Bio-Inspired Hybridization of Artificial Neural Networks for Various Classification Tasks

Ouail MJAHED[1]*, Salah EL HADAJ[1], El Mahdi EL GUARMAH[1,2], Soukaina MJAHED[1]

[1] Faculty of Sciences and Technology, Department of Computer Sciences, L2IS, Marrakech, Morocco
mjahed.ouail97@gmail.com (**Corresponding author*), elhadajs@yahoo.fr,
guarmah@gmail.com, soukaina.mjahed@edu.uca.ma

[2] Royal Air School, Mathematics and Informatics Department, Marrakech, Morocco

Abstract: Recently, in order to optimize artificial neural networks (ANNs), several bio-inspired metaheuristic algorithms have been successfully applied. Moreover, these hybrid ANNs were operated using no more than two or three metaheuristic algorithms at a time. Additionally, the classification field is so rich that some issues were not sufficiently addressed. The main contribution of this paper is related to the use of several ANN hybridizations at the same time, while taking into account the datasets for which the ANNs or their hybridizations have been rarely explored. Thus, seven hybridized ANNs with bio-inspired metaheuristic algorithms such as particle swarm optimization (PSO-ANN), genetic algorithm (GA-ANN), differential evolution (DE-ANN), cultural algorithm (CA-ANN), harmony search (HS-ANN), black hole algorithm (BH-ANN) and ant lion optimizer (ALO-ANN) were considered for classifying four kinds of datasets. After a back-propagation neural network (BPNN) was designed, the connection weights and biases of neurons were optimized by using the seven metaheuristic algorithms mentioned above. The four selected data types belong to different domains and differ with regard to the number of classes, variables and examples. As performance measurement is concerned; the efficiencies, purities and F-measure are analysed. For all simulation runs, it can be noticed that metaheuristic algorithms were able to reach optimal efficiencies and that all the PSO-ANN-based networks obtained higher values for efficiency. For this analysis, the dependence of the obtained results on certain metaheuristic parameters was taken into account.

Keywords: Classification, Metaheuristics, Neural Networks, Optimization.

1. Introduction

The general goal of artificial intelligence is to duplicate the aptitude of human brain to perceive, evaluate, acquire, and make a decision, particularly for complex problems. A significant quantity of research in the field of Machine Learning (ML) is concerned with developing methods that systematize classification tasks (Sesmero et al., 2015) in several real-world applications, in such fields as civil engineering (Wang et al., 2020), medicine (Bhardwaj & Tiwari, 2015), high energy physics (Mjahed, 2005), investment (Del Vecchio et al., 2019), and marketing (Kaefer et al., 2005).

The use of metaheuristic algorithms to improve ML method allows reaching faster convergence with minimum iterations, which consequently enhances the efficiency of an algorithm. Recently, and with the purpose of improving ML algorithms like the artificial neural network (ANN), numerous bio-inspired metaheuristic algorithms have been established and effectively applied.

Artificial Bee Colony (ABC) (Karaboga et al., 2007) and whale optimization algorithm (Aljarah et al., 2018), and Particle Swarm Optimization (PSO), grasshopper optimization algorithm (GOA) and grey wolf optimization (GWO) (Braik et al., 2021) are applied to search optimal connection weights in ANNs. Back-propagation algorithm is replaced by PSO to train neural networks (Shah, 2018). GA is employed for optimizing ANN architectures (Sexton et al., 1998; Rahman & Setu, 2015). GA and grammatical evolution (GE) are evolved to simultaneously improve the topology and the connection weights of ANNs (Ahmadizar et al., 2015). Black Hole algorithm (BH) is proposed as a new training algorithm for ANNs (Pashaei & Pashaei, 2021). In (Khan & Sahai, 2012) a comparative analysis between gradient descent algorithms and GA, PSO, and Bat algorithm is performed.

For a medical diagnosis, a hybrid approach mixing GA and BPNN is suggested (Karegowda et al, 2011). GA and ANNs are combined to optimize the Higgs Boson search (Mjahed, 2006). By replacing standard back-propagation with PSO, thermal properties have been estimated using an ANN (Lazzús, 2013). Four hybrid ANNs for predicting the heating load of buildings' energy productivity based on ABC, PSO, imperialist competitive algorithm (ICA) and GA are suggested in (Le et al, 2019). In (Enache & Sgârciu, 2015), an improved Bat Algorithm combined to Support Vector Machines for intrusion detection is proposed. Chen et al.

https://doi.org/10.24846/v31i3y202202

(2015) proposed a ANN hybrid approach using PSO and Cuckoo Search (CS), whose results outclassed either PSO or CS alone. A hybrid method using PSO and Gravitational Search Algorithm (GSA) for training ANNs is proposed and the comparison results showed its superior performance to that of the basic PSO and GSA alone in terms of convergence speed and local minima avoidance (Mirjalili et al, 2012).

In brief, in the various works cited above, bio-inspired metaheuristic algorithms are recognized as useful in training ANNs, and the hybridized ANNs were able to develop solutions which improved the expected classification objectives. Furthermore, these hybrid ANNs were designed using no more than two or three metaheuristic algorithms at a time. The classification domain is so broad that some issues (data or methods) are not enough addressed. Indeed, some metaheuristics such as CA, BH or HS are not sufficiently processed.

Thus, the main contribution of this paper lies in:

a. the use of several hybridizations at the same time,
b. taking into account datasets where ANNs, or their hybridizations, are rarely used.

Accordingly, the proposed work focuses on seven metaheuristic optimization algorithms and four datasets. Among the metaheuristic methods, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evolution (DE), Cultural Algorithm (CA), Harmony Search (HS), Black Hole optimization (BH) and Ant Lion Optimization (ALO), are explored in order to produce more optimal solutions for ANN, moderate premature convergence and circumvent local minima.

These hybrid ANNs classifiers are applied to four datasets relevant to classification and prediction taken from public Machine Learning Repository such as Mammographic mass dataset (Elter et al., 2007), Seed dataset (Charytanowicz et al., 2010), Bearing fault data (Ouadine et al., 2018), and the Higgs Boson experiment (ATLAS Collaboration, 2014).

Performance-based fitness function using F-measure and computed from the confusion matrices is explored. In addition, the dependence of the results on certain metaheuristic-ANN parameters (such as the size of the ANN, the iterations number as well as the metaheuristic population size) is considered.

The remainder of this paper is organized as follows. Section 2 describes the ANN principle and the seven chosen metaheuristic optimization algorithms and presents the four considered datasets. In Section 3 the comparative results for Back-propagation Neural Network (BPNN) and the seven hybrid ANNs (PSO-ANN, GA-ANN DE-ANN, CA-ANN, HS-ANN, BH-ANN and ALO-ANN) are given. Section 4 sets forth the conclusion of this paper.

2. Methodology and Data

As it was specified in the Introduction, the proposed work consists in optimizing a neural network by the use of metaheuristic methods. This section provides an overview of the methods adopted as well as the data which was the subject of various applications.

2.1 Back-propagation Neural Network (BPNN)

The architecture of a multilayered neural network is organized into levels of neurons: one input layer, one output layer and one or several hidden layers. Each neuron in a level is connected to all the neurons of the previous layer and produces a response by computing a weighted sum of the outputs of the neurons to which it is connected. This sum is then processed through a nonlinear sigmoid function.

According to a supervised learning algorithm using the "gradient descent method", the network seeks to minimize, for each pattern p, a quadratic error E (equation 1) existing between the current value $o_l^{(p)}$ and the desired value $d_l^{(p)}$ of the output neuron (Bishop, 1995).

$$E=\sum_p (o_l^{(p)} - d_l^{(p)})^2 \quad (1)$$

2.2 Particle Swarm Optimization (PSO)

PSO (Kennedy et al., 2001), is a metaheuristic algorithm based on the swarm intelligence concept. A PSO process is initialized with a random population of solutions. The prospective solutions (particles) move via the problem space by following the current optimal solutions (Rini et al., 2011).

Every particle has a fitness value to evaluate by the objective function, and a velocity which directs its flying. With each iteration, the particle swarm optimizer updates its velocity and position using the two best attributes: the best solution it has reached (p_{best}) and the global best value (g_{best}), as given in (Kennedy et al., 2001).

2.3 Genetic Algorithm (GA)

GA, developed by John Holland, is a heuristic search based on Charles Darwin theory of natural evolution and selection ″the one that is best endowed, survives″ (Holland, 1975; Goldberg, 1989). It is useful for solving several kinds of problems with high complexity, and is presented as a graph of three main operators: selection, crossover and mutation. The GA algorithm is detailed in (Goldberg, 1989).

2.4 Differential Evolution Algorithm (DE)

DE is a population-based metaheuristic search algorithm (Price et al., 2005). In differential evolution, solutions are known as genomes or chromosomes. Each chromosome starts with a mutation followed by recombination. A new child is created using target vector which is employed to generate a donor vector that is then used during the recombination to obtain the trial vector. DE operates as illustrated in (Price et al., 2005).

2.5 Cultural Algorithm (CA)

CA (Reynolds, 1994) is inspired by a representation of multiple belief spaces consisting of individual and group mappa. Each individual can be distinguished in terms of traits or behaviors and a generalized depiction (called mappa) of their experience. Traits can be modified and exchanged between individuals by means of a variety of socially motivated operators. Similarly, individual mappa can be combined and improved to form “group mappa”. The updated belief space is mediated based on existing means of communication. The detailed Cultural Algorithm is described in (Reynolds, 1994).

2.6 Harmony Search Algorithm (HS)

HS Algorithm (Gao et al., 2015) is a metaheuristic search algorithm which tries to imitate the inventiveness process of musicians to elaborate a pleasing harmony. This algorithm uses harmony memory with a process similar to that of selecting the best suited individuals in other metaheuristic techniques. In addition, a harmony memory accepting parameter (reaccept) is assigned. If reaccept is too weak (close to 0), only a few better harmonies are selected leading to too slow convergence. If reaccept is high (close to 1), some harmonies will be weakly explored, thus leading to potentially erroneous solutions. The HS steps are defined in (Gao et al., 2015).

2.7 Black Hole Algorithm (BH)

BH algorithm (Hatamlou, 2013) is motivated by the phenomenon of the black hole and attempts to imitate its properties of attracting other stars in space. The algorithm starts with a population of initial solutions. At any time, the best solution is considered as a black hole and all other solutions must move towards it, as explained in (Hatamlou, 2013).

2.8 Ant Lion Optimization (ALO)

ALO algorithm (Mirjalili, 2015) is based on ant lions hunting mechanism. It consists in random walk exploration and random agent selection based on five main hunting steps: random walk of agents, construction traps, entrapment of ants in the trap, catching prey and reconstruction traps. The ALO optimizer roulette wheel and random ants’ walks can disregard local optima. The detailed ALO steps are described in (Mirjalili, 2015).

2.9 Data

As it was mentioned in the Introduction, the proposed approach is explored for four types of data: Mammographic Mass Dataset (Elter, 2007), Seed Dataset (Charytanowicz et al., 2010), Bearing Fault Dataset (Ouadine et al., 2018) and the Higgs Boson experiment (ATLAS Collaboration, 2014).

The choice of these datasets is dictated by several considerations, such as the concern for diversity and relevance, the number of classes, attributes and examples. The characteristics of these datasets are given in Table 1.

Table 1. Characteristics of the chosen datasets

Dataset	Classes and instances	Attributes (N_{VAR})	Training Data	Test Data
Mammographic Mass Data (Elter, 2007)	2 classes: 516 benign, 445 malignant	5 including 3 BIRADS Attributes	387 benign, 334 malignant	129 benign, 111 malignant
Seed Data (Charytanowicz et al., 2010)	3 classes: 70 Kama, 70 Rosa, 70 Canadian	7 Seed shape attributes	56 Kama, 56 Rosa, 56 Canadian	14 Kama, 14 Rosa, 14 Canadian
Bearing Fault Data (Ouadine et al., 2018)	4 classes (4000 signals): 1000 Healthy, 3×1000 Faulty	8 features: frequencies and amplitudes of the first 4 peaks of the Welch spectrum	700 Healthy, 3×700 Faulty	300 Healthy, 3×300 Faulty
Higgs Boson Data (ATLAS Collaboration, 2014)	2 classes (200000 events): Higgs Boson: 68340, Background: 131660	10 selected from the initial 30 features (Mjahed et al., 2020)	Higgs Boson: 54672, Background: 105328	Higgs Boson: 13668, Background: 26332

2.10 Implementation of the proposed approach

At the beginning, the considered data is classified using BPNN, followed by classifications using hybridized neural networks as it is described in Algorithm 1 and based on tuning parameters shown in Table 2.

Table 2. The tuning parameters for the employed algorithms

Algorithm	Parameter Tuning
All Algorithms	Number of features = 5-10 Number of runs = 50 Maximum number of iterations = 1000 Population size = 20-100 Fitness function, $f = 1\text{-}F$ (8)
PSO	Inertia weight = 0.72 Inertia weight damping ratio = 0.99 Personal learning coefficient = 1.49 Global learning coefficient = 1.49
GA	Crossover probability = 0.8 Mutation rate = 0.02 Mutation probability = 0.3 Selection pressure = 8
DE	Crossover probability = 0.85 Differential Mutation factor = 0.75
CA	Acceptance rate = 0.35 Constant parameter = 0.2
HS	Harmony memory rate = 0.95 Pitch-adjusting rate = 0.3 Fret width or bandwidth = 0.2
BH	Translation parameter = 0.25 Translation probability = 0.75
ALO	Number of search agents = 1000

To get an unbiased valuation of the performances of the algorithms, all datasets are distributed into a training sample (70-80%) and test sample (30-20%), as it can be seen in Table 1.

For all the neural networks considered, BPNN and metaheuristic-based ANN, the input layer contains N_{var} neurons, and the output layer only one neuron. The desired response of this output neuron is coded according to the number of classes. In the case of 2-class data (as Mammographic mass and Higgs Boson data), d_1 is set to -1 for class C_1 and +1 for C_2. If the data contains 3 classes (Seed data), d_1 is chosen to be equal to -1, 0, +1 for classes C_1, C_2 and C_3, respectively. For data with 4 classes (as Bearing fault data), d_1 = -2, -1, 1 and 2 for classes C_1, C_2, C_3 and C_4, respectively.

For every data element p_0, decision rules are proposed, for 2-class, 3-class and 4-class problems, respectively, according to the neural output o_1 as indicated in the systems (2)-(4).

$$\begin{cases} if\ o_1(p_0)\le 0\ then\ p_0\in C_1 \\ else\ \ p_0\in C_2 \end{cases} \tag{2}$$

$$\begin{cases} if\ o_1(p_0)\le -0.5\ then\ p_0\in C_1 \\ else\ if -0.5< o_1(p_0)\le 0.5\ then\ \ p_0\in C_2 \\ else\ \ p_0\in C_3 \end{cases} \tag{3}$$

$$\begin{cases} if\ o_1(p_0)\le -1.5\ then\ p_0\in C_1 \\ else\ if -1.5< o_1(p_0)\le 0\ then\ \ p_0\in C_2 \\ else\ if\ 0< o_1(p_0)\le 1.5\ then\ \ p_0\in C_3 \\ else\ \ p_0\in C_4 \end{cases} \tag{4}$$

It should be noticed that efficiencies β_i, *purities* γ_i and errors ε_i corresponding to the classifications are computed from the confusion matrix $A(A_{ij})$, (A_{ij} being the value of examples of genuine *class* C_i classified as *class* C_j). For each class C_i composed of N_i individuals, the following is obtained.

$$\beta_i = \frac{A_{ii}}{N_i},\ \ \gamma_i = \frac{A_{ii}}{\sum_j A_{ji}},\ \ \ \varepsilon_i = 1-\beta_i \tag{5}$$

Thus, the global values of these parameters become as it is specified in equation (6).

$$\beta = \frac{\sum_i N_i \beta_i}{\sum_i N_i}, \quad \gamma = \frac{\sum_i N_i \gamma_i}{\sum_i N_i}, \quad \varepsilon = 1 - \beta \qquad (6)$$

Similarly, a global F-measure (F), and F-measure (F_i) for each class C_i, are defined in equation (7).

$$F_i = \frac{2\beta_i \gamma_i}{\beta_i + \gamma_i}, \quad F = \frac{\sum_i N_i F_i}{\sum_i N_i} \qquad (7)$$

It is also important to emphasize that for all the hybridized ANNs, and in order to maximize efficiencies β, purities γ and F-measure F, the same fitness function f was used (equation 8).

$$f = 1 - F \qquad (8)$$

Algorithm 1. BPNN and Metaheuristic-ANN Implementation
1: Training and validation data reading, 2: Extraction of dimensions (N_{var}, N_i number of individuals of each class...) 3: Preparing the architecture of ANN with 1, 2 or 3 hidden layers (N_{var}:n_1:1) or (N_{var}:n_1:n_2:1) or (N_{var}:n_1: n_2: n_3:1), (n_1, n_2, n_3 ranging from 5 to 35 neurons) 4: For the BPNN approach, optimize weights and thresholds according to subsection 2.1. 5: Return the results of each BPNN with its structure, error and efficiency 6: Find the best BPNN 7: For the hybrid metaheuristic ANN-based approach (PSO-ANN, GA-ANN, CA-ANN, ALO-ANN, HS-ANN, BH-ANN or DE-ANN), optimize weights and thresholds according to the respective metaheuristic algorithm. 8: Return the results of each metaheuristic-ANN with its structure, error and efficiency 9: Find the best metaheuristic-ANN

In the case of the BPNN classification, N_W weights and N_T thresholds, for all multilayer architectures, are developed thanks to the calculations provided in section 2.1. In the case of ANN hybridized by PSO, GA, DE, CA, BH, HS and ALO, the N_W weights and N_T thresholds are the ingredients (particles, ant lions or genes) to be handled until the end of the iterations. All structures with 1, 2 and 3 hidden layers, containing between 5 and 35 neurons each, are explored, with 100 to 1000 iterations and populations ranging from 20 to 100 individuals.

The computation code for these different analyses was compiled under the Matlab environment, R2017b, on an Intel Core i7 3.0 GHz processor, with 16 GB of RAM. Some existing functions in Mathworks library (blocksets) were used to implement the optimization tasks. For practical adaptations, other Matlab codes have been developed.

3. Results and Discussion

In this section the results for the implementation of the proposed approach are presented.

The complexity of the task performed by a neural network is related to the number of its weights and thresholds to be optimized. For the case of a 3-hidden layer ANN architecture (N_{var}:n_1: n_2: n_3: 1), knowing that the input layer contains $Nvar$ neurons, with n_1, n_2 and n_3 hidden neurons (in the hidden layers) and the output layer of a single neuron, the N_W weights and N_T thresholds are as specified in equation (9).

$$\begin{aligned} N_W &= N_{\text{var}} n_1 + n_1 n_2 + n_2 n_3 + n_3 \\ N_T &= n_1 + n_2 + n_3 + 1 \end{aligned} \qquad (9)$$

This means that for the architecture (5: 10: 25: 20: 1), for example, the values of 820 weights and 56 biases have to be optimized.

The first observation to be made concerns the number of ANNs to be explored. Figure 1 illustrates the histogram of ANN architectures as a function of the number of hidden neurons used.

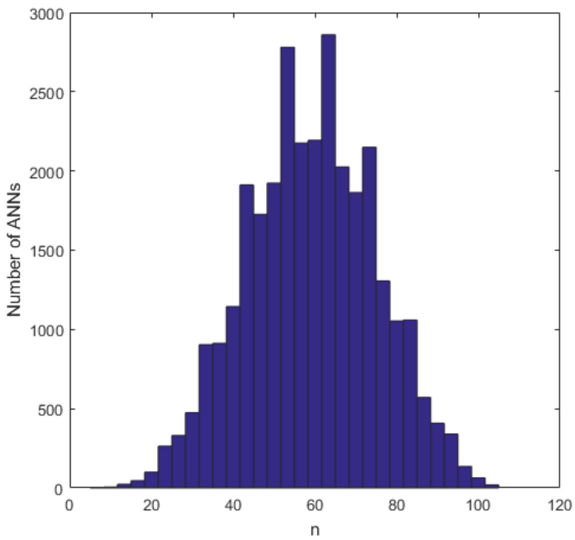

Figure 1. Histogram of ANN architectures as a function of the number of hidden neurons

The second point worth noting is the reliance of the classification efficiency on the number of neurons (whatever their distribution in the neural network). This observation is valid regardless of the optimization approach used.

This property is illustrated in Figure 2, where classification efficiencies are plotted with respect to the number of neurons in the network for PSO-ANN and for Mammographic mass and Higgs Boson data.

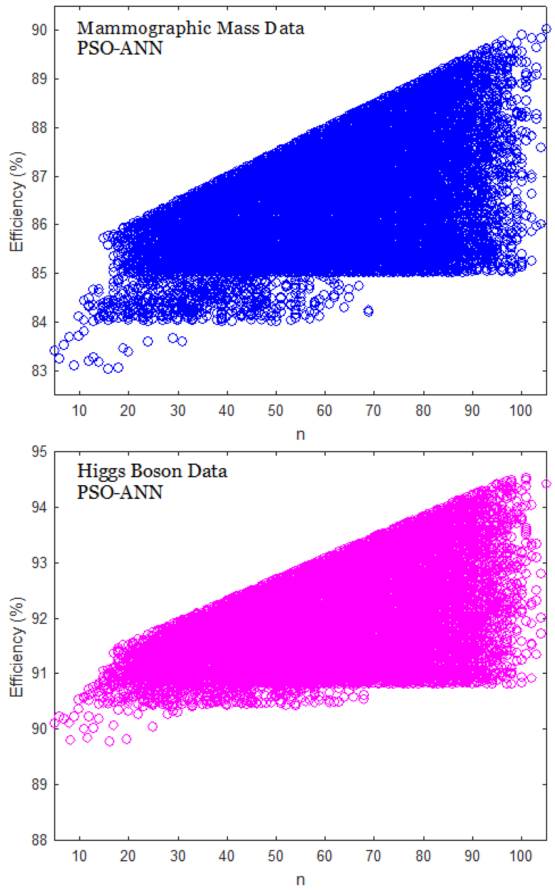

Figure 2. PSO-ANN Efficiency vs the Number of Neurons *n* for Mammographic Mass and Higgs Boson Datasets

The same result is illustrated in Figure 3, where the histogram of PSO-ANN efficiencies is shown, again for the Mammographic Mass and Higgs Boson datasets.

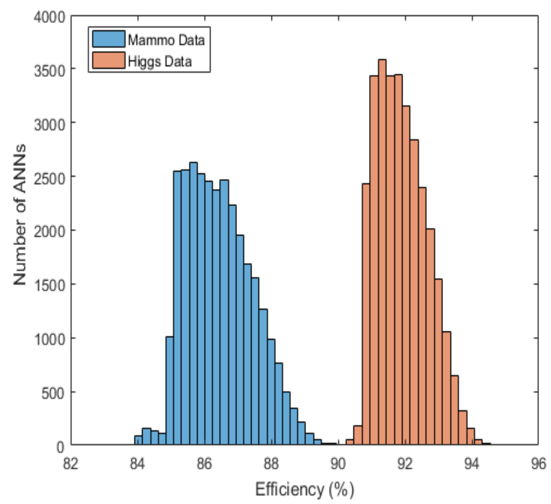

Figure 3. Histogram of PSO-ANN Efficiencies for Mammographic Mass and Higgs Boson Datasets

Tables 3 and 4 illustrate the Efficiencies and Purities measures obtained with the eight employed algorithms and for all datasets, for both training and testing respectively. *Best* and *Mean* respectively designate the best value and the average value of the Efficiencies and Purities measure for the 50 runs and a maximum number of iterations set to 200. Standard Deviation (*Std*) represents the standard deviation of the reached values.

From another point of view, the third topic worth citing in this analysis concerns the dependence of the different ANNs optimization results on the number of iterations chosen during the learning stage. As it is illustrated in Figure 4, produced on the ANN architecture (5:34:30:1), a comparison between the different approaches can be made, despite the fact that the results depend on the purely stochastic character of the different considered metaheuristics. It can be seen that the PSO-ANN algorithm converges faster towards the expected minimum than the other ANNs. It should be noticed that for any hybridization, the efficiency can be improved by 2-3 points if the number of iterations is increased to 1000.

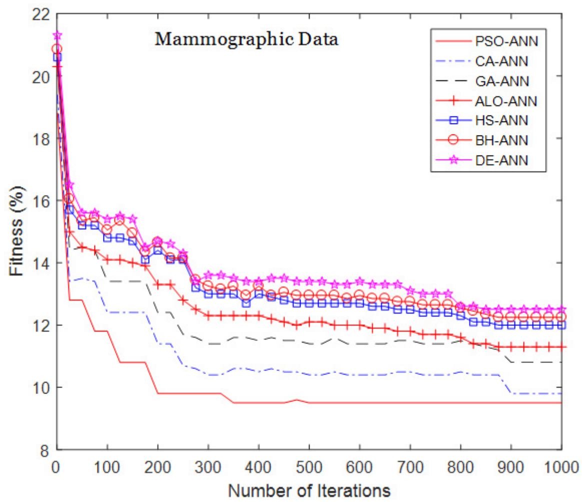

Figure 4. Fitness Function vs Number of Iterations for different ANNs with architecture (5:34:30:1) (Mammographic Mass Data)

Another point of interest regards the dependence of the results on the population's size. In Figure 5, produced on the same ANN architecture (5:34:30:1), the best efficiencies obtained for different values of population's size are illustrated. The obtained results again show the dominance of PSO-ANN approach. Moreover, an average population of 50 individuals seems to provide better results.

Table 3. Training Efficiencies β (%) and Purities γ (%) for all Datasets and all Algorithms

Algorithm		Mammographic Mass		Higgs Boson		Seed Data		Bearing Fault	
		β (%)	γ (%)	β (%)	γ (%)	β (%)	γ (%)	β (%)	γ (%)
BPNN	*Best*	86.55	87.75	88.89	89.38	96.02	96.32	99.13	99.18
	Mean	85.34	85.67	86.67	86.89	94.82	94.76	97.34	98.32
	Std	01.01	01.31	01.21	01.63	00.99	00.94	01.43	01.32
PSO-ANN	*Best*	90.54	92.34	94.23	96.33	99.93	99.98	100.00	99.99
	Mean	87.31	87.66	93.15	95.35	98.33	99.73	98.72	98.96
	Std	01.21	01.27	01.43	01.46	00.97	00.96	01.33	01.23
GA-ANN	*Best*	88.41	88.77	91.72	91.79	98.52	98.73	99.95	99.95
	Mean	86.95	86.83	89.11	89.93	97.13	96.16	98.04	98.78
	Std	00.91	00.91	01.23	01.93	00.96	00.95	01.63	01.75
DE-ANN	*Best*	86.98	87.67	89.13	89.93	96.66	97.63	99.94	99.77
	Mean	85.60	86.73	87.73	88.76	94.92	95.93	98.01	98.36
	Std	01.34	01.34	01.76	01.36	00.98	00.89	01.51	01.43
CA-ANN	*Best*	89.08	89.56	92.77	92.89	98.82	98.72	99.97	99.98
	Mean	87.14	87.84	90.41	91.71	97.98	97.96	97.78	97.73
	Std	01.08	01.08	00.97	00.93	00.96	00.89	01.21	01.45
HS-ANN	*Best*	88.06	88.96	90.02	91.12	97.39	98.69	99.93	99.98
	Mean	86.79	86.83	89.71	89.81	96.22	97.32	97.64	97.93
	Std	00.98	00.98	01.23	01.63	00.97	00.94	01.04	01.33
BH-ANN	*Best*	88.17	88.97	89.83	89.96	97.52	98.13	99.95	99.96
	Mean	86.12	88.13	89.73	89.73	96.42	97.32	97.74	98.73
	Std	01.43	01.68	00.99	01.39	01.63	01.43	01.19	01.66
ALO-ANN	*Best*	88.31	88.31	92.55	92.89	97.58	97.88	99.96	99.98
	Mean	86.86	87.83	89.76	89.53	95.72	96.33	98.63	98.36
	Std	01.43	01.46	01.27	01.67	01.23	01.11	01.23	01.07

Table 4. Test Efficiencies β (%) and Purities γ (%) for all Datasets and all Algorithms

Algorithm		Mammographic Mass		Higgs Boson		Seed Data		Bearing Fault	
		β (%)	γ (%)	β (%)	γ (%)	β (%)	γ (%)	β (%)	γ (%)
BPNN	*Best*	86.12	87.56	88.75	89.12	95.98	96.23	99.01	99.04
	Mean	85.31	85.38	86.43	86.67	94.45	94.45	97.12	98.22
	Std	01.03	01.23	01.43	01.41	00.94	00.97	01.34	01.43
PSO-ANN	*Best*	90.23	92.12	94.12	96.12	99.45	99.38	99.79	99.68
	Mean	87.12	87.45	93.35	95.32	98.23	99.23	98.67	98.21
	Std	01.44	01.22	01.33	01.22	00.79	00.98	01.23	01.11
GA-ANN	*Best*	88.17	88.47	91.55	91.56	98.41	98.55	99.55	99.57
	Mean	86.34	86.56	89.05	89.67	97.15	96.22	98.01	98.50
	Std	00.96	00.90	01.21	01.64	00.93	00.98	01.22	01.23
DE-ANN	*Best*	86.74	87.56	89.02	89.65	96.24	97.12	99.34	99.57
	Mean	85.42	86.36	87.34	88.43	94.79	95.56	98.01	98.02
	Std	01.21	01.44	01.36	01.26	00.95	00.86	01.21	01.01
CA-ANN	*Best*	89.01	89.55	92.23	92.55	98.62	98.44	99.66	99.44
	Mean	87.01	87.67	90.21	91.43	97.77	97.67	97.44	97.22
	Std	01.04	01.06	00.95	00.95	00.97	00.88	01.13	01.14
HS-ANN	*Best*	88.01	88.55	89.78	91.05	97.22	98.56	99.45	99.59
	Mean	86.45	86.67	89.34	89.44	96.21	97.23	97.44	97.34
	Std	00.97	00.96	01.53	01.23	00.99	00.98	01.02	01.23
BH-ANN	*Best*	87.99	88.69	89.56	89.76	97.42	98.02	99.67	99.56
	Mean	86.82	88.22	89.34	89.61	96.32	97.22	97.44	98.45
	Std	01.34	01.45	00.96	01.13	01.21	01.13	01.01	01.23
ALO-ANN	*Best*	88.21	88.23	92.24	92.58	97.18	97.67	99.56	99.45
	Mean	86.67	87.67	89.43	89.55	95.22	96.22	98.43	98.11
	Std	01.33	01.44	01.21	01.34	01.11	01.03	01.23	01.01

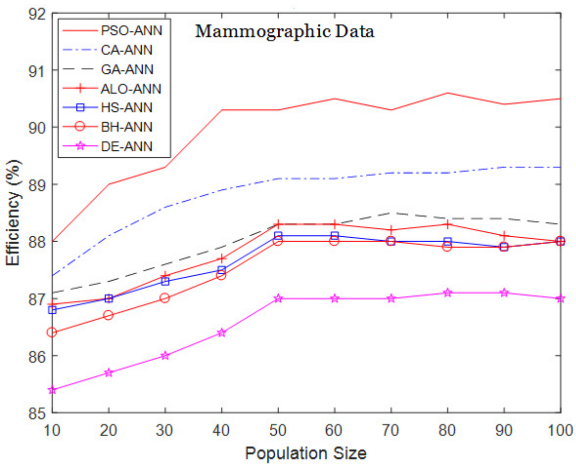

Figure 5. Efficiency vs Population's size for different ANNs with architecture (5:34:30:1) (Mammographic Mass Data)

From the results depicted by Figures 1 to 3, it can be understood that optimal ANN architectures are rare. However, increasing the number of iterations or the populations' size also improves the performance of certain architectures (Figures 4 and 5). From the obtained results (Tables 3 and 4 and Figures 2 to 5), it is worth highlighting a significant preference for the PSO-ANN approach, followed by hybridizations CA-ANN, GA-ANN, ALO-ANN, HS-ANN, BH-ANN and DE-ANN.

The run times for all the considered approaches, performed offline during Bearing Fault data training step, for (8:25:22:1) ANN architecture, with 1000 iterations, are grouped in Table 5. Considering Table 5 and previous results, the PSO-ANN approach features the best efficiency and computational cost.

Table 5. Run times in seconds for all ANNs approaches using Bearing Fault dataset with architecture (8:25:22:1) (Training step with 1000 iterations)

Method	BPNN	GA-ANN	BH-ANN
CPU Time (s)	133.14	278.76	262.79
Method	HS-ANN	DE-ANN	PSO-ANN
CPU Time (s)	545.28	546.89	591.18
Method	ALO-ANN	CA-ANN	
CPU Time (s)	1696.85	665.84	

It should be noted that Bearing fault data and Higgs Boson data, have been extensively analysed, also implying the selection of the best attributes, as in (Ouadine et al., 2018) for Bearing fault data, and in (Mjahed et al., 2020; Tang, 2021; Azhari et al., 2020) for Higgs Boson data. For Bearing fault diagnosis and with the use of PSO-ANN approach, one obtained generally better results than those featured by other works, as in (Chomphan & Kingrattanaset, 2014; Ouadine et al., 2018). Regarding the results of the Higgs Boson dataset, PSO-ANN outperforms the results published in (Mjahed et al., 2020; Tang, 2021; Azhari et al., 2020). On the other hand, the other datasets (Seed data and Mammograhic mass data) did not undergo any preprocessing, or any research on other possibly more discriminating attributes. Moreover, the number of attributes is low, which does not allow a real search for the best among them. Despite these observations, the results obtained compare favorably with those featured in similar works for Seed data segmentation (Ayse, 2020), and for Mammographic mass data (Nirmala & Suresh, 2020). Looking more closely at the obtained test results, high values for *F-measure* were reached. With PSO-ANN, this parameter is 0.9116, 0.9931, 0.9511 and 0.9973 for Mammographic Mass, Seed, Higgs Boson and Bearing fault datasets, respectively. Note that these latter values can be obtained using the results of Table 4 and the formula defined in equation 7.

The small differences between test and train results (Tables 3 and 4) prove that the proposed hybrid ANN has learned the characteristics existing in the employed datasets. Using a F-measure based fitness function has achieved the desired results. In a parallel study, the use of error also lead to similar results. This is because minimizing error is equivalent to maximizing efficiency or purity or F-measure. The results obtained, based on the four types of data tend towards the same conclusions. This is particularly the case for the observations presented above and illustrated by Figures 3 to 5.

4. Conclusion

With the goal of using multiple hybridizations at the same time, and taking into account datasets where ANNs, or their hybridizations, are not often used, the conclusions for this paper are as follows.

The quality of learning depends on the parameters adopted for metaheuristic algorithms, which was a good reason to run the computational code several times (50 runs). Moreover, the number of iterations is here of great importance, without forgetting the precision-computation time dilemma. The results are also initialization-dependent; consequently several runs are necessary to achieve the expected performance.

All of the bio-inspired methods have given promising results. The combinations of ANNs with bio-inspired metaheuristic methods (PSO, CA, GA and ALO) confirm the superiority of their results compared to those of BPNN. PSO-ANN appears to be more efficient than other hybridized ANNs for the datasets chosen and the runs performed. In comparison with the classical BPNN, the Hybrid Neural Networks PSO-ANN, CA-ANN and GA-ANN gave good results with up to 5% more efficiency.

For the PSO-ANN approach in particular, better results than those featured by published works were generally obtained for the 4 datasets chosen.

As possible future work, metaheuristic algorithms could be hybridized between them, so as to combine their advantages. Additionally, other types of neural network models could be explored and hybridized with other metaheuristic algorithms.

REFERENCES

Ahmadizar, F., Soltanian, K., Akhlaghian T., F. & Tsoulos, I. (2015). Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm, *Engineering Applications of Artificial Intelligence*. *39,* 1-13.

Aljarah, I., Faris, H. & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Computing, 22*, 1–15.

ATLAS Collaboration (2014). *Dataset from the ATLAS Higgs Boson machine learning challenge 2014*, CERN Open Data Portal, DOI 10.7483/OPENDATA.ATLAS.ZBP2.M5T8

Ayse, E. (2020). An Application of Deep Neural Network for Classification of Wheat Seeds, *European Journal of Science and Technology, 19,* 213-220.

Azhari, M., Abarda, A., Ettaki, B., Zerouaoui, J. & Dakkon, M. (2020). Higgs Boson Discovery using Machine Learning Methods with Pyspark, *Procedia Computer Science, 170,* 1141–1146.

Bhardwaj, A. & Tiwari, A. (2015). Breast cancer diagnosis using Genetically Optimized Neural Network model, *Expert Systems with Applications*, *42*(10), 4611–4620.

Bishop. C.M. (1995). *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press.

Braik, M., Al-Zoubi, H. & Al-Hiary, H. (2021). Artificial neural networks training via bio-inspired optimisation algorithms: modelling industrial winding process, case study, *Soft Computing, 25*(6)*,* 4565-4569.

Charytanowicz, M. Niewczas, J. Kulczycki, P. Kowalski, P. A. Lukasik, S. & Zak. S. (2010). UCI Machine Learning Repository. Seeds Dataset. Available at: <https://archive.ics.uci.edu/ml/datasets/seeds>.

Chen, J.-F., Do, Q. & Hsieh, H.-N. (2015) Training artificial neural networks by a hybrid PSO-CS algorithm, *Algorithms 8*(2), 292–308.

Chomphan, S. & Kingrattanaset, T. (2014). An Analysis of Sound for Fault Engine, *American Journal of Applied Sciences, 11*(*6*), 1005-1009.

Del Vecchio, C., Fenu, G., Pellegrino, F.A., Di Foggia, M., Quatrale, M., Benincasa, L., Iannuzzi, S., Acernese, A., Correra, P. & Glielmo, L. (2019). Support Vector Representation Machine for superalloy investment casting optimization, *Applied Mathematical Modelling*, *72*, 324–336.

Elter, M., Schulz-Wendtland R. & T. Wittenberg. (2007). *UCI Machine Learning Repository*. Mammo graphic mass Dataset. Available at: <http://archive.ics.uci.edu/ml/datasets/mammographic+mass>

Enache, AC., Sgârciu, V. (2015). An Improved Bat Algorithm Driven by Support Vector Machines for Intrusion Detection. In *Proceedings of the International Joint Conference. CISIS 2015*, *vol 369* (pp. 41-51).

Gao, X. Z., Govindasamy, V., Xu, H., Wang, X. & Zenger, K. (2015). Harmony Search Method: Theory and Applications, *Computational Intelligence and Neuroscience,* Article ID 258491. Available at: <https://doi.org/ 10.1155/2015/258491>

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley Publishing Company, Inc.

Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering, *Information Sciences, 222*, 175-184.

Holland. J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press.

Hongping H., Yangyang L., Yanping B., Juping Z. & Maoxing L. (2019). The Improved Antlion Optimizer and Artificial Neural Network for Chinese Influenza Prediction, *Complexity*, 1-12, (Article ID 1480392).

Kaefer, F., Heilman, C.M. & Ramenofsky, S.D. (2005). A neural network application to consumer classification to improve the timing of direct marketing activities. *Computers and Operations Research*, *32*(10), 2595–2615.

Karaboga, D., Akay, B. & Ozturk, C. (2007). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. *Modeling Decisions for Artificial Intelligence*, pp.318–329, Springer, Berlin, Heidelberg.

Karegowda, A. G., Manjunath, A. & Jayaram, M. (2011). Application of genetic algorithm optimized neural network connection weights for medical diagnosis of PIMA Indians diabetes, *International Journal of Soft Computing, 2*(2), 15–23.

Kennedy, J., Eberhart, R. C. & Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

Khan, K. & Sahai, A. (2012). A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context, *International Journal of Intelligent Systems and Applications, 4*(7), 23–29.

Lazzús, J. A. (2013). Neural network-particle swarm modeling to predict thermal properties, *Mathematical and Computer Modelling*, *57*(9–10), 2408-2418.

Le, L.T., Nguyen, H., Dou, J. & Zhou, J. (2019). A Comparative Study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in Estimating the Heating Load of Buildings' Energy Efficiency for Smart City Planning, *Applied Sciences, 9*(*13*), 2630.

Mirjalili, S. (2015). The ant lion optimizer, *Advances in Engineering Software*, *83,* 80-98.

Mirjalili, S., Hashim, S. Z. M., & Sardroudi, H. M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm, *Applied Mathematics and Computation, 218*(22), 11125–11137.

Mjahed, M. (2005). Higgs Search at LHC by Neural Networks, *Nuclear Physics B* (Proceedings Supplements), *140,* 799-801.

Mjahed, M. (2006). Search for the Higgs boson at LHC by using Genetic Algorithms, *Nuclear Instruments and Methods A: Accelerators, Spectrometers, Detectors and Associated Equipment, 559*(1), 172-176.

Mjahed, S., El Hadaj, S., Bouzaachane, K. & Raghay, S. (2020). Hybridization of Fuzzy and Hard Semi-Supervised Clustering Algorithms Tuned with Ant Lion Optimizer Applied to Higgs Boson Search. *Computer Modeling in Engineering & Sciences*, *125*(2), 459-494.

Nirmala, G. & Suresh, K.P. (2020). Deep Convolutional Neural Network for Breast Mass Classification from Mammogram, *Bioscience Biotechnology Research Communications* (Special Issue), *13*(13), 203-208.

Ouadine, A. Y., Mjahed, M., Ayad H. & El Kari, A. (2018). Aircraft Air Compressor Bearing Diagnosis Using Discriminant Analysis and Cooperative Genetic Algorithm and Neural Network Approaches, *Applied Sciences*, *8*(*11*), 2243.

Pashaei, E. & Pashaei, E. (2021). Training Feedforward Neural Network Using Enhanced Black Hole Algorithm: A Case Study on COVID19 Related ACE2 Gene Expression Classification, *Arabian Journal for Science and Engineering, 46*, 3807–3828.

Price, K. V., Storn R. M. & Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Springer.

Rahman, M. & Setu, T. A. (2015). An Implementation for Combining Neural Networks and Genetic Algorithms. *International Journal of Computer Science and Technology*. *6*(3), 218-222.

Reynolds, R. (1994). An Introduction to Cultural Algorithms. In *Proceedings of the Third Annual Conference on Evolutionary Programming* (Eds.), (Sebald A. V. & Fogel L. J.), (pp. 131-139).

Rini, P., Shamsuddin, M., & Yuhaniz. S. (2011). Particle Swarm Optimization: technique, system and challenges, *International Journal of Computer Applications 14*(1)*,* 19–27.

Sesmero, M. P., Alonso-Weber, J. M., Gutierrez, G., Ledezma, A. & Sanchis, A. (2015). An ensemble approach of dual base learners for multi-class classification problems, *Information Fusion*, *24*, 122–136.

Sexton, R. S., Dorsey, R. E. & Johnson, J. D. (1998). Toward global optimization of neural networks: a comparison of the genetic algorithm and back-propagation, *Decision Support Systems 22(*2), 171-185.

Shah, U. (2018). A Hybrid Approach of ANN with PSO for Classification Problems. *In TENCON 2018 - IEEE Region 10 Conference*, (pp. 0611-0616), DOI: 10.1109/TENCON.2018.8650351.

Tang, H. (2021). Application of Artificial Neural Network (ANN) and Feature Selection Algorithm (FSA) on the ATLAS Experiment Data to Identify Higgs Boson, *Journal of Physics: Conference Series, 1873*, 012072

Wang, Q., Li, Q., Wu, D., Yu, Y., Tin-Loi, F., Ma, J. & Gao, W. (2020). Machine learning aided static structural reliability analysis for functionally graded frame structures, *Applied Mathematical Modelling*, *78*, 792–815.