	0.87375
	WORST	0.87484	0.77358	0.67742	0.74667	0.76031
FPA	AVG	0.92433	0.86730	0.77742	0.90444	0.83816
	STD	0.00836	0.01506	0.03425	0.02351	0.01676
	BEST	0.94065	0.88679	0.83871	0.96000	0.86559
	WORST	0.90538	0.83962	0.70968	0.86667	0.80215
BAT	AVG	0.91209	0.83553	0.81075	0.84578	0.82617
	STD	0.04299	0.05471	0.05272	0.08269	0.04071
	BEST	0.94796	0.89623	0.90323	0.93333	0.87841
	WORST	0.77247	0.68868	0.67742	0.62667	0.68533
FF	AVG	0.94072	0.86384	0.87911	0.82688	0.85250
	STD	0.00434	0.00810	0.01259	0.01794	0.00860
	BEST	0.94968	0.87736	0.90667	0.83871	0.86559
	WORST	0.92946	0.84906	0.85333	0.77419	0.82540
MBO	AVG	0.92346	0.85660	0.78387	0.88667	0.83203
	STD	0.01766	0.02796	0.06460	0.05083	0.02750
	BEST	0.94882	0.90566	0.87097	0.98667	0.89515
	WORST	0.87957	0.76415	0.64516	0.73333	0.77598
BBO	AVG	0.93875	0.86730	0.82473	0.88489	0.85403
	STD	0.00660	0.01106	0.02896	0.01836	0.01285
	BEST	0.95097	0.88679	0.87097	0.93333	0.87547
	WORST	0.92645	0.84906	0.74194	0.84000	0.83163

5.3 Experiments setup

All the fair experiments for compared optimizers were conducted using MATLAB 7.10 (R2010a) on a 64-bit Windows server 2012 operating system with two Intel Xeon processor at 1.70 Ghz and a RAM of 64 GB. No commercial GOA-based software was utilized in this study.

For training and testing of the developed model, all datasets are partitioned into two parts: 66.66% of the dataset is utilized for training, and 33.33% of the data is employed for testing. Moreover, stratified sampling is applied in this partitioning to preserve the original class distribution of the data in the training and testing processes. In order to get statistically meaningful results, the process of training and testing is repeated 30 independent times; then, the average measurements are calculated. This number of independent runs was performed in many previous studies in the literature (Islam et al. 2003; Chan et al. 2011; Yao and Liu 1999; Faris et al. 2018b). Note also that all features are normalized into the interval [0, 1] using the max–min normalization method as given in Eq. (18). $\min_A$ and $\max_A$ are the minimum and maximum values of a given feature A. This process is very important before applying neural networks. The reason is to ensure that all features have equal weights and to avoid the influence of features that have values that fall in a wider range than other features.

$$A_i' = \frac{A_i - \min_A}{\max_A - \min_A} \tag{18}$$

To verify the results of the proposed GOA-based trainer, it is compared with a set of well-regarded and recent meta-heuristic algorithms. The well-established algorithms are the GA, PSO, biogeography-based optimization (BBO) (Mirjalili et al. 2014; Aljarah et al. 2018b), bat algorithm (BAT/BA) (Yang and Gandomi 2012), the firefly algorithm (FF/FA) (Yang 2010), and artificial bee colony (ABC) (Karaboga et al. 2007) algorithms. The recent algorithms are the monarch butterfly optimization (MBO) (Wang et al. 2015) and the flower pollination algorithm (FPA) (Yang et al. 2014). In all experiments, the population size and number of iterations of optimizers are set to 50 and 250, respectively. The rest of parameters are set as recommended and utilized in the specialized literature (Yang et al. 2011; Goldberg 1989; Simon 2008; Karaboga and Basturk 2007; Mirjalili and Lewis 2016; Mirjalili 2016; Wang et al. 2015; Saremi et al. 2017). Regarding the determination of the number of hidden neurons in the MLP network, there are many rules proposed in the literature and there is no agreement between researchers on which rule is the best to apply. However, in this work, a common rule adopted by many previous works like those in Wdaa et al. (2008), Mirjalili et al. (2014) is selected. The rule set the number of hidden neurons as 2 $\times$ number of features + 1.

5.4 Experimental results and discussion

5.4.1 Breast cancer problem results

Evaluation results of the proposed GOAMLP and other MHA-based MLP models are shown in Table 2. The average, standard deviation, best, and worst of each evaluation measure are reported in the table and denoted as AVG, STD, BEST, and WORST, respectively. According to the attained results, it can be noticed that all algorithms can achieve to the high ratios in all measurements. In addition, it can be observed that the results of the GOA, FPA, FF, and BBO are slightly better than those of other algorithms in the majority of the measurements. In Fig. 8, the convergence trends of all evaluated algorithms and a boxplot of the results in terms of MSE are demonstrated in detail. Inspecting the convergence curves, it can be clearly noticed that the GOA is capable of outperforming other algorithms in terms of the convergence speed. Checking the boxplot, it can be seen that the GOA-

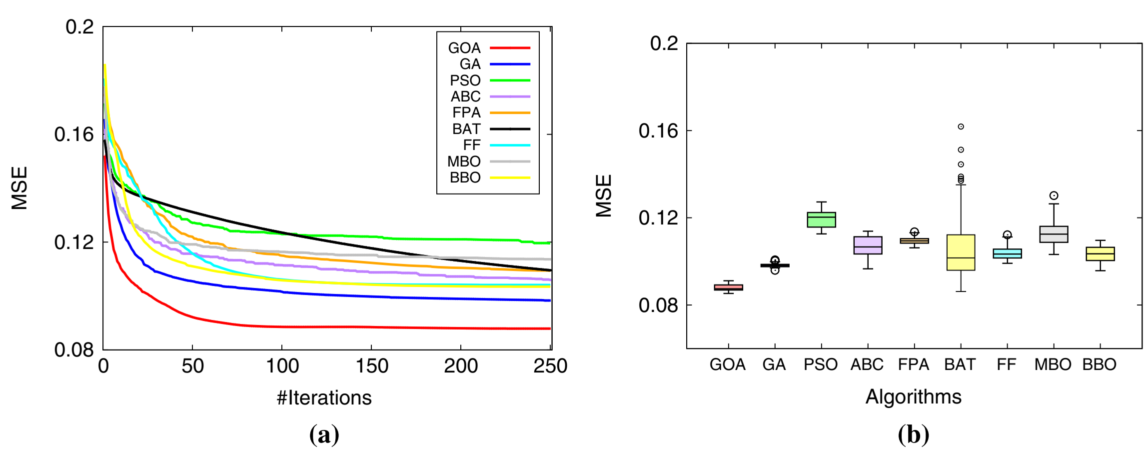

Fig. 12 Convergence curve and boxplot chart for Vertebral dataset. **a** Convergence curve. **b** Boxplot

Table 7 The attained p values of the Wilcoxon statistical test for GOAMLP and other algorithms on all benchmark datasets (there is no $p \geq 0.05$ to be underlined)

Benchmark	GA	PSO	ABC	FPA	BAT	FF	MBO	BBO
Breast	3.82E−10	3.02E−11	3.02E−11	8.15E−11	4.95E−11	5.49E−11	3.02E−11	3.02E−11
Diabetes	2.89E−11	2.90E−11	2.90E−11	2.90E−11	2.90E−11	2.90E−11	2.90E−11	2.90E−11
Parkinsons	2.15E−6	3.02E−11	3.34E−11	8.10E−10	3.26E−7	4.08E−11	3.02E−11	3.02E−11
Saheart	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.34E−11	3.02E−11	3.02E−11	3.02E−11
Vertebral	3.02E−11	3.02E−11	3.02E−11	3.02E−11	4.18E−9	3.02E−11	3.02E−11	3.02E−11

based trainer has the smallest MSE average and it has one of the most compact boxes, which indicates the stability of the proposed training algorithm.

5.4.2 Parkinsons problem results

The results of identifying persons with Parkinsons disease using the MHA-based MLP networks are shown in Table 3. The results show that GOA was capable of achieving the highest ratios in two measurements: classification accuracy and G-mean with values of 90.746 and 0.83084%. The advantage of G-mean is that it indicates a good balance between the specificity and sensitivity. For example, the BBO optimizer has the best sensitivity of 98.497%; however, it performed poorly in terms of specificity, which is only 55.208%. The convergence curves and the boxplot shown in Fig. 9 confirm that the GOA has the fastest convergence trend, the lowest error, and acceptable stability.

5.4.3 Diabetes problem results

Evaluation results for the diabetes problem are shown in Table 4. The GOA shows the highest ratios in AUC, accuracy, and specificity. In addition, it is very competitive in terms of G-mean measurement. Although BAT algorithm shows competitive results compared to GOA, it has high variability in terms of MSE (as shown in Fig. 10) and consequently indicates low stability. In addition, MBOMLP and ABCMLP cannot maintain a good balance among exploration and exploitation mechanisms and show unstable performances. In contrast, GOA has a more compact box which proves its stability. Inspecting the convergence curves in Fig. 10, we can see the fast convergence rate of GOA compared to the other algorithms, while methods such as PSO and ABS have stagnated to LO. The reason is that the GOA-based trainer improves the quality of all search agents (MLPs) during the search and it is capable of performing a smoother transition from broad exploration to extensive exploration over the course of iterations.

5.4.4 Saheart

In identifying the coronary heart disease cases, this dataset is a very challenging one, and all MHA-based models including the proposed GOAMLP have demonstrated lower detection rates compared to the previously discussed problems as shown in Table 5. However, the GOA-based technique can attain better results in competition with all the other trainers in terms of accuracy and sensitivity with values of 73.122 and 85.449%. Other competitive algorithms in terms of G-mean are MBO; however, the algorithm has shown high variability in the boxplots in Fig. 11. In terms of convergence speed, Fig. 11 also shows that the GOA-based trainer can demonstrate a faster convergence rate compared to the other algorithms.

5.4.5 Vertebral

For this dataset, the meta-heuristic algorithms are used to optimize neural networks to classify orthopedic patients into two classes: normal or abnormal. The results of the evaluations for this experiment are shown in Table 6. As it can be seen, best results in terms of AUC, accuracy, and G-mean were achieved by GOA, FF, and BBO. Although their evaluation results were very close and competitive, GOA showed faster convergence as can be seen in Fig. 12. Also, the same figure shows that GOA and GA have more compact boxes which indicate their stability, but GA has the lowest error in terms of MSE.

Table 7 shows the obtained p values from the Wilcoxon rank-sum test of MSE results for GOA and other competitors for all benchmark datasets. The statistical results also confirm that the MSE results of the GOA are significantly improved compared to other optimizers for all datasets.

Although previous methods can show some advantages such as competitive convergence rates, the main drawback of all studied optimizers is that they cannot maintain a more stable balance between exploration and exploitation tendencies compared to the GOAMLP. Therefore, they cannot escape from LO and stagnation problem can be observed in the results of BATMLP, FFMLP, FPAMLP, GAMLP, ABCMLP, MBOMLP, PSOMLP, and BBOMLP methods. However, the proposed GOAMLP can demonstrate several unique advantages in finding the optimal structure. For instance and as the first advantage, it is equipped with simple yet effective adaptive mechanisms that assists it in avoiding LO and find accurate results with respect to the number of iterations. In addition, the GOA can divide a problem into several bi-parameter subproblems, which is a mechanism for simplifying the core MLP-based task. This unique mechanism assists GOA in avoiding immature convergence and stagnation to LO.

The comparative results in terms of monitored metrics for different datasets reveal that the GOA has substantially improved the classification accuracy of the MLP. In contrast to other algorithms such as the GA, PSO, ABC, FPA, BAT, FF, MBO, and BBO, it was observed that it can demonstrate high-quality solutions and the performance of GOAMLP does not degrade proportionally to the number of features, training samples, and/or test samples. This is due to the flexibility of this algorithm in handling a large number of location solutions. The GOA breaks a problem into several bi-parameter subproblems, a mechanism for simplifying the main problem. This mechanism allows GOA to avoid immature convergence drawbacks and local optima.

Another cause for improvements in accuracy results is that the search agents in GOA can perform several abrupt, large-step jumps in the early phases of the searching procedure, which enable them for higher levels of exploration of the unexplored areas. In addition, the GOA algorithm has only one adaptive parameter that assists it in well-balancing exploration and exploitation regarding the complexity of the feature space. The original adaptive curve for the parameter was beneficial in training MLPs as well, which resulted in avoiding local optima and finding more accurate optimal values for the structural parameters of MLPs to achieve high classification rate.

6 Conclusions and future directions

This work employed the recently proposed GOA algorithm to find the optimal values for the structural parameters of MLPs. After introducing an encoding mechanism and objective function based on MSE, the GOAMLP was applied to five challenging real case studies: breast cancer, Parkinson, diabetes, coronary heart disease, and orthopedic patients. It was observed that the GOAMLP is capable of efficiently classifying these datasets with a high classification accuracy. A comparative study with eight recent and well-known algorithms was conducted that showed the superiority of the GOA algorithm.

Quantitative and qualitative results were collected and analyzed. It was found that the proposed GOAMLP is beneficial for accurately classifying data sets of different sizes and complexities. The assessment of the results showed that this is mainly due to the simple yet effective adaptive mechanism of GOA which assists this algorithm to avoid local solutions and find accurate results proportional to the number of iterations.

The trainer proposed in this paper can be a reliable alternative to the current algorithms on a set of classification problems as per the NFL theorem. Therefore, we recommend researchers to apply GOAMLP to their classification problems. The exploratory behavior of GOA makes GOAMLP potentially capable of providing promising results on the used datasets. For future works, GOA can be applied to other types of NNs or large-scale datasets. In addition, researchers can investigate the relationship between attribute noise and performance of used MLP-based approaches.

Compliance with ethical standards

Conflict of interest All authors declare that there is no conflict of interest.

Ethical standard This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alba E, Chicano J (2004) Training neural networks with GA hybrid algorithms. In: Genetic and evolutionary computation—GECCO 2004. Springer, pp 852–863

Aljarah I, Faris H, Mirjalili S (2016) Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Comput 22:1–15

Aljarah I, Al-Zoubi AM, Faris H, Hassonah MA, Mirjalili S, Saadeh H (2018a) Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cognit Comput 10:1–18

Aljarah I, Faris H, Mirjalili S, Al-Madi N (2018b) Training radial basis function networks using biogeography-based optimizer. Neural Comput Appl 29(7):529–553

Almonacid F, Fernandez EF, Mellit A, Kalogirou S (2017) Review of techniques based on artificial neural networks for the electrical characterization of concentrator photovoltaic technology. Renew Sustain Energy Rev 75:938–953

Ata R (2015) Artificial neural networks applications in wind energy systems: a review. Renew Sustain Energy Rev 49:534–562

Blum C, Socha K (2005) Training feed-forward neural networks with ant colony optimization: an application to pattern classification. In: Fifth international conference on hybrid intelligent systems, 2005. HIS'05. IEEE, p 6

Braik M, Sheta A, Arieqat A (2008) A comparison between GAs and PSO in training ANN to model the TE chemical process reactor. In: AISB 2008 convention communication, interaction and social intelligence, vol 1, p 24

Chan KY, Ling S-H, Dillon TS, Nguyen HT (2011) Diagnosis of hypoglycemic episodes using a neural network based rule discovery system. Expert Syst Appl 38(8):9799–9808

Chaudhuri BB, Bhattacharya U (2000) Efficient training and improved performance of multilayer perceptron in pattern classification. Neurocomputing 34(1):11–27

Chen J-F, Do QH, Hsieh H-N (2015) Training artificial neural networks by a hybrid PSO-CS algorithm. Algorithms 8(2):292–308
Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst (MCSS) 2(4):303–314
Ding S, Li H, Su C, Yu J, Jin F (2013) Evolutionary artificial neural networks: a review. Artif Intell Rev 39(3):251–260
Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118
Faris H, Aljarah I, Al-Madi N, Mirjalili S (2016a) Optimizing the learning process of feedforward neural networks using lightning search algorithm. Int J Artif Intell Tools 25(06):1650033
Faris H, Aljarah I, Mirjalili S (2016b) Training feedforward neural networks using multi-verse optimizer for binary classification problems. Appl Intell 45(2):322–332
Faris H, Mafarja MM, Heidari AA, Aljarah I, Al-Zoubi AM, Mirjalili S, Fujita H (2018a) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl Based Syst 154:43–67
Faris H, Aljarah I, Mirjalili S (2018b) Improved monarch butterfly optimization for unconstrained global search and neural network training. Appl Intell 48(2):445–464
Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc, Boston
Gupta JND, Sexton RS (1999) Comparing backpropagation with a genetic algorithm for neural network training. Omega 27(6):679–684
Hamidzadeh J, Moradi M (2018) Improved one-class classification using filled function. Appl Intell 1–17
Hamidzadeh J, Namaei N (2018) Belief-based chaotic algorithm for support vector data description. Soft Comput 1–26
Hamidzadeh J, Monsefi R, Yazdi HS (2012) DDC: distance-based decision classifier. Neural Comput Appl 21(7):1697–1707
Hamidzadeh J, Monsefi R, Yazdi HS (2014) LMIRA: large margin instance reduction algorithm. Neurocomputing 145:477–487
Hamidzadeh J, Monsefi R, Yazdi HS (2015) IRAHC: instance reduction algorithm using hyperrectangle clustering. Pattern Recognit 48(5):1878–1889
Hamidzadeh J, Monsefi R, Yazdi HS (2016) Large symmetric margin instance selection algorithm. Int J Mach Learn Cybern 7(1):25–45
Hamidzadeh J, Sadeghi R, Namaei N (2017) Weighted support vector data description based on chaotic bat algorithm. Appl Soft Comput 60:540–551
Hamidzadeh J, Zabihimayvan M, Sadeghi R (2018) Detection of web site visitors based on fuzzy rough sets. Soft Comput 22(7):2175–2188
Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 11(1):1–18
Heidari AA, Abbaspour RA (2018) Enhanced chaotic grey wolf optimizer for real-world optimization problems: a comparative study. In: Handbook of research on emergent applications of optimization algorithms. IGI Global, pp 693–727
Heidari AA, Delavar MR (2016) A modified genetic algorithm for finding fuzzy shortest paths in uncertain networks. ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci XLI–B2:299–304
Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with lévy flight for optimization tasks. Appl Soft Comput 60:115–134
Heidari AA, Abbaspour RA, Jordehi AR (2017) An efficient chaotic water cycle algorithm for optimization tasks. Neural Comput Appl 28(1):57–85
Heidari AA, Abbaspour RA, Jordehi AR (2017) Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems. Appl Soft Comput 57:657–671
Ilonen J, Kamarainen J-K, Lampinen J (2003) Differential evolution training algorithm for feed-forward neural networks. Neural Process Lett 17(1):93–105
Islam MM, Yao X, Murase K (2003) A constructive algorithm for training cooperative neural network ensembles. IEEE Trans Neural Netw 14(4):820–834
Jianbo Y, Wang S, Xi L (2008) Evolving artificial neural networks using an improved PSO and DPSO. Neurocomputing 71(4):1054–1060
Jordehi AR, Jasni J (2013) Parameter selection in particle swarm optimisation: a survey. J Exp Theor Artif Intell 25(4):527–542
Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
Karaboga D, Akay B, Ozturk C (2007) Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In: International conference on modeling decisions for artificial intelligence. Springer, pp 318–329
Krogh A (2008) What are artificial neural networks? Nat Biotechnol 26(2):195–197
Lee S, Choeh JY (2014) Predicting the helpfulness of online reviews using multilayer perceptron neural networks. Expert Syst Appl 41(6):3041–3046
Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM et al (2007) Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. Biomed Eng OnLine 6(1):23
Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Al-Zoubi AM, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl Based Syst 145:25–45
Mallipeddi R, Suganthan PN, Pan Q-K, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl Soft Comput 11(2):1679–1696
Mangasarian OL, Setiono R, Wolberg WH (1990) Pattern recognition via linear programming: theory and application to medical diagnosis. Large Scale Numer Optim 22–31
McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133
Mirjalili S (2015) How effective is the grey wolf optimizer in training multi-layer perceptrons. Appl Intell 43(1):150–161
Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133
Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
Mirjalili S, Mirjalili SM, Lewis A (2014) Let a biogeography-based optimizer train your multi-layer perceptron. Inf Sci 269:188–209
Mirjalili SZ, Saremi S, Mirjalili SM (2015) Designing evolutionary feedforward neural networks using social spider optimization algorithm. Neural Comput Appl 26(8):1919–1928
Moghaddam VH, Hamidzadeh J (2016) New hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier. Pattern Recognit 60:921–935
Ojha VK, Abraham A, Snášel V (2017) Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng Appl Artif Intell 60:97–116
Sadeghi R, Hamidzadeh J (2018) Automatic support vector data description. Soft Comput 22(1):147–158
Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
Seiffert U (2001) Multiple layer perceptron training using genetic algorithms. In: Proceedings of the European symposium on artificial neural networks ESANN, Bruges, Blgica

Sexton RS, Gupta JND (2000) Comparative evaluation of genetic algorithm and backpropagation for training neural networks. Inf Sci 129(1):45–59
Sexton RS, Dorsey RE, Johnson JD (1999) Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing. Eur J Oper Res 114(3):589–601
Shanker MS (1996) Using neural networks to predict the onset of diabetes mellitus. J Chem Inf Comput Sci 36(1):35–41
Siddique MNH, Tokhi MO (2001) Training neural networks: backpropagation vs. genetic algorithms. In: International joint conference on neural networks, 2001. Proceedings. IJCNN'01, vol 4. IEEE, pp 2673–2678
Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
Slowik A, Bialko M (2008) Training of artificial neural networks using differential evolution algorithm. In: 2008 Conference on human system interactions. IEEE, pp 60–65
Socha K, Blum C (2007) An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Comput Appl 16(3):235–247
Trujillo MCR, Alarcón TE, Dalmau OS, Ojeda AZ (2017) Segmentation of carbon nanotube images through an artificial neural network. Soft Comput 21(3):611–625
Wang G-G, Deb S, Cui Z (2015) Monarch butterfly optimization. Neural Comput Appl 1–20
Wang L, Zeng Y, Chen T (2015) Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Syst Appl 42(2):855–863
Wdaa ASI (2008) Differential evolution for neural networks learning enhancement. PhD thesis, Universiti Teknologi Malaysia
Whitley D, Starkweather T, Bogart C (1990) Genetic algorithms and neural networks: optimizing connections and connectivity. Parallel Comput 14(3):347–361
Wienholt W (1993) Minimizing the system error in feedforward neural networks with evolution strategy. In: ICANN93. Springer, pp 490–493
Wolberg WH, Mangasarian OL (1990) Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc Natl Acad Sci 87(23):9193–9196
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
Yang X-S, Deb S, Fong S (2011) Accelerated particle swarm optimization and support vector machine for business optimization and applications. In: International conference on networked digital technologies. Springer, pp 53–66
Yang X-S (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio Inspired Comput 2(2):78–84
Yang X-S, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483
Yang X-S, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multiobjective optimization. Eng Optim 46(9):1222–1237
Yao X, Liu Y (1999) Neural networks for breast cancer diagnosis. In: Proceedings of the 1999 congress on evolutionary computation, 1999. CEC 99, vol 3. IEEE, pp 1760–1767
Yi-Chung H (2014) Nonadditive similarity-based single-layer perceptron for multi-criteria collaborative filtering. Neurocomputing 129:306–314
Zhang J-R, Zhang J, Lok T-M, Lyu MR (2007) A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Appl Math Comput 185(2):1026–1037

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.