Iranian Journal of
Marine Technologies
Marine Science University of Imam Khomeini

A Modified Grey Wolf Optimizer by Individual Best Memory and Penalty Factor for Sonar and Radar Dataset Classification

Mohammad Taghavi, Mohammad Khishe

m_khishe@alumni.iust.ac.ir

Department of Electrical and Communication Engineering, Marine Sciences University of Imam Khomeini

Abstract: Meta-heuristic Algorithms (MA) are widely accepted as excellent ways to solve a variety of optimization problems in recent decades. Grey Wolf Optimization (GWO) is a novel MA that has been generated a great deal of research interest due to its advantages such as simple implementation and powerful exploitation. This study proposes a novel GWO-based MA and two extra features called Individual Best Memory (IBM) and Penalty Factor (PF) to train Feed-forward Neural Network (FNN) for the classification of Sonar and Radar datasets. Besides, FNN is accompanied by Feature Selection (FS) using GWO. Experiments were done on Sonar and Radar datasets obtained from the University of California, Irvin (UCI) to evaluate the performance of the proposed MA; the results demonstrated the proposed MA is markedly better than GWO in terms of classification accuracy, avoiding local optima stagnation, and convergence speed without higher computational complexity. This framework can be applied to naval navigation systems or atmospheric research.

Key words: Classification, Feature Selection, Grey Wolf Optimization, Meta-heuristic Algorithms, Radar, Sonar.

Received Date : 1397/06/12

Accepted Date : 1397/11/11

1. Introduction

There are plenty of algorithms have been utilized to solve optimization problems, where MAs have generated considerable research interest recently [1-8]. These algorithms have become remarkably common due to four reasons: simplicity, flexibility, derivation-free mechanism, and local optima avoidance [9]. Specifically ,Grey Wolf Optimization (GWO) is considered as one of the best-known MAs due to its remarkable further advantages such as few parameters, scalability, and powerful exploitation [10].

The prosperity of the GWO algorithm has motivated other researchers to apply the algorithm for solving different types of optimization problems. Specially, it has been widely used in a variety of machine learning problems such as Feed-forward Neural Network (FNN) and SVM classifiers [7, 11, 12], clustering algorithms [13, 14], and Feature Selection(FS) tasks [15, 16]. Further, researchers have used it to solve engineering problems like the unit commitment problem [17], combined heat and dispatch problem [18], and power system stabilizer design [19].

Nevertheless, GWO can't be efficient enough to solve all optimization problems, and when the search space dimension is growing, there is a possibility of slow convergence or getting trapped in local optima [20]. Because of this fact, researchers have modified GWO in order to solve a wider range of optimization problems. Rodriguez et al. [21] proposed a dynamic adaptation of GWO coefficients using fuzzy logic and reached more accurate answers than GWO. Saremi et al. [22] applied Evolutionary Population Dynamic (EPD) in GWO to eliminate the worst individuals in the population and achieved more accurate answers than GWO as well. Dudani and Chudasama [23] adopted a strategy for updating the position of the wolves based on incorporating a step size to detect partial discharge in the transformer. This method attains faster convergence with less parameter dependency than GWO.

However, in spite of the popularity of GWO, few researchers have addressed the classification problems using any kind of enhanced GWO. Classification is a form of data analysis that can be used to extract models describing important data classes [24]. Specifically, according to [25] and [26], Sonar and Radar (ionosphere) datasets [27] are two of demanding datasets owing to their complexity.

Therefore, due to the mentioned challenges, we are encouraged to promote GWO for the purpose of Sonar and Radar classification. It is well-known that GWO features great exploitation [20], so in order to achieve better balance between exploration and exploitation, we focused on exploration enhancement; we refine GWO by Individual Best Memory (IBM), a new update mechanism, and Penalty Factor (PF), a new eliminator of the worst solutions. This MA was used in FNN, a common classifier in many frameworks [4, 6, 28-32], to find the optimum combination of FNN's weights and biases. Furthermore, we assume that using MA simultaneously in FNN and FS methods will improve the results, so we used an FS method that deploys GWO to select an optimal feature subset used in FNN.

The rest of the paper is organized as follows: Section 2 gives an overview of the concepts of FNN and GWO. In section 3, we outline our framework that is an FNN classification task accompanied by an FS method. Firstly, IBM and PF are developed. Then the pseudo code of our MA is presented that is based on GWO, IBM and PF. Final steps are explaining FNN training and FS. Section 4 provides experimental results and comparisons between proposed MA and previous MAs, and section 5 is to draw Conclusions.

2. Related works

2.1. FNN Training

The FNN is an artificial neural network wherein connections between the units do not form a cycle. As such, it is different from recurrent neural networks [6]. There are several ways to train multilayer FNN using MAs. A common way is to find the optimum combination of the network's weights and biases using MAs in order to achieve the best result. The network structure remains unchanged in this way[1].

Further, there are some ways to demonstrate neuron weights and node biases. A regular way is to use vectors; a vector represents a particle of an MA. Each vector's element is used to store a neuron weight or node bias. To clarify this concept, the vector of a three-layer FNN shown in Fig. 1 is demonstrated in Equation(1) [1]:

$$\text{particle} = \left[w_{13} w_{23} w_{14} w_{24} w_{15} w_{25} w_{36} w_{46} w_{56} \theta_1 \theta_2 \theta_3 \theta_4 \right] \quad (1)$$

Where w_{ij} means the weight of the neuron that connects nodesi and j, and $\theta_i$is bias of the node i.

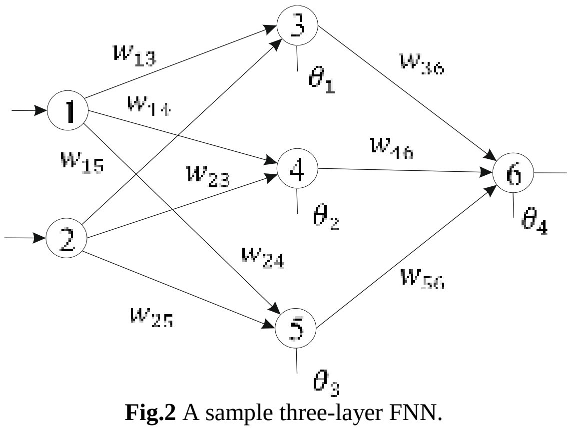

Fig.2 A sample three-layer FNN.

2.2. GWO

The GWO was proposed by Mirjalili et al. [9] in 2014. The fittest, second fittest and third fittest solutions are called α, β andδ respectively in the GWO mathematical model. The other solutions are called ω. The GWO algorithm utilizes α, β andδ to modify other solutions. The formulae that obtain the best solution are provided in Equations (2-9):

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) + \vec{A}.\vec{D} \tag{2}$$

where $\vec{D}$ is defined in (5). $\vec{A}$, $\overrightarrow{X_p}(t)$, $\vec{X}$ and tare coefficient vector, the prey position, the grey wolf position, and iteration number, respectively.

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{3}$$

where $\vec{C}$ is another coefficient vector. $\vec{A}$ and $\vec{C}$ are calculated as follow:

$$\vec{A} = 2\vec{A}.\vec{r_1} - \vec{a} \tag{4}$$

$$\vec{C} = 2\vec{r_2} \tag{5}$$

where $\vec{r_1}$ and $\vec{r_2}$ are random vectors that range between 0 and 1. $\vec{a}$ decreases linearly from 2 to 0 over the course of iterations. $\vec{a}$ sets a tradeoff between exploration and exploitation. It is formulated as follow:

$$\vec{a} = 2 - t.\frac{2}{\text{Maxiter}} \tag{6}$$

where Maxiter is the maximum number of iteration in GWO.

As it is mentioned above, positions of solutions are evolved by the best solutions α, β and δ, since it is assumed that these solutions have better knowledge about the best solution. Therefore, formulae that change the positions of solutions are as Equation (7-9). The pseudo-code of the GWO algorithm is presented in Fig.2.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right|,\ \overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}\right|,\ \overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right| \tag{7}$$

$$\overrightarrow{X_1} = \left|\overrightarrow{X_\alpha} + \overrightarrow{A_1}.\overrightarrow{D_\alpha}\right|,\ \overrightarrow{X_2} = \left|\overrightarrow{X_\beta} + \overrightarrow{A_2}.\overrightarrow{D_\beta}\right|,\ \overrightarrow{X_3} = \left|\overrightarrow{X_\delta} + \overrightarrow{A_3}.\overrightarrow{D_\delta}\right| \tag{8}$$

$$\vec{X}(t+1) = \left(\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}\right)/3 \tag{9}$$

GWO Algorithm
Initialize the grey wolf population $X_i (i = 1,2, \ldots, n)$
Initialize a, A and C.
Calculate the fitness of each search agent
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
While (t < Max number of iterations)
For each search agent
Update the position of the current search agent by equation (9)
End for
Update a, A and C
Calculate the fitness of search agents
Update X_α, X_β and X_δ
$t = t + 1$
End while
Return X_α

Fig. 2 Pseudo code of GWO [9]

3. Methodology

The proposed method is described in this section. This method contained two phases: FNN training and FS; both phases used an MA. The proposed method uses a novel MA that is a modified version of GWO. The modified GWO exploits two new well-designed approaches to improve the result of classification: IBM and PF. The proposed algorithm was named IPG (abbreviation of IBM-PF-GWO). GWO enhanced by IBM was called IG, and PG pointed to the GWO that was enhanced by PF. In this study, IPG is employed in the FNN training step so as to address the GWO drawbacks in the field of finding accurate neuron weights and node biases, and the algorithm that is chosen to exploit the best feature subsets involved in FS is GWO.

This section is set out as follows: First of all, IBM and PF are explained. Then, the pseudo-code of IPG algorithm is presented. Finally, FFN training and FS methods are discussed.

3.1. IPG

3.1.1. IBM

It is widely recognized that exploration and exploitation phases play crucial roles in optimal behavior of MAs. In regular GWO, α, β and δ, the three best solutions so far, guide the other agents to reach the global optima, which leads to powerful exploitation. However, there is possibility of finding a suboptimal solution without enough searches in the solution space, which results in poor exploration.

Promoting of update mechanism is a widespread way to improve the performance of GWO [23, 33]. So, we intend to incorporate the best memory of each agent into the update mechanism, named IBM, so as to compensate for the poor exploration and boost the final solution. More to the point, IBM compared each agent's new position to its previous positions using fitness. If the new position didn't have better fitness than that of previous positions, the agent's position used in Equation (7) wouldn't change, and if the new position had better fitness than that of previous positions, the agent's position used in Equation (7) would change to its new value. Hence, Equations (7-9) converted to Equations (10-12):

$$\overrightarrow{D'_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \overrightarrow{BP_X}\right|,\ \overrightarrow{D'_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \overrightarrow{BP_X}\right|,\ \overrightarrow{D'_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \overrightarrow{BP_X}\right| \tag{10}$$

$$\overrightarrow{X'_1} = \left|\overrightarrow{X_\alpha} + \overrightarrow{A_1}.\overrightarrow{D'_\alpha}\right|,\ \overrightarrow{X'_2} = \left|\overrightarrow{X_\beta} + \overrightarrow{A_2}.\overrightarrow{D'_\beta}\right|,\ \overrightarrow{X'_3} = \left|\overrightarrow{X_\delta} + \overrightarrow{A_3}.\overrightarrow{D'_\delta}\right| \tag{11}$$

$$\vec{X}(t+1) = \left(\overrightarrow{X'_1} + \overrightarrow{X'_2} + \overrightarrow{X'_3}\right)/3 \tag{12}$$

where $\overrightarrow{BP_X}$ was the agent's position that had the best individual fitness so far.

IBM was based on this assumption that the individual best memory of an agent has better knowledge about the optimum solution than current agent. This feature was developed to improve the exploration of the optimum solution through the entire solution space. It should be noticed that IBM developed in this method is remarkably different from IBM of PSO. In PSO, the distance between IBM and current position of an agent is used to direct an agent. In IPG, the IBM was used instead of current position of an agent to find a new answer, so the proposed IBM features the advantage of less computation complexity.

3.1.1. PF

It is previously discussed that both good exploration and exploitation are of the essence in order to have an efficient MA, and conventional GWO may suffer from poor exploration. Adding new operation is another common trend to improve the performance of MAs. PF was another efficient modification done on GWO.

First, a group of agents that had the worst fitness was collected in a vector ($\left(X_{\varepsilon 1}, X_{\varepsilon 2}, \ldots, X_{\varepsilon n_\varepsilon}\right) \in \overrightarrow{X_\varepsilon}$) that n_ε was the number of the worst search agents that should be penalized. This vector was updated each iteration. There was a Penalty_i corresponding to X_i, that kept the number of consecutive iterations that X_i remained in $\overrightarrow{X_\varepsilon}$. After each iteration, if X_i left $\overrightarrow{X_\varepsilon}$, then Penalty_i set to zero, but if X_i remained in $\overrightarrow{X_\varepsilon}$, the Penalty_i was increased by one. Finally, if Penalty_i exceeded the desired iteration number n_p, the corresponding agent X_i would be omitted and a new agent would be initialized randomly.

This decision relied on two assumptions. Firstly, agents with the worst fitness have weak knowledge about the optimum solution, so they can be omitted. Besides, the proposed PF cannot have a negative impact on MA, since if the new random answers aren't good, these answers won't influence the other solutions, and the random answers will be guided by three global best answers. This feature was utilized to improve the exploration of GWO as well.

It should be noticed that PF of IPG is not similar to PF of Artificial Bee Colony algorithm (ABC). In ABC, if answer fitness does not improve after certain iterations, the answer will be removed. In IPG, if an answer remains in the worst answers' group after certain iterations, it will be removed. Also, PF of IPG isn't as PF of [22]. Indeed, a subtle change between them made them fundamental different tools. To be precise, the worst solutions are removed in PF of IPG and [22] similarly, whilst new solutions are chosen based on the three best solutions and numerous formula in[22]. This decision leads to better exploitation. However, PF of IPG creates new solutions randomly to trigger better exploration as well as less complexity and time consumption.

3.1.3. Pseudo-code of IPG Algorithm

IBM and PF modifications were introduced and IPG can be presented now. The pseudo-code of IPG algorithm is presented in Fig. 3.

3.2. FNN Training and FS

FNN training and FS are implemented to address the classification problems. The IPG was proposed to improve the power of FNN, and GWO is elected to perform FS. The whole procedure represented in Fig. 4 is explained in detail as follow:

3.2.1. Scree test

Feature reduction and feature construction are good ways to achieve higher classification accuracy and reduce computational time. Scree test is a feature reduction method which is used in this study [34] which also makes the input nodes of FNN substantially plunge. The Scree test plots the components as the X axis and the corresponding eigenvalues as the Y-axis. As one sample moves to the right, toward later components, the eigenvalues drop. When the drop ceases and the curve makes an elbow toward less steep decline, Scree test says to drop all further components after the one starting the elbow. Feature construction can be done according to each dataset.

3.2.2. FS

When the desired feature number is achieved, FS may be done. In the first iteration, the population is initialized randomly.

IPG Algorithm

```
Initialize the grey wolf population X_i, (i = 1,2, ..., n)
Calculate the fitness of each search agent F_{X_i}, (i = 1,2, ..., n)
Initialize each agent's best known position BP_{Xi} and fitness F_{BP_{Xi}}
by its initial position and fitness BP_{Xi} = X_i , F_{BP_{Xi}} = F_{X_i},
(i = 1,2, ..., n)
X_α = the best search agent
X_β = the second best search agent
X_δ = the third best search agent
Initialize a, A, C and Penalty→ (Penalty_1, ..., Penalty_n) ∈
Penalty→ = (0, ...,0)
Initialize the worst agents (X_ε1, X_ε2, ..., X_εn_ε) ∈ X_ε→
While (t < Max number of iterations)
        For each search agent
      Update the position of the current search agent by
equation (12)
   End for
   Update a, A and C
   Calculate the fitness of search agents
   If F_{X_newi} > F_{BP_{Xi}}
      BP_{Xi} = X_newi
      F_{BP_{Xi}} = F_{X_newi}
    End if
   UpdateX_α, X_β and X_δ through agents' best known positions
   Update X_ε→
   For each agent
      If X_i ∈ X_ε→
         Penalty_i = Penalty_i + 1
         If Penalty_i = n_p
            Randomly initialize a new agent instead of the
            old one
         End if
      Else
         Penalty_i = 0
      End if
   End for
      t = t + 1
End while
Return X_α
```

Fig. 3 Pseudo code of IPG.

In the other iterations, it uses GWO to search the solution space (dataset) in order to achieve the best feature subset. GWO changes the selected features, based on fitness values calculated by FNN classifier. Actually, this FS method is wrapper based, because classification accuracy is used as the evaluation criterion. Therefore, the feature subsets are provided to be evaluated in the FNN.

3.2.3. FNN

In this step, the feature subsets are ready to be classified and evaluated by FNN. The number of input nodes was calculated already, and according to[35], the number of hidden nodes is achieved through Equation (13):

$$H = 2I + 1 \tag{13}$$

Where H and I are the number of hidden nodes and input nodes, respectively. One output node is enough because there are two classes.

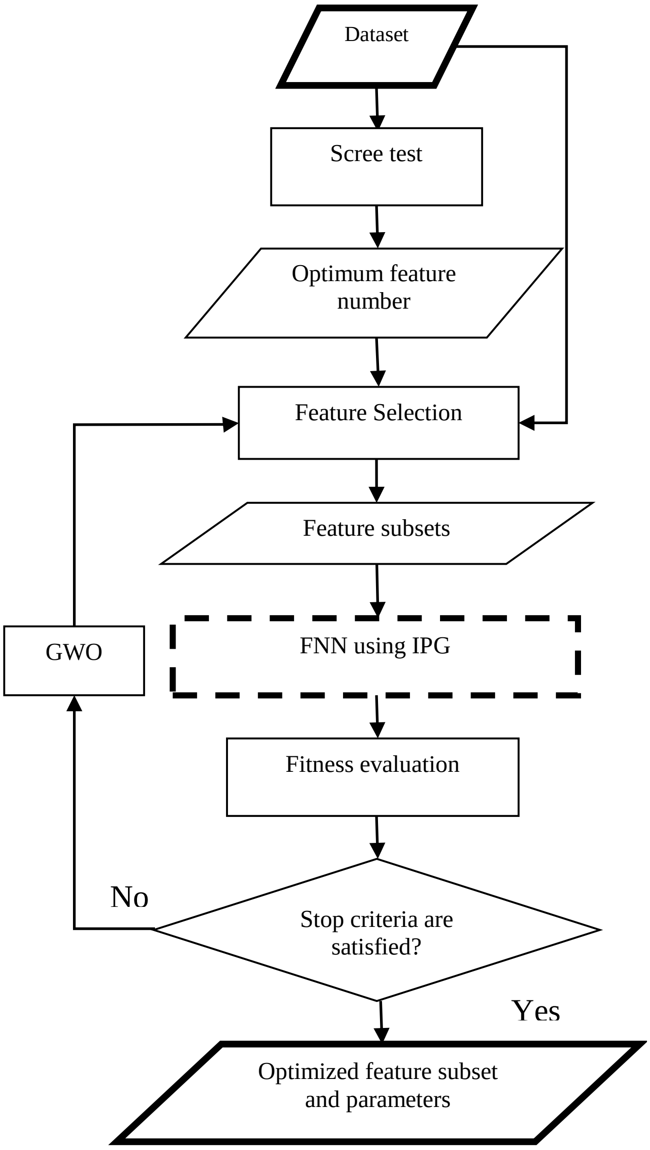

Fig.4 Flowchart of the FS method

The compulsory input of FNN is a dataset containing only selected features that are called feature subset. A feature subset is classified by FNN separate from the other feature subsets. IPG was nominated to train FNN using a group of agents. More to the point, the vector of each agent (constructed using equation (1)) that was used in IPG-FNN referred to one combination of neuron weights and node biases; these vectors were initialized randomly, and then evolved by IPG to reach the best combination. The fitness values of agents are classification accuracies. The output of FNN is the feature subset as well as its classification accuracy. The flowchart of FNN with IPG is demonstrated in Fig. 5.

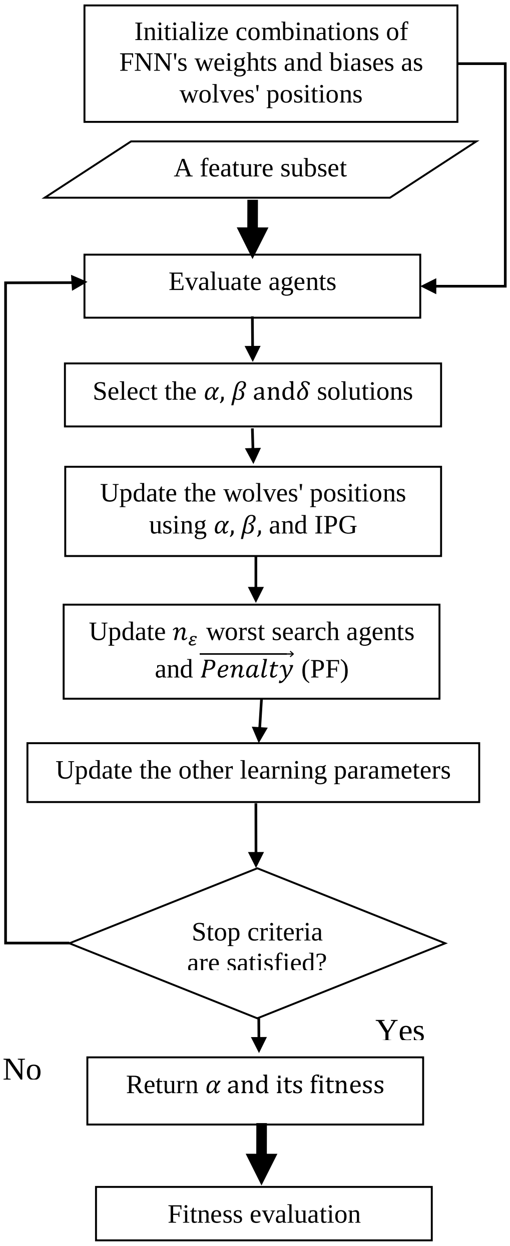

Fig. 5 Flowchart of FNN with IPG (the input is a feature subset, and the output is the feature subset with optimum combination of network's weights and biases).

3.2.4. Fitness Evaluation and Stop Criteria

In this point, all feature subsets were classified by IPG-FNN. Therefore, their fitness evaluation is possible. The classification accuracy is the evaluation criterion. The higher the classification accuracy is, the superior the feature subset will be. These fitness values are used to decide whether the stop criteria are satisfied. The stop criteria are: 1) reaching 100% classification accuracy, or 2) exceeding the maximum allowed iteration. If the stop criteria don't satisfy, the GWO deploys feature subsets and their fitness to improve them.

4. Discussion

First of all, the hardware and software characteristics used for experiments need to be introduced in this section. Then experimental parameters are shown. Next, experimental results are explained to evaluate the accuracy of the proposed method in classification. It must be considered that this study aims to improve the FNN classification accuracy, avoiding local optima stagnation, and convergence speed by modifications of MAs used in FNN training phase. Finally, computational complexity of the proposed MA is compared to its basic MA which is GWO.

The experiments were done on a computer with an Intel Core i5 CPU running Windows at 2.30GHz, with memory 2GB. The software used to run MAs and use scree test were MATLAB R2017 and SPSS16, respectively. Sonar and Radar datasets which are used in this article were obtained from the University of California, Irvin UCI [27], and Table 1 points to their main characteristics.

Table 1 UCI datasets**.**

Datasets	Number of features	Number of classes	Number of instances
Sonar	60	2	208
Radar	32	2	351

4.1. Experimental parameters

Parameters of MAs and FNN need to be set before doing any experiment. Firstly, feature numbers participated in the classification process were calculated. In this article, Scree test is used for feature reduction. Finding the elbow is subjective in this test. The feature numbers used in FS phase are achieved by the component number related to the elbow of Scree plot that they are 4 and 3 features for Sonar and Radar datasets, respectively. Scree graphs are presented in Fig. 6 and 7.

Also, based on [36], Probability Density Function (PDF) of mine and clutter are one-dominant-plus-Rayleigh and k-distribution respectively and according to [1], skewness (3rd central moment) and kurtosis (4th central moment) of data will change, if PDF change. So, skewness and kurtosis of data are used in Sonar dataset as two features, and the other two features have remained that should be chosen randomly or by MAs among 60 features of Sonar dataset.

Additionally, it should be noticed that MAs were used in two phases: FS and FNN training; so, parameter settings related to both phases were prepared in Tables 2 and 3.

4.2. Experimental results

It's good to restate that 2 features were selected in advance in Sonar dataset. The 3rd and 4th moments of features of each sample were calculated and considered as new features.

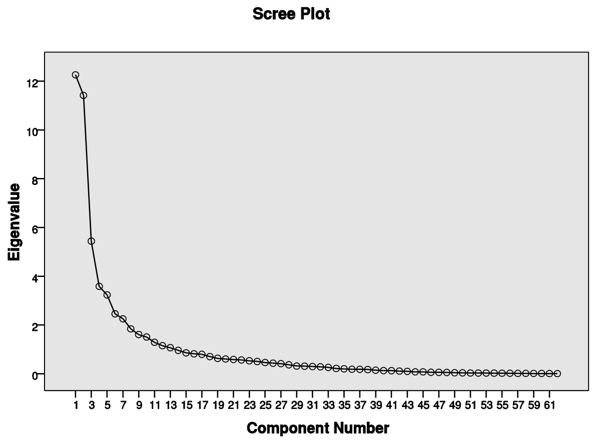

Fig.6 Scree test in Sonar dataset.

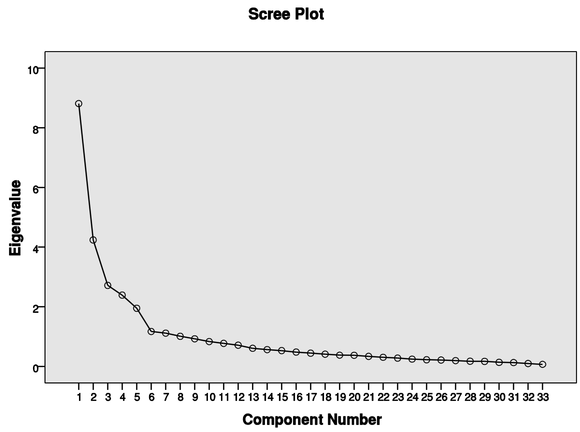

Fig.7 Scree test in Sonar dataset.

Table 2 MAs parameters of FS: (vmin: lower bound of decision variables, vMax: upper bound of decision variables, sizev: matrix size of decision variables, npar = number of particles, maxiter: max iteration, wMax: maximum inertia weight, wmin: minimum inertia weight, c1 and c2: constants used in updating equation of PSO).

Parameters	PSO	GWO
v_{min}	0	
v_{Max}	1	
$size_v$	Scree value	
n_{par}	7	
maxiter	50	
w_{Max}	0.9	NA
w_{min}	0.4	NA
c_1	2	NA
c_2	2	NA

Table 3 Parameters of FNN and MAs used in it: (I: number of input nodes, H: number of hidden nodes, O: number of output nodes, DP: dimensions of particles, NTS: number of training samples, n_p: number of allowed iteration in PF, n_ε: number of punished particles in PF, maxiterf: max iteration in FNN, Nparf = number of particles in FNN)

Parameters	AGPSO	GWO	PG	IG	IPG
I	Number of features				
H	2*(N)+1				
O	1				
D_P	(N+O+1)*H+O				
N_{TS}	Number of entire samples				
w_{Max}	0.9	NA			
w_{min}	0.4	NA			
c_1	$(-2t^3+T^3)+2.5$	NA			
c_2	$(2t^3+T^3)+0.5$	NA			
$\mathbf{n_p}$	NA			10	10
$\mathbf{n_\varepsilon}$	NA			5	5
$maxiter_f$	300				
N_{parf}	50				

The other 2 features needed to be selected using MAs. The proposed method (IPG) was compared to AGPSO (Particle Swarm Optimization with Autonomous Groups) [37] and GWO. AGPSO is a novel enhancement of PSO, and GWO is a state-of-the-art algorithm. One further point, IPG was compared to IG and PG to find out whether the combination of IBM and PA is better than IBM and PA.

Tables 4 and 5 are with regard to the classification accuracy of each method related to each dataset. There is an extra option in the FS method in Sonar dataset: Random FS. Since classification, accuracy is 100% in other options, and MAs' effectiveness cannot be analyzed without Random FS.

Table 4 Average classification accuracy (%) related to Sonar dataset

		FNN training				
		AGPSO	GWO	PG	IG	IPG
FS	Random FS	99.17	98.89	98.94	99.42	99.75
	Without FS	100	100	100	100	100
	PSO	100	100	100	100	100
	GWO	100	100	100	100	100

Table 5 Average classification accuracy (%) related to Radar dataset

		FNN training				
		AGPSO	GWO	PG	IG	IPG
FS	Without FS	77.82	68.77	67.36	79.72	80.71
	PSO	91.01	90.62	91.00	91.03	91.30
	GWO	92.10	91.88	91.93	92.83	92.86

This option assisted in finding the proposed method's advantages. It is completely obvious that the proposed method outperforms the other methods. Among FS methods, GWO is an appreciably better choice than PSO, and PSO is better than the situation that all features were used (without FS option). IPG is the most efficient method for FNN training phase, and it proves that IBM and PA combination works well. IG or PG will result in fairly better classification accuracy than ordinary GWO, and AGPSO is just better than GWO and PG for FNN training. Therefore, using GWO and IPG for FS and FNN training respectively is the best choice to achieve the highest classification accuracy.

However, average classification accuracy may not be a comprehensive evaluation. Since a method may have some answers so far from the average, or an MA may not converges to a suboptimum answer. In addition, the miss rate and precision will be ignored that give valuable information about ignoring and reliability of a positive answer. So, the standard deviation of classification accuracy, precision, miss rate, the convergence curve and a box plot of each method are calculated to reveal comprehensive information about the proposed method's advantages.

Standard deviations of classification accuracy are reported in Table 6 and 7. A low standard deviation indicates that the data points tend to be close to the expected value of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. SD of IPG is overwhelmingly lower than the others in Sonar dataset and the first and second row of Radar dataset. SD of PG is negligibly lower than IPG in the last row of Radar dataset.

Also, there are two other parameters that are really good to evaluate the methods and compare the results: Positive Predictive Value (PPV) that is also called precision and False Negative Rate (FNR) that is called miss rate [38]. PPV and FNR are calculated as follow:

Table 6 Standard deviations of classification accuracy (%) related to Sonar dataset

		FNN training				
		AGPSO	GWO	PG	IG	IPG
FS	Random FS	1.15	1.23	1.43	0.99	0.60
	Without FS	00.00	00.00	00.00	00.00	00.00
	PSO	00.00	00.00	00.00	00.00	00.00
	GWO	00.00	00.00	00.00	00.00	00.00

Table 7 Average classification accuracy (%) related to Radar dataset

		FNN training				
		AGPSO	GWO	PG	IG	IPG
FS	Without FS	22.94	26.07	26.21	24.07	20.51
	PSO	1.52	1.72	1.77	1.60	1.23
	GWO	1.03	1.22	0.83	1.01	0.84

$$\mathrm{PPV} = \frac{\text{True positive}}{(\text{True positive} + \text{False positive})} \tag{14}$$

$$\mathrm{FNR} = \frac{\text{False negative}}{(\text{False negative} + \text{True positive})} \tag{15}$$

High PPV means the probability that the predicted positive is true is high. It shows how much the user can rely on the predicted positive. Low FNR means miss rate is low and the probability that a true positive is ignored is low. The true positive means mine and inappropriate ionosphere in Sonar and Radar dataset, respectively. If a true positive is mislabeled, the consequence will be a huge mistake in these datasets. Fig. 8 plots the PPV and FNR of each method. The highest PPV and the least FNR belongs to IPG in the random FS option of Sonar dataset. Also, the combination of IPG-FNN training and GWO-FS has the highest PPV in the Radar dataset, but the combination of PG-FNN training and GWO-FS has the least FNR in the Radar dataset. In Radar dataset, all FNN training methods have the best PPV and FNR when they use GWO-FS, and the worst PPV and FNR are related to the situation in which FNN training methods are not accompanied by any FS.

Analysis of convergence curve is a popular way to evaluate the behavior of MAs. Fig. 9 represents the convergence curve of each method. The curve of IPG manifests the best convergence in all situations except one. The curve of GWO shows the best convergence in Radar dataset when there is not any FS. This superiority of GWO is marginal because even curves of IG and PG are more convergent than GWO in other situations.

Box plots of MSE are shown in Fig. 10. They display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. The spacing between the different parts of the box indicates the degree of dispersion (spread) and skewness in the data, and show outliers. When it comes to classification of Sonar dataset using PSO-FS, GWO-FS or without FS, the average classification accuracies are 100%, so their box plots are ignored. Two important points can be seen from Fig. 10(a): firstly, the box of IPG is considerably lower and smaller than other boxes and another point is that the first and second quartiles of IPG box are set to zero. It means that MSEs of IPG spread in a significantly smaller amount than the others did and at least half of the MSEs are zero. This situation makes IPG have obviously better classification accuracy. Fig. 10(b) shows superiority of GWO and IG over the other FNN training methods. Fig. 10(c) and 10(d) have the same description as each other. Boxes of IPG are dramatically smaller and lower than other boxes in each Figure that results in higher classification accuracy. Finally, it can result that the proposed method (IPG) has higher classification accuracy, lower standard deviation of accuracy, higher reliability and lower miss rate than the other methods when FNN is used to classify Sonar and Radar datasets. In the second place, IPG converges faster than the other MAs and has the least MSE.

4.2. The Computational Complexity of Proposed Algorithm

O(n,m,…) is defined to analyze the computational complexity in an algorithm. Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. In computer science, big O notation is used to classify algorithms according to how their running time or space requirements grow as the input size grows. Besides, (n,m,…) are the sizes of the inputs. If there are many inputs, the most important ones are used and the others are eliminated. The most important factors in calculation of O are the maximum iteration and recursive functions.

Computational complexity in hunting-based classifiers like GWO depends on the number of the hunters, the number of the variables, maximum iteration, and the procedure of sorting in each iteration. Because the quicksort method is used to sort hunters, the order of the algorithm is O(nlogn) and O(n2) in the worst and the best situations respectively. For these algorithms, computational complexity is calculated as:

$$\begin{aligned} &\mathrm{O(Hunting-based)} \\ &= \mathrm{O(t(O(Quick\ sort) + O(Position\ update))} \\ &= \mathrm{O(t(n^2 + nd)) = O(tn^2 + tnd)} \end{aligned} \tag{16}$$

where n, t, and d are the number of the hunters, maximum iteration, and the number of the variables respectively. Besides, the changes that convert GWO to IPG do not affect Eq. (16).

5. Conclusion

This research proposes an enhanced MA based on GWO for FNN training in a classification task. FS phase is added to lower time costs and make classification accuracies higher. FS of the current research uses an MA, too. Experimental results corroborated the fact that IPG has dominant superiority over IG, PG, GWO, and AGPSO. In addition, using FS makes the classification more accurate with fewer features in comparison with the classification without FS. Finally, the proposed method can assist ships and submarines to avoid accidents and help scientists to be aware of the amount of ionosphere degradation.

Therefore, if a user needs to work with Sonar and Radar datasets, these should be considered:1) the combination of IBM and PA has a better result than IBM or PA, 2) IBM and PF have positive effects on GWO, and 3) AGPSO is better than GWO. Further, GWO is a better choice than PSO for FS, and using PSO is better than the situation that there is not any FS.

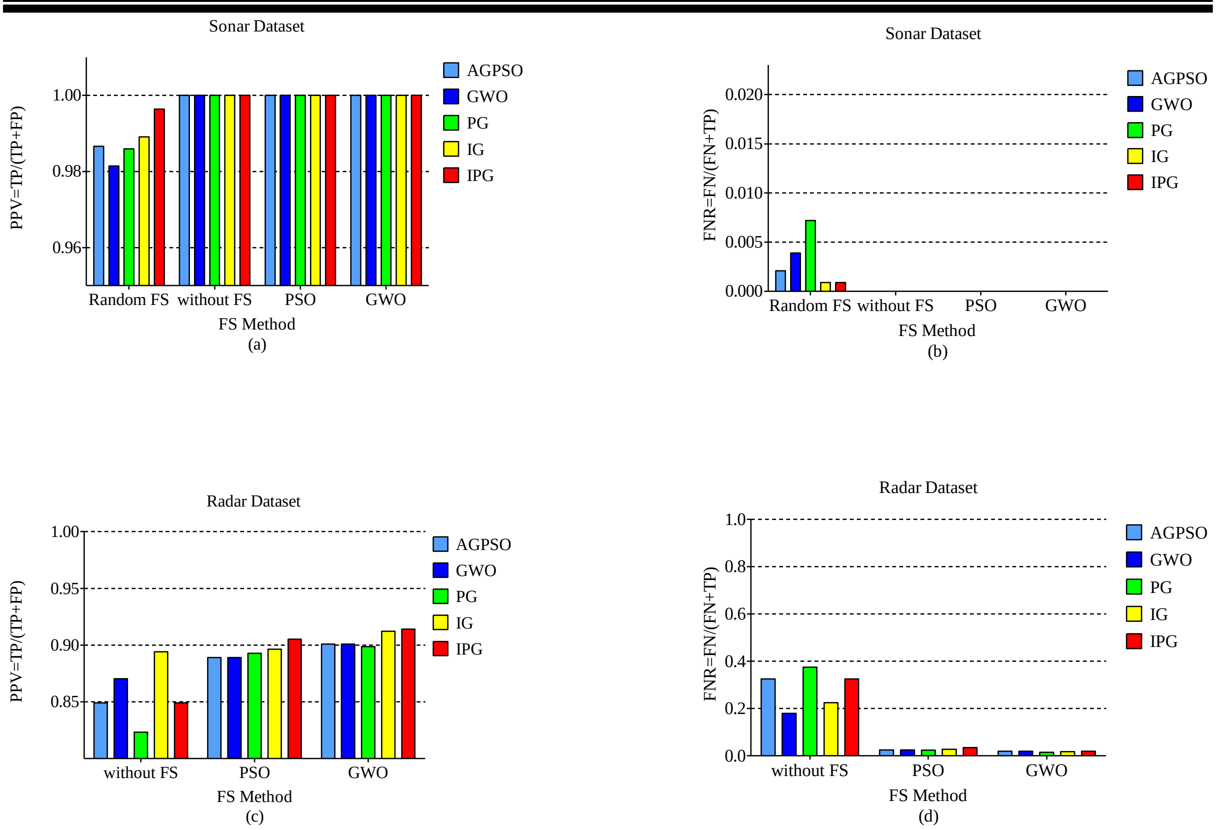

Fig.8 a) Precisions in Sonar dataset, b) miss rates in Sonar dataset, c) precisions in Radar dataset, and d) miss rate in Radar dataset.

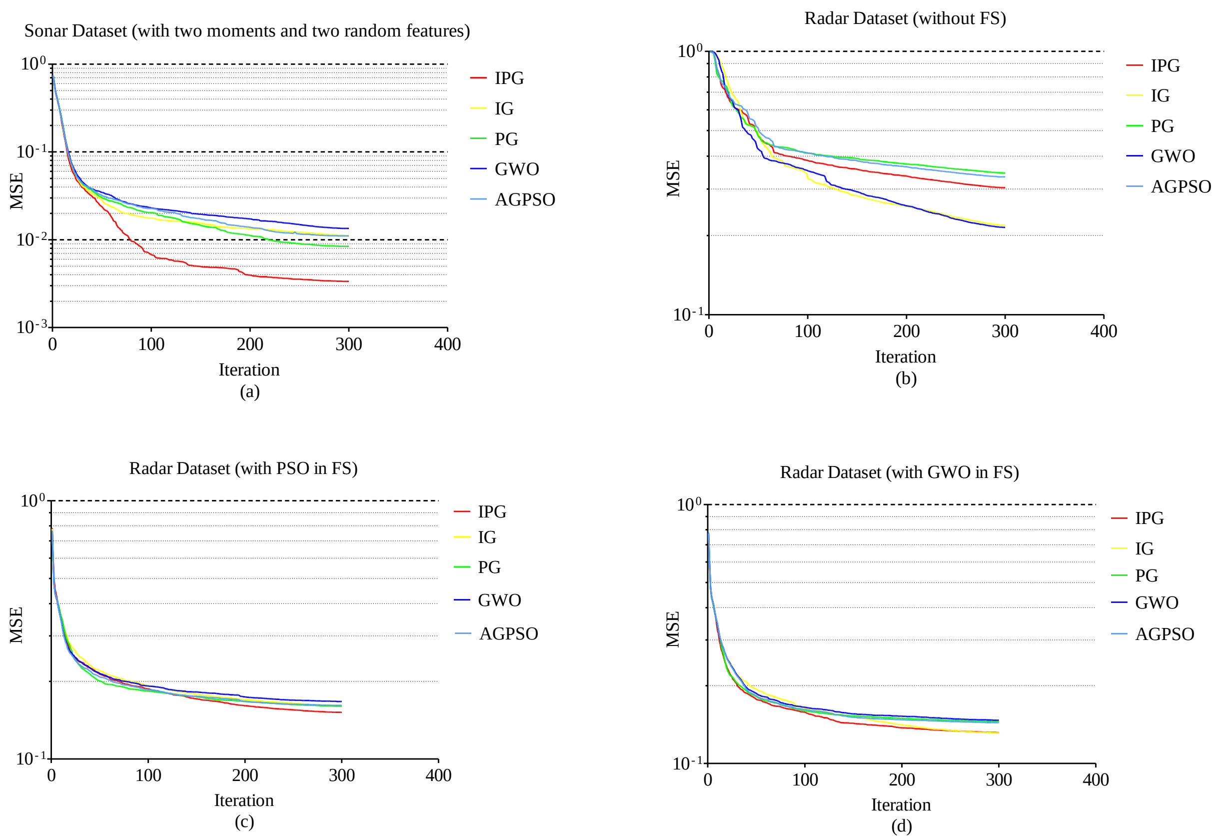

Fig.9 Convergence curves of MAs in a) Sonar dataset (random FS), b) Radar dataset (without FS), c) Radar dataset (PSO in FS phase), and d) Radar dataset (GWO in FS phase).

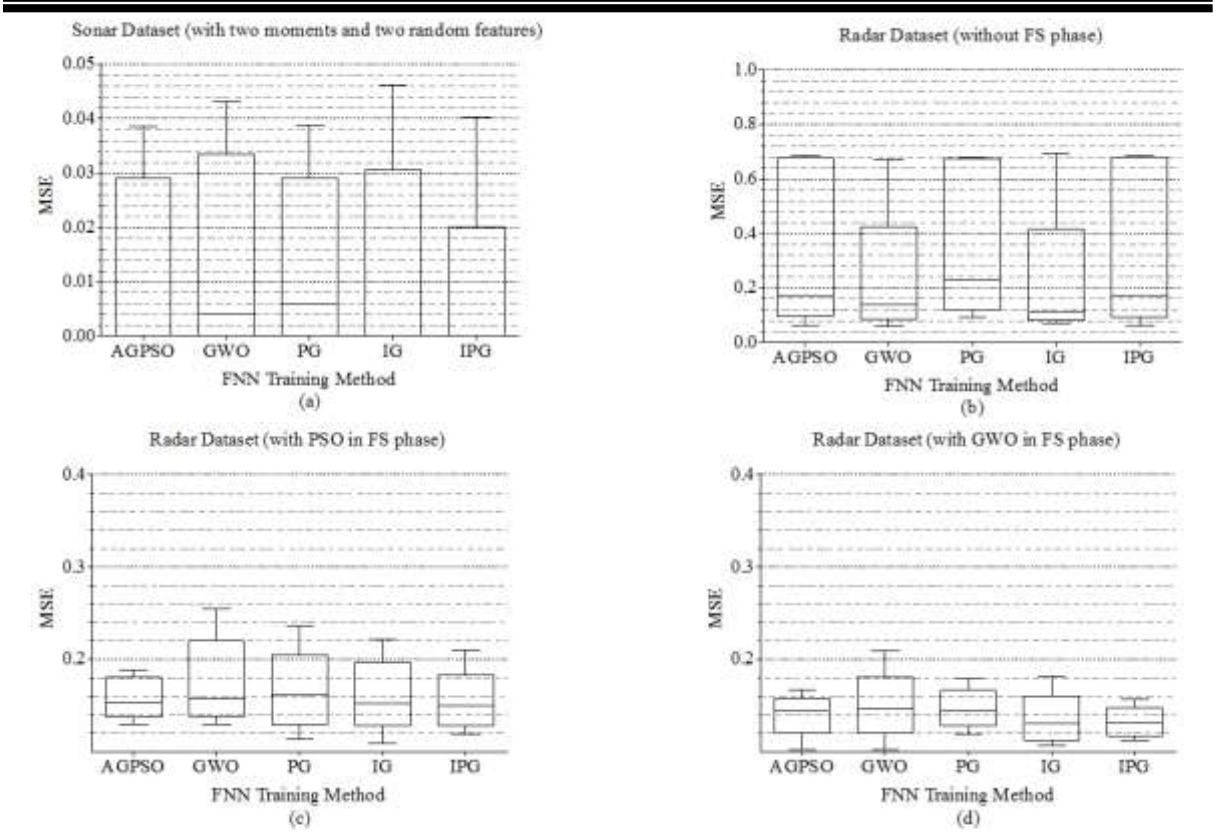

Fig.10 MSE box plots a)Sonar dataset, b) Radar dataset (without EA in FS phase), c) Radar dataset (with PSO in FS phase), and d) Radar dataset (with GWO in FS phase).

6. Reference

[1] M. R. Mosavi, M. Khishe, and A. Moridi, “Classification of Sonar Target using Hybrid Particle Swarm and Gravitational Search,” Iraninan Journal of Marine Technology, Vol.3, No.1, pp.13, 2016.

[2] H. Opena, and J. Yusiong, “Automated Tomato Maturity Grading using ABC-Trained Artificial Neural Networks”, Malaysian Journal of Computer Science, Vol.30,No.1, pp.12-26, 2017.

[3] M. Khishe, M. R. Mosavi, and M. Kaveh, “Improved Migration Models of Biogeography-based Optimization for Sonar Dataset Classification by using Neural Network,” Applied Acoustics, Vol.118, pp.15-29, 2017.

[4] A. Heidari, H. Faris, I. Aljarah, and S. Mirjalili, “An Efficient Hybrid Multilayer Perceptron Neural Network with Grasshopper Optimization,” Soft Computing, pp.18, 2018.

[5] N. SadatJaddi, S. Abdullah, and A. Hamdan, “Multi-population Cooperative Bat Algorithm-based Optimization of Artificial Neural Network Model,” Information Sciences, Vol.294, pp.19, 2015.

[6] I. Aljarah, H. Faris, and S. Mirjalili, “Optimizing Connection Weights in Neural Networks using the Whale Optimization Algorithm,” Soft Computing, Vol.22, No.1, pp.15, 2018.

[7] M. Mosavi, M. Khishe, and A. Ghamgosar, “Classification of Sonar Data Set using Neural Network Trained by Gray Wolf Optimization,” N. Network World, Vol.26, No.4, pp.23, 2016.

[8] K. C. Lin, and Y. H. Hsieh, “Classification of Medical Datasets using SVMs with Hybrid Evolutionary Algorithms based on Endocrine-based Particle Swarm Optimization and Artificial Bee Colony Algorithms,” Journal of Medical Systems, Vol.39, No.119, pp.9, 2015.

[9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” Advances in Engineering Software, Vol.69, pp.24, 2015.

[10] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, “Grey Wolf Optimizer: a Review of Recent Variants and Applications,” Neural Computing and Applications, Vol.30, No.2, pp.23, 2017.

[11] S. Eswaramoorthy, N. Sivakumaran, and S. Sekaran, “Grey Wolf Optimization based Parameter Selection for Support Vector Machines,” The International J. for Computation and Mathematics in Electrical and Electronic Engineering, Vol.35, No.5, pp.11, 2016.

[12] E. Elhariri, N. El-Bendary, A. Hassanien, and A. Abraham, “Grey Wolf Optimization for One-Against-One Multi-class Support Vector Machines,” in 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp. 7–12, 2015

[13] V. Kumar, J. Chhabra, and D. Kumar, “Grey Wolf Algorithm-based Clustering Technique,”

Journal of Intelligent Systems, Vol.26, No.1, pp.168, 2017.

[14] H. Yang , and J. Liu, "A Hybrid Clustering Algorithm based on Grey Wolf Optimizer and K-Means Algorithm," Journal of Jiangxi University of Science and Technology, Vol.5, pp.015, 2015.

[15] E. Emary, H. Zawbaa, and A. Hassanien, "Binary Grey Wolf Optimization Approaches for Feature Selection," Neurocomputing, Vol.172, pp.11, 2016.

[16] E. Emary, H. Zawbaa, C. Grosan, and A. Hassenian, "Feature Subset Selection Approach by Gray-Wolf Optimization," in Afro-European Conference for Industrial Advancement, , pp.13, 2015

[17] V. K. Kamboj, "A Novel Hybrid PSO-GWO Approach for unit Commitment Problem," Neural Computing and Applications, Vol.27, No.6, pp.1655, 2016.

[18] V. K. Kamboj, S. K. Bath, and J. S. Dhillon, "Solution of Non-Convex Economic Load Dispatch Problem using Grey Wolf Optimizer," Neural Computing and Application, Vol.27, No.5, pp.1316, 2016.

[19] M. R. Shakarami, and I. F. Davoudkhani, "Wide-Area Power System Stabilizer Design based on Grey Wolf Optimization Algorithm Considering the Time Delay," Electric Power System Research, Vol.133, pp.159, 2016.

[20] W. Long, J. Jiao, X. Liang, and M. Tang, "An Exploration-Enhanced Grey Wolf Optimizer to Solve High-Dimensional Numerical Optimization," Engineering Applications of Artificial Intelligence, Vol.63, pp.80, 2018.

[21] L. Rodrı´guez, O. Castillo, and J. Soria, "Grey Wolf Optimizer with Dynamic Adaptation of Parameters using Fuzzy Logic," in 2016 IEEE Congress on Evolutionary Computation (CEC), , pp.3116–3123, 2016

[22] S. Saremi, S. Mirjalili, and S. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," Neural Computing and Applications, Vol.26, No.5, pp.7, 2015.

[23] K. Dudani, and A. Chudasama, "Partial Discharge Detection in Transformer using Adaptive Grey Wolf Optimizer based Acoustic Emission Technique," Cogent Engineering, Vol.3, No.1, pp.1256083, 2016.

[24] J. Han, and M. Kamber, Data Mining (Concepts and Techniques), Seond ed., p.285: Morgan Kaufman, 2006.

[25] M. Khishe, M. R. Mosavi, and A. Moridi, "Chaotic Fractal Walk Trainer for Sonar Data Set Classification using Multi-Layer Perceptron Neural Network and its Hardware Implementation," Applied Acoustics, Vol.137, pp.15, 2018.

[26] S. J. Anderson, and Y. I. Abramovich, "Detection, Classification and Correction of Ionospheric Distortion in HF Skywave (OTH) Radar Systems", IEEE, pp. 280-284, 1996.

[27] D. Dua, and E. Karra Taniskidou, "UCI Machine Learning Repository", 2017.

[28] M. Bildirici, and O. Ersin, "Forecasting Volatility in Oil Prices with a Class of Nonlinear Volatility Models: Smooth Transition RBF and MLP Neural Networks Augmented GARCH Approach," Petroleum Science, Vol.12, No.3, pp.19, 2015.

[29] H. Boughrara, M. Chtourou, C. Ben Amar, and L. Chen, "Facial Expression Recognition based on a Mlp Neural Network using Constructive Training Algorithm," Multimedia Tools and Applications, Vol.75, No.2, pp.23, 2014.

[30] Y. Chand, M. A. Alam, and Y. R. S. N. Tejaswini, "Performance Comparison of Artificial Neural Networks Learning Algorithms and Activation Functions in Predicting Severity of Autism," Network Modeling Analysis in Health Informatics and Bioinformatics, Vol.4, No.1, pp.23, 2015.

[31] H. Chen, S. Grant-Muller, L. Mussone, and F. Montgomery, "A Study of Hybrid Neural Network Approaches and the Effects of Missing Data on Traffic Forecasting," Neural Computing & Applications, Vol.10, No.3, pp.10, 2001.

[32] M. A. Ghorbani, H. Ahmadzadeh, M. Isazadeh, and O. Terzi, "A Comparative Study of Artificial Neural Network (MLP, RBF) and Support Vector Machine Models for River Flow Prediction," Environmental Earth Sciences, Vol.75, No.476, pp.14, 2016.

[33] N. Mittal, U. Singh, and B. S. Sohi, "Modified Grey Wolf Ooptimizer for Global Engineering Optimization," Applied Computational Intelligence and Soft Computing, Vol.8, 2016.

[34] R. Cattell, "The Scree Test for the Number of Factors," Multivariate Behavioral Research, Vol.1, No.2, pp.32, 1966.

[35] S. Mirjalili, Hybrid Particle Swarm Optimization and Gravitational Search Algorithm for Multilayer Perceptron Learning: Universitiy Teknologi Malaysia, 2011.

[36] C. A. M. Moll, M. A. Ainslie, F. P. A. Lam, and J. Janmaat, "The Most Likely Distribution of Target Echo Amplitudes," in 2nd International Conference & Exhibition on Underwater Acoustic Measurements: Technologies & Results, pp. 3, 2010

[37] M. R. Mosavi, and M. Khishe, "Training a Feed Forward Neural Network using Particle Swarm Optimizer with Autonomous Groups for Sonar Target Classifcation," J. of Circuits, Systems, and Computers, Vol.26, No.11, pp.20, 2017.

[38] D. M. W. Powers, "Evaluation: From Precision, Recall and F-factor to ROC, Informedness, Markedness & Correlation," International Journal of Machine Learning Technology, Vol.2, No.1, pp.27, 2011.