IUST
Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper

Fuzzy Grasshopper Optimization Algorithm: A Hybrid Technique for Tuning the Control Parameters of GOA Using Fuzzy System for Big Data Sonar Classification

A. Saffari*, S. H. Zahiri*(C.A.), and M. Khishe**

Abstract: In this paper, multilayer perceptron neural network (MLP-NN) training is used by the grasshopper optimization algorithm with the tuning of control parameters using a fuzzy system for the big data sonar classification problem. With proper tuning of these parameters, the two stages of exploration and exploitation are balanced, and the boundary between them is determined correctly. Therefore, the algorithm does not get stuck in the local optimization, and the degree of convergence increases. So the main aim is to get a set of real sonar data and then classify real sonar targets from unrealistic targets, including noise, clutter, and reverberation, using GOA-trained MLP-NN developed by the fuzzy system. To have accurate comparisons and prove the GOA performance developed with fuzzy logic (called FGOA), nine benchmark algorithms GOA, GA, PSO, GSA, GWO, BBO, PBIL, ES, ACO, and the standard backpropagation (BP) algorithm were used. The measured criteria are concurrency speed, ability to avoid local optimization, and accuracy. The results show that FGOA has the best performance for training datasets and generalized datasets with 96.43% and 92.03% accuracy, respectively.

Keywords: Classification, Fuzzy System, Grasshopper Optimization Algorithm, MLP-NN, Sonar.

1 Introduction

AT present, due to the increase and diversity of data, analysis and classification should be done more accurately [1]. This approach makes safer decisions to improve efficiency and reduce costs. One example of this data considered among the big data family is sonar data [2].

The complicated physical features of sonar targets, the classification of real targets, and avoiding unreal targets have become an essential practical field for active researchers and artisans [3]. Due to this complexity and heterogeneous sound circulation situation in seawater, many properties for the classification and differentiation of sonar targets should be extracted. As the feature vector dimensions increase, data dimensions increase as well [4]. Therefore, this paper will discuss the classification of real sonar targets from unreal sonar targets (noise, clutter, reverberation).

There are two general approaches for classifying data with high dimensions. The first one is to use the deterministic method. This method's reliability is so high, so it leads to the best answer; however, this method's challenging point appears when data dimensions are so high, followed by an increase in the spatial and temporal complexity [5]. Indeed, this method has no suitable practicality for the data, which is considered big data. The second approach is the stochastic method. These methods provide a quasi-optimal answer [6]. Besides, they have fewer spatial and temporal complexities in comparison with deterministic methods [7]. Artificial Neural Networks (ANN) are among the best stochastic methods used in big data in

Iranian Journal of Electrical and Electronic Engineering, 2022.
Paper first received 17 March 2021, revised 30 June 2021, and accepted 06 July 2021.
* The authors are with the Faculty of Electrical Engineering and Computer, University of Birjand, Birjand, Iran.
E-mails: abbas.saffari@birjand.ac.ir and hzahiri@birjand.ac.ir.
** The author is with the Department of Electronic, Imam Khomeini Marine Science University, Nowshahr, Iran.
E-mail: m_khishe@alumni.ac.ir.
Corresponding Author: S. H. Zahiri.
https://doi.org/10.22068/IJEEE.18.1.2131

the real world.

As discussed in the following section, one of the methods of MLP-NN training is the use of metaheuristic algorithms. All metaheuristic algorithms consist of two phases of exploration and exploitation. The exploration phase explores a dedicated search space, and the exploration phase convergence the algorithm to reliable responses. In many cases, because there is no clear boundary between the two phases, the algorithm gets stuck in local optimizations. The problem of getting stuck in local options is exacerbated when working with big data. GOA is one of the metaheuristic algorithms. In this paper, an improved version of GOA with a fuzzy system is used to teach MLP-NN in sonar data classification.

The paper is organized in such a way that Section 2 deals with Background and Related works. Section 3 explains general issues for GOA. The proposed method for teaching MLP-NN is presented in Section 4. Section 5 will present the data set. Section 6 describes the feature extraction model. Section 7 deals with setting parameters and simulation. Simulation results will be discussed in Section 8.

2 Background and Related Works

MLP-NNs are one of the most widely used tools for soft computing [4]. Non-linear problems can be solved by using these networks. Learning is a common and essential part of all neural networks, divided into supervised learning and unsupervised learning. For MLP-NNs (in most applications), backpropagation algorithms or standards [8] are used as the learning method from the supervised learning family. The backpropagation algorithm is based on a gradient with drawbacks such as slow convergence [9] and is therefore unreliable for practical applications.

The ultimate goal of the neural network learning process is to find the best combination of weighted edges and their bias to have the least amount of error in network training and test samples. However, most MLP-NN errors will be large for a long time during the learning process, and the learning algorithm will reduce them. This is quite common in gradient-based learning processes, such as backpropagation algorithms. Also, the convergence of the backpropagation algorithm depends largely on the initial values of the learning rate and the size of the motion. Improper values of these variables can even cause the algorithm to diverge. Many studies have been done to solve this problem of the backpropagation algorithm [10], but not enough optimization has been achieved, and each method has only its side effects.

In recent years, we have witnessed the increasing use of meta-heuristic algorithms for the subject of neural network training. In the following (Table 1), some studies related to neural network training using various metaheuristic algorithms are discussed.

GA and SA are likely to reduce local optimization but converge at a slower rate. This works poorly in applications that require real-time processing. PSO is faster than evolutionary algorithms but usually cannot compensate for the quality of the solution by increasing the number of iterations. PSOGSA is very complex, and its performance is not suitable for solving high-dimensional problems. BBO has time-consuming calculations. GWO, SCA, and IWT, despite their low complexity and high convergence speed, fall into the trap of local optimization, so they are not suitable for problems with local optimization. Many regulatory parameters and high complexity in the SSA algorithm are among the weaknesses of this algorithm. In addition to having many control parameters, DOA has time-consuming calculations that are not suitable for real-time use.

One of the problems that GOA has in some issues is getting stuck in local optimizations. The main reason for getting stuck in local optimizations is the imbalance between the two phases of exploration and extraction. Reference [21] reviews the various methods of developing and improving GOA. One of the techniques to improve the performance of this algorithm is the use of fuzzy systems. Optimum sizing and placement of distributed generations shunt capacitors and electric vehicle charging stations [22], Cells segmentation in

Table 1 some related studies on the use of metaheuristic algorithms in neural network training.

Paper	Neural Network	Application	Algorithms	Period
[11]	Feedforward	sonar image classification	Genetic Algorithm (GA)	1989
[12]	MLP	detection of the magnetic body in magnetic field	simulated annealing (SA)	1994
[13]	MLP	Prediction of gas solubility in polymers	Particle Swarm Optimizer (PSO)	2013
[14]	MLP	Parkinson's Disease identification	Spider Community Algorithm (SSA)	2014
[15]	MLP	Classification of sonar dataset	Gray Wolf Optimization (GWO)	2016
[3]	Feed-Forward	Classification of sonar targets	Particle Swarm Optimizer (PSO)	2017
[16]	MLP	Big data	Biogeography-based optimization (BBO)	2018
[17]	MLP	Classification of sonar targets	Dragonfly optimization algorithm (DOA)	2019
[18]	MLP	Sonar datasets classification	Improved whale	2019
[19]	MLP	Classification of EEG signals	PSOGSA	2020
[20]	deep Convolutional	COVID19 diagnosis	Sine–cosine (SCA)	2021

histopathological images [23], and photovoltaic maximum power point tracking technique [24] are examples of using a fuzzy system to improve GOA performance. The first fuzzy technique requires much time to run and is not suitable for real-time applications. The second and third fuzzy techniques are suitable for data with small dimensions but do not appropriately respond by increasing the dimensions.

This section will explain the case of "No Free Lunch (NFL)" [25, 26]. This proposition proves logically that there is no metaheuristic method that is suitable for settling all optimization troubles. In other words, one metaheuristic method can work promisingly and adequately on a particular set of problems while the same method acts poorly on another set of problems. The NFL activates this field of study and causes to improve current methods and proposition of new metaheuristic methods yearly. According to the NFL theory and the problems mentioned above and the inability of GOA to work with big data, the proposed method for improving GOA by the fuzzy system for MLP-NN training and finally, the classification of real sonar targets from unreal targets (clutter, noise, and reverberation) will be presented.

3 Grasshopper Optimization Algorithm

GOA was introduced in 2017 by Saremi *et al*. [27]. Inspired by nature, this algorithm divides the search space into two phases of exploration and exploitation. The exploratory phase causes the search agent's sudden movements, and the exploitation phase causes the search agent's local movements. The following mathematical model has been used to simulate grasshopper's social behavior [24]:

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

where X_i shows i-th grasshopper position, A_i shows the wind, S_i shows the social interaction and G_i shows the gravity imposed on the i-th grasshopper. To show the random behavior of grasshoppers, Eq. (1) becomes (2) with the addition of random factors:

$$X_i = r_1 A_i + r_2 S_i + r_3 G_i \tag{2}$$

where r_1, r_2, and r_3 are random numbers in the range [0,1]. Social interaction is obtained from (3).

$$S_i = \sum_{\substack{j=1 \\ i \neq j}}^{N} s\left(d_{ij}\right) d_{ij} \tag{3}$$

where N is the number of grasshoppers, d_{ij} shows the distance between i-th and j-th grasshopper and d_{ij} is a unit vector from the i-th grasshopper to the j-th grasshopper and is calculated from (4) and (5), respectively.

$$d_{ij} = \left| x_j - x_i \right| \tag{4}$$

$$d_{ij} = \left(\frac{x_j - x_i}{d_{ij}} \right) \tag{5}$$

As a result, social interaction is calculated using the following equation.

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s\left(\left| x_j - x_i \right|\right) \left(\frac{x_j - x_i}{d_{ij}} \right) \tag{6}$$

The force of gravity on each grasshopper is calculated using (7).

$$G_i = -g e_g \tag{7}$$

where g is the gravitational constant and e_g shows a unity vector towards the center of the earth [23].

Equation (8) also shows the effect of wind force on the i-th grasshopper.

$$A_i = u e_w \tag{8}$$

where u is a constant drift and e_w is a unit vector in the direction of the wind.

s is a function that expresses the power of social interaction and is calculated by the following equation:

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} \tag{9}$$

where f is the absorption intensity constant, and l is the constant length of the absorption length. In [23] does not discuss the gravity operator and always considers the wind's direction towards the target. In the following, Eq. (1) becomes the following relation:

$$X_i^d = C1 \left(\sum_{\substack{j=1 \\ j \neq i}}^{M} C2 \frac{ub_d - lb_d}{2} \cdot s\left(\left| x_j^d - x_i^d \right|\right) \frac{x_j - x_i}{d_{ij}} \right) + T_d \tag{10}$$

Respectively ub_d and lb_d are the highest and lowest limits of d-th dimension. T_d is the value of the d-th dimension in the target (best solution found so far). Eq. (10) indicates that the grasshopper's next position is calculated according to the other grasshopper's current position and position and the target position. Parameter $C1$ is very similar to the weight of inertia in PSO, reduces movement around the target, and balances the two phases of exploration and exploitation. $C2$ decreases the comfort zone, repulsion zone, and attraction zone between grasshoppers.

To extract some effective fuzzy rules, at first, a linguistic description of the effects of the GOA parameters on its search process is presented as a

subsection.

3.1 The Linguistic Description of the Effect of GOA Parameters on its Search Process

The role of the C1 is very important in GOA convergence behavior. The parameter makes a compromise between the global and local exploration abilities of the grasshopper. A large C1 facilitates global exploration (searching new areas), while a small one tends to facilitate local exploration.

A suitable value of the C1 usually provides a balance between global and local exploration abilities, and consequently, a more efficient search to locate the optimum solution can be done. A general rule of thumb suggests that it is better to initially set the C1 to a large value to make better global exploration of the search space and gradually decrease it to get more refined solutions.

The GOA search process is non-linear and complicated, and linearly decreasing or increasing C1 changes the search from global to local and local to global linearly. However, many problems require the search algorithm to have non-linear searchability. Deriving some statistical features from the obtained results in each iteration may help to understand the GOA search quality and calculate a proper C1 for the next iteration.

The *C*2 parameter also plays a critical role in GOA convergence and decreases the attraction zone, comfort zone, and repulsion zone between grasshoppers. Also decreases the space that the grasshoppers should explore and exploit. It can be interpreted that grasshoppers first seek food in a large search space (exploration phase). After finding the right area, they search carefully for food (exploitation phase). In other words, search agents search to find the general optimality after exploration in the search space. *C*2 is one of the most influential parameters in convergence. It can be said that parameter *C*2 must first be set to the maximum value, and as the number of iterations increases, its value must decrease.

4 Training a Multilayer Neural Network Using FGOA

In this article, the FGOA is used to find the best combination of the edge weight and the node's bias with the least amount of error in MLP-NN. Due to the non-complexity of MLP-NN, the vector method is used to show the weight of the edges and bias nodes. The MATLAB general toolbox has not been used to reduce running time. The following equation is an example of the coding method for the MLP-NN, and Fig. 1 shows how to training MLP-NN using FGOA.

$$Position = [w_{13}\ w_{14}\ w_{15}\ w_{16}\ w_{23}\ w_{24}\ w_{25}\ w_{26}\ w_{37}\ w_{47}\ w_{57}\ w_{67}\ \theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5]$$

4.1 Design of The Fuzzy Logic Controller for Parameters GOA

All optimization algorithms have essential and influential parameters. These parameters play an influential role in the search process. For example, factors such as early convergence, convergence rate, local optimization, exploration, and exploitation depend on these influential parameters' correct and dynamic adjustment. As shown in Section 3.1 and Eq. (10), *C*1 and *C*2 are two important GOA parameters. In references [28, 29], fuzzy inference has been used as an automatic adjuster of control parameters to upgrade the algorithm, modify the search process, prevent getting stuck in local optimizations, etc. According to studies, a fuzzy system is a suitable candidate for intelligent control of control parameters.

It should be noted that efficiency and complexity in most intelligent systems based on metaheuristic algorithms are directly related. The higher the efficiency, the more complex the parallel. Therefore, the use of an additional intelligent controller (fuzzy controller) increases the computation's complexity, which improves the performance of the high-dimension feature space. Therefore, in this paper, the fuzzy inference is used to develop GOA. To extract some effective fuzzy rules, in Section 3.1. linguistic description of the effects of the GOA parameters on its search process is presented.

The proposed fuzzy system has three stages of fuzzification, fuzzy inference, and de-fuzzification. The first step is fuzzy construction. In this step, the inputs are converted to the fuzzy model for processing based on the fuzzy system. In the second step, the fuzzy system evaluates and infers the rules using the Mamdani inference algorithm. The last step is called

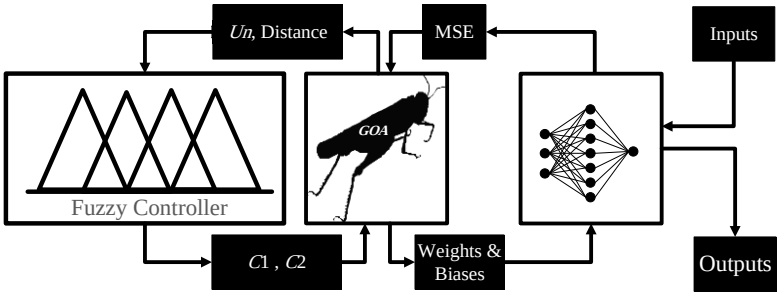
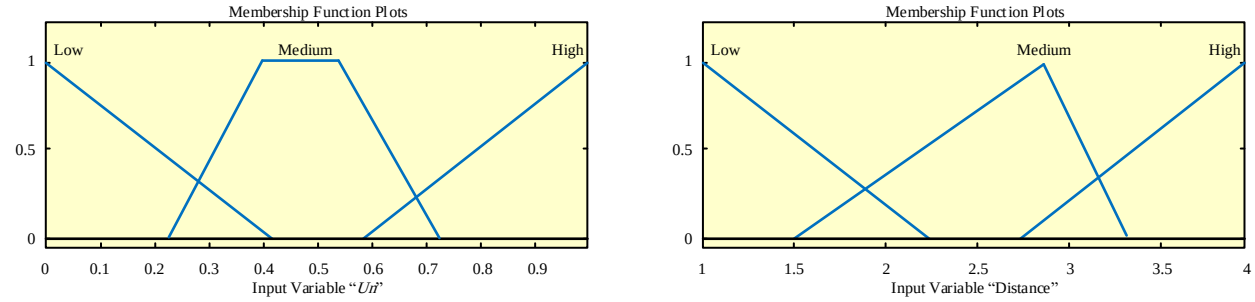


Fig. 1 How to train a neural network using FGOA.

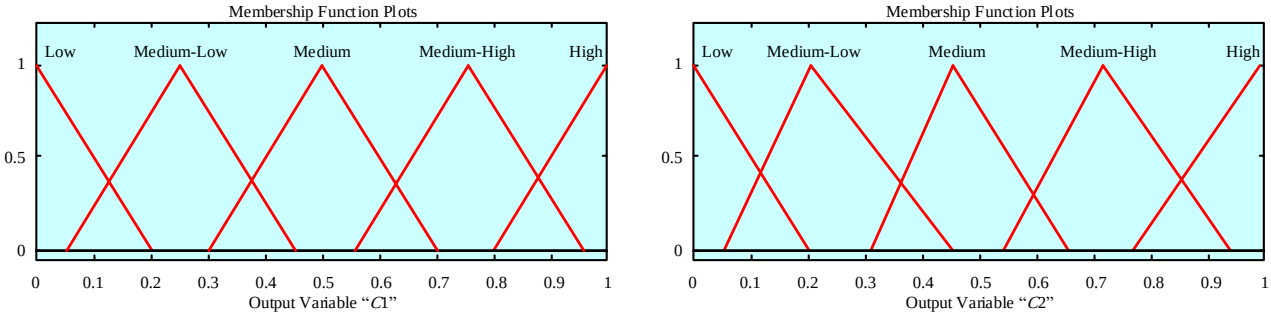


(a) (b)

Fig. 2 The membership functions for fuzzification; a) Membership function of "*Un*" parameter and b) Membership function of "Distance" parameter.

Table 2 the rules used in the fuzzy system.

Rules
If (*Un* is Low) and (Distance is Low), then (*C*1 is Medium_High) (*C*2 is Medium_Low)
If (*Un* is Low) and (Distance is Medium), then (*C*1 is Medium_High) (*C*2 is Medium)
If (*Un* is Low) and (Distance is High), then (*C*1 is High) (*C*2 is High)
If (*Un* is Medium) and (Distance is Low), then (*C*1 is Medium) (*C*2 is Medium_Low)
If (*Un* is Medium) and (Distance is Medium), then (*C*1 is Medium) (*C*2 is Medium)
If (*Un* is Medium) and (Distance is High), then (*C*1 is Medium) (*C*2 is Medium_High)
If (*Un* is High) and (Distance is Low), then (*C*1 is Low) (*C*2 is Low)
If (*Un* is High) and (Distance is Medium), then (*C*1 is Medium_Low) (*C*2 is Medium)
If (*Un* is High) and (Distance is High), then (*C*1 is Medium_Low) (*C*2 is Medium_High)



(a) (b)

Fig. 3 The membership functions for de-fuzzification; a) Membership function of "*C*1" parameter and b) Membership function of "*C*2" parameter.

de-fuzzification, in which the results of fuzzy inference, which are in the form of fuzzy sets, are converted into quantitative data and information.

In this model, the membership functions for fuzzification (It consists of two inputs shown in Fig. 2 are):

Un: is the number of repetitions in which the value of the fitness function is unchanged. The "*Un*" membership function is shown in Fig. 2(a).

Distance: indicates the distance between the grasshoppers, which is normalized in the range [1, 4] in each iteration. The "Distance" membership function is shown in Fig. 2(b).

After fuzzy input data, a set of rules is presented using the Mamdani inference algorithm according to Table 2.

In the following, the Mamdani inference output will be de-fuzzification by the two membership functions of parameters *C*1 and *C*2. The de-fuzzification phase consists of two parameters *C*1 and *C*2, which, if properly adjusted, determine the boundary between the exploration and exploitation phases.

Parameters *C*1, *C*2: These parameters lead to convergence or divergence of search agents. Fig. 3 shows the membership function of parameters *C*1 and *C*2.

Fig. 4 shows the diagram block of the proposed method.

5 Data Set

The complex environment of sound propagation at sea has prepared reliable sonar datasets, always one of the most challenging parts of sonar research [18]. Therefore, in this article, a set of experimental sonar data was prepared.

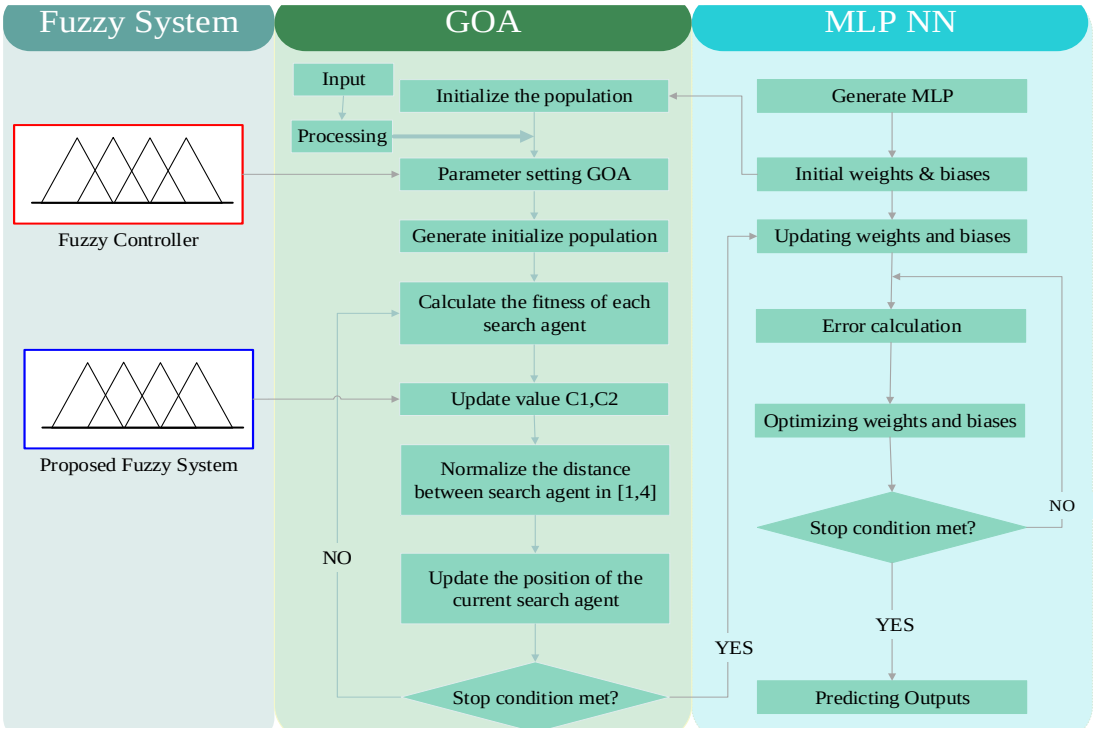


Fig. 4 Flowchart related to the GOA used in a neural network.

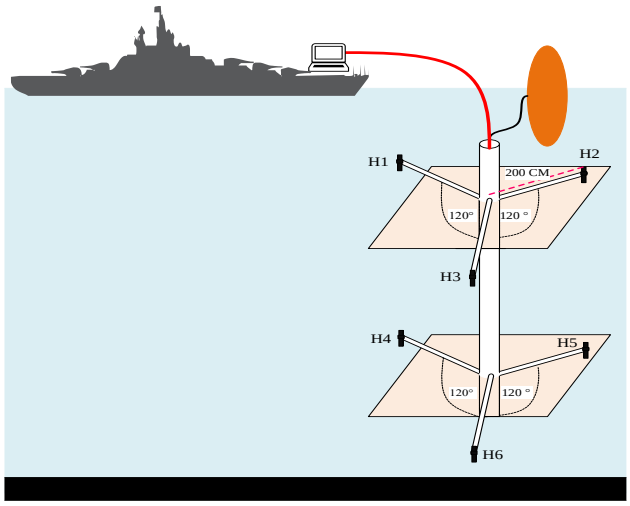


Fig. 5 Location of hydrophones.

Fig. 6 Hydrophone model of 8103.

Fig. 7 DT9816 model data acquisition module.

Obtaining sonar data set in summer 2020 and different regions of the Persian Gulf has been done. Most recordings were made in the south of Qeshm Island (26°74'25.4 "N 56°14'39.1 "E) with a depth of fewer than 60 meters.

The equipment used to obtain real samples of the sound at sea includes the following: Base for the placement of hydrophones, computer with at least 2.5 GHz processor, six hydrophones, data acquisition module, USB 2.0 to fiber media converter ICRON USB Ranger 2324, fiber optic cable.

Six hydrophones mounted on an aluminum stand at two different depths have been used to increase the recording's dynamic range. As shown in Fig. 5, Each hydrophone's distance to the central axis is 200 cm.

The hydrophones used are B&K model 8103 (shown in Fig. 6). These hydrophones have a frequency response range of 0.1-180 kHz and work well up to 40 atmospheres.

In this experiment, the DT9816 model (shown in Fig. 7) of ECONseries modules was used. These modules are a flexible and cost-effective set of multi-purpose data collection modules that can be stored and edited using modern hardware and flexible software in addition to an LCD.

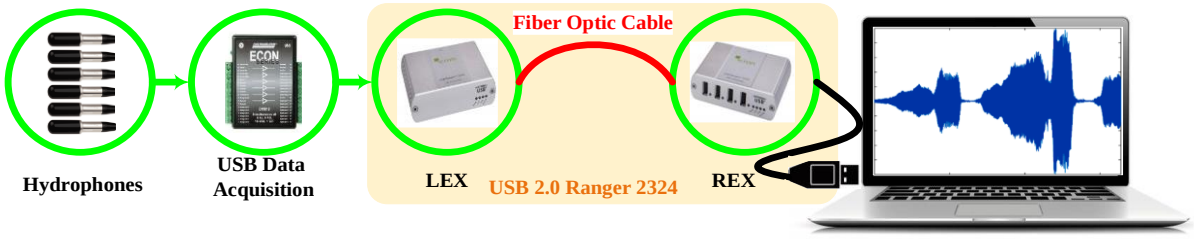


Fig. 8 General schematic of the real sonar data acquisition system.

Equipment used underwater is connected to ship-based equipment using a fiber optic cable. Fiber optic cable model for power and data transmission MacArtney Underwater Technology Model 3444. Three fibers for data transfer is a fiber for data storage. Two pairs of 1 mm^2 coated copper wire insulated with polyolefin are responsible for conducting 24 volts, two amps. The length of optical fiber is 150 meters. The USB 2.0 Ranger 2324 extends high-speed USB 2.0 connections up to 500 m using multimode fiber. A general schematic of the actual sonar data acquisition system is shown in Fig. 8.

6 Feature Extraction

After the pre-processing section and receiving the detected frames containing the audio corresponding to the received signals [17], at this stage, the detected sounds whose effects of synthetic phenomena have been removed and taken to the frequency domain (named $S(k)$) are delivered to the feature extraction section. At this stage, the energy of the signal spectrum is first computed by (11):

$$\left|S\left(k\right)\right|^2 = S_r^2\left(k\right) + S_i^2\left(k\right) \tag{11}$$

In this relation, $S_r(k)$ and $S_i(k)$ are the real and imaginary parts of the Fourier transform of the detected signal, respectively. The spectral energy of $|S(k)|^2$ is then filtered by Mel-scaled triangular filters. The output energy of the l filter is obtained by the relation (12).

$$E\left(l\right) = \sum_{k=0}^{N-1}\left|S\left(k\right)\right|^2 H_l\left(k\right) \tag{12}$$

In this relation, N is the number of discrete frequencies used in the FFT conversion of the pre-processing step and $H_l(k)$ is a filtered transfer function, where, l = 0, The dynamic range of Mel-Filtered Energy Spectrum is compressed by the logarithm function as (13).

$$E\left(l\right) = \log\left(E\left(l\right)\right) \tag{13}$$

Finally, mel-frequency cepstral coefficients (MFCC) are returned to the time domain by (14) and discrete cosine transform (DCT) [30].

$$C\left(n\right) = \sum_{l=1}^{M} e\left(l\right)\cos\left(n\left(l-\frac{1}{2}\right)\frac{\pi}{M}\right) \tag{14}$$

In this case and for any explicit purpose, the feature vector will be related (15).

$$X_m = \left[c\left(0\right).c\left(1\right)....c\left(P-1\right)\right]^{\mathrm{T}} \tag{15}$$

The steps mentioned in the pre-processing and feature extraction steps are shown as a block diagram in Fig. 9.

In this section, 140 features have been extracted. Given that the number of samples is 500, the size of the data is 500×140, of which 140 will be the number of input nodes (n) of the neural network, and given the relationship ($2\times n+1$), the number of neurons in the hidden layer will be 281. Therefore, despite such large data sets, computational and deterministic methods have a very high time complexity. Therefore random methods are considered as the best solution for this kind of problem.

7 Adjusting Parameters and Performing Simulations

In this chapter, to test the FGOA algorithm's efficiency in training MLP-NN, this network is trained and the FGOA algorithm by GWO, GSA, ACO, PSO, BBO, GOA, GA, ES, PBIL, BP benchmark algorithms. The parameters and initial values of these algorithms are shown in Table 3.

Initially designed classifiers are applied to sonar data with the size of 140*500, and the classifiers are tested in terms of classification rate, avoidance of local minimization, and convergence speed. Training and testing rates are estimated at 70-30%, respectively. The reason for this choice is to prevent the two phenomena of under-fitting and over-fitting. Each algorithm is executed 30 times, and the classification rate, the mean and standard deviation of the minimum error, and P-value are shown in Table 4 and Fig. 10. The classification rate represents the accuracy of the designed classifier. The mean values and standard deviation of the minimum error and P-value represent the algorithmic power in avoiding local optimization. Fig. 10 also shows a comprehensive comparison of the

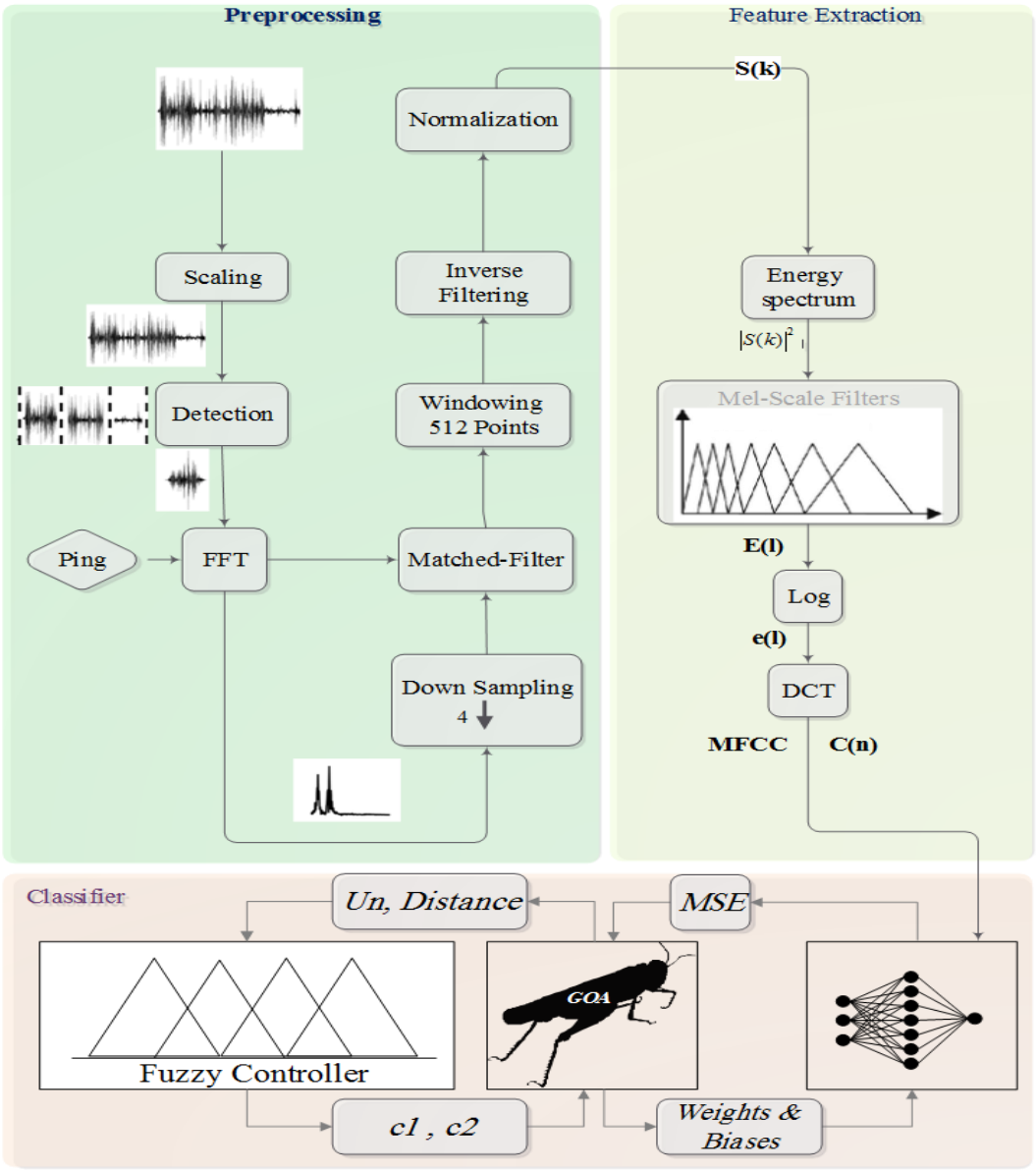


Fig. 9 Block diagram of the pre-processing and feature extraction steps.

convergence rate and method and the classifier's final error.

As shown in Figs. 11 and 12, the proposed algorithm (FGOA), an improved version of GOA, has a higher convergence rate than GOA and the other eight algorithms. As shown in Table 3, algorithm FGOA with 96.43% has the best performance, and algorithm BP with 61.27% has the weakest performance in the big data sonar classification. It should be noted that FGOA, with 95.89% accuracy, achieved the best performance for the generalized data set. The most important reason for FGOA's optimal performance for big data sonar is that the designed fuzzy system can determine the boundary of the exploration and exploitation phase by tuning the *C*1 and *C*2 parameters. As shown in Table 3 and the standard deviation and P-value values, FGOA performs best to avoid getting stuck in the local minimum.

8 Conclusion

In this paper, high-dimensional sonar data were classified using multilayer neural networks. GWO, FGOA, GOA, GA, PSO, ACO, BBO, GSA, ES, PBIL, and BP algorithms train the MLP-NN classifier. As can be seen, FGOA can correctly identify the boundary between exploration and exploitation phases by properly tuning the control parameters. That is why it sticks to local optimization and proves its ability to find big data Sonar's global optimization. The results show that FGOA, an improved GOA version, performs best with 96.43% compared to GOA and eight other algorithms. Also, the convergence rate of this algorithm is better than the other five algorithms.

Table 3 Parameters and initial values of training algorithms.

Algorithm	Parameter	Value
GWO	Population size	60
	The number of the gray wolf	13
GOA	Highest value (*cmax*)	1
	Lowest value (*cmin*)	0.00001
FGOA	*C*1	Tuning by Fuzzy system
	*C*2	Tuning by Fuzzy system
GA	Selection	Tournament
	Recombination	Single-point (1)
	Mutation	0.01
PSO	*C*1	1.1
	*C*2	1.1
	W	0.4
ACO	$\tau 0$	0.000001
	Q	20
	*q*0	1.1
	Pg	0.8
	Pt	0.6
	a	2
	β	6
GSA	α	20
	Limit down	-30
	Limit up	30
	$G_{\circ}$	1
	The initial speed of the masses	[0, 1]
	The initial value of the acceleration	0
BBO	The initial value of mass	0
	The probability of correcting the habitants	1
	The probability range for migrating into for each gene	[0, 1]
	Step size for the probability numerical integral	1
	Maximum migration into (*I*) and migrating out of (*E*) coefficient	1
	Mutation probability	0.005
ES	Weighting factor (*F*)	0.5
	Crossover constant (*CR*)	0.5
PBIL	Mutation probability	0.1
	Mutation shift	0.4
	Learning rate	0.05

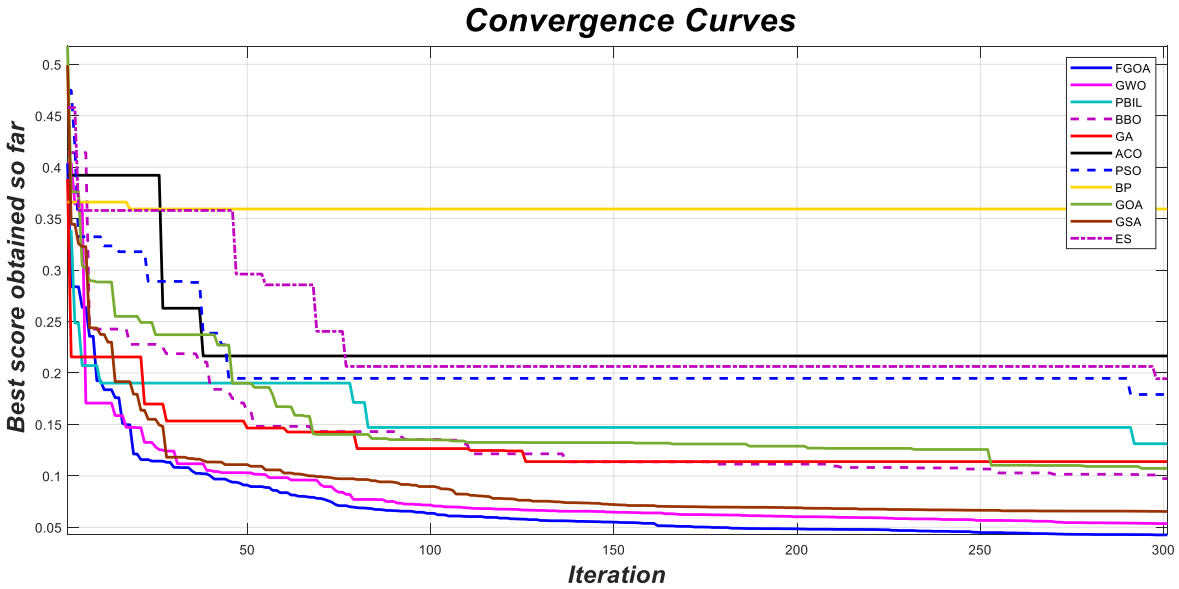


Fig. 10 Convergence diagram of different training algorithms.

Table 4 Results of applying different training algorithms in designing sonar purpose classifier.

Classifier		MLP-GWO	MLP-BBO	MLP-FGOA	MLP-GSA	MLP-GOA	MLP-PSO	MLP-GA	MLP-ACO	MLP-ES	MLP-PBIL	MLP-BP
Accuracy	Best	93.55	92.84	96.43	93.12	92.10	84.35	90.41	73.45	79.45	85.37	61.27
	Worst	89.07	84.64	93.86	89.10	86.14	80.43	85.73	68.33	73.16	78.09	55.19
	Mean	92.64	88.79	95.68	91.98	88.13	83.71	87.45	71.34	75.55	81.09	58.86
MSE	Best	0.1387	0.1810	0.1147	0.3008	0.3744	0.6433	0.5610	0.8115	0.7567	0.5933	0.9246
	Worst	0.1693	0.2125	0.1492	0.3212	0.4103	0.7005	0.5993	0.8928	0.7980	0.6345	0.9677
	Mean	0.1529	0.2011	0.1383	0.3112	0.3895	0.6874	0.5828	0.8460	0.7828	0.6118	0.9461
	STD	0.0375	0.2016	8.271×10^{-4}	0.2685	0.3542	0.5912	0.4982	0.6112	0.6215	0.5912	0.6833
P-Values	Best	0.0019	0.116310^{-2}	0.0067	3.1599×10^{-3}	4.0019×10^{-1}	7.1244e-08	5.0573×10^{-5}	8.2245×10^{-6}	7.2564×10^{-2}	6.1119×10^{-4}	8.9205×10^{-1}
	Worst	0.0349	0.2185×10^{-2}	0.0109	2.9864×10^{-7}	4.4177×10^{-5}	8.0151×10^{-6}	5.6982×10^{-9}	9.1255×10^{-1}	7.7864×10^{-4}	6.3939×10^{-7}	9.6324×10^{-6}
	Mean	0.0029	0.1451×10^{-3}	0.0091	3.3349×10^{-1}	4.1149×10^{-6}	7.8339×10^{-1}	5.1229×10^{-9}	8.9152×10^{-2}	7.3152×10^{-2}	6.1239×10^{-5}	9.0041×10^{-2}

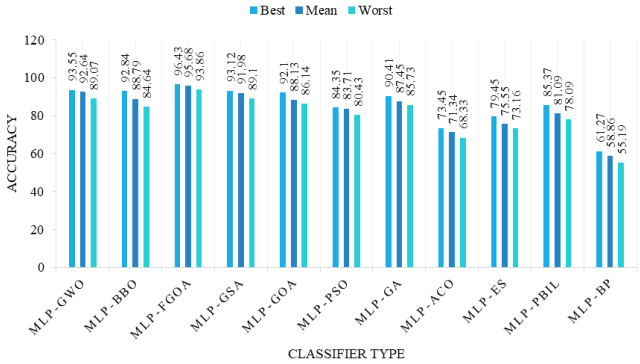


Fig. 11 Accuracy comparison chart in designed classifiers for training data.

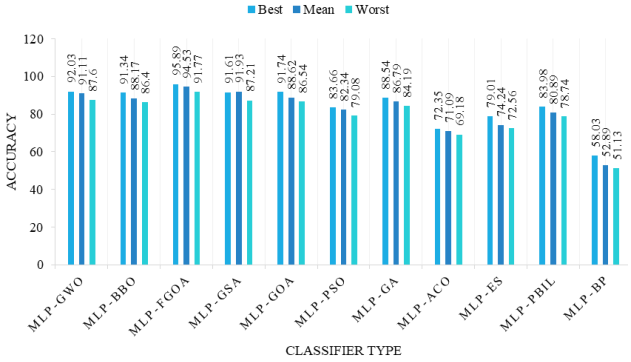


Fig. 12 Accuracy comparison chart in designed classifiers for generalized data.

Intellectual Property

The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property.

Funding

No funding was received for this work.

CRediT Authorship Contribution Statement

A. Saffari: Original draft preparation, Idea & conceptualization, Software and simulation, Revise & editing. **S. H. Zahiri:** Project administration, Methodology, Revise & editing. **M. Khishe:** Idea & conceptualization, Data curation, Verification.

Declaration of Competing Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be made by this journal. All authors have approved the manuscript and agree with its submission to "Iranian Journal of Electrical and Electronic Engineering".

References

[1] R. He, B. Ai, A. F. Molisch, G. L. Stuber, Q. Li, Zh. Zhong, and J. Yu., "Clustering enabled wireless channel modeling using big data algorithms," *IEEE Communications Magazine*, Vol. 56, No. 5, pp. 177–183, 2018.

[2] T. Berthold, A. Leichter, B. Rosenhahn, V. Berkhahn, and J. Valerius, "Seabed sediment classification of side-scan sonar data using convolutional neural networks," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, Vol. 2018, pp. 1–8, 2018.

[3] M. R. Mosavi and M. Khishe, "Training a feed-forward neural network using particle swarm optimizer with autonomous groups for sonar target classification," *Journal of Circuits, Systems and Computers*, Vol. 26, No. 11, p. 1750185, 2017.

[4] A. S. Mahboob and S. H. Zahiri, "Automatic and heuristic complete design for ANFIS classifier," Network: Computation in Neural Systems, Vol. 30, No. 1–4, pp. 31–57, 2019.

[5] P. Koturwar, S. Girase, and D. Mukhopadhyay, "A survey of classification techniques in the area of big data," *arXiv preprint arXiv:1503.07477*, 2015.

[6] P. Wang, X. Yu, and J. Lü, "Identification and evolution of structurally dominant nodes in protein-protein interaction networks," *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 8, No. 1, pp. 87–97, 2014.

[7] G. Gupta and K. Lakhwani, "Big data classification techniques: A systematic literature review," *Journal of Natural Remedies*, Vol. 21, No. 2, pp. 1–13, 2020.

[8] S. C. Ng, C. C. Cheung, S. H. Leung, and A. Luk, "Fast convergence for backpropagation network with magnified gradient function," in *IEEE Proceedings of the International Joint Conference on Neural Networks*, Vol. 3, pp. 1903–1908, 2003.

[9] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, "On the alleviation of local minima in backpropagation," *Nonlinear Analysis: Theory, Methods and Applications*, Vol. 30, 1997.

[10] M. A. Simaan, "Simple explanation of the no-free-lunch theorem and its implications," *Journal of Optimization Theory and Applications*, Vol. 115, No. 3, pp. 549–570, 2003.

[11] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proceedings of the 11th international joint conference on Artificial intelligence - Volume 1*, pp. 762–767, Aug. 1989.

[12] C. S. Koh and S. Y. Hahn, "Detection of magnetic body using artificial neural network with modified simulated annealing," *IEEE Transactions on Magnetics*, Vol. 30, No. 5, pp. 3644–3647, 1994.

[13] M. Li, X. Huang, H. Liu, B. Liu, Y. Wu, A. Xiong, and T. Dong, "Prediction of gas solubility in polymers by back propagation artificial neural network based on self-adaptive particle swarm optimization algorithm and chaos theory," *Fluid Phase Equilibria*, Vol. 356, pp. 11–17, 2013.

[14] L. A. M. Pereira, D. Rodrigues, P. B. Ribeiro, J. P. Papa, and S. A. T. Weber, "Social-spider optimization-based artificial neural networks training and its applications for Parkinson's Disease identification," in *IEEE 27th International Symposium on Computer-Based Medical Systems*, pp. 14–17, 2014.

[15] M. R. Mosavi, M. Khishe, and A. Ghamgosar, "Classification of sonar data set using neural network trained by gray wolf optimization," *Neural Network World*, Vol. 26, No. 4, pp. 393–415, 2016.

[16] X. Pu, S. Chen, X. Yu, and L. Zhang, "Developing a novel hybrid biogeography-based optimization algorithm for multilayer perceptron training under big data challenge," *Scientific Programming*, 2018.

[17] M. Khishe and A. Safari, "Classification of Sonar targets using an MLP neural network trained by dragonfly algorithm," *Wireless Personal Communications*, Vol. 108, No. 4, pp. 2241–2260, 2019.

[18] M. Khishe and M. R. Mosavi, "Improved whale trainer for sonar datasets classification using neural network," *Applied Acoustics*, Vol. 154, pp. 176–192, 2019.

[19] S. Afrakhteh, M. R. Mosavi, M. Khishe, and A. Ayatollahi, "Accurate classification of EEG signals using neural networks trained by hybrid population-physic-based algorithm," *International Journal of Automation and Computing*, Vol. 17, No. 1, pp. 108–122, 2020.

[20] C. Wu, M. Khishe, M. Mohammadi, S. H. Taher Karim, and T. A. Rashid, "Evolving deep convolutional neural network by hybrid sine-cosine and extreme learning machine for real-time COVID19 diagnosis from X-ray images," Soft Computing, Vol. 6, 2021.

[21] Y. Meraihi, A. B. Gabis, and S. Mirjalili, "Grasshopper optimization algorithm : theory, variants, and applications," Vol. 4, pp. 1–27, 2021.

[22] S. Rao, K. Jasthi, P. Goli, D. Das, and R. C. Bansal, "Grasshopper optimization algorithm based two stage fuzzy multiobjective approach for optimum sizing and placement of distributed generations, shunt capacitors and electric vehicle charging stations," *Journal of Energy Storage*, Vol. 27, p. 101117, 2020.

[23] V. Tiwari and S. C. Jain, "Histopathological cells segmentation using exponential grasshopper optimisation algorithm-based fuzzy clustering method," *International Journal of Intelligent Information and Database Systems*, Vol. 13, No. 2–4, pp. 118–138, 2020.

[24] L. Bhukya and S. Nandiraju, "A novel photovoltaic maximum power point tracking technique based on grasshopper optimized fuzzy logic approach," *International Journal of Hydrogen Energy*, Vol. 45, No. 16, pp. 9416–6427, 2020.

[25] S. P. Adam, S. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No free lunch theorem : A review," *Approximation and Optimization*, pp. 57–82, 2019.

[26] K. H. Lai, Z. Zainuddin, and P. Ong, "A study on the performance comparison of metaheuristic algorithms on the learning of neural networks," *AIP Conference Proceedings*, Vol. 1870, No. 1, p. 040039, 2017.

[27] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm : Theory and application," *Advances in Engineering Software*, Vol. 105, pp. 30–47, 2017.

[28] S. H. Zahiri and S. A. Seyedin, "Swarm intelligence based classifiers," *Journal of the Franklin Institute*, Vol. 344, No. 5, pp. 362–376, 2007.

[29] H. Askari and S. H. Zahiri, "Data classification using fuzzy-GSA," in *IEEE 1st International eConference on Computer and Knowledge Engineering (ICCKE)*, pp. 6–11, 2011.

[30] T, Giannakopoulos, A, Pikrakis, *Introduction to audio analysis*. Academic Press; 1st Edition, Apr. 2014.

A. Saffari was born in Touyserkan, Iran, in 1992. He received his B.Sc. degree in 2013 and M.Sc. degree in 2016 from Imam Khomeini Marine University, Department of Electrical Engineering. He is currently a Ph.D. Candidate in Electronic Engineering at the University of Birjand. His research interests are in neural networks, meta-heuristic algorithms, video and image processing, sonar, and classification.

S. H. Zahiri received the B.Sc., M.Sc., and Ph.D. degrees in Electronics Engineering from Sharif University of Technology, Tehran, Tarbiat Modares University, Tehran, and Mashhad Ferdowsi University, Mashhad, Iran, in 1993, 1995, and 2005, respectively. Currently, he is a Professor with the Department of Electronics Engineering, University of Birjand, Birjand, Iran. His research interests include pattern recognition, evolutionary algorithms, swarm intelligence algorithms, and soft computing.

M. Khishe was born in 1985 in Nahavand, Iran. He received the B.Sc. degree in Maritime Electronics and Communication Engineering from Imam Khomeini Maritime Sciences University, Nowshahr, Iran in 2007 and also, M.Sc. and Ph.D. degrees in Electronic Engineering from Islamic Azad University, Qazvin branch and Iran University of Science and Technology, Tehran, in 2012 and 2017, respectively. Since 2011, he is a faculty member of Maritime Electrical and Electronic Engineering at Imam Khomeini Maritime Sciences University, Nowshahr, Iran. His research interests are digital signal processing, artificial neural networks, meta-heuristic algorithms, sonar and radar signal processing, and FPGA design.