Sonar Data Classification using Neural Network Trained by Hybrid Dragonfly and Chimp Optimization Algorithms

F. Mousavipour
Iran University of Science and Technology

Mohammad Reza Mosavi (m_mosavi@iust.ac.ir)
Iran University of Science and Technology https://orcid.org/0000-0002-2389-644X

Research Article

Keywords: Classifier, MLP NN, Hybrid Dragonfly Trainer, Sonar

Posted Date: June 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1673592/v2

Sonar Data Classification using Neural Network Trained by Hybrid Dragonfly and Chimp Optimization Algorithms

F. Mousavipour[1] and M. R. Mosavi[2*]
[1,2] Department of Electrical Engineering, Iran University of Science and Technology, Narmak, Tehran 13114-16846, Iran
[1] Ph.D. Student, F_Mousavipour@elec.iust.ac.ir
[2*] Professor, Corresponding Author, m_mosavi@iust.ac.ir

Abstract: This paper proposes a hybrid Dragonfly Algorithm (DA) for training Multi-Layer Perceptron Neural Network (MLP NN) to design the classifier for solving complicated problems and distinguishing the real target from liars' targets in sonar applications. Due to improving the cost computation and reducing the waste of time, a modified low-cost DA is designed for evaluation. To assess the accuracy of the technique, some well-known meta-heuristic trainers include Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), DA, and Chimp Optimization Algorithm (ChoA) compared to show the accuracy of similar algorithms. DA and ChoA algorithms have remarkable features and a hybrid algorithm of them is proposed. The performance of the proposed classifier will be evaluated by two standard benchmark datasets. The results show that the modified hybrid DA-ChoA has 15% less time consuming and 4% better performance rather than the original dragonfly method.

Keywords: Classifier, MLP NN, Hybrid Dragonfly Trainer, Sonar.

1. Introduction

Classification of underwater targets has received much attention in recent years, including distinguishing between the real target and liar target objects such as fish tribes or background clutter. This procedure is one of the most complex and challenging in this research field because of its variant characteristics [1-3]. In underwater environments, some different echo classes exist. Noises, reverberation, and clutter are some classes of underwater echoes [4]. Classification techniques such as statistical processing [5], signal processing [6,7], and feature extraction algorithms [8] were classical schemes. In recent years, Neural Networks (NNs) are considered for their prominent specifications [9] such as a high degree of precision, adaptability, an inherently parallel structure which is very appropriate in hardware implementation and real-time processing.

Multi-Layer Perceptron (MLP) NNs are the most significant tools for data classification [10]. Learning is the main and important feature of this tool that without this essential block, algorithm cannot improve primary results. From a technical manner, the method that provides learning for NNs is called trainer [11]. Trainers can be divided into two main types: supervised and unsupervised learning [12]. Classic supervised methods such as gradient descent and Newton have poor quality and disadvantages such as local optima stagnation and the need to derivate the search space [13].

Meta-heuristic algorithms are readily applicable to problems in different fields especially trainers for NNs in high complex problems [14-16]. There are different categories for meta-heuristic methods. Two main categories are based on the inspiration of an algorithm and the number of generated solutions in each step of the algorithm. The former methods such as swarm intelligence base [17], physical-based [18], evolutionary-based [19], and human-related. The last classification divides the algorithms into two classes: individual-based and population-based algorithms. In individual classes, one solution generates randomly and improves over the iterations. The latter class, trainer algorithm generates a multitude of solutions randomly and improves them during the procedure.

There is a basic question in the optimization domain as if and why we need more optimization techniques. By the No Free Lunch (NFL) theorem [20] the answer is found. NFL proves that no one can propose a method for solving all optimization problems. This means that no special optimization algorithm is well-fitted to overcome all optimization challenges. In other words, almost the optimization techniques perform equally on average when intending all optimization challenges despite the superior performance on a special optimization problem. This doctrine allows researchers to improve and modify existing algorithms for problems in different fields. The theoretic researches in this context can be divided into three basic directions: introducing the existing methods, mixing and hybridizing different algorithms, and developing new algorithms. In the first direction, different mathematical or stochastic operators have been used to augment the performance such as chaotic maps [21,22], evolutionary operators [23-25], and local searches [26,27]. In the second direction, different algorithms have been mixed and made hybrid algorithms to overcome some shortages or bottlenecks. Some hybrid meta-heuristic techniques are mentioned in the literature such as PSO-ACO [28], PSO-DE [29], and ACO-DE [30]. Latter direction, the proposal of new algorithms is a popular research domain for many scientists. Inspiration of a new method can be from physical rules, evolutionary events, and cooperative behavior of living things.

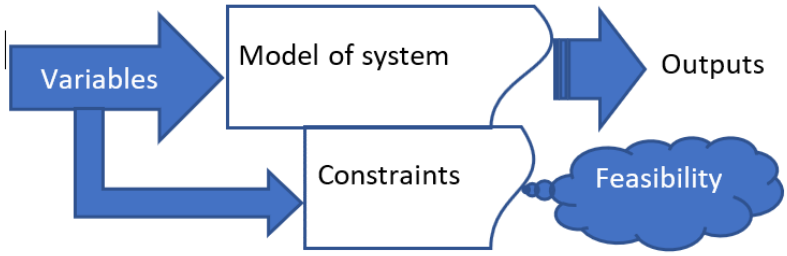

Fig.1: Main elements of an optimization system.

The elements involved in the optimization process are parameters and constraints. Parameters are the unknown variables of the system that have to be optimized. As shown in Fig. 1, variables are primary inputs and constraints are secondary inputs addressed from variables. The feasibility of the obtained results is defined by constraints.
The paper is organized as follows. Section 2, discusses the closely related works. Section 3, describes MLP NN training algorithms. Section 4, describes an overview of the standard Dragonfly Algorithm (DA) and Chimp Optimization Algorithm (ChoA) that, how to use it for training a MLP NN. In Section 5, sonar datasets and their simulations are presented. Finally, conclusions are presented in Section 6.

2. Closely Related Works

Population-based meta-heuristic optimization techniques share a common feature despite their nature. The search space is divided into two discrete phases: “exploration” and “exploitation” [31,32]. The algorithm must have operators that globally search and explore the whole of the search space. Exploration is the ability of an optimizer to have extremely random behavior changing the solutions. If solutions have big changes, the consequent exploration ability will be greater. The exploitation phase follows the exploration phase and expressed as the phase of detailed investigation of the promising areas of search space.
There must be no clear border between exploitation and exploration phases. Finding an appropriate balance between exploitation and exploration is the main challenge in a meta-

heuristic optimization simultaneously. These two phases are correlative that augmenting one phase, in castrating another phase.
DA algorithm is a powerful meta-heuristic optimizer introduced in recent years. This algorithm has two main purposes: hunting and migration. The hunting procedure is called the static swarm and the migration procedure is called dynamic swarm. DA algorithm mimics the five primitive principles of the swarming behavior of dragonflies. Separation behavior was applied to avoid collisions between insects in the tribe. Alignment is a tool for matching the velocity of individuals rather than other swarms. Cohesion refers to the tendency of individuals toward the average neighborhood. Attraction means dragonflies move towards the food sources and for survival distract outward enemies. In [33], DA algorithm has investigated the effect of the original updating mechanisms of the coefficients on exploration and exploitation phases. After that, three modified updating techniques are proposed for coefficients of the algorithm. These modifications improve results slightly, but increase the complexity of the algorithm remarkably.
In [34], the great number of DA algorithm variants are mentioned. These variants include modified DA, hybridization DA, and multi-objective DA. Merits and disadvantages of each variant are analyzed and some applications of these variants are mentioned. Classification of sonar targets in real environments is needed low cost and accurate algorithm. Original DA is complex and has medium accuracy rather than some other algorithm such as MFO. For improving the performance of algorithm, the hybridization technique can be used, but with consideration of low complexity rather than original DA. The literature indicates that combining the chaotic local search with meta-heuristic algorithm is one of the low computational-cost methods for improving both exploration and exploitation phases. The motivation of this research is utilizing chaotic local search to improve the performance of the DA optimizer. Chaotic local search has stochastic behavior also is a deterministic method. To have a real-time procedure, considering the parallel structure of the DA optimizer and MLP NN.

3. Multi-Layer Perceptron Neural Network

MLP NNs are the most practical and common type of NNs. Each neuron is interconnected to other neurons in mono-directional mode. Fig. 2 shows a MLP NN schematic with two layers and *p* input nodes, *h* neurons in the only hidden layer and *m* neurons in the output layer.

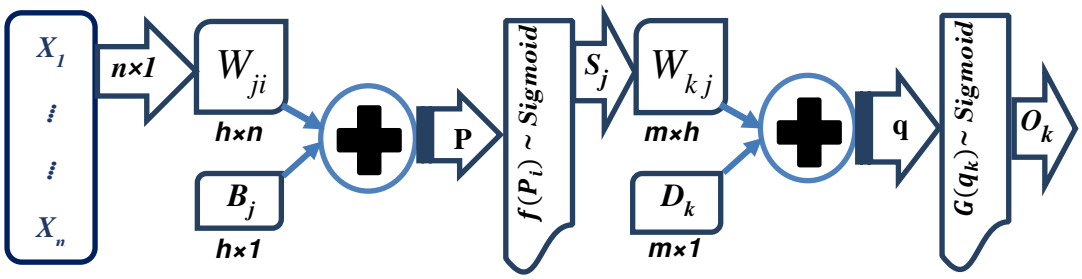

Fig.2: MLP NN with one hidden layer.

In this MLP NN, two layers handle the procedure that the *j*th input of *P* (hidden layer) and its outputs is shown as follows:

$$P_j = \sum_{i=1}^{n} x_i * w_{ji} + b_j \quad j = 1,2, \dots, h. \tag{1}$$

where x_i is the *i*th input, w_{ji} represents the weight from the *i*th input to the *j*th neuron and b_j is the bias of the *j*th neuron.

$$S_j = F(P_j) \qquad j = 1,2,\dots,h. \tag{2}$$

F is the activation function that calculate the output of hidden layer's neurons in Eq. (2).

$$q_k = \sum_{j=1}^{h} S_j * w_{kj} + D_k \quad k = 1,2,\dots,m \tag{3}$$

where w_{mj} is the synaptic weight relating the output of the *j*th neuron in the hidden layer to the *m*th output layer neuron and D_m is the bias of the *m*th jth neuron output neuron.

$$O_k = G(q_k) \qquad k = 1,2,\dots,m. \tag{4}$$

G function, calculates the final outputs of MLP NN in Eq. (4). Connection weights and biases of each neuron are the most important segments of the MLP NN. In sonar target classification, real target presented by 1 and liar target by zero. For error evaluation, use Eq. (5) for minimizing the error.

$$E_k = \sum_{i=1}^{m} \left(O_i^k - d_i^k\right)^2 \tag{5}$$

To find the optimum values of weights and biases to achieve the best fitness (minimum error), the teaching algorithm such as gradient descent and Newton is applied.

$$w_i(n+1) = w_i(n) - \eta \frac{\partial E_k(n)}{\partial w_i(n)} \tag{6}$$

The training process of updating weights next time is shown in Eq. (6). η is the learning coefficient of $w_i(n+1)$ and n is the current time step.
In most cases, three methods represent the configuration of biases and weights of the MLP NN as a particle of meta-heuristic trainers: (1) vector, (2) matrix, and (3) binary state. In vector trainer, each particle presents only one vector. In matrix presentation, a matrix presents each particle and in binary type, each particle is expressed in the form of a string of bits. In this research, vector trainer is used because of its simplicity [35-38].
Sejnowski [39] and Iris [40] datasets are used in this research. Sejnowski classified the sonar signals using a NN. The task is to train a network to distinguish sonar signals of a metal cylinder from sandy rocks. The transmitted sonar signal is a frequency-modulated chirp. The Sejnowski dataset contains 97 patterns obtained from rocks under similar conditions. 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions. These (i.e., 208) signals were obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. Each pattern is a set of 60 numbers normalized to 1. Each number represents the energy within a particular frequency band, integrated over an absolute period of time.
This means the input number of MLP NN is equal to dimensions of problem (i.e., 60). Output neurons are two that present real and liar targets. Iris dataset has 4 dimensions and 3 output neurons including real target, liar target, and clutter. In Sejnowski, the dataset number of inputs are too many that increase the complexity of the problem for selecting the number of hidden neurons of MLP NN. For determining hidden layer numbers, some theoretical and empirical methods are considered.

Table 1: References and proposed formulations for estimating the number of nerves of hidden layers.

Proposed Ref.	Formulation
[41]	$N_h = N_i - 1$
[42]	$N_h = \sqrt{N_i N_o}$
[43]	$N_h = log(N_i - 1) - N_o$
[44]	$N_h = \frac{4N_i^2 + 3}{N_i^2 - 8}$

In Table 1, some papers and proposed formulations are presented. Ref [42] is used for selecting number of hidden neurons equal to 11.

4. Trainer Algorithms

This section is intended to provide necessary information to DA and ChoA trainers which are used for designing the classifier.

4.1. Dragonfly Algorithm

As mentioned before, DA is a novel population-based meta-heuristic optimization algorithm in which their search agents have two behaviors [45]. Hunting and migration are the main phases of this algorithm. In hunting (static swarm), local movement and unexpected changes in the flying path are the main characters. In migration (dynamic swarm), a large group of dragonflies make the swarm for transferring to long distances. Static and dynamic swarm is very similar to the main phases of meta-heuristic algorithms: exploration and exploitation.
Separation, alignment, and cohesion addition food, and enemy are the main operators of algorithm. These five main operators, adjust the position updating of individuals in the swarm. Each of these operators is mathematically modelled as follows in Table 2.

Table 2: DA operators and formulations.

Operator	Formula
Separation	$S_i = -\sum_{j=1}^{N} X - X_j$
Alignment	$A_i = \frac{\sum_{j=1}^{N} \Delta x_j}{N}$
Cohesion	$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X$
Food	$F_i = X^+ - X$
Enemy	$E_i = X^- + X$

Where X is the position of the current individual, X_j shows the position jth neighboring, and N is the number of population. Δx_j is the step of jth neighboring individual similar in PSO algorithm and X^+ shows the position of the food source. The position of the enemy is X^-. All the five operators update in each time iteration and then make the next position of each individual by position and step vectors. Equation (7) represents the step vector.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \tag{7}$$

All the lower-case alphabet such as: *s, a, c* are weights of basic operators and calculated randomly. The position vectors are calculated as follows:

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{8}$$

In static swarm, alignment is very low and coherence to food is very high. If alignment is very high and coherence is low, the exploitation phase is operated. For transiting the exploration phase to exploitation, the radius of neighbors should be increased by iterations. In simulation Eq. (9) is used for calculating the initial radius of neighbors.

$$r_d = \frac{UB_d - LB_d}{10} \tag{9}$$

UB and *LB* are upper band and lower band of search space in dimension of *d*. Convergence of algorithm is guaranteed if weights of DA changed adaptively for transiting from exploration to exploitation of the search space.

4.2. Chimp Optimization Algorithm

ChoA [46] inspired by the individual intelligence and sexual motivation of chimps in their group haunting. Some animal tribes such as chimps are fission-fusion colonies. The size of such colony changes as time passes and members moved throughout the environment. Each tribe of the chimp attempts to explore the search space with exclusively and independent strategy. In each group, chimps have not uniform ability and alertness. The ability of each chimp can be useful in special situations.

There are four types of chimps in a tribe: attacker, chaser, barrier, and driver. They have not equal abilities but can do hunting successfully together. Drivers scare the prey and follow it. Barriers place themselves in a tree to prevent prey moving back. Chaser is nimble and catches the prey. Finally, attackers hunt the prey. The prey hunted during the exploration and exploitation phases. For modelling the behavior of chimp hunting, Eq. (10) and Eq. (11) are proposed.

$$\boldsymbol{d} = \left|\boldsymbol{c}.\boldsymbol{X}_{prey}(t) - \boldsymbol{m}.\boldsymbol{X}_{chimp}(t)\right| \tag{10}$$

$$\boldsymbol{X}_{chimp}(t+1) = \boldsymbol{X}_{prey}(t) - \boldsymbol{a}.\boldsymbol{d} \tag{11}$$

Where *t* is the number of current iteration, $\boldsymbol{a}$, $\boldsymbol{m}$, and $\boldsymbol{c}$ are coefficient vectors, $\boldsymbol{X}_{prey}$ and $\boldsymbol{X}_{chimp}$ are the position of prey and chimp respectively. Equations (12) to (14) calculate the $\boldsymbol{a}$, $\boldsymbol{m}$, and $\boldsymbol{c}$ vectors.

$$\boldsymbol{a} = 2.\boldsymbol{f}.\boldsymbol{r}_1 - \boldsymbol{f} \tag{12}$$
$$\boldsymbol{c} = \boldsymbol{2}.\boldsymbol{r}_2 \tag{13}$$
$$\boldsymbol{m} = \boldsymbol{chaotic\_value} \tag{14}$$

Where $\boldsymbol{f}$ is reduced non-linearly from 2.5 to 0 through the iteration process. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors and chaotic values calculated based on the various chaotic map. This operator shows the sexual motivation of chimps in hunting.

The first attacker is the best solution in meta-heuristic problems. Driver, barrier, and chaser are better informed about the position of prey. Four of the best solutions are saved in memory and others try to follow these four solutions. This relationship is expressed by Eq.s (15) to (17).

$$\begin{aligned} &\boldsymbol{d}_{Attacker} = |\boldsymbol{C}_1.\boldsymbol{X}_{Attacker} - \boldsymbol{m}_1.\boldsymbol{X}|, \quad \boldsymbol{d}_{Barrier} = |\boldsymbol{C}_2.\boldsymbol{X}_{Barrier} - \boldsymbol{m}_2.\boldsymbol{X}|, \\ &\boldsymbol{d}_{Chaser} = |\boldsymbol{C}_3.\boldsymbol{X}_{Chaser} - \boldsymbol{m}_3.\boldsymbol{X}|, \quad \boldsymbol{d}_{Driver} = |\boldsymbol{C}_4.\boldsymbol{X}_{Driver} - \boldsymbol{m}_4.\boldsymbol{X}|. \end{aligned} \tag{15}$$

$$\boldsymbol{X_1} = \boldsymbol{X}_{Attacker} - \boldsymbol{a_1}(\boldsymbol{d}_{Attacker}), \quad \boldsymbol{X_2} = \boldsymbol{X}_{Barrier} - \boldsymbol{a_2}(\boldsymbol{d}_{Barrier}),$$
$$\boldsymbol{X_3} = \boldsymbol{X}_{Chaser} - \boldsymbol{a_3}(\boldsymbol{d}_{Chaser}), \quad \boldsymbol{X_4} = \boldsymbol{X}_{Driver} - \boldsymbol{a_4}(\boldsymbol{d}_{Driver}). \tag{16}$$

$$\boldsymbol{X}(t+1) = \frac{\boldsymbol{X_1}+\boldsymbol{X_2}+\boldsymbol{X_2}+\boldsymbol{X_3}+\boldsymbol{X_4}}{4} \tag{17}$$

Chaotic behavior in final stage of algorithm helps chimps to overcome two basic meta-heuristic problems: entrapping in local optima and low convergence rate for solving complex and high-dimensional problems. To model two behaviors simultaneously (normal updating or chaotic model) mathematical model is expressed by Eq. (18):

$$\boldsymbol{X}_{Chimp}\ (t+1) = \begin{cases} \boldsymbol{X}_{prey}(t) - \boldsymbol{a}.\boldsymbol{d} & if \quad \mu < 0.5 \\ \boldsymbol{Chaotic\_Value} & if \quad \mu > 0.5 \end{cases} \tag{18}$$

In sonar target classification, the sinusoidal map has the best performance in low-dimensional problems.

4.3. Hybrid DA-ChoA

DA and ChoA algorithms are newly developed and have good performance as classification rate and convergence. They use complex operators that need more processing power and memory rather than some old meta-heuristic algorithms such as PSO and GA. ChoA has strong local search ability under the guidance of four main chimps, and its convergence rate is fast. The DA owns excellent global search capability because of its procedure of being distracted by outward enemies and attracted towards food sources, but its convergence rate is lower and has high potential trapping in local optima. The hybrid algorithm combined with two excellent algorithms can avoid the disadvantages of each trainer in the process of optimization and enhance the exploration and exploitation of the algorithm simultaneously. For better performance, some novel constraints are applied to hybrid algorithm to guarantee the feasibility of solutions. By using flow chart, the operation mechanism of the proposed hybrid DA-ChoA trainer is shown in Fig. 3.

Three rules are assigned to the hybrid algorithm for avoiding fluctuation in convergence cure. Rules are described in Table 3.

Table 3: Rules for hybrid DA_ChoA flow chart.

Rule #	description
Rule 1	If best local map score is degraded rather than the last score, repeat the score and replace DA's last best position.
Rule 2	If best ChoA score is degraded rather than last score, repeat score and replace DA with the last best position.
Rule 3	If beat DA score is degraded rather than last score, repeat score and replace ChoA's last position.

In hybrid DA-ChoA setting of k_{max} equal 6 and DA has high level priority to run. DA algorithm quests entire search space in the optimization process and generate the best global solutions for the next time iterations. If DA cannot update the optimal solutions for $0.5 * k_{max}$ times, ChoA algorithm is executed.

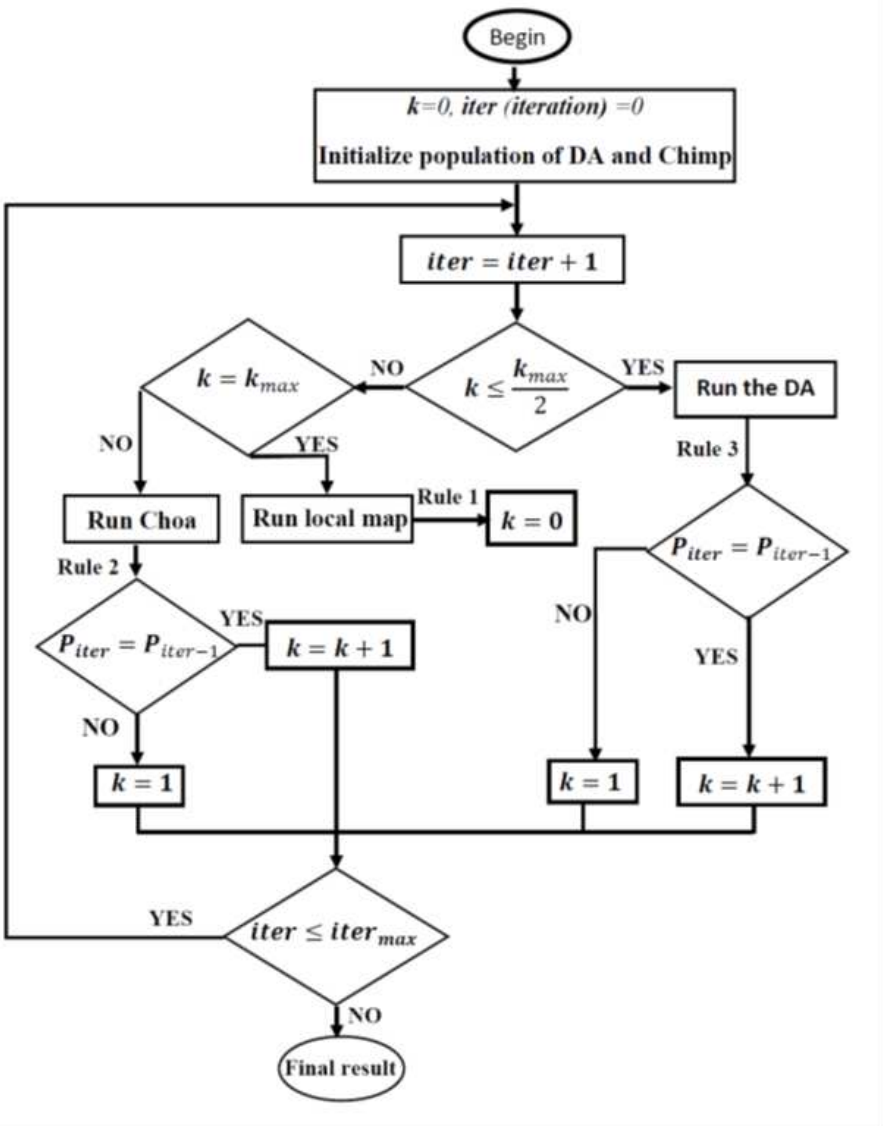

Fig.3: Flow chart of proposed hybrid algorithm.

This procedure is beneficial for fine optimization with the aid of its' great local search ability and low-cost computation. For more useful improvement of performance of the algorithm, chaotic local search technique is applied to the hybrid algorithm as an extra option. Chaotic local search generate candidate accurate solutions as the iteration is progressed. By ChoA, hybrid algorithm can settle the high dimensional optimization problems. This configuration of algorithms and methods with consideration rules can compensate the drawbacks of each algorithm greater and highlight their advantages.

5. Training MLP NNs using the Hybrid Trainer

Typically, the configuration of the parameters (weights and biases) of MLP NN is represented by three methods: (1) vectors, (2) matrix, and (3) binary state.

There are two important phases in training an MLP NN using the meta-heuristic trainer: first the representation of the problem's parameters by the search agents of meta-heuristic trainers and second, choosing the fitness function. In vector mode, each search agent of meta-heuristic algorithm (particle) presents only one vector. In matrix presentation, each search agent is shown by a matrix and in binary mode, each search agent is presented in the form of a string of binary bits.

The first method is simple and typically used in NNs. In this paper, the vector method is applied to the MLP NN because its structure is not complicated. Fig. 4 shows the vector based configuration of MLP NN parameters.

Fig.4: Assigning the problem's parameters to meta-heuristic's searching agent.

After defining the problem's parameters, a fitness function for the meta-heuristic trainer must be defined. The final goal in training an MLP NN is to obtain the highest testing accuracy as Fig. 5.

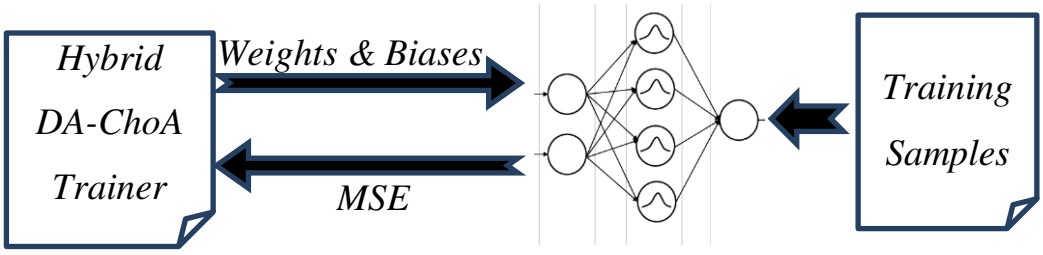

Fig.5: Hybrid DA-ChoA provides optimal weights and biases for MLP NN and gets MSE for all training samples.

Mean Square Error (MSE) is one of the most common metrics for the estimation of fitness in MLP NNs. MSE is formulated as Eq. (18). d_i^k shows desired output in dimension of k and o_i^k is ith calculated output from the MLP NN:

$$MSE = \sum_{i=1}^{m} (o_i^k - d_i^k)^2 \tag{19}$$

5.1. Setting Parameters

For evaluating the efficiency of hybrid DA-ChoA in training MLP NN, some famous trainers such as PSO, GSA, and GWO are benchmarked. The essential parameters and values of these trainers are presented in Table 5. Population size and maximum number of iteration in each algorithm is equal. In PSO algorithm, the basic version is applied and also GSA. If maximum number of iteration was less than 500, the results was not clear for evaluating the convergence and some other behavior of algorithms.

Table 5: The primary parameters of the benchmark algorithms.

Algorithms	Parameters	Value
DA	r	$(UB - LB)/10$
	Maximum number of iterations	500
	Population size	30
PSO	Layout	Full connection
	Cognitive constant (C1)	2
	Social constant (C2)	2
	Local constant (W)	0.2
	Population size	30
GSA	Number of masses	30
	a	1
	Gravitational constant	1
	Maximum number of iterations	500
ChoA	a	Linearly decreased from 2 to 0
	Maximum number of iterations	500
	Population size	30

6. Simulation Results and Analysis

Mentioned classifiers in Table 5 are applied on Sejnowski and Iris datasets and performances are evaluated in term of the processing time, classification rate, convergence speed and stuck avoiding in local optimum.

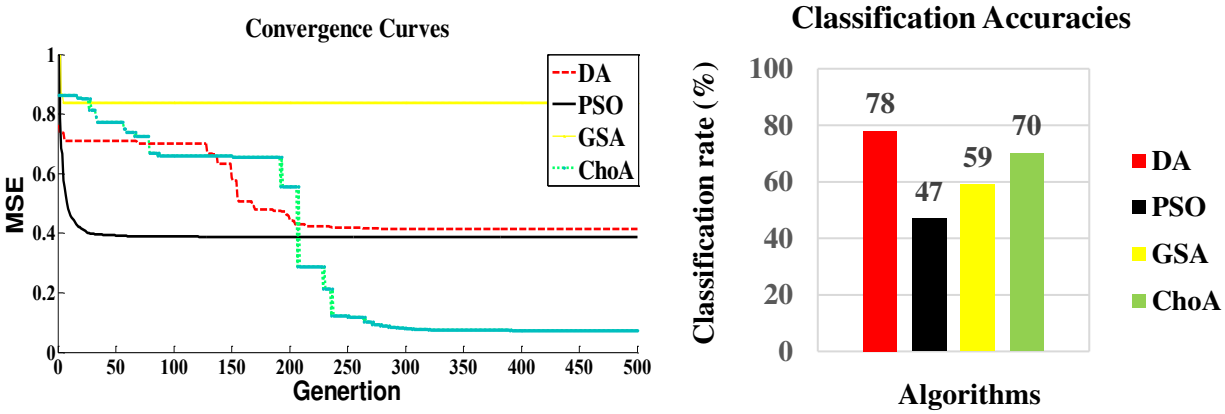

Fig.7: Comparison of convergence curves and classification rates for Iris dataset.

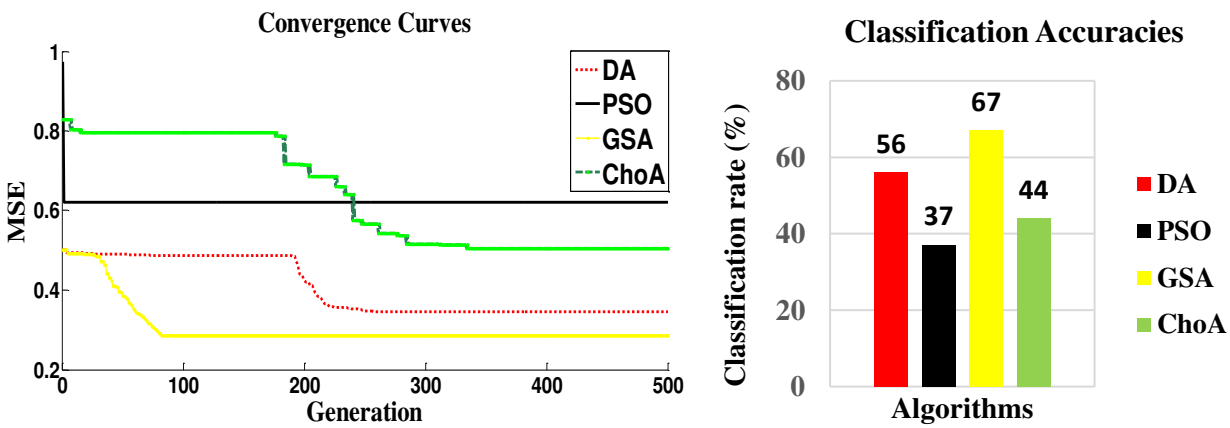

Fig.8: Comparison of convergence curve and classification rate for Sejnowski dataset.

The simulations were performed in MATLAB using a PC with a 1.8 GHz processor and 4GB RAM. Each dataset is divided into a training set (70 percent of data) and a test set (30 percent). Each classifier is tested 10 times and mean results are shown in Fig.s (7) and (8). Tables 6 and 7 show the statistical results in Iris and Sejnowski datasets respectively.

In Fig. 7, the Iris dataset is used for showing the classification accuracy and convergence rate. The designed classifier with DA trainer has the best performance than other trainers in term of classification rate. After that, ChoA has better accuracy rate. GSA and PSO have not acceptable score. In term of convergence speed, PSO and GSA are fast, but they are entrapped in local minima and their performance of MSE are not acceptable. The best performance in term of MSE is for ChoA but it converges slower and needs more iterations (generation).

Fig. 8 shows results of designed classifier in Sejnowski dataset benchmark. This dataset is more complex than Iris and has 60 dimensions. After GSA, DA has good accuracy in terms of classification. The convergence rate of DA and ChoA is very close to GSA.

The Average (AVE) and Standard Deviation (STD) of MSE in Tables 6 and 7 of the results are shown. Note that the best results are highlighted in bold type in Tables 6 and 7. To see whether the obtained results differ from other benchmark trainers in a statistically significant way, Wilcoxon's rank-sum test, a non-parametric statistical test, was accomplished at 5%

significance level. Although, P-values less than 0.05 are considered as powerful evidence against the null hypothesis.
Another comparative criterion shown in the results are classification rates and time consumed for each run.

Table 6: Simulation results of benchmark trainers for Iris dataset.

Algorithm	MSE (AVE ± STD)	P-values	Execution Time (Second)	Classification rate %
DA	0.5345 ± 0.0885	9.94 e-84	135	**78**
PSO	0.3965 ± 0.0774	1.26e-83	**81**	47
GSA	0.8407 ± 0.0491	**6.78e-110**	84	59
ChoA	**0.0794 ± 0.0112**	0.0079	118	70

In Table 6, the Iris dataset is used for evaluating each trainer. The best AVE and STD is for Choa. Hence, DA and ChoA are the slowest algorithms that need near 1.5 times more delay (execution time) rather than PSO and GSA. All the trainers have acceptable P-value.

Table 7: Simulation results of benchmark trainers for the Sejnowski dataset.

Algorithm	MSE (AVE ± STD)	P-values	Execution Time (Second)	Classification rate %
DA	0.4043 ± 0.0685	3.58 e-84	382	56
PSO	0.6212 ± 0.0158	**1.56e-110**	**142**	37
GSA	**0.3077 ± 0.0563**	1.49e-97	200	**67**
ChoA	0.777 ± 0.0249	8.04e-84	375	44

Sejnowski dataset is applied to trainers and indicated in Table 7. After GSA, DA has minimum average MSE and its delay is 2 times bigger than PSO and GSA.
In both datasets, DA and ChoA have large time-consuming mechanisms and by hybrid, DA-ChoA can decrease execution time problem and improve the classification accuracy.
In Table 8, both datasets indicated that execution time is decreased and hybrid trainer has the minimum execution time in both datasets. In the MATLAB software, we do not use pipeline or parallel structure and in high dimensional Sejnowski dataset, reduction in time consumption is not clear rather than Iris.

Table 8: Simulation results of benchmark trainers.

Algorithm	Dataset	MSE (AVE ± STD)	Execution Time (Second)	Classification rate %
DA	Iris	0.5345 ± 0.0885	135	78
ChoA	Iris	0.0794 ± 0.0112	118	70
Hybrid DA-ChoA	Iris	**0.359 ± 0.305**	**116**	**82**
DA	Sejnowski	**0.4043 ± 0.0685**	382	**56**
ChoA	Sejnowski	0.777 ± 0.0249	375	44
Hybrid DA-ChoA	Sejnowski	0.6388 ± 0.133	**360**	50

In the Iris dataset, classification rate of hybrids DA-ChoA has improved 4% rather than DA and 12% rather than ChoA. This improvement indicates that hybrid algorithm is powerful in exploration and exploitation mechanism and prevents to entrap in local minima. In the Sejnowski dataset, hybrid trainer cannot improve classification rate rather than DA but is 6% better than ChoA trainer.
For more clear evaluation, sonar dataset classification results of DA, ChoA, and hybrid trainer are shown in Fig. 9 and Fig. 10. In the Iris dataset, convergence curve indicates that the hybrid algorithm has the best performance. In lower iterations, ChoA has lower MSE, but at last, has

not had enough performance rather than hybrid. These results indicate that hybrid algorithm has an extraordinary ability in small size dataset classification because of good exploration in the entire search space. Also, hybrid has the higher speed in convergence and better performance in escaping from local minimum traps.

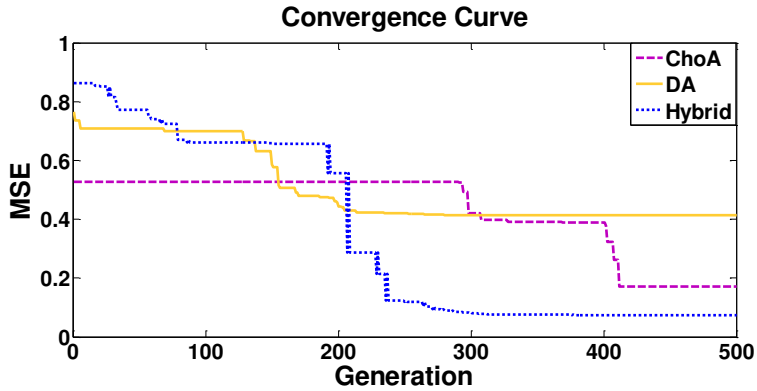

Fig.9: convergence curve of Iris dataset.

In the Sejnowski dataset, the dimensions of the dataset are 60, and Fig. 10 shows the performance of DA is better than other trainers. This is because of nature of hybrid that derives from ChoA, but after 300 iterations close to DA. The high slope of the hybrid curve shows it has greater exploration rather than two other trainers.

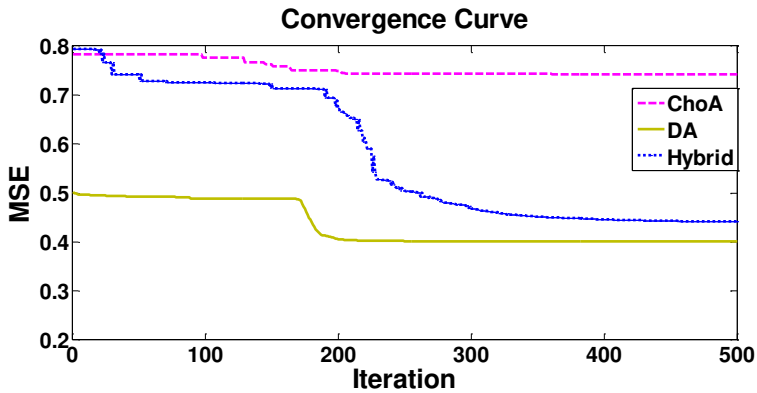

Fig.10: convergence curve of Sejnowski dataset.

7. Conclusions

A hybrid DA-ChoA classification algorithm based on MLP NN has been simulated by MATLAB simulator software. Proposed hybrid algorithm, improve the exploration ability of ChoA for training an MLP NN for the first time.

To evaluate their performance, two benchmark famous sonar datasets were utilized. Sejnowski and Iris are imported to hybrid and original classification systems and the results show that hybrid DA-ChoA has merit compared to other conventional meta-heuristic algorithms. This hybrid technique has clear advantages in terms of classification rate, local minima avoidance, and convergence speed, especially for low dimensional problems. The time consuming of process is lesser than original DA and ChoA algorithms.

Declarations

- **Funding**

The authors did not receive any funding for this study.

- **Conflict of Interest**

The authors declare that there is no confict of interest regarding the publication of this article.

- **Data availability**

The datasets generated during and/or analyzed during the current study are available.

References

[1] Mosavi MR, Khishe M, Ebrahimi E. Classification of sonar targets using OMKC, genetic algorithms and statistical moments. J Adv Comput Res 2015;7(1):50–9.

[2] Pearce SK, Bird JS. Sharpening side scan sonar images for shallow-water target and habitat classification with a vertically stacked array. IEEE J Oceanic Eng 2013;38(3):455–69.

[3] Fei T, Kraus D, Zoubir AM. Contributions to automatic target recognition systems for underwater mine classification. IEEE Trans Geoscience and Remote Sensing 2015;53(1):505–18.

[4] Mosavi MR, Khishe M, Aghababaee M, Mohammadzadeh, F. Approximation of active sonar clutter's statistical parameters using array's effective beam-width. Iranian Journal of Marine Science and Technology 2015;73(1):11–22.

[5] Miles DA, Kirk D, Clarke T. A statistical analysis of the detection performance of a broadband split-beam passive sonar. IEEE Journal of Oceanic Engineering 2006;31(4):986–996.

[6] Solomon ISD, Knight AJ. Spatial processing of signals received by platform mounted sonar. IEEE Journal of Oceanic Engineering 2002;27(1):57–65.

[7] Liu D, Liu Y, Cai H, Wang Y, Zhang H. Linear frequency-modulated continuous wave active sonar signal processing. In: IEEE Conference on Oceans-TAIPEI; 2014 p. 1–5.

[8] Fandos R, Zoubir AM, Siantidis K. Unified design of a feature based ADAC system for mine hunting using synthetic aperture sonar. IEEE Transaction Geoscience Remote Sensing 2014;52(5):2413–2426.

[9] Geethalakshmi SN, Subashini P, Ramya S. A study on detection and classification of underwater mines using neural networks. Int J Soft Comput Eng 2011;1(5):150–7.

[10] Moody J, Darken C. Fast learning in networks of locally-tuned processing units. Neural Comput 1989;1(1):289–303.

[11] Mosavi MR, Khishe M, Ghamgosar A. Classification of sonar data set using neural network trained by gray wolf optimization. J Neural Network World 2016;26(4):393–415.

[12] Oja E. Unsupervised learning in neural computation. Theoretical Computer Science 2002; 287(1):187–207.

[13] Simpson AR, Dandy GC, Murphy LJ. Genetic algorithms compared to other techniques for pipe optimization. J Water Resource Plan Management 1994;120:423–443.

[14] Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Adv Eng Software 2014;69:46–61.

[15] Mirjalili S, Hashim SZM, Sardroudi HM. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Appl Math Comput 2012;218:25–37.

[16] Mosavi MR, Khishe M, Akbarisani M. Neural network trained by biogeography based optimizer with chaos for sonar data set classification. Wireless Pers Commun 2017:1–20.

[17] Parpinelli RS, Lopes HS. New inspirations in swarm intelligence: a survey. In J Bio Inspired Comput 2011;3:1–16.

[18] Biswas A, Mishra K, Tiwari S, Misra A. Physics-inspired optimization algorithms: a survey. J Optim 2013:(2013).

[19] Fonseca CM, Fleming PJ. A novel review of evolutionary algorithms in multi objective optimization. Evol Comput 1995;3:1–16.

[20] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1:67–82.

[21] Wang GG, Guo L, Gandomi AH, Hao GS, Wang H. Chaotic krill herd algorithm. Information Science 2014;274:17–34.

[22] Wang GG, Deb S, Gandomi AH, Zhang Z, Alavi AH. A novel cuckoo search with chaos theory and elitism scheme. In: Conference on Soft Computing and Machine Intelligence-ISCMI;2014 p. 64–69.

[23] Wang G, Guo L, Wang H, Duan H, Liu L, Li J. Incorporating mutation scheme into krill herd algorithm for global numerical optimization. Neural Comput Appl 2014;24(5):853–871.
[24] Zhang JW, Wang GG. Image matching using a bat algorithm with mutation. Appl Mech Mater 2012;203(1):88–93.
[25] Chen S. Particle swarm optimization with pbest cross over. In: IEEE Congress on Evolutionary Computation-CEC 2012;1–6.
[26] Premalatha K, Natarajan A. A new approach for data clustering based on PSO with local search. Comput Inf Sci 2008;1(4):139-145.
[27] Noman N, Iba H. Accelerating differential evolution using an adaptive local search. IEEE Trans Evol Comput 2008;12(1):107–125.
[28] Holden N, Freitas AA. A hybrid PSO/ACO algorithm for discovering classification rules in data mining. J Artif Evol 2008;(2):2-11.
[29] Niu B, Li L. A novel PSO-DE based hybrid algorithm for global optimization. Conference on Intelligent Computing 2008;156–163.
[30] Duan H, Yu Y, Zhang X, Shao S. Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm. Simul Model Pract Theory 2010;18(8):1104–1115.
[31] Olorunda O, Engelbrecht AP. Measuring exploration/exploitation in particle swarms using swarm diversity. congress on evolutionary computation, CEC 2008; 1128–34.
[32] Alba E, Dorronsoro B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans Evol Comput 2005;9:126–42.
[33] Hammouri AI, Mafarja M, Al-Betar MA, Awadallah MA, Abu-Dou I. An improved Dragonfly Algorithm for feature selection. Knowledge-Based Systems 2020;203(3):106-131.
[34] Meraihi I, Ramdane A, Acheli D, Mahseur M. Dragonfly algorithm: a comprehensive review and applications. Neural Comp and App 2020;32:16625-16646.
[35] Khishe M, Mosavi MR, Kave M. Improved migration models of biogeography-based optimization for sonar dataset classification by using neural network. Applied Acoustics 2017;118:15-29.
[36] Khishe M, Mosavi MR, Moridi A. Chaotic fractal walk trainer for sonar data set classification using multi-layer perceptron neural network and its hardware implementation. Applied Acoustics 2018;137:121-139.
[37] Khishe M, Mosavi MR. Improved whale trainer for sonar datasets classification using neural network. Applied Acoustics 2019;154:176-192.
[38] Honghui Y, Junhau L, Meiping S. Underwater acoustic target multi-attribute correlation perception method based on deep learning. Applied Acoustics 2022;190:108644.
[39] Gorman RP, Sejnowski TJ. Analysis of hidden units in a layered network trained to classify sonar targets. Neural networks 1998;1:75-89.
[40] http://archive.ics.uci.edu/ml/datasets.
[41] Tamura s, Tateishi M. Capability of four-layered neural network: four layers versus three. IEEE Trans. Neural Net 1997;8:251–255.
[42] Shibata K, Ikeda Y. Effect of number of hidden neurons on learning in large-scale layered neural networks. in Proceedings of the ICROS-SICE International Joint Conference 2009;5008–5013.
[43] Hunter D, Hao Y, Pukish M.S, Kolbusz J, Wilamowski B.M. Selection of proper neural network sizes and architectures - a comparative study. IEEE Trans. Ind. Inform 2012;8:228-240.
[44] Gnana Sheela K, Deepa S. N. Review on methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering 2013;2013:1-11.
[45] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput and Applic 2015;27(4):1053-1073.
[46] Khishe M, Mosavi MR. Chimp Optimization Algorithm. Expert Systems and Applications 2020;149: Art no. 113338.