Comparative Analysis of Performances of an Improved Particle Swarm Optimization and a Traditional Particle Swarm Optimization for Training of Neural Network Architecture Space

Emre Çomak*, Gürhan Gündüz

Department of Computer Engineering, Faculty of Engineering, Pamukkale University, Kinikli Campus, Camlaralti Neighbourhood, 20070 Denizli, Türkiye; ecomak@pau.edu.tr, ggunduz@pau.edu.tr

* Correspondence author

Abstract: Many studies evaluating the performance of various optimization methods for training Artificial Neural Networks (ANNs) have produced conflicting results. This discrepancy often arises due to the limited application of these methods across a narrow spectrum of ANN architectures and training parameter values. In response to this gap, our study introduces an enhanced Particle Swarm Optimization (PSO) technique, denoted as Reverse Direction Supported Particle Swarm Optimization (RDS-PSO), specifically designed for ANN training. RDS-PSO incorporates two novel parameters, namely alpha and beta, allowing the creation of four distinct RDS-PSO types including the original PSO. Unlike many existing studies, we comprehensively evaluate the performance of these four RDS-PSO types across a diverse set of criteria. These criteria include the architectural space of ANN, training depths for ANN, inertia weight direction for RDS-PSO, and adaptation approaches for the two novel parameters of RDS-PSO. Through 100 iterations for each training case, we conduct an extensive and intricate analysis of ANN training performance on three medical datasets. Our experimental findings reveal that RDS-PSO_3, featuring decreasing inertia weight and cosine adaptation, consistently outperforms other RDS-PSO types. Furthermore, RDS-PSO_3 demonstrates greater reliability, as evidenced by lower standard deviation values, across most ANN architectures.

Keywords: neural network training; global searching; particle swarm optimization; improved particle swarm optimization

1 Introduction

Artificial Neural Networks (ANNs) are based on emulating neuron functions within the human nervous system through mathematical representations. To manage the abilities related to some learning tasks, ANNs have been designed over many neurons connected to other neurons in determined orders [1]. Such an order between neurons is constructed over the structures called as layer. General performance of ANNs is dependent on following three main settings regardless of which application is implemented [2]:

1-Preferred network architecture (model) for ANN: Number of layers, number of neurons at each layer, direction and sparseness of connections between layers are critical factors for an ANN network design. Number of layers is determined as at least 3 (including one hidden layer) while there is no mathematical procedure to determine the number of neurons at each layer. Both fully connected and sparsely connected models may be preferred while an ANN model is constructed.

2-Preferred training algorithm, training parameter setting and training depth for ANN: Algorithms used for ANN training may be divided into two main groups called as global and local searching techniques [3]. Global searching techniques which are based on machine learning algorithms can search whole space although they do not guarantee the global optimal solution exactly. Dependency to initial training parameter values in global techniques is less than those of the local ones. On the other hand, local searching techniques contain gradient descent based solutions [4]. Contrary to global ones, these techniques guarantee only the local optimal. Finally, training depth is a parameter indicating limit of the iteration number in training algorithm to be executed.

3-Dataset: Some properties such as form of input and output spaces, and number of samples create the stylemark of a data set. Stylemarks of the datasets directly affects both training and test classification performances of ANN.

The study [5] claimed that generalization ability of ANNs could be enhanced by using Generalized Operational Perceptrons with modified back propagation. The study [6] used the Teaching Learning-Based Optimization for training the hybrid Functional Fuzzy Wavelet Neural Network and tested it on five medical datasets. The mussels wandering optimization was improved and applied to training of ANNs and comparable results were obtained on real-world data in the study of [7]. The study [8] suggested the Delta Associative Memory using social networking and collaborative learning to diagnose diseases. However, Multi-Layer Perceptron (MLP) with Back Propagation (BP) has been mostly used version of ANN in medical applications because of its effectiveness [9], and also PSO has been one of the mostly used optimization techniques in different areas.

PSO indicating global searching characteristics is proposed for optimization problems as a member of Swarm Intelligence methods [10]. PSO has been used for solution of many problems from different areas including training of ANNs due to its flexible and uncomplicated algorithm compared to a number of other global searching techniques [11-13].

In literature, some studies have been suggested to compare ANN performances based on various global and local searching techniques in different ways.

The study [14] evaluated PSO and BP methods in terms of computational complexity when both methods had the same Mean Square Error (MSE) over nonlinear function approximation. PSO based MLP results overrode BP one. The study [15] trained an ANN with single hidden layer by means of BP and PSO for imbalanced medical datasets. Results showed that BP based classifier produced better classification accuracy than that of PSO based one. The study [16] introduced a hybrid algorithm based on PSO with time-varying parameter and BP algorithm to train ANNs. They compared their algorithm with some other popular algorithms and obtained reasonable results on eight datasets from UCI medical datasets [17]. The study [18] trained an MLP by an improved PSO, i.e. Centripetal Accelerated Particle Swarm Optimization (CAPSO). They indicated superior success of this classifier on nine standard medical datasets. The study [19] reduced computational cost of the Polynomial Neural Network (PNN) by using PSO and gradient descent algorithms. There was no significant difference between PSO and gradient descent algorithms according to these results. The study [20] developed a novel algorithm, which was called as Action Dependent Heuristic Dynamic Programming (ADHDP), in order to discriminate the symbols and text automatically in medical images. They compared their experimental results with BP-ANN, PSO-ANN, GA-ANN, K Nearest Neighbor (KNN) and Support Vector Machine (SVM). Their algorithm provided a small improvement in Receiver Operating Characteristic (ROC) curve and Area Under Curve (AUC) values of ROC. The study [21] used a modified gravitational search algorithm optimized MLP to separate benign and malicious internet traffic. They claimed that their algorithm was superior to PSO and error back propagation optimized MLP. The study [22] proposed PSO, Differential Evolution and back propagation based hybrid optimization methods to train ANN for classifying clinical datasets. Differential Evolution with back propagation indicated the best results. [23] introduced Vortex Search optimization algorithm to set weights and biases of an MLP. They conducted their learning system on six different datasets including few medical datasets and obtained competitive results. The study [3] proposed an extended evaluation approach to compare the performances of BP and PSO algorithms in MLP training.

Given the existing discrepancies in prior literature studies, this paper advocates for a rigorous and thorough analysis to attain a robust and well-founded judgment on the subject. The study in [3] was also proposed to address these discrepancies. However, it employed only original PSO for training of ANN. In this work, we employ RDS-PSO for training ANN. RDS-PSO is an enhanced and generalized version of original PSO and it was introduced by us as a function optimizer. RDS-PSO includes total of four PSO versions, three of them are new PSO versions and one of them is original PSO. Thus in this study, BP and four PSO algorithms were compared with the study in [3] for MLP training on the same medical datasets. As it is explained in above paragraphs, general performance of ANN depends on its network architecture (model), training algorithm/parameters/depth, and the dataset. The same model providing accurate and in-depth evaluation in the study of [3] was preferred again in this study for a reliable comparison. The preferred model was constructed on an architecture space containing 41 configurations from the simplest one to the most complex one. Another factor affecting the general performance is the training algorithm. Generalization on PSO algorithm was provided by an improved PSO and an extended comparison among PSO and improved three PSO algorithms was introduced in this study. In tests, MSE metric for training and Classification Error (CE) metric were measured over 41 ANN models as in [3]. Iteration limits namely training depth for our searching algorithms were selected as 200 and 2000 in this study as in [3] where they were called shallow and deep respectively. The final factor is the dataset. To compare this study with [3], the same three medical datasets were used in our experiments.

Remaining of this paper is organized as follows. The experimental methods are detailed in Section 2, while Section 3 elucidates the setup and outcomes of the conducted experiments. The conclusions are then assessed in Section 4.

2 Methods

2.1 The Original PSO

PSO is a technique simulating biological populations in terms of sociological perspective for solving multidimensional and nonlinear problems [10]. Thus, a population formed by N particles is set and each particle is designed by D numerical values. By a mathematical way, $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ indicates the i_{th} particle and $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ indicates its velocity vector. Two types of memory are operated in PSO. The first one of them is the best visited position of i_{th} particle indicated mathematically by $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and called as the personal best. The second type is the best visited position of all particles (i.e. swarm) or local neighbors of P_i indicated mathematically by $p_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$ and called as the global best. PSO changes the velocities and indirectly positions of the particles iteratively by using equations (1) and (2).

$$v_{iD} = w * v_{iD} + c_1 * \text{rand1}() * (p_{iD} - x_{iD}) + c_2 * \text{rand2}() * (p_{gD} - x_{iD}) \tag{1}$$

$$x_{iD} = x_{iD} + v_{iD} \tag{2}$$

Where c_1 and c_2 are coefficients affecting personal and global best parts of the equation (1), respectively. The other coefficients of these parts are rand1 and rand2 functions generating random numbers in [0, 1]. D and i indicate dimension and index of the particles, respectively. W is a coefficient used for balancing between personal and global converging. Finally; p_{iD}, p_{gD}, x_{iD} and v_{iD} represent the personal best, global best, current position and velocity of particles, respectively.

2.2 Reverse Direction Supported PSO (RDS-PSO) Algorithms

Although PSO has some superior aspects in optimization problems, it has some shortcomings as well. Two shortcomings of PSO can be summarized; the first one is parameter settings which may result in large performance deviations, and the second one is extremely large losses in diversity at final iterations because of guiding with respect to only personal and global best particles.

The velocities of particles in PSO are updated with respect to only two references, namely personal best and global best vectors. Such an updating causes degradation of diversity among particles, premature convergence and local optimum problem especially in nonlinear problems with many local optimal points [24]. As a result of this way of updating, although in the first iterations a quick convergence is observed, improving in convergence to optimal point is inadequate especially in the last iterations of PSO.

To avoid degradation of diversity in PSO, RDS-PSO algorithms were suggested in our earlier work[25] as a function optimizer. As distinct from PSO, both the global worst and the personal worst vectors were considered to compute updated velocity values of particles in RDS-PSO. Two new constants, namely alpha and beta in [0, 1] interval, were added to the original PSO to design RDS-PSO. These constants provide a trade-off between the personal-global best particles and the personal-global worst particles. When the alpha closes to 0, effect of the global worst on the velocity update equation increases. Due to similar reason, the beta parameter is added to this equation to increase or decrease effect of the personal worst against the personal best vector. When both the alpha and the beta are assigned by 1 in RDS-PSO, a traditional PSO can be derived. Thus, four type of RDS-PSO can be derived as follows:

RDS-PSO_1: In this version, alpha = beta = 1 and it presents the same solution with original PSO.

RDS-PSO_2: This RDS-PSO uses the global worst as a supporting information to the global best by regulating alpha Є [0, 1] while it doesn’t use the personal worst (always beta = 1).

RDS-PSO_3: This RDS-PSO uses the personal worst as supporting information to the personal best by regulating beta Є [0, 1] while it doesn’t use the global worst (always alpha = 1).

RDS-PSO_4: This RDS-PSO type uses both the global worst and the personal worst as supporting information to the global best and the personal best, respectively.

For example, in RDS-PSO_4 both the global and the personal worst particles are inserted into the velocity updating equation as follows.

$$\begin{aligned} v_{iD} &= w * v_{iD} + beta * c_1 * rand1() * (p_{iD} - x_{iD}) + (1 - beta) * c_1 \ldots \\ &* rand1() * (x_{iD} - p_{iwD}) + alpha * c_2 * rand2() * (p_{gD} - x_{iD}) + \ldots \\ &(1 - alpha) * c_2 * rand2() * (x_{iD} - p_{gwD}) \end{aligned} \tag{3}$$

Equation (3) substitutes equation (1) used in PSO for velocity updating. Equation (3) represents general case of velocity updating for RDS-PSO. For example, when we write 1 for alpha in this equation, RDS-PSO_3 can be obtained.

3 Experimental Setup

As it has been mentioned in the introduction part of this study, each methods' performances should not be compared on only one or a few MLP architecture. Thus, an MLP architecture space that was proposed in [3] for a reliable comparison was preferred in this study too. They presented minimum and maximum neuron numbers by R_{min} and R_{max} terms. As it was implemented in study [3], R_{min} was set to one for both the first and the second hidden layers while R_{max} was set to eight and four for the first and the second hidden layers, respectively. Thus, totally 41 variations of model for MLP architecture space came to exist as in [3]. The first and the last numbers, for instance nine and two depend on dataset. We executed four types of RDS-PSO algorithms on 41 different ANN architecture indexes for three medical datasets.

Table 1

Parameter values for all RDS-PSO types

Parameter	Value
Population size	40
Maximal iteration	200/2000
Maximal inertia weight value	1.2
Minimal inertia weight value	0.1
C_1	2.0
C_2	2.0
Dimension	Depended on problem and NN index
Error goal	$1*10^{-6}$
Alpha	Depended on RDS-PSO type
Beta	Depended on RDS-PSO type
Step Size	0.05
Threshold	0.05

Two groups of evaluations were conducted together over different NN and PSO structures in this study. The first group was inspired by study[3] and contains maximal iteration number (200 iterations for shallow and 2000 iterations for deep training), MLP architecture space and medical datasets. The second group was set with respect to RDS-PSO algorithm and contains changing of inertia weight (increasing or decreasing value) and adaptation approach for alpha and/or beta parameter (max-min or cosine) selection in order to expand analysis. Thus; RDS-PSO_2, RDS-PSO_3 and RDS-PSO_4 executed eight times over each data sets and MLP model while RDS-PSO_1 executed four times over the same data sets and MLP model in order to conduct these evaluations. Totally, we got 28 results by four RDS-PSO types for all architectures of each data set. We also 100 times executed each algorithm with different random initial values for all RDS-PSO types as in the reference study [3]. Parameter values for all RDS-PSO types related to these executions are indicated in Table 1. Alpha, Beta, Step Size and Threshold parameters are only available in RDS-PSO algorithms. Alpha and Beta parameters are added to equation of PSO to set a balance between individual/social best and worst values for particles. Step Size and Threshold parameters adjust changing of Alpha and/or Beta parameters' values. For instance, value of Beta is always 1 in RDS-PSO_2 while value of Alpha parameter must be changed adaptively. Cosine or Max-Min approach controls this adaptation automatically by changing value of Alpha parameter. Values of Alpha and Beta parameters are determined adaptively by RDS-PSO algorithm. Step size and Threshold parameters determines how often the values of Alpha and Beta are changed adaptively. Their values were optimized as 0.05 with respect to experimental studies in our previous work [25]. When we set the values lower than 0.05, the computational cost will increase. Conversely, when the values are set higher than 0.05, the computed Alpha and Beta values deteriorate. Other parameters without Alpha, Beta, Step Size, Threshold and Maximal iteration in Table 1 were determined based on the general use in literature.

To contribute a reliable improvement to the reference study [3], data set was separated only two groups; training and testing. By combining training and validation group, only one group, i.e. training set was obtained. Besides, ten-fold cross validation approach was applied for a fair evaluation.

3.1 Datasets and Application Results

3.1.1 Breast Cancer Dataset and Application Results of It

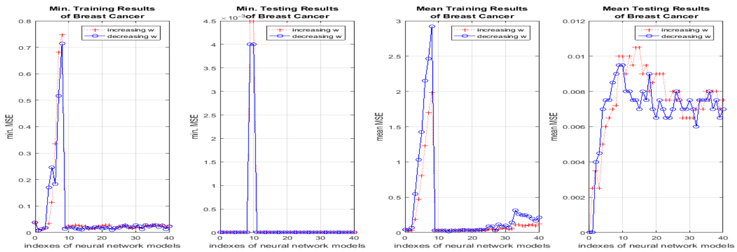

Figure 1
The results of shallow RDS-PSO_1

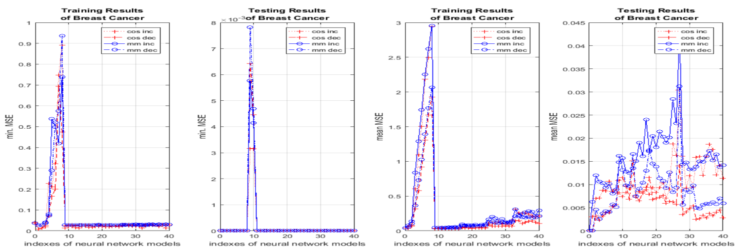

Figure 2
The results of shallow RDS-PSO_2

This data set was created by collecting of 699 samples including 458 benign and 241 malignant types. Important features of breast cancer dataset can be summarized as follows: each datum is consisted with nine inputs and two outputs, whole dataset has been partitioned into three parts of which 350 data for training, 175 data for validation and 174 data for testing process. This study prefers the same organization for partition. We can assert that, for only the mean training results, RDS-PSO_1 shallow with increasing inertia weight is superior to RDS-PSO_1 with decreasing inertia weight in all indexes (Fig 1). However, for minimum training, minimum testing, and mean testing results, it is inconclusive whether one inertia weight is superior to the other in all indexes. Additionally, RDS-PSO_2 shallow with increasing inertia weight has demonstrated almost identical values to those with decreasing inertia weight for minimum training and testing results (Fig 2). The similar comparisons have been observed in RDS-PSO_3 for the minimum training and testing results (Fig 3). In the mean training results of RDS-PSO_2, there is no significant difference between increasing and decreasing inertia weights. However, RDS-PSO_2, employing the cosine adaptation approach, has demonstrated better values than RDS-PSO_2 using the max-min adaptation approach in both mean training and testing results.

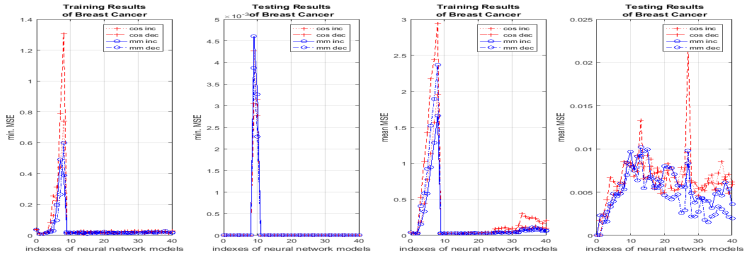

Figure 3

The results of shallow RDS-PSO_3

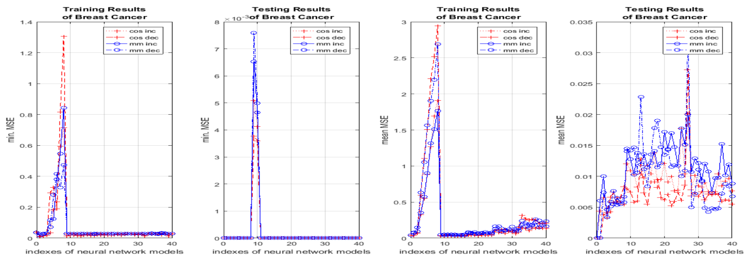

Figure 4

The results of shallow RDS-PSO_4

The mean testing results of RDS-PSO_2 have indicated that decreasing inertia weight is superior to increasing one for both cosine adaptation and max-min adaptation in many of indexes. Contrary to RDS-PSO_2, RDS-PSO_3 using cosine adaptation approach has indicated worse values than RDS-PSO_3 using max-min adaptation one in both the mean training and testing results of RDS-PSO_3. When we examine the results of RDS-PSO_4 shallow for breast cancer data set, there is no significant difference between both increasing and decreasing inertia weights, and cosine and max-min adaptations for the minimum training, minimum testing and the mean training (Fig 4). However, decreasing one is superior to increasing one for the mean testing results.

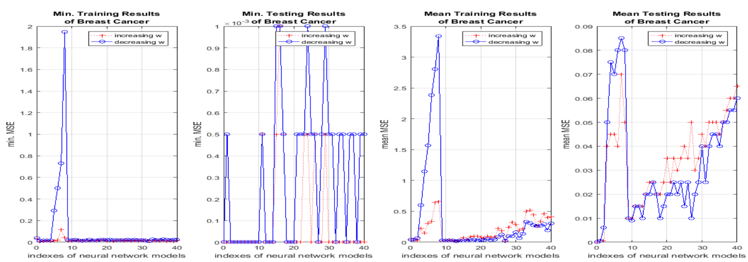

Figure 5

The results of deep RDS-PSO_1

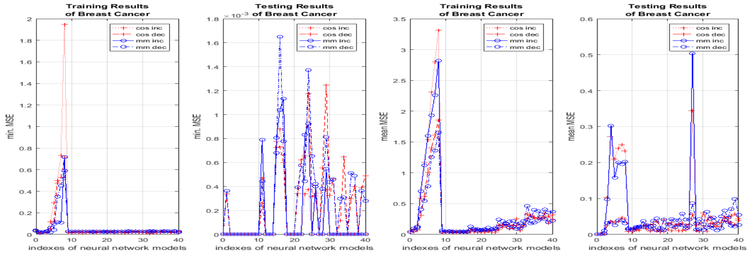

Figure 6

The results of deep RDS-PSO_2

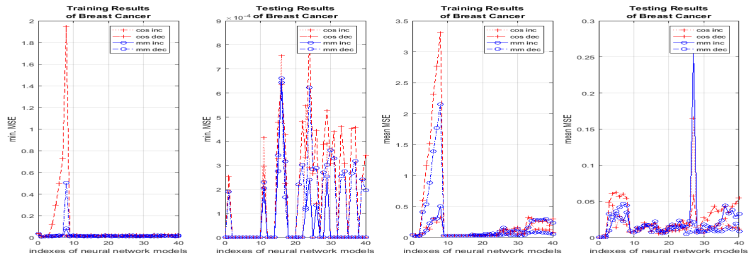

Figure 7

The results of deep RDS-PSO_3

RDS-PSO_1 deep with increasing inertia weight is superior to decreasing inertia weight one in indexes between 0 and 9 for both minimum and mean training/testing results while RDS-PSO_1 deep with decreasing inertia weight is superior or equal in remaining indexes (Fig. 5). In the indexes where considerable fluctuations are available, RDS-PSO_2 with max-min adaptation is superior to cosine adaptation one while there is no important difference between them in other indexes for deep minimum training values (Fig. 6). In these indexes, RDS-PSO_4 with increasing inertia weight types is somewhat worse than decreasing ones (Fig. 8). A similar situation to RDS-PSO_2 is valid for minimum training values of RDS-PSO_3 (Fig. 7). In minimum testing, RDS-PSO_2 with increasing inertia weight is somewhat superior to decreasing one for both max-min and cosine adaptation. On the contrary, the increasing one is somewhat worse than the decreasing one in RDS-PSO_4. In the minimum testing results of RDS-PSO_3, increasing inertia weight is superior to the decreasing one for only a few indexes.

In many of the indexes there is no important difference between increasing inertia weight and decreasing inertia weight for both max-min and cosine adaptation. RDS-PSO_2 with cosine adaptation is superior to max-min in many of indexes for the mean training and mean testing results. Increasing inertia weight with cosine adaptation and decreasing inertia weight with cosine adaptation have indicated somewhat better results than others for deep mean training and the mean testing results of RDS-PSO_3. RDS-PSO_4 with max-min adaptation is superior to the cosine one in the indexes between 0 and 9 while the cosine one is somewhat superior to max-min one in other indexes.

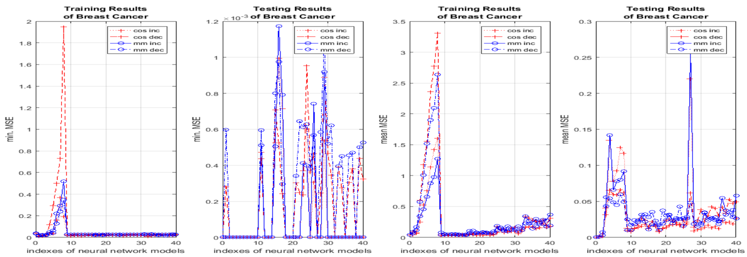

Figure 8

The results of deep RDS-PSO_4

3.2.2 Diabetes Dataset and Application Results of It

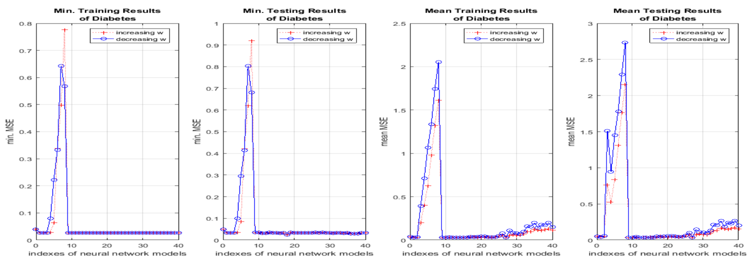

Figure 9

The results of shallow RDS-PSO_1

This dataset includes 768 measurements. 268 samples were labeled with diabetes, while 500 samples without diabetes. Each datum is represented by 1 x 10 vector of which eight values form inputs and two values form outputs. Whole dataset has been partitioned into three parts of which 384 data for training, 192 data for validation and 192 data for testing process. RDS-PSO_1 shallow with increasing inertia weight is generally superior to RDS-PSO_1 with decreasing one without a few indexes for minimum training/testing and mean training/testing (Fig. 9). Except of few indexes for minimum training/testing results, RDS-PSO_2 with cosine adaptation has indicated somewhat better results than max-min adaptation one for minimum training/testing and mean training/testing (Fig. 10). RDS-PSO_3 with max-min adaptation types is superior to cosine adaptation one in nearly all indexes for minimum training/testing and mean training/testing (Fig. 11). RDS-PSO_4 with max-min adaptation is superior to cosine adaptation one except of few indexes for minimum training/testing results (Fig. 12). In mean training/testing, RDS-PSO_4 with increasing inertia weight is superior to RDS-PSO_4 with decreasing one.

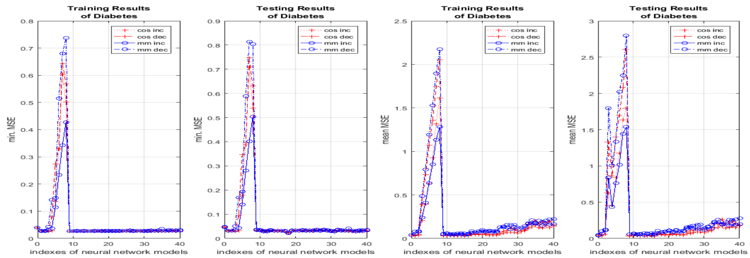

Figure 10

The results of shallow RDS-PSO_2

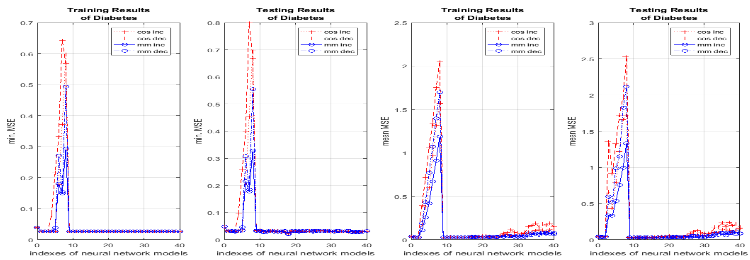

Figure 11

The results of shallow RDS-PSO_3

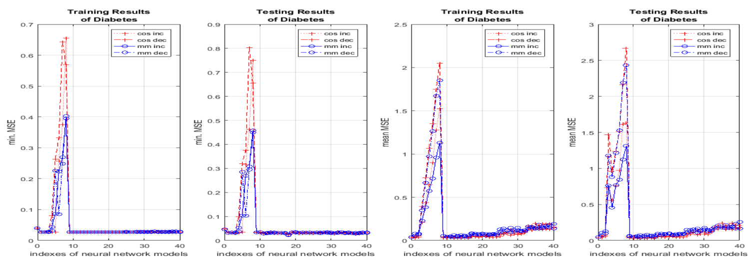

Figure 12

The results of shallow RDS-PSO_4

Except of indexes 3 and 4 of the mean testing results, RDS-PSO_1 deep with increasing inertia weight for diabetes is superior than with decreasing inertia weight one in indexes between 0 and 9 for both minimum and mean training/testing results as in RDS-PSO_1 for breast cancer data set (Fig. 13). In other indexes, RDS-PSO_1 with decreasing inertia weight has indicated similar or better performance than with increasing inertia weight one again.

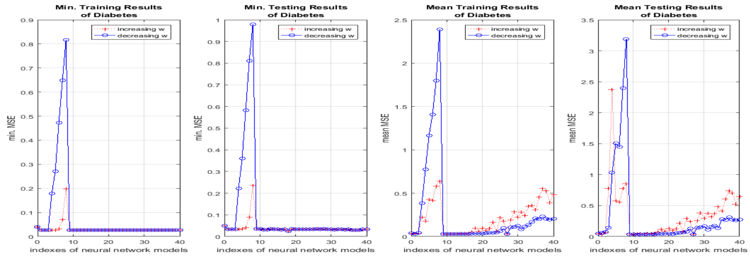

Figure 13

The results of deep RDS-PSO_1

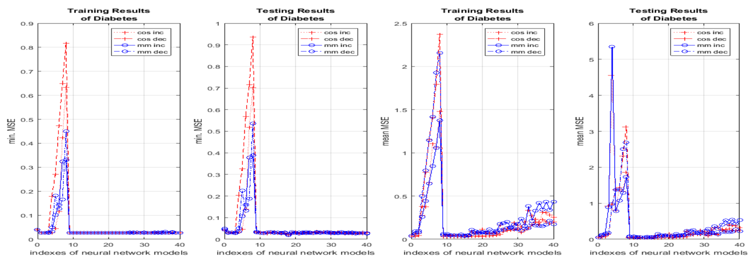

Figure 14

The results of deep RDS-PSO_2

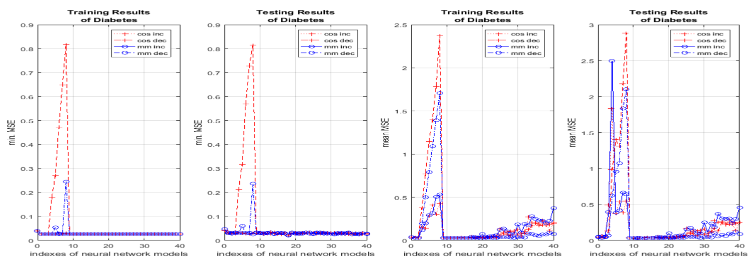

Figure 15

The results of deep RDS-PSO_3

RDS-PSO_2 deep with max-min adaptation is superior to cosine adaptation one for minimum training and testing results (Fig. 14). For mean training and testing results, RDS-PSO_2 with decreasing inertia weight has indicated better values than RDS-PSO_2 with increasing inertia weight one. RDS-PSO_3 types with decreasing inertia weight have indicated worse values than types with increasing inertia weight ones in indexes between 4 and 8 for minimum training and testing results (Fig. 15). In other indexes, we cannot say that one RDS-PSO_3 type is superior to any other RDS-PSO_3 type. In the indexes between 0 and 8, RDS-PSO_3 with increasing inertia weight is superior to the one with decreasing inertia weight while in indexes between 25 and 40 RDS-PSO_3 with decreasing inertia weight and max-min adaptation is superior to other types for the mean training results. In other indexes, we cannot say that one RDS-PSO_3 type is superior to any other RDS-PSO_3 type. There is not a considerable difference between RDS-PSO_3 types in indexes between 0 and 8. On the other hand, RDS-PSO_3 with decreasing inertia weight and max-min adaptation is superior to other types in other indexes for mean testing results. RDS-PSO_4 with increasing inertia weight and max-min adaptation is superior to other types in indexes between 5 and 9 (Fig. 16). In other indexes, we cannot say that there is a considerable difference between one type of RDS-PSO_4 and any other one.

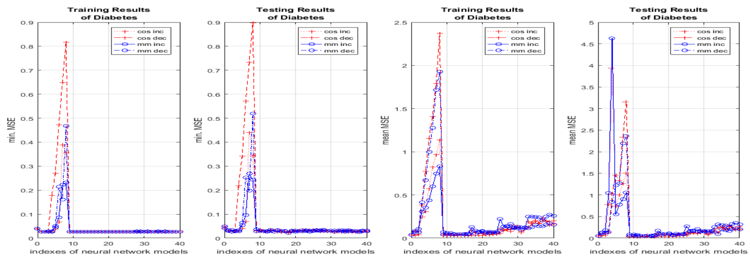

Figure 16
The results of deep RDS-PSO_4

3.2.3 Heart Disease Dataset and Application Results of It

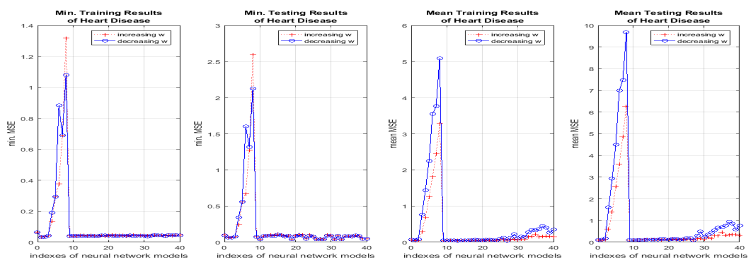

Figure 17
The results of shallow RDS-PSO_1

Original version of heart disease dataset had been created with 920 samples and 35 inputs before the second version was produced by some data preprocessing techniques. Finally, 297 data were remained with 13 inputs and 2 outputs in it. Whole dataset has been partitioned into three parts of which 149 data for training, 74 data for validation and 74 for testing. In minimum training/testing and mean training/testing shallow results of RDS-PSO_1, increasing inertia weight is superior to decreasing one in all indexes without only two exceptions (Fig. 17). RDS-PSO_2 with increasing inertia weight for mean training/testing results have indicated better values than the decreasing inertia weight one however, we cannot say that they are superior to decreasing inertia weight one in all indexes for minimum training/testing shallow results (Fig. 18). In indexes between 3 and 9, RDS-PSO_3 types with increasing inertia weight is superior to decreasing one. In other indexes, max-min adaptation types have indicated better results than the cosine adaptation ones for mean training and testing while there is no superiority between them for minimum training and testing (Fig. 19).

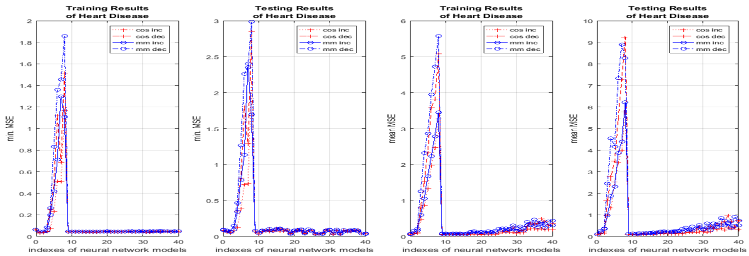

Figure 18

The results of shallow RDS-PSO_2

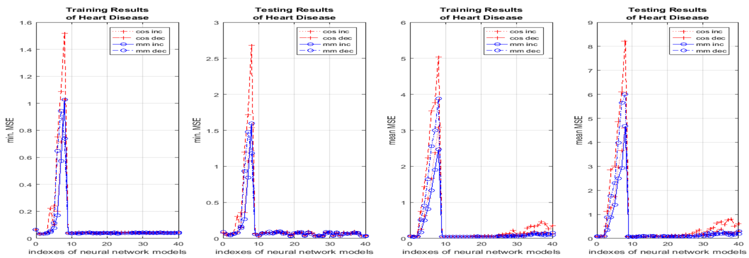

Figure 19

The results of shallow RDS-PSO_3

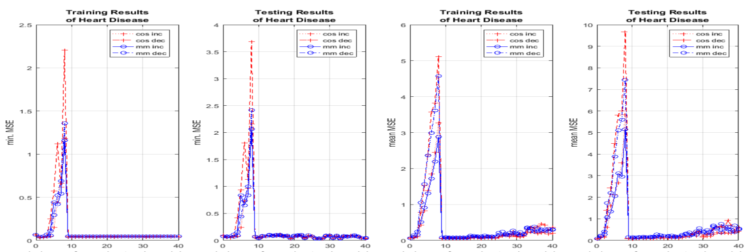

Figure 20

The results of shallow RDS-PSO_4

RDS-PSO_4 types have indicated that decreasing inertia weight with cosine adaptation is worse than other types in indexes between 3 and 9 for minimum training and testing while types with increasing inertia weight is superior to decreasing ones in the same indexes for mean training and testing (Fig. 20).

RDS-PSO_1 deep with increasing inertia weight has less MSE value than the decreasing one in all indexes without one exception for the minimum training and testing results (Fig. 21). The increasing one is superior to decreasing one in the indexes between 0 and 8 for the mean training and testing results while decreasing is superior to increasing one in other indexes for the mean training and testing results.

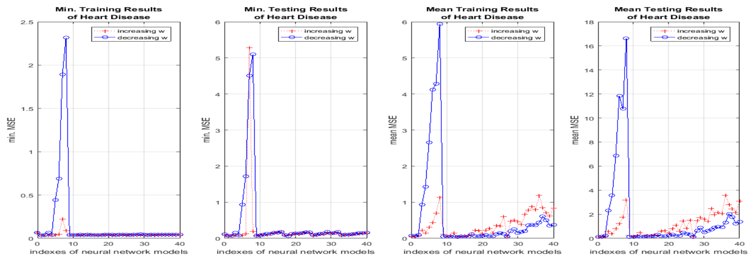

Figure 21
The results of deep RDS-PSO_1

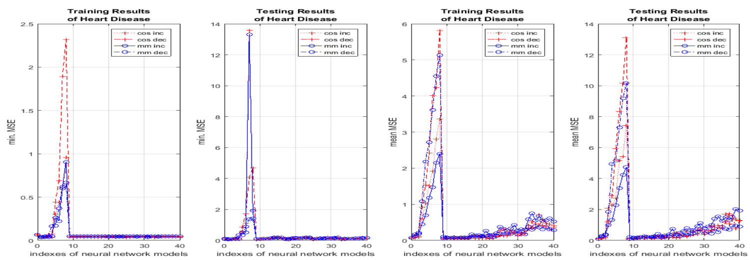

Figure 22
The results of deep RDS-PSO_2

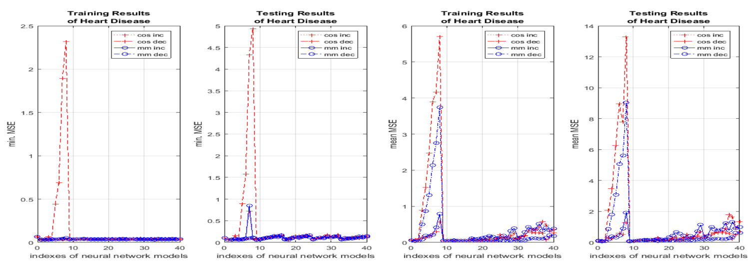

Figure 23
The results of deep RDS-PSO_3

In the indexes between 0 and 8, RDS-PSO_2 types with max-min adaptation have better results than the types with cosine adaptation one whereas in other indexes we cannot say that the max-min adaptation is better than the cosine one (Fig. 22).

Except indexes between 3 and 9 of mean training and testing, RDS-PSO_3 with decreasing inertia weight and max-min adaptation types have indicated the best performance among all deep results of RDS-PSO_3 types (Fig. 23). RDS-PSO_4 types with increasing inertia weight have indicated better results than the types with decreasing ones in indexes between 0 and 8 for the minimum training/testing and the mean training/testing results without one index (Fig. 24).

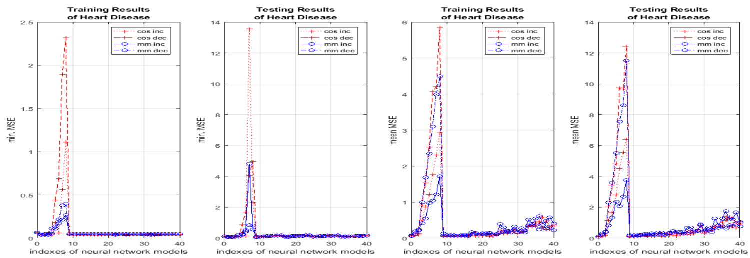

Figure 24
The results of deep RDS-PSO_4

Conclusions

In contrast to most literature studies, the study by [3] compared performances of gradient and heuristic-based methods within an ANN configuration space, encompassing both deep and shallow training depths. Their findings indicate that backpropagation (BP) outperforms particle swarm optimization (PSO) in terms of training errors, while PSO excels over BP in terms of testing errors.

This study presents a comprehensive analysis, evaluating four distinct PSO algorithms, including three improved variants, namely the RDS-PSO types, for training ANN using medical datasets. We have conducted the extended analysis by changing maximal iteration number (200 for shallow and 2000 for deep training), the direction of inertia weight value change (increasing and decreasing), the adaptation approach for alpha and/or beta parameter (max-min or cosine) and the RDS-PSO type (totally four types) criteria. This analysis was repeated 100 times across 41 different feed-forward and fully connected ANN architectures, mirroring the approach taken in the study by [3]. Minimum, average and standard deviation results have been calculated with respect to these criteria. Breast Cancer, Heart Disease and Diabetes datasets have been chosen as medical data sets from Proben1 repository [26].

The overall performance of ANNs is contingent upon factors such as the ANN architecture, training algorithm/parameters, and the dataset, as detailed in the introduction section. Experimental results have proved this idea as indicated in Table 2. Standard deviation values of RDS-PSO types generally higher than BPs values while their minimum values are generally smaller than BPs values. These results have indicated superior searching ability of RDS-PSO types with respect to BP and original PSO.

Table 2
Overall Results of Study
(**1**:RDS-PSO_1; **2**:RDS-PSO_2; **3**:RDS-PSO_3; **4**:RDS-PSO_4)

Data Set	Training method	Training depth	Min. test CE statistics			Mean test CE statistics		
			min	average	sd	min	average	sd
Breast cancer	BP (Ince et. al. 2010)	Shallow	**0**	0.0045	0.0024	0.0003	0.0101	0.0024
	1 inciw		**0**	**0.0002**	0.0010	**0**	0.0074	0.0022
	1 deciw		**0**	**0.0002**	0.0009	**0**	0.0070	**0.0019**
	2inciw_cos		**0**	**0.0002**	**0.0007**	**0**	0.0109	0.0050
	2inciw_mm		**0**	**0.0002**	0.0011	**0**	0.0157	0.0066
	2deciw_cos		**0**	0.0003	0.0012	**0**	0.0058	0.0034
	2deciw_mm		**0**	0.0003	0.0014	**0**	0.0087	0.0055
	3inciw_cos		**0**	**0.0002**	0.0008	**0**	0.0064	**0.0021**
	3inciw_mm		**0**	**0.0002**	0.0009	**0**	**0.0056**	0.0023
	3deciw_cos		**0**	**0.0002**	**0.0007**	**0**	0.0064	0.0032
	3deciw_mm		**0**	**0.0002**	**0.0007**	**0**	**0.0043**	0.0024
	4inciw_cos		**0**	**0.0002**	0.0010	**0**	0.0089	0.0034
	4inciw_mm		**0**	0.0003	0.0012	**0**	0.0111	0.0041
	4deciw_cos		**0**	**0.0002**	0.0008	**0**	0.0071	0.0038
	4deciw_mm		**0**	0.0003	0.0014	**0**	0.0099	0.0060
	BP (Ince et. al. 2010)	Deep	**0**	0.0055	0.0036	**0**	0.0176	**0.0064**
	1 inciw		**0**	**0.0001**	0.0002	**0**	0.0344	0.0177
	1 deciw		**0**	0.0003	0.0003	**0**	0.0323	0.0230
	2inciw_cos		**0**	**0.0001**	0.0002	**0**	0.0537	0.0876
	2inciw_mm		**0**	0.0002	0.0003	**0**	0.0563	0.0968
	2deciw_cos		**0**	0.0003	0.0003	**0**	0.0311	0.0141
	2deciw_mm		**0**	0.0003	0.0004	**0**	0.0376	0.0200
	3inciw_cos		**0**	**0.0001**	0.0002	**0**	**0.0164**	0.0248
	3inciw_mm		**0**	**0.0001**	**0.0001**	**0**	0.0231	0.0387
	3deciw_cos		**0**	**0.0002**	**0.0002**	**0**	0.0275	0.0184
	3deciw_mm		**0**	**0.0002**	**0.0002**	**0**	**0.0147**	0.0112
	4inciw_cos		**0**	**0.0001**	0.0002	**0**	0.0326	0.0436
	4inciw_mm		**0**	**0.0001**	0.0003	**0**	0.0383	0.0440
	4deciw_cos		**0**	**0.0002**	0.0003	**0**	0.0326	0.0436
	4deciw_mm		**0**	0.0003	0.0003	**0**	0.0283	0.0136
Diabetes	BP (Ince et. al. 2010)	Shallow	0.1875	0.2002	**0.0063**	0.2101	0.2175	**0.0112**
	1 inciw		0.0240	0.0801	0.1708	0.0354	0.2419	0.4650

	1 deciw		0.0239	0.0853	0.1654	0.0353	0.3455	0.6407
	2inciw_cos		0.0221	0.0729	0.1398	0.0344	0.2286	0.4055
	2inciw_mm		0.0239	0.0617	0.0984	0.0451	0.2467	0.3502
	2deciw_cos		0.0235	0.0817	0.1560	0.0298	0.3263	0.5972
	2deciw_mm		0.0225	0.0920	0.1858	0.0416	0.3920	0.6542
	3inciw_cos		**0.0219**	0.0625	0.1221	**0.0290**	0.2131	0.4066
	3inciw_mm		0.0225	**0.0472**	0.0572	0.0304	**0.1478**	0.2730
	3deciw_cos		0.0225	0.0818	0.1624	0.0297	0.3157	0.5840
	3deciw_mm		**0.0214**	**0.0547**	0.0926	**0.0289**	**0.2174**	0.4652
	4inciw_cos		0.0237	0.0667	0.1338	0.0373	0.2258	0.3782
	4inciw_mm		0.0237	0.0574	0.0844	0.0418	0.2205	0.2947
	4deciw_cos		0.0235	0.0828	0.1625	0.0294	0.3260	0.6009
	4deciw_mm		0.0222	0.0566	0.0840	0.0439	0.3341	0.5568
	BP (Ince et. al. 2010)	Deep	0.1719	0.1941	0.0129	0.2135	0.2246	**0.0068**
	1 inciw		0.0230	0.0540	0.0822	0.0468	0.3651	0.7470
	1 deciw		0.0228	0.0553	0.0884	0.0466	0.3325	0.6883
	2inciw_cos		0.0214	0.0614	0.1270	0.0463	0.4042	0.7679
	2inciw_mm		0.0218	0.0525	0.0783	0.0480	0.4427	0.8542
	2deciw_cos		0.0223	0.0936	0.1933	0.0329	0.3134	0.6305
	2deciw_mm		**0.0197**	0.0544	0.0863	0.0491	0.3275	0.5942
	3inciw_cos		0.0217	0.0301	**0.0031**	0.0341	**0.1845**	0.2976
	3inciw_mm		0.0220	**0.0300**	0.0038	**0.0321**	0.2343	0.3947
	3deciw_cos		0.0225	0.0908	0.1820	0.0313	0.3047	0.5950
	3deciw_mm		0.0204	**0.0361**	**0.0322**	**0.0271**	**0.2054**	0.4588
	4inciw_cos		**0.0212**	0.0485	0.0788	0.0452	0.3243	0.6597
	4inciw_mm		0.0214	0.0432	0.0495	0.0460	0.3542	0.7226
	4deciw_cos		0.0239	0.0939	0.1915	0.0306	0.3176	0.6399
	4deciw_mm		0.0224	0.0530	0.0852	0.0462	0.2942	0.5241
Heart disease	BP (Ince et. al. 2010)	Shallow	0.1757	0.1928	**0.0100**	0.1893	**0.2222**	**0.0087**
	1 inciw		0.0383	0.1970	0.4374	0.0888	0.6164	1.3091
	1 deciw		0.0378	0.2092	0.4324	0.0880	1.0614	2.1474
	2inciw_cos		0.0323	0.1609	0.3478	0.0731	0.6009	1.1773
	2inciw_mm		0.0332	0.2193	0.4658	0.0905	0.7205	1.2573
	2deciw_cos		0.0314	0.2306	0.5366	0.0737	0.9616	1.9576
	2deciw_mm		0.0343	0.2929	0.6638	0.0956	1.1890	2.1769
	3inciw_cos		0.0316	0.1419	0.2765	0.0650	0.4773	0.9848
	3inciw_mm		**0.0307**	**0.1268**	0.2641	**0.0627**	0.4060	0.9008
	3deciw_cos		**0.0300**	0.2033	0.4971	0.0703	0.8538	1.7167
	3deciw_mm		0.0304	**0.1473**	0.3027	**0.0630**	0.6088	1.3838

	4inciw_cos		0.0308	0.1626	0.3594	0.0700	0.5672	1.1036
	4inciw_mm		0.0337	0.1730	0.3526	0.0879	0.6117	0.9875
	4deciw_cos		0.0317	0.2517	0.6400	0.0657	0.9525	1.9649
	4deciw_mm		0.0343	0.1778	0.3977	0.0735	0.9237	1.6094
	BP (Ince et. al. 2010)	Deep	0.1486	0.1773	**0.0171**	0.2150	**0.2340**	**0.0096**
	1 inciw		0.0608	0.2407	0.7981	0.0861	1.1531	0.9771
	1 deciw		0.0596	0.4100	1.0333	0.0992	1.7152	3.4742
	2inciw_cos		0.0577	0.5351	2.0879	0.0804	1.0637	1.5266
	2inciw_mm		0.0608	0.5117	2.0439	0.0803	1.0311	1.0429
	2deciw_cos		0.0582	0.3831	0.9495	0.0830	1.4125	2.8136
	2deciw_mm		0.0623	0.2207	0.2967	0.0921	1.3032	2.3912
	3inciw_cos		0.0570	**0.1160**	0.0995	0.0765	0.3303	0.2412
	3inciw_mm		**0.0540**	0.1172	0.1183	**0.0579**	0.5060	0.4205
	3deciw_cos		0.0556	0.3924	0.9956	**0.0761**	1.3827	2.7355
	3deciw_mm		0.0557	**0.1072**	0.0296	0.0808	0.7781	1.7757
	4inciw_cos		0.0638	0.5145	2.0913	0.1074	0.9016	1.3963
	4inciw_mm		0.0684	0.2625	0.7237	0.0866	0.8143	0.7804
	4deciw_cos		**0.0549**	0.3884	0.9777	0.0791	1.3820	2.7689
	4deciw_mm		0.0647	0.1784	0.1648	0.0779	1.3148	2.4645

RDS-PSO_3 types have indicated the best results on heart disease data set except of one obtained by RDS-PSO_4 deep training depth with decreasing inertia weight for minimum testing. RDS-PSO_3 with decreasing inertia weight and max-min adaptation has indicated better results than other RDS-PSO_3 types for shallow training depth while increasing inertia weight and cosine adaptation has better than others for deep training depth. RDS-PSO_3 types have also indicated the best results on diabetes data set without two exceptional case including RDS-PSO_2 deep training depth with decreasing inertia weight for minimum testing and RDS-PSO_4 deep training depth with increasing inertia weight for minimum testing. RDS-PSO_3 with decreasing inertia weight and max-min adaptation has the best results among all RDS-PSO_3 types. Beside these data sets, RDS-PSO_3 types have indicated the best results on breast cancer data set again although experimental results of all RDS-PSO types are close each other. In the mean testing, RDS-PSO_3 types have indicated superior performance to all remaining RDS-PSO types for all data sets. RDS-PSO_3 with decreasing inertia weight is superior to increasing one in terms of standard deviation. When we compare the development between shallow and deep training depths, RDS-PSO_3 and RDS-PSO_4 with increasing inertia weight have given the best rates.

As the index number in the ANN architecture space increases, fluctuations in the resulting graphics also increase for ANNs with more than one hidden layer. Conversely, there are substantial fluctuations in ANN results with only one hidden layer, particularly concerning the neuron number.

By combining factors affecting the overall performance of ANNs and extended results of this study indicated in Table 2, the key findings can be summarized as in Table 3.

Table 3
Key findings of the study

Factor affecting the overall performance of ANNs	Relationship with application results
1-Architecture of ANN	While the neuron number in a layer increase for ANNs with more than one hidden layer, a slight increase was seen in fluctuations of minimum values and a greater increase was seen in fluctuations in the mean values.
2-Datasets used in experiments	Based on the experiments in this study, the distances among the results of different algorithms are very close to each other in Breast Cancer dataset. These distances are a bit farther and much farther in Diabetes and Heart Disease datasets, respectively.
3-Training algorithm and its parameters	Although the distances among the results of different algorithms change in different datasets, RDS-PSO_3 generally indicates better results in all datasets. When we compare the results of RDS-PSO_3 versions among themselves, max-min adaptation for parameter setting generally provides equal or better results than cosine one.

The study in [3] has conducted a computational complexity analysis for BP and PSO. They have claimed that PSO algorithm has much greater computational cost than BP algorithm. We have not conducted a computational complexity analysis in this study because there is no considerable difference between computational costs of RDS-PSO types and original PSO.

Researchers have studied on many evolutionary algorithms in literature. As a result, they have introduced many original and improved versions of these algorithms. As a future work, adapting these algorithms into training of various ANN types and architectures will provide both better classification performances and the better comparisons than the existing literature studies. In addition, an adaptive setting method for Step Size and Threshold parameters can be proposed. Furthermore, an adaptive setting method for Alpha and Beta parameters without Step Size and Threshold can be proposed to improve this study.

Acknowledgement

This work was not supported by any institution.

References

[1] Nguyen V. A., Starzyk J. A., Goh W. B. and Jachyra D. Neural network structure for spatio-temporal long-term memory. IEEE Transactions, Neural Network Learning Systems, Vol. 23, No. 6, pp. 971-983, 2012

[2] Lin D., Vasilakos A. V., Tang Y. and Yao Y. Neural networks for computer-aided diagnosis in medicine: A review. Neurocomputing, Vol. 216, pp. 700-708, 2016

[3] Ince T., Kiranyaz S., Pulkkinen J. and Gabbouj M. Evaluation of global and local training techniques over feed-forward neural network architecture spaces for computer-aided medical diagnosis. Expert Systems with Applications, Vol. 37, pp. 8450-8461, 2010

[4] Rumelhart D., Hinton G. and Williams R. Learning internal representation by error propagation. MIT Press Cambridge, MA, USA. Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1, pp. 318-362, 1986

[5] Kiranyaz S., Ince T., Iosifidis A. and Gabbouj M. Progressive Operational Perceptrons. Neurocomputing, Vol. 224, pp. 142-154, 2017

[6] Alneamy J. S. M., Alnaish Z. A. H., Hashim S. Z. M. and Alnaish R. A. H. Utilizing hybrid functional fuzzy wavelet neural networks with a teaching learning-based optimization algorithm for medical disease diagnosis. Computers in Biology and Medicine, Vol. 112, Article Number 103348, 2019

[7] Abusnaina A. A., Abdullah R. and Kattan A. Self-Adaptive mussels wandering optimization algorithm with application for artificial neural network training. Journal of Intelligent Systems, Vol. 29, pp. 345-363, 2020

[8] Perez M. A., Marquez C. Y., Nieto O. C., Yanez I. L. and Cruz A. J. A. Collaborative learning based on associative models: Application to pattern classification in medical datasets. Computers in Human Behavior, Vol. 51, pp. 771-779, 2015

[9] Lisboa P. J. and Taktak A. F. G. The use of artificial neural networks in decision support in cancer: A systematic review. Neural Networks, Vol. 19, No. 4, pp. 408-415, 2006

[10] Kennedy J. and Eberhart R. C. Particle swarm optimization. In Proceedings of the IEEE conference on neural networks, Perth, Australia, pp. 1942-1948, 1995

[11] Meissner M., Schmuker M. and Schneider G. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics, Vol. 7, Article Number 125, 2006

[12] Carvalho M. and Ludermir T. B. Particle swarm optimization of neural network architectures and weights. In Proceedings of the 7th international conference on hybrid intelligent systems, Washington DC, pp. 336-339, 2007

[13] Yu J., Xi L. and Wang S. An improved particle swarm optimization for evolving feedforward artificial neural networks. Neural Processing Letters, Vol. 26, No. 3, pp. 217-231, 2007

[14] Kolen J. F. and Pollack J. B. Back propagation is sensitive to initial conditions. Complex Systems, Vol. 4, pp. 860-867, 1990

[15] Mazurowski M. A., Habas P. A., Zurada J. M., Lo J. Y., Baker J. A. and Tourassi G. D. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks, Vol. 21, pp. 427-436, 2008

[16] Yaghini M., Khoshraftar M. M. and Fallahi M. A hybrid algorithm for artificial neural network training. Engineering Applications of Artificial Intelligence, Vol. 26, pp. 293-301, 2013

[17] UCI Repository of machine learning databases. Department of Information and Computer Science. University of California, Irvine, 2011

[18] Beheshti Z., Shamsuddin S. M. Hj., Beheshti E. and Yuhaniz S. S. Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis. Soft Computing, Vol. 18, pp. 2253-2270, 2014

[19] Misra B. B., Dehuri S., Dash P. K. and Panda G. A reduced and comprehensible polynomial neural network for classification. Pattern Recognition Letters, Vol. 29, pp. 1705-1712, 2008

[20] Cheng B., Stanley R. J., Antani S. K. and Thoma G. R. A novel computational intelligence-based approach for medical image artifacts detection. In Proceedings of the 2010 International Conference on Artificial Intelligence and Pattern Recognition. Orlando, FL, pp. 113-120, July 2010

[21] Sheikhan M. and Jadidi Z. Flow-based anomaly detection in high-speed links using modified GSA-optimized neural network. Neural Computing & Applications, Vol. 24, pp. 599-611, 2014

[22] Leema N., Nehemiah H. K. and Kannan A. Neural network classifier optimization using Differential Evolution with Global Information and Back Propagation algorithm for clinical datasets. Applied Soft Computing, Vol. 49, pp. 834-844, 2016

[23] Sağ T. and Jalil Z. A. J. Vortex search optimization algorithm for training of feed-forward neural network. International Journal of Machine Learning and Cybernetics, Vol. 12, pp. 1517-1544, 2021

[24] Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. In Proceedings of Evolutionary Programming Conference, San Diego, USA, Springer-Verlag, pp. 601-610, 25-27 March 1998

[25] Çomak E. A generalized particle swarm optimization using reverse direction information. Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 24, pp. 639-655, 2016

[26] Prechelt L. Proben1 – A set of neural network benchmark problems and benchmark rules. Technical Report 21/94, Faculty of Information, University of Karlsruhe, Germany, 1994