Soft Comput (2014) 18:2253–2270
DOI 10.1007/s00500-013-1198-0

METHODOLOGIES AND APPLICATION

Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis

Zahra Beheshti · Siti Mariyam Hj. Shamsuddin · Ebrahim Beheshti · Siti Sophiayati Yuhaniz

Published online: 15 December 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract In recent decades, artificial neural networks (ANNs) have been extensively applied in different areas such as engineering, medicine, business, education, manufacturing and so on. Nowadays, ANNs are as a hot research in medicine especially in the fields of medical disease diagnosis. To have a high efficiency in ANN, selection of an appropriate architecture and learning algorithm is very important. ANN learning is a complex task and an efficient learning algorithm has a significant role to enhance ANN performance. In this paper, a new meta-heuristic algorithm, centripetal accelerated particle swarm optimization (CAPSO), is applied to evolve the ANN learning and accuracy. The algorithm is based on an improved scheme of particle swarm algorithm and Newton's laws of motion. The hybrid learning of CAPSO and multi-layer perceptron (MLP) network, CAPSO-MLP, is used to classify the data of nine standard medical datasets of Hepatitis, Heart Disease, Pima Indian Diabetes, Wisconsin Prognostic Breast Cancer, Parkinson's disease, Echocardiogram, Liver Disorders, Laryngeal 1 and Acute Inflammations. The performance of CAPSO-MLP is compared with those of PSO, gravitational search algorithm and imperialist competitive algorithm on MLP. The efficiency of methods are evaluated based on mean square error, accuracy, sensitivity, specificity, area under the receiver operating characteristics curve and statistical tests of *t*-test and Wilcoxon's signed ranks test. The results indicate that CAPSO-MLP provides more effective performance than the others for medical disease diagnosis especially in term of unseen data (testing data) and datasets with high missing data values.

Keywords Artificial neural networks · Multi-layer perceptron network · Hybrid learning · Particle swarm optimization · Centripetal accelerated particle swarm optimization

1 Introduction

Artificial neural networks (ANNs) (Fausett 1994; Gurney 1997) are nonlinear sophisticated modeling techniques inspired by biological nervous systems. ANNs are one of the powerful tools effectively employed in various fields (Anderson 2006) such as time series prediction, classification, pattern recognition, system identification and control, function approximation, signal processing and so on. The performance of ANNs depends on selecting proper architecture and learning algorithm.

During the recent decades, there have been many studies for solving the problem of ANN learning and architecture optimization. The Back-Propagation (BP) algorithm (Rumelhart and McClelland 1986) was introduced and applied widely to train multilayer networks. However, algorithms based on gradient-descent methods like BP easily trap into local minimum (Brent 1991; Gori and Tesi 1992). Also, the proposed traditional learning algorithms suffer from some shortcomings such as slow convergence rate and long learn-

Communicated by D. Liu.

Z. Beheshti (✉) · S. M. Hj. Shamsuddin · S. S. Yuhaniz
Soft Computing Research Group, Faculty of Computing,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
e-mail: bzahra2@live.utm.my

S. M. Hj. Shamsuddin
e-mail: mariyam@utm.my

S. S. Yuhaniz
e-mail: sophia@utm.my

E. Beheshti
Software Research Group, Nowpardaz Engineering Company,
Isfahan, Iran
e-mail: beheshty@gmail.com

ing time. Hence, many researchers have employed meta-heuristic algorithms for ANN learning to overcome the mentioned limitations. They have been used both in training and obtaining appropriate ANN architecture. The main advantages of using these algorithms over the traditional ones are their ability of escaping from local minima and adapting with their environment.

Genetic algorithm (GA) (Tang et al. 1996), artificial immune system (AIS) (Farmer et al. 1986), ant colony optimization (ACO) (Dorigo et al. 1996), particle swarm optimization (PSO) (Kennedy and Eberhart 1995; Shi and Eberhart 1998), artificial bee colony (ABC) (Karaboga 2005), imperialist competitive algorithm (ICA) (Atashpaz-Gargari and Lucas 2007) and gravitational search algorithm (GSA) (Rashedi et al. 2009) are in the class of meta-heuristic algorithms. The study in (Yao 1999) offers a general framework of the algorithms for ANN learning.

Samanta et al. (2003) compared the performance of bearing fault detection using two different classifiers of ANN and support vector machine (SMV) where the classifier parameters were optimized by GA. In the method, the time-domain vibration signals of a rotating machine with normal and defective bearings were processed for feature extraction. The extracted features from original and preprocessed signals were applied as inputs to the classifiers for two-class (normal or fault) recognition. In another research, GA and ANN were developed for loading pattern optimization of advanced gas-cooled reactors (AGR) (Ziver et al. 2004). Oreski et al. (2012) proposed a hybrid system of GA and ANN for retail credit risk assessment. The method provided a feature selection technique for finding an optimum feature subset to improve the accuracy of ANN classification.

Yao et al. (2000) applied the hybrid learning of PSO and feed-forward neural network (FFNN) to classify two problems in medical domain. The result showed that the method has better accuracy in classified data compared to other algorithms (Bennett and Mangasarian 1992; Setiono and Hui 1995; Prechelt 1995). Also, an adapted particle swarm optimization (PSO) model was offered to train MLP for predicting the outcome of construction claims in Hong Kong (Chau 2007). The successful prediction rate was up to 80 % and the method had more accurate results than its counterparts of a benchmarking BP. In other study, Huang et al. (2010) proposed an ANN classifier with entropy based feature selection for Breast Cancer diagnosis. Levenberg–Marquardt (LM) was used for training and PSO algorithm was applied for obtaining the proper weights of ANN.

Further, a hybrid approach based on ACO and ANN (Socha and Blum 2007) was presented for feature selection of several medical datasets. The proposed method had better performance than BP and GA for ANN. In another study, a hybrid version of continuous ACO with classical gradient techniques was applied to train ANN for medical pattern classification (Sivagaminathan and Ramakrishnan 2007).

Moreover, a comparative study was conducted on chronic obstructive pulmonary and pneumonia diseases diagnosis using the hybrid of AIS and different ANN classification models (Er et al. 2009). The results of the study were compared with those of the pervious similar studies reported focusing on the diseases diagnosis (Ashizawa et al. 1999; Heckerling et al. 2004; Coppini et al. 2007). It was concluded that AIS and ANN could be successfully used to diagnose chronic obstructive pulmonary and pneumonia diseases. A hybrid learning of MLP and AIS also was presented to predict the output from a grid-connected photovoltaic system (Sulaiman et al. 2012). In the proposed method, AIS was selected as the optimizer for the training process of MLP to optimize ANN architecture.

Besides, Ozkan et al. (2011) designed an ANN for modeling daily reference evapotranspiration (ET_0) using ABC algorithm. The accuracy of proposed model was compared with those of the ANN models trained with LM, standard BP algorithms and some existing empirical models (Kisi and Ozturk 2007). The proposed model had better performance in estimation of ET_0.

Ahmadi et al. (2013) offered a model based on MLP to predict permeability of the reservoir and used ICA to decide the initial weights of the network. The performance and the generalization capability of proposed method were compared with those of models developed with the common technique of BP. The results showed that the method outperformed the gradient descent-based neural network. Also, Tayefeh-Mahmoudi et al. (2013) proposed a method to evolve ANN architecture using grammar encoding and ICA. The hybrid method was evaluated on four known regression problems and compared against the state of the art methods: GA, resilient BP, Minimum Finder (MinFinder), Broyden–Fletcher–Goldfarb–Shanno (BFGS), and neural network construction (NNC) as optimization methods (Tsoulos et al. 2008). The experimental results showed that the proposed grammar encoding method using ICA outperformed the other methods.

Although many meta-heuristics have been extensively applied in ANN learning, none of them has shown a good performance for all applications. Also, the existing meta-heuristics have several disadvantages (Leung et al. 1997; Hrstka and Kučerová 2004; Liang et al. 2006; Gao et al. 2012; Moslemipour et al. 2012) such as slow convergence speed, trapping into local minima, long computational time, tuning many parameters and difficult encoding scheme. Hence, the performance enhancement of previous meta-heuristics or proposing new ones would seem necessary to increase the efficiency of ANN learning in different domains.

In this paper, the CAPSO algorithm (Beheshti and Shamsuddin 2013a) is used to improve ANN learning. Several medical datasets (Kuncheva 2005; Bache and Lichman 2013) are selected in order to evaluate the proposed method. The performance metrics of mean square error (MSE), correct classification rate or accuracy (ACC), sensitivity, specificity and area under the receiver operating characteristics curve (AUC) are employed to compare the results of proposed algorithm with PSO, GSA and ICA on MLP learning.

The structure of this paper is organized as follows. Section 2 briefly describes a background of MLP and blending some meta-heuristic algorithms used in this study for MLP learning. The details of proposed approach are elucidated in Sect. 3. Section 4 demonstrates the experimental results of neural network classification on medical datasets using the algorithms. Finally, a conclusion is given in Sect. 5.

2 Background materials

The basic background materials required for a deep understanding of the proposed method are elucidated in this section. At first, MLP as a FFNN is described after that the algorithms of PSO, GSA and ICA are explained in order to compare their results with the proposed algorithm for MLP learning in the classification problems. Later, the classification problem is briefly explained.

2.1 MLP network

FFNN is one of the most common ANNs and maintains a high level of research interest because of its ability to map any function to an arbitrary degree of accuracy. This kind of network has been used for both MLP network (Hornik et al. 1989) and radial basis function (RBF) network (Park and Sandberg 1991). Due to their similarity in functional mapping ability, MLP and RBF networks share the same problem domains.

Multi-layer perceptron has been utilized in a broad range of science such as finance, medicine, engineering, geology, physics and biology. In medicine, it has been extensively employed in medical diagnosis, detection and evaluation of medical conditions and treatment cost estimation (Handels et al. 1999; Folland et al. 2004; Karabatak and Ince 2009; Ari and Saha 2009; Jerez et al. 2010; Abbod et al. 2011). Also, it has been used in replacing conventional pattern recognition methods for the disease diagnosis systems (Kayaer and Yildirim 2003; Delen et al. 2005; Temurtas 2007; Elveren and Yumuşak 2011). As Fig. 1 demonstrates, MLP contains one input layer which receives external inputs, one or more hidden layers, and one output layer which shows the results. All layers except the input layer are made of processing nodes and activation function. Data presented at the input layer and the network nodes carry out calculations in the sequential layers until an output value is acquired at each output node.

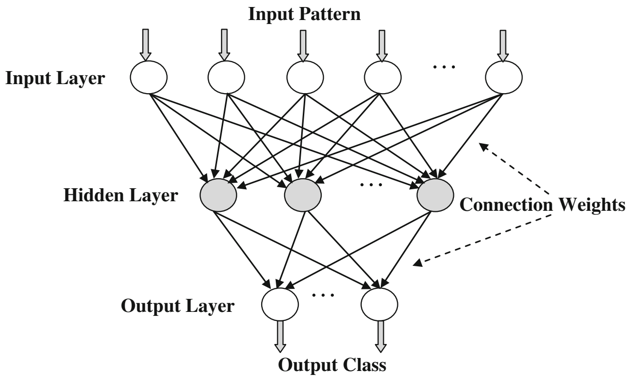

Fig. 1 Architecture of MLP network

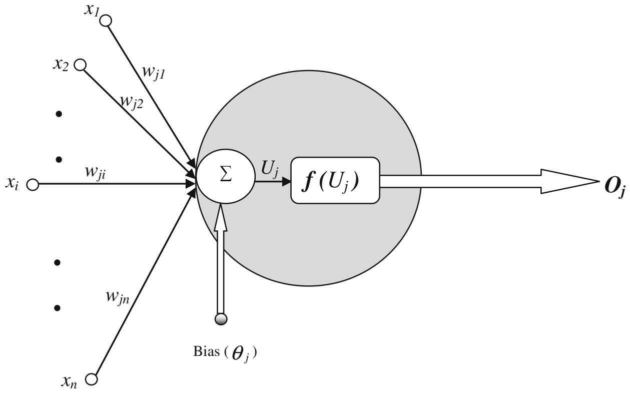

Fig. 2 An artificial neuron of MLP

Neurons are the processing units of MLP. An artificial neuron has been simulated in Fig. 2. As shown, each neuron or node computes the sum of the inputs weighted at the presence of a bias, and passes this sum through an activation function (like sigmoid function) so that the output is obtained. This process can be expressed as Eq. (1).

$$U_j = \sum_{i=1}^{n} w_{ji} x_j + \theta_j,$$
$$O_j = f_j(U_j) \tag{1}$$

where w_{ji} is the weight connected between node i and j, U_j is the linear combination of inputs, O_j and θ_j are the output and bias of node j respectively and $f(.)$ is an activation function.

f_j (.) is usually considered as a sigmoid function and O_j is represented by:

$$O_j = f_j(U_j) = \frac{1}{(1 + e^{-U_j})}. \tag{2}$$

Proper design of MLP architecture is very complex because many elements affect the performance of MLP such as the number of nodes in input, hidden and output layer, distribution of layers, interconnection between neurons and layers (weights), error function and activation function.

Also, selecting a proper learning algorithm is another problem of using MLP. In the following sections, some meta-heuristic algorithms applied in MLP learning are explained and finally their performances are compared together.

2.2 Particle swarm optimization

Particle swarm optimization is a population based meta-heuristic algorithm inspired by social behavior of bird flocking or fish schooling. In the algorithm, each solution is called particle and particles can be presented by a group of vectors as $(\vec{X}_i, \vec{V}_i, \vec{P}_i)$ in a *d-dimensional* search space, where $\vec{X}_i$, $\vec{V}_i$ and $\vec{P}_i$ are the position, velocity and the personal best position of the ith particle respectively. The vectors are defined as follows:

$$\vec{X}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \quad for\ i = 1, 2, \ldots, N, \tag{3}$$

$$\vec{V}_i = (v_{i1}, v_{i2}, \ldots, v_{id}) \quad for\ i = 1, 2, \ldots, N, \tag{4}$$

$$\vec{P}_i = (p_{i1}, p_{i2}, \ldots, p_{id}) \quad for\ i = 1, 2, \ldots, N, \tag{5}$$

where N is the number of particles.

The velocity and position of particles are randomly initialized and the next velocity and position of every particle are computed as Eqs. (6) and (7):

$$v_{id}(t+1) = w(t) \times v_{id}(t) + C_1 \times rand \times (p_{id}(t) - x_{id}(t)) + C_2 \times rand \times (p_{gd}(t) - x_{id}(t)), \tag{6}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{7}$$

where $v_{id}(t)$ and $x_{id}(t)$ are the current velocity and position of ith particle respectively. $w(t)$ is inertia weight, C_1 and C_2 are acceleration coefficients. *rand* is uniformly random number in the interval of [0, 1]. In addition, $v_{id}(t) \in [-v_{\max}, v_{\max}]$ and v_{max} is set to a constant based on the bounds of the solution space by users. In Eq. (6), the second and the third terms are called cognition and social term respectively. p_{gd} is the best position achieved by the entire swarm so far in the dth dimension. For the population, this position is defined as follows:

$$\overrightarrow{Pg} = (p_{g1}, p_{g2}, \ldots, p_{gd}). \tag{8}$$

Although PSO is easy to implement, it has several disadvantages due to its poor exploration, such as facing the slow convergence rate, selecting parameters and getting easily trapped in a local optimum (Leung et al. 1997; Hrstka and Kučerová 2004; Liang et al. 2006; Gao et al. 2012; Moslemipour et al. 2012).

2.3 Gravitational search algorithm

GSA is a meta-heuristic algorithm inspired by the Newtonian gravity and motion laws. According to the gravity law (Schutz 2003), all particles in the universe attract each other by a gravity force. This force is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. Therefore, GSA defines a system with N agents (masses) based on the law in a *d-dimensional* search space and at the beginning, agent positions are initialized randomly. The position of the ith agent is illustrated as follows:

$$\vec{X}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \quad for\ i = 1, 2, \ldots, N, \tag{9}$$

where x_{id} is the position of the ith agent in the dth dimension.

Each mass presents a solution and the heaviest mass shows an optimum solution in the search space. By lapse of time, masses are attracted by the heaviest mass. The gravitational force of mass j on mass i at a specific time t is computed as follows:

$$F_{ij,d}(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}(x_{jd}(t) - x_{id}(t)), \tag{10}$$

where M_i and M_j are the masses of agent i and agent j respectively. ε is a small constant. $G(t)$ is gravitational constant at time t initialized at the beginning and decreased by time t to control the search accuracy. R_{ij}(t) is the Euclidean distance between probe i and j at time t:

$$R_{ij}(t) = \left\| x_i(t), x_j(t) \right\|_2. \tag{11}$$

M_i in Eq. (10) is calculated as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \tag{12}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}, \tag{13}$$

where $fit_i(t)$ is the fitness value of the agent i, *best(t)* and *worst(t)* are the best and the worst values of fitness functions at time t.

Also, the total force acting on agent i at time t in the dth dimension is computed by Eq. (14):

$$F_{id}(t) = \sum_{j \in kbest, j \neq i}^{N} rand_j \times F_{ij,d}(t), \tag{14}$$

where $rand_j$ is a random number in the range of [0, 1]. *Kbest* is a function of time, with the initial value K_0 at the beginning and it is linearly decreased step by step. In fact, *Kbest* is the set of first K agents with the best fitness value and the biggest mass.

The force accelerates the agent i as Eq. (15) and the agent moves from current position to new position:

$$a_{id}(t) = \frac{F_{id}(t)}{M_i(t)}, \tag{15}$$

$$v_{id}(t+1) = v_{id}(t) + rand_i \times a_{id}(t), \tag{16}$$

$$x(t+1) = x_{id}(t) + v_{id}(t+1), \tag{17}$$

where $a_{id}(t)$ represents the acceleration of the ith agent, $v_{id}(t)$ and $x_{id}(t)$ are the current velocity and position of the ith agent. Also, $v_{id}(t + 1)$ and $x_{id}(t + 1)$ show the next velocity and position of the agent i in dimension d at time t respectively.

Similar to other meta-heuristic algorithms, GSA also has some weaknesses such as using complex operators and taking long time to run (Beheshti and Shamsuddin 2013a, b).

2.4 Imperialist competitive algorithm

ICA is a global search optimization algorithm inspired by imperialistic competition. The algorithm is population-based including countries and imperialists. Each individual of the population is called a country. Some of the best countries with the best cost are chosen as imperialist states and others named colonies are divided among the imperialists according to their costs. Empires are formed by imperialist states and their colonies. After forming the empires, imperialistic competition is started among them to collapse weak empires and to remain the most powerful empire. To divide the colonies among the imperialists, the cost and power of each imperialist are normalized and initial colonies are allocated to the empires as Eqs. (18)–(20):

$$C_n = c_n - \max_i \{c_i\}, \tag{18}$$

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right|, \tag{19}$$

$$N \cdot C_n = round\{p_n . N_{col}\}, \tag{20}$$

where c_n is the cost of the nth imperialist and C_n and p_n are the normalized cost and power of the imperialist respectively. Also, $N.C_n$ is the initial number of colonies related to the nth empire and N_{col} is the total number of initial colonies.

To form the nth empire, the $N.C_n$ of the colonies are randomly selected and allocated to the nth imperialist. In real world, the imperialist states try to make their colonies as a part of themselves. This process called assimilation is modelled by moving all of the colonies toward the imperialist as illustrated in the Fig. 3. In this figure, θ and x are random angle and number with uniform distribution respectively, also; d is the distance between the imperialist and colony.

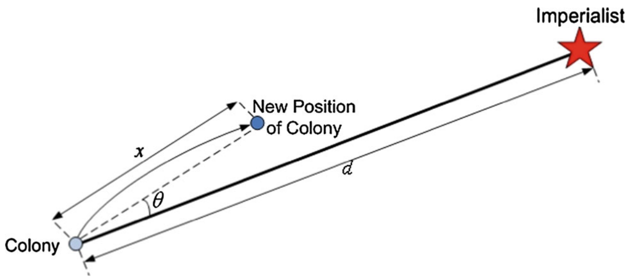

Fig. 3 Movement of colonies toward their relevant imperialist

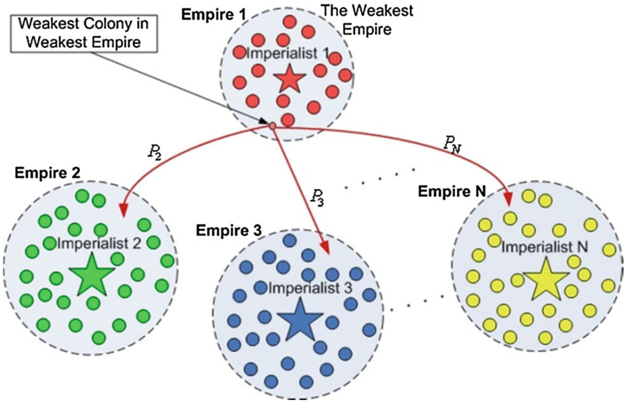

Fig. 4 Imperialistic competition

$$x \sim U(0, \beta \times d), \tag{21}$$

$$\theta \sim U(-\gamma, \gamma), \tag{22}$$

where β, γ are parameters which cause colonies to move their relevant imperialist in a randomly deviated direction.

While a colony moves toward an imperialist, it might achieve a better position than the imperialist. In this case, the colony and the imperialist change their positions with each other. Also, it is possible that a revolution takes place among colonies which changes the power or organizational structures of the empire. This revelation is based on a revolution rate which shows the percentage of colonies that randomly change their position.

Finally, empires compete together in order to possess and control other empires' colonies as shown in Fig. 4. A colony of the weakest empire is selected and the possession probability of each empire, Pp, is computed as Eq. (24). The normalized total cost of an empire, $N.T.C_n$, is acquired by Eq. (23) and used to obtain the empire possession probability.

$$N.T.C_n = T.C_n - \max_i \{T.C_i\}. \tag{23}$$

$$p_{p_n} = \left| \frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_i} \right|. \tag{24}$$

Vector P is formed in order to divide the mentioned colonies among empires:

$$P = \left[p_{p_1}, p_{p_2}, p_{p_3}, \ldots, p_{p_{N_{imp}}}\right]. \tag{25}$$

Vector R with the same size as P whose elements are uniformly distributed random numbers is created:

$$\begin{aligned} &R = \left[r_1, r_2, r_3, \ldots, r_{N_{imp}}\right] \\ &r_1, r_2, r_3, \ldots, r_{N_{imp}} \in U(0, 1). \end{aligned} \tag{26}$$

Vector D is formed so that the mentioned colony (colonies) is given to an empire whose relevant index in D is maximized.

$$D = P - R = \left[D_1, D_2, D_3, \ldots, D_{N_{imp}}\right]. \tag{27}$$

According to vector D, the process of choosing an empire is similar to the roulette wheel in GA. However, this method is faster because the selection is based on probabilities values.

The competition affects the power of empires and an empire power will be weaker or stronger. Therefore, all colonies of the weak empire are owned by more powerful empires and the weaker one is eliminated. The total power (cost) of an empire is modelled by adding the power of imperialist country (cost) and a percentage of mean power of its colonies (colonies costs) as:

$$\begin{aligned} T.C_n = &\, Cost(imprialist_n) \\ &+ \xi \; mean\left\{Cost(colonies\; of empire_n)\right\}, \end{aligned} \tag{28}$$

where $T.C_n$ is the total cost of the nth empire and ξ is a positive small number.

The competition will go on until remaining one empire or reaching determined maximum iteration. ICA also has some drawbacks such as using complex operators, tuning several parameters and taking long time for running (Beheshti and Shamsuddin 2013a, b). These limitations do not allow that ICA is easily used in different problems.

The described algorithms are applied for MLP learning to classify some medical datasets. In the next section, classification concept is explained to utilize in medical diseases diagnosis.

2.5 Classification

Classification is used to understand the existing data and to predict how new instances will behave. In other words, the objective of data classification is to classify data in different classes. Classification techniques include neural networks (Bishop 1995), decision tree induction (Breiman et al. 1984; Quinlan 1993) and support vector machines (Cristianini and Shawe-Taylor 2000). Researchers have employed the methods to classify different problems (Chen et al. 2007; Mitra and Wang 2008; Reynolds and Iglesia 2009; Saxena and Saad 2007; Qasem and Shamsuddin 2011; Awad and Motai 2008; García et al. 2009; Fana et al. 2011; Ekici 2012).

The correct classification is defined by the number of true positives (TP) and the number of true negatives (TN), whereas the misclassification is defined by the number of false positives (FP) and the number of false negatives (FN). Also, the accuracy of predicting events is measured by Sensitivity computed as *TP/total actual positive* and Specificity is a measure of accuracy for predicting nonevents based on *TN/total actual negative of a classifier* for a range of cutoffs. Receiver operating characteristic (ROC) curve is another concept to analyze the performance of algorithms in classification problem. ROC is a graphical plot to demonstrate the quality of classifiers. In other words, the curve shows the true positive rate (sensitivity) and false positive rate (1-specificity). AUC is the area under the curve (ROC). Since, the AUC is a portion of the area of the unit square, its value is between 0.0 and 1.0. Analyses of sensitivity and specificity of algorithms are carried out using the following equations (Fawcett 2006):

$$Sensitivity = \frac{TP}{TP + FN}(\%), \tag{29}$$

$$Specificity = \frac{TN}{TN + FP}(\%), \tag{30}$$

where $TP + FN$ is the number of positive patterns and $TN + FP$ is the number of negative patterns in a dataset.

According to the concept, *Accuracy* can be defined as follows:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}(\%) \tag{31}$$

In the next section, a new meta-heuristic algorithm is introduced to enhance the ANN learning and used to train MLP for classification problems.

3 Methods

This section introduces CAPSO as a new global optimization algorithm to improve the ANN accuracy and follows by proposing the hybrid learning of CAPSO and MLP (CAPSO-MLP).

3.1 CAPSO algorithm

The CAPSO algorithm is an efficient optimization algorithm utilizing mechanics motion laws and PSO algorithm. According to the law, if an object with initial velocity v_1 moves from position x_1 to position x_2 and its velocity modifies to v_2 during the time step of Δt, the object will be accelerated. Therefore, the new velocity and position are obtained as follows:

$$v_2 = v_1 + a \times \Delta t, \tag{32}$$

$$x_2 = x_1 + \frac{1}{2} \times a \times \Delta t^2 + v_1 \times \Delta t, \tag{33}$$

where a is the object acceleration.

Regarding the concept, each solution candidate, called a particle, in CAPSO algorithm has four specifications: position, velocity, acceleration and centripetal acceleration. If a *d-dimensional* search space with N particles is considered, the position and velocity of the ith particle are as follows:

$$\vec{X_i} = (x_{i1}, x_{i2}, \ldots, x_{id}) \quad \textit{for } i = 1, 2, \ldots, N. \tag{34}$$

$$\vec{V_i} = (v_{i1}, v_{i2}, \ldots, v_{id}) \quad \textit{for } i = 1, 2, \ldots, N. \tag{35}$$

Also, the personal best position of the ith particle, $\vec{Pi}$, and the best position explored by population so far, $\vec{Pg}$, are illustrated using Eqs. (36) and (37):

$$\vec{P_i} = (p_{i1}, p_{i2}, \ldots, p_{id}) \quad \textit{for} \quad i = 1, 2, \ldots, N. \tag{36}$$

$$\vec{Pg} = (p_{g1}, p_{g2}, \ldots, p_{gd}). \tag{37}$$

Every particle updates its velocity based on current velocity, acceleration and centripetal acceleration:

$$v_{id}(t+1) = v_{id}(t) + a_{id}(t) + A_{id}(t), \tag{38}$$

where $vi_d(t+1)$ and $v_{id}(t)$ are the next and current velocity respectively.

Additionally, $a_{id}(t)$ and $A_{id}(t)$ represent the acceleration and centripetal acceleration as follows:

$$a_{id}(t) = rand \times (p_{id}(t) - x_{id}(t)) + rand \times (p_{gd}(t) - x_{id}(t)), \tag{39}$$

$$A_{id}(t) = E_i(t) \times rand \times (p_{id}(t) - p_{md}(t) - x_{id}(t)), \tag{40}$$

where *rand* is a random number with uniform distribution in the interval of [0, 1]. $x_{id}(t)$ is the current position, $p_{id}(t)$ shows the personal best position of the ith particle and $p_{gd}(t)$ is the global best position explored in the dth dimension. Also, $p_{md}(t)$ is the current median position of the swarm in dimension d and $E_i(t)$ is acceleration coefficient computed by Eq. (42). Moreover, according to Eq. (33) the next position of the ith particle is acquired by Eq. (43) as follows:

$$e_i(t) = fit_i(t) - GWfit(t), \tag{41}$$

$$E_i(t) = \frac{e_i(t)}{\sum_{j=1}^{N} e_j(t)}, \tag{42}$$

$$x_{id}(t+1) = x_{id}(t) + \tfrac{1}{2} \times a_{id}(t) + v_{id}(t+1) \tag{43}$$

where $fit_i(t)$represents the fitness value of particle i and $GWfit(t)$ is the worst fitness value explored so far by the swarm.

3.2 Analysis and design of the proposed algorithm

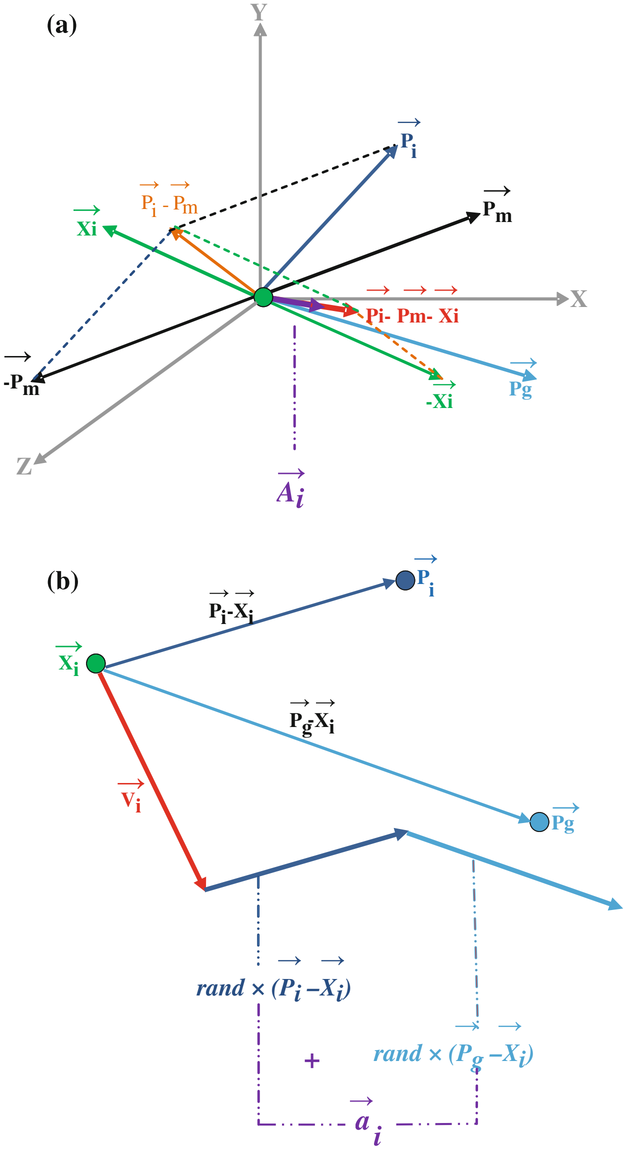

Fig. 5 Graphical representation of **a** $\vec{A}_i$ and **b** $\vec{a}_i$

In CAPSO algorithm, two terms of acceleration, $\vec{a}_i$, and centripetal acceleration, $\vec{A}_i$, play a key role to increase the convergence rate. As shown in Fig. 5, if a particle is far from the solutions obtained so far by the swarm ($\vec{P}_g$, $\vec{P}_i$, $\vec{P}_m$), these terms help to move the particle toward the solutions. Also, $\vec{A}_i$ helps to escape the algorithm from local optima. Fig. 5a demonstrates the term of $\vec{A}_i$ computed based on Eq. (40) and in Fig. 5b, the graphical representation of $\vec{a}_i$ has been illustrated according to Eq. (39).

Figure 6 demonstrates the flowchart of proposed method and the CAPSO pseudo code has been illustrated in Fig. 7. As shown in these figures, the velocities and positions of particles are randomly initialized and each particle is evaluated based on its fitness value. $\vec{P_i}$ is computed for each particle and the best position explored by the swarm is chosen as $\vec{Pg}$. The next particles velocity and position will be acquired

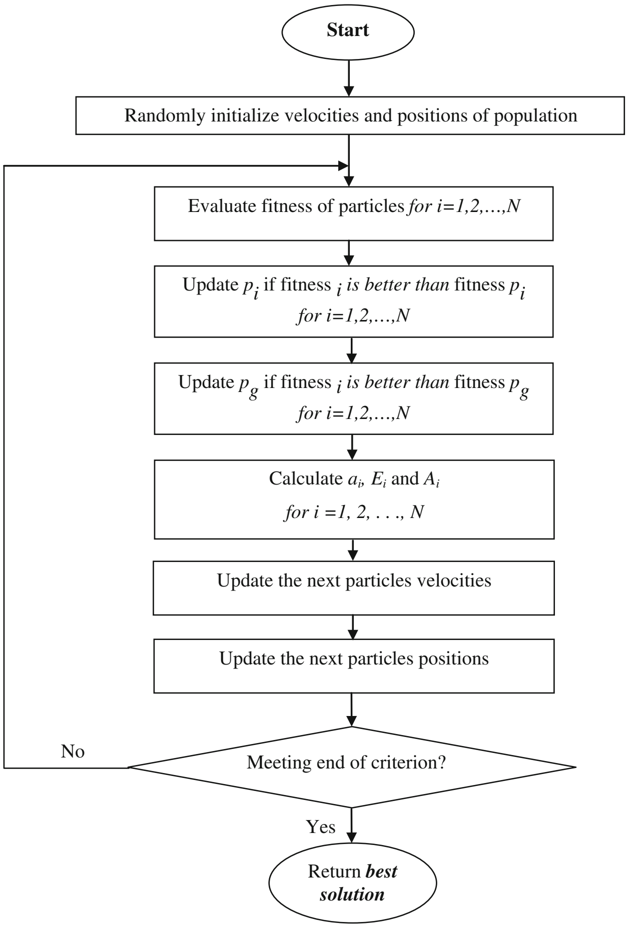
Start
Randomly initialize velocities and positions of population
Evaluate fitness of particles for i=1,2,…,N
Update p_i if fitness $_i$ is better than fitness p_i for i=1,2,…,N
Update p_g if fitness $_i$ is better than fitness p_g for i=1,2,…,N
Calculate a_i, E_i and A_i for i =1, 2, . . ., N
Update the next particles velocities
Update the next particles positions
Meeting end of criterion?
No
Yes
Return best solution

Fig. 6 Flowchart of CAPSO

```
Step 1:  Starts.
Step 2:  Initialize the velocities and positions of population randomly.
Step 3:  Evaluate fitness values of particles.
Step 4:  Update Pi if particle fitness value $_i$ is better than particle best fitness
         value $_i$, for i = 1, 2, . . ., N.
Step 5:  Update Pg if particle fitness value $_i$ is better than global best fitness
         value, for i = 1, 2, . . ., N.
Step 6:  Calculate $a_i$, $E_i$ and $A_i$ for i = 1, 2, . . ., N.
Step 7:  Update the next velocities of particles.
Step 8:  Update the next positions of particles.
Step 9:  Repeat steps 3 to 8 until the stop criterion is reached.
Step 10: Return the best solution.
```

Fig. 7 CAPSO pseudo code

based on Eqs. (38)–(43). These steps will go on until stopping criterion is met and the best solution is returned by the algorithm.

3.3 Hybrid learning of CAPSO and MLP network

In this section, CAPSO is used for MLP learning. The approach is the hybrid learning of CAPSO and MLP network (CAPSO-MLP) to enhance the ability of the network in term of accuracy. The algorithm will simultaneously determine the set of weights and its corresponding accuracy by training the network. The MLP network can be represented as a vector with dimension D containing the network weights. The vector for MLP is defined as Eq. (44). To optimize the MLP weights using CAPSO algorithms, the dimension of each particle is considered as the vector D:

$$D = (Input \times Hidden) + (Hidden \times Output) + Hidden_{bias} + Output_{bias}, \tag{44}$$

where *Input*, *Hidden* and *Output* are referred the number of input, hidden and output neurons of MLP network respectively. Also, $Hiddenb_{ias}$ and $Output_{bias}$ are the number of biases in hidden and output layers.

Figure 8 shows the flowchart of hybrid leaning of CAPSO and MLP (CAPSO-MLP). The CAPSO-MLP is started by collecting, normalizing and reading a dataset. This is followed by setting the desired number of input, output and hidden neurons to determine the dimension of the particles as Eq. (44). The population is initialized and after MLP training, the training error is calculated as an objective function. According to training error, every particle updates its velocity and position. The new positions are the new weights of MLP network which should minimize the objective function. The

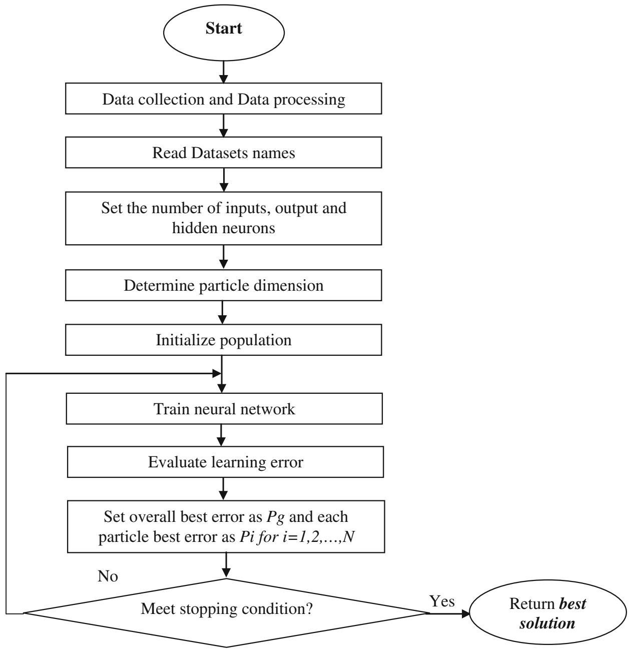
Start
Data collection and Data processing
Read Datasets names
Set the number of inputs, output and hidden neurons
Determine particle dimension
Initialize population
Train neural network
Evaluate learning error
Set overall best error as Pg and each particle best error as Pi for i=1,2,…,N
No
Meet stopping condition?
Yes
Return best solution

Fig. 8 Flowchart of hybrid learning of CAPSO and MLP network

Table 1 Description of datasets

Dataset	Attribute	Class	Sample	Missing data	Input	Hidden	Output
Hepatitis	19	2	155	Yes	19	39	1
Heart Disease	13	2	270	No	13	27	1
Pima Diabetes	8	2	768	Yes	8	17	1
Breast Cancer	32	2	198	Yes	32	65	1
Parkinson's Disease	22	2	197	No	22	45	1
Echocardiogram	12	2	132	Yes	12	25	1
Liver Disorders	6	2	345	No	6	13	1
Laryngeal 1	16	2	213	No	16	33	1
Acute Inflammations	6	2	120	No	6	13	1

Table 2 MSE of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP for all datasets

MSE	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Hepatitis								
Mean	0.101	0.197	0.120	0.242	0.149	0.238	0.130	0.210
Min	0.084	0.153	0.118	0.234	0.145	0.217	0.129	0.209
Max	0.122	0.252	0.121	0.246	0.160	0.254	0.131	0.214
Heart Disease								
Mean	0.134	0.141	0.167	0.186	0.244	0.244	0.191	0.204
Min	0.123	0.119	0.160	0.176	0.237	0.237	0.189	0.200
Max	0.153	0.168	0.175	0.196	0.252	0.251	0.193	0.205
Diabetes								
Mean	0.156	0.200	0.157	0.202	0.217	0.254	0.174	0.217
Min	0.154	0.193	0.156	0.200	0.207	0.240	0.172	0.211
Max	0.159	0.204	0.159	0.205	0.221	0.267	0.176	0.222
Breast Cancer								
Mean	0.145	0.170	0.148	0.173	0.179	0.167	0.154	0.177
Min	0.140	0.167	0.146	0.170	0.172	0.160	0.153	0.173
Max	0.148	0.175	0.149	0.179	0.187	0.190	0.155	0.180
Parkinson's Disease								
Mean	0.135	0.076	0.136	0.078	0.198	0.162	0.149	0.080
Min	0.123	0.058	0.134	0.076	0.188	0.134	0.148	0.079
Max	0.138	0.081	0.138	0.080	0.221	0.208	0.150	0.082
Echocardiogram								
Mean	0.066	0.089	0.076	0.089	0.178	0.192	0.110	0.111
Min	0.060	0.083	0.073	0.085	0.159	0.162	0.109	0.109
Max	0.069	0.108	0.078	0.093	0.194	0.217	0.111	0.112
Liver Disorders								
Mean	0.190	0.193	0.187	0.197	0.251	0.250	0.218	0.216
Min	0.179	0.175	0.177	0.187	0.242	0.240	0.216	0.213
Max	0.200	0.211	0.192	0.205	0.283	0.293	0.219	0.218
Laryngeal 1								
Mean	0.105	0.159	0.109	0.171	0.225	0.254	0.139	0.173
Min	0.099	0.148	0.107	0.163	0.208	0.222	0.135	0.168
Max	0.111	0.172	0.112	0.182	0.235	0.275	0.144	0.175

Table 2 continued

MSE	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Acute Inflammations								
Mean	0.029	0.017	0.027	0.017	0.253	0.259	0.190	0.177
Min	0.026	0.014	0.025	0.015	0.250	0.244	0.188	0.173
Max	0.034	0.020	0.030	0.019	0.258	0.272	0.192	0.181
Training rank	1		2		4		3	
Testing rank	1		2		4		3	

Table 3 Comparison of datasets in term of accuracy (%)

ACC (%)	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Hepatitis								
Mean	87.02	71.29	85.00	68.71	82.18	67.74	82.74	64.52
Min	83.06	67.74	83.87	67.74	81.45	67.74	82.26	64.52
Max	90.32	77.42	85.48	70.97	82.26	67.74	83.06	64.52
Heart Disease								
Mean	82.22	81.85	75.65	71.11	55.83	56.67	72.82	66.67
Min	78.24	72.22	74.07	66.67	48.61	51.85	71.30	62.96
Max	84.72	87.04	77.78	74.07	62.96	61.11	73.61	68.52
Diabetes								
Mean	78.06	72.99	78.09	72.40	67.36	56.43	73.57	64.61
Min	77.69	71.43	77.20	70.78	67.26	55.84	72.64	63.64
Max	78.83	74.68	78.83	74.03	68.08	56.49	74.76	66.23
Breast Cancer								
Mean	79.49	80.25	78.54	78.25	75.51	79.25	78.16	78.25
Min	78.48	77.50	77.85	77.50	74.05	72.50	77.22	77.50
Max	80.38	82.50	79.75	80.00	77.22	80.00	78.48	80.00
Parkinson’s Disease								
Mean	83.97	92.56	83.91	92.56	73.65	82.05	82.63	92.31
Min	82.69	89.74	83.33	92.31	73.08	82.05	81.41	92.31
Max	85.90	94.87	84.62	94.87	73.72	82.05	83.33	92.31
Echocardiogram								
Mean	92.45	88.46	89.62	84.62	71.32	65.38	89.06	84.62
Min	92.45	88.46	88.68	84.62	67.92	61.54	87.74	84.62
Max	92.45	88.46	90.57	84.62	86.79	84.62	89.62	84.62
Liver Disorders								
Mean	72.75	72.32	73.70	72.17	53.19	55.65	70.29	67.68
Min	70.29	69.57	71.38	68.12	42.75	39.13	68.48	65.22
Max	75.00	75.36	76.81	73.91	57.61	62.32	72.46	71.01
Laryngeal 1								
Mean	84.18	81.63	85.00	78.37	65.29	56.05	83.47	74.88
Min	83.53	76.74	82.94	76.74	64.12	53.49	80.59	74.42
Max	85.88	86.05	86.47	79.07	67.65	65.12	84.71	76.74
Acute Inflammations								
Mean	100.00	100.00	100.00	100.00	50.42	45.00	84.17	94.17
Min	100.00	100.00	100.00	100.00	48.96	41.67	82.29	87.50

Table 3 continued

ACC (%)	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Max	100.00	100.00	100.00	100.00	51.04	58.33	87.50	95.83
Training rank	1		2		4		3	
Testing rank	1		2		4		3	

Table 4 Comparison of datasets in term of sensitivity (%)

Sensitivity	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Hepatitis								
Mean	87.47	72.82	85.64	72.32	82.24	67.74	82.66	66.67
Min	84.62	71.43	84.75	72.00	82.11	67.74	82.26	66.67
Max	89.47	75.00	86.21	73.08	82.26	67.74	82.93	66.67
Heart Disease								
Mean	80.59	74.63	75.74	68.24	29.15	46.33	71.99	64.82
Min	75.79	65.52	73.26	63.64	0.00	0.00	70.24	60.00
Max	84.62	77.42	80.00	70.83	85.71	100.00	74.07	68.42
Diabetes								
Mean	71.14	85.14	71.62	83.25	22.18	4.00	76.92	89.41
Min	70.19	81.58	68.71	78.95	0.00	0.00	72.41	83.33
Max	72.90	91.18	73.83	86.49	64.71	40.00	80.26	94.12
Breast Cancer								
Mean	78.06	52.33	69.45	43.00	34.38	7.86	70.00	43.00
Min	72.73	40.00	66.67	40.00	0.00	0.00	63.64	40.00
Max	81.82	66.67	75.00	50.00	100.00	50.00	72.73	50.00
Parkinson’s Disease								
Mean	83.34	93.96	82.69	93.96	73.76	82.05	82.29	93.94
Min	82.84	93.75	82.01	93.94	73.55	82.05	81.62	93.94
Max	85.50	94.12	83.46	94.12	74.34	82.05	82.48	93.94
Echocardiogram								
Mean	100.00	88.89	88.88	80.00	16.94	26.00	82.79	80.00
Min	100.00	88.89	86.67	80.00	0.00	0.00	80.00	80.00
Max	100.00	88.89	90.00	80.00	94.44	100.00	84.85	80.00
Liver Disorders								
Mean	72.31	76.57	73.27	76.27	55.66	49.17	69.52	71.75
Min	70.43	74.42	71.58	73.81	0.00	0.00	67.84	70.45
Max	74.59	77.78	76.40	78.05	100.00	67.50	71.13	75.00
Laryngeal 1								
Mean	86.38	82.30	86.41	78.22	65.29	55.02	83.64	73.28
Min	85.22	76.00	85.47	76.00	64.12	53.49	80.65	73.08
Max	88.07	86.96	87.61	79.17	69.57	61.11	84.75	74.07
Acute Inflammations								
Mean	100.00	100.00	100.00	100.00	40.73	33.26	98.63	99.00
Min	100.00	100.00	100.00	100.00	0.00	0.00	92.11	90.00
Max	100.00	100.00	100.00	100.00	51.04	41.67	100.00	100.00
Training rank	1		2		4		3	
Testing rank	1		2		4		3	

Table 5 Comparison of datasets in term of specificity (%)

Specificity	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Hepatitis								
Mean	80.68	65.33	74.40	53.00	0.00	0.00	60.00	0.00
Min	57.14	50.00	66.67	50.00	0.00	0.00	0.00	0.00
Max	100.00	100.00	83.33	60.00	0.00	0.00	100.00	0.00
Heart Disease								
Mean	83.55	90.21	75.61	73.34	57.29	57.12	73.40	67.78
Min	80.17	80.00	74.62	68.75	52.60	54.17	71.97	64.71
Max	85.00	100.00	77.10	77.78	73.81	63.64	75.20	69.70
Diabetes								
Mean	80.44	69.27	80.28	69.00	67.39	56.48	73.14	61.77
Min	80.13	68.07	79.57	68.07	67.26	56.38	72.31	61.15
Max	80.84	70.00	80.70	70.09	68.17	56.49	73.98	62.77
Breast Cancer								
Mean	79.62	83.38	79.37	83.00	75.77	80.34	78.80	83.00
Min	78.91	82.86	78.77	82.86	75.32	80.00	78.23	82.86
Max	80.56	83.78	80.42	83.33	77.46	81.82	78.91	83.33
Parkinson’s Disease								
Mean	88.28	85.48	93.32	85.00	5.00	0.00	84.95	83.33
Min	81.82	71.43	82.61	83.33	0.00	0.00	80.00	83.33
Max	94.74	100.00	100.00	100.00	50.00	0.00	89.47	83.33
Echocardiogram								
Mean	90.00	88.24	89.93	87.50	71.80	65.37	92.07	87.50
Min	90.00	88.24	89.47	87.50	67.92	61.54	90.54	87.50
Max	90.00	88.24	90.79	87.50	93.94	87.50	92.96	87.50
Liver Disorders								
Mean	73.75	65.49	74.64	65.33	34.79	18.65	72.17	59.96
Min	69.57	60.00	70.59	59.26	0.00	0.00	70.13	56.00
Max	77.78	70.83	78.82	69.57	52.94	60.00	75.61	64.00
Laryngeal 1								
Mean	79.97	80.97	82.20	78.60	48.10	58.57	83.09	77.43
Min	77.05	77.78	76.67	77.78	0.00	0.00	80.43	76.47
Max	86.27	85.00	86.54	78.95	100.00	100.00	85.71	81.25
Acute Inflammations								
Mean	100.00	100.00	100.00	100.00	12.29	16.67	76.02	91.60
Min	100.00	100.00	100.00	100.00	0.00	0.00	73.44	82.35
Max	100.00	100.00	100.00	100.00	48.96	58.33	79.66	93.33
Training rank	1		2		4		3	
Testing rank	1		2		4		3	

function is computed based on mean square error (MSE) as Eq. (45). These steps will go on until meeting stop conditions.

$$MSE = \frac{1}{P}\sum_{j=1}^{P}(t_j - O_j)^2, \tag{45}$$

where t_j and O_j are the desired output and the network output respectively, and P is the number of patterns.

4 Experimental studies

To evaluate the performance of CAPSO-MLP, nine medical datasets (Kuncheva 2005; Bache and Lichman 2013) are selected as detailed in Table 1. As shown in this table, the selected datasets of Hepatitis, Heart Disease, Pima Indian Diabetes, Wisconsin Prognostic Breast Cancer, Parkinson’s disease and Echocardiogram (Heart attack), Liver Disorders,

Table 6 Comparison of datasets in term of AUC

AUC	CAPSO-MLP		PSO-MLP		GSA-MLP		ICA-MLP	
Dataset	Train	Test	Train	Test	Train	Test	Train	Test
Hepatitis	0.834	0.837	0.753	0.821	0.496	0.441	0.748	0.792
Heart Disease	0.897	0.876	0.841	0.791	0.624	0.621	0.785	0.738
Diabetes	0.837	0.808	0.834	0.806	0.642	0.628	0.812	0.787
Breast Cancer	0.810	0.630	0.794	0.605	0.667	0.551	0.769	0.567
Parkinson's Disease	0.804	0.912	0.775	0.898	0.562	0.596	0.720	0.865
Echocardiogram	0.960	0.902	0.913	0.857	0.464	0.477	0.894	0.858
Liver Disorders	0.733	0.710	0.740	0.708	0.452	0.386	0.709	0.659
Laryngeal 1	0.931	0.870	0.925	0.860	0.607	0.606	0.904	0.847
Acute Inflammations	1.00	1.00	1.00	1.00	0.325	0.310	0.873	0.953

Laryngeal 1 and Acute Inflammations are in the class of binary classification problems. Each dataset will be represented as input patterns and outputs in hybrid learning of MLP network. Therefore, the effectiveness and performance of MLP network learning have been verified using different datasets. CAPSO, PSO, GSA and ICA algorithms are used for MLP learning on these datasets to assess the ability of proposed approach. In this context, the hybrids of the algorithms and MLP network are called PSO-MLP, GSA-MLP and ICA-MLP respectively.

4.1 Datasets design

The Hepatitis dataset problem is a complex and noisy data and has a large number of missing data (167 missing values in total dataset). The classification should predict whether a patient with hepatitis will live or die. There are 19 attributes/inputs and the output classes' distribution is live: 123 and die: 32.

The absence or presence of heart disease is predicted by Statlog (Heart Disease) dataset. This dataset contains 13 attributes/inputs and the output classes' distribution is absence: 150 and presence: 120.

The Pima Indian Diabetes dataset diagnoses a person based on personal data and medical examination has Diabetes or not. There are 8 attributes/inputs and the output class distribution is Diabetes positive: 268 and Diabetes negative: 500.

The aim of the Wisconsin Prognostic Breast Cancer (WPBC) dataset is to predict whether a tumor is recurrence or non-recurrence. Each record represents a follow-up data for one breast cancer case. There are 33 attributes/inputs and the output class distribution is recurrence: 47 and non-recurrence: 151.

The objective of Parkinson's disease dataset is to discriminate healthy people from those with Parkinson's disease. This dataset is composed of a range of biomedical voice measurements with 22 attributes/inputs and the output class distribution is health status: 147 and Parkinson's disease: 48.

In the dataset of Echocardiogram, all patients have suffered from heart attacks at some points in the past. Some of them are still alive and some are not. The dataset has 132 samples and 12 attributes/inputs. The number of missing values in total dataset is 132 and the distribution of output attribute is alive: 43, dead: 88 and missing value: 1.

The main aim of the Liver Disorders dataset is to diagnose a patient as healthy (normal) or abnormal. It has six attributes/inputs and two output classes (abnormal or normal). The first five variables are all blood tests which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. This dataset contains 345 examples in, where 200 are negative examples (normal) and 145 are positive examples (abnormal). Each line in the Liver Disorders dataset constitutes the record of a single male individual.

The Laryngeal 1 dataset is applied to diagnose the laryngeal pathology as normal and pathological voices. The number of attributes/inputs is 16 (all continuous-valued). The total instances is 213 and the distribution of output attribute is 81 Normal and 132 Pathological voices.

The idea of Acute Inflammations dataset is to diagnose two diseases of urinary system: Inflammation of urinary bladder and Nephritis of renal pelvis origin. The basis for rules detection is Rough Sets Theory. The dataset has 120 samples and 6 attributes/inputs. In this paper, the algorithms are evaluated to classify the Inflammation of urinary bladder output. The distribution of output attribute is Positive: 59 and Negative: 61.

The datasets are processed for better generalization of error and accuracy. Normally, real world data are noisy because they are incomplete and inconsistent. The datasets are normalized and missing data values are separately replaced by the mean of attributes values.

4.2 Experimental setup

The mentioned datasets are partitioned into two as training and testing sets. The training set is used to train the net-

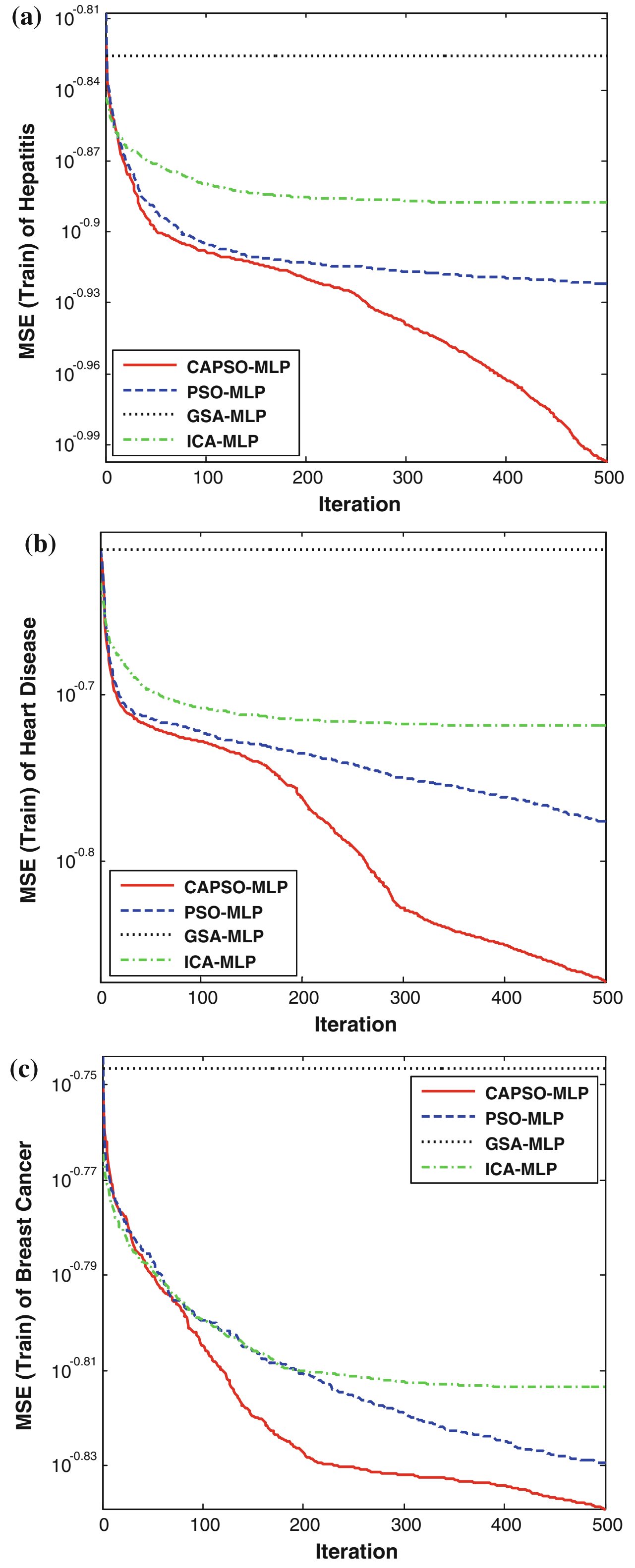

Fig. 9 Training errors (MSE) of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP for **a** Hepatitis, **b** Heart Disease and **c** Breast Cancer

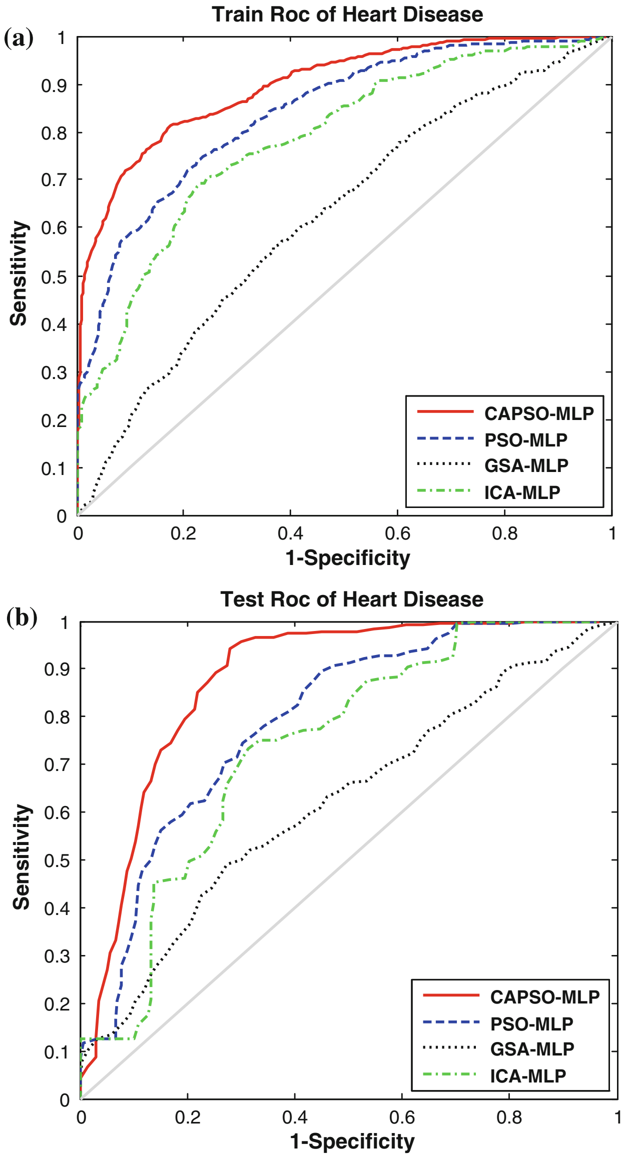

Fig. 10 ROC curve of **a** training data for Heart Disease and **b** testing data for Heart Disease

work in order to achieve the optimal weights. The testing set is applied on unseen data to test the generalization performance of meta-heuristic algorithms on MLP network. For each dataset, 80 % of data are employed for the training set and the rest 20 % for the testing set.

The number of input and output neurons in MLP network is problem-dependent and the number of hidden nodes is computed as Kolmogorov theorem (Hecht-Nielsen 1987) based on input neurons:

$$Hidden = 2 \times Input + 1. \tag{46}$$

The number of iterations is set to 500 for all datasets. The population size is considered 40 in CAPSO, PSO and GSA; also, the number of countries and imperialist are 160 and 8 in ICA respectively. In PSO, w is set to 0.7 and linearly decreased to 0.4 and V_{max} is set to 0.5. The acceleration coefficients C_1 and C_2 are equal to 2. In ICA, revolution rate = 0.5, $\beta = 2$, $\gamma = 0.5$ and $\xi = 0.02$ (Tayefeh-Mahmoudi et al. 2013) are considered. Besides, G in GSA is computed as Eq. (47):

$$G(t) = G_0 e^{\frac{-\alpha t}{T}} \tag{47}$$

where G_0 and α are set to 100 and 20 respectively; also, T is set to the maximum iteration. At the beginning, *Kbest* in Eq. (14) is initialized with 40 (population size) and then linearly decreased to 1.

4.3 Results and discussion

This section presents the results of CAPSO, PSO, GSA and ICA adapted on MLP network. The experiments are conducted using the nine medical datasets. In this study, all datasets have been partitioned into two sets: training set and testing set. The training set is used to train the network in order to achieve the optimal weights. The testing set is applied on unseen data to test the generalization performance of meta-heuristic algorithms on MLP network.

The algorithms are independently run 10 times and the results are shown in Tables 2, 3, 4, 5 and 6. In these tables, the Mean, Min and Max indicate the average, minimum and maximum value respectively. Also, the algorithms are ranked based on the mean solution in each row of tables and the rank of algorithms is calculated in the each column of table.

Table 2 demonstrates the training and testing error (MSE) of MLP network based on the mentioned algorithms on Hepatitis, Heart Disease, Diabetes, Breast Cancer, Parkinson's disease, Echocardiogram, Liver Disorders, Laryngeal 1 and Acute Inflammations datasets. In the table, the training and testing ranks signify that CAPSO-MLP has better convergence rate in comparison with other algorithms. In contrast, GSA-MLP shows the worst convergence rate among the algorithms. Also, Fig. 9 illustrates the training error (MSE) of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP on the Hepatitis, Heart Disease and Breast Cancer datasets. As observed, CAPSO-MLP provides the lower error training for all datasets in contrast; GSA-MLP shows the worst MSE among the algorithms.

Table 3 presents the classification accuracy of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP over ten independent runs on the datasets. The accuracy refers to the percentage of correct classification on training and testing data respectively. As depicted in this table, CAPSO-MLP provides the best performance on unseen data (testing data): Hepatitis with 71.29 %, Heart Disease with 81.85 %, Diabetes with 72.99 %, Wisconsin Prognostic Breast Cancer with 80.25 %, Parkinson's disease with 92.56 %, Echocardiogram (Heart attack) with 88.46 %, Liver Disorders with 72.32 %, Laryngeal 1 with 81.63 % and Acute Inflammations with 100 %. Also, it is noticeable that the proposed method has the highest classification accuracy for Hepatitis and Echocardiogram datasets, containing missing data, compared to the other algorithms. Also, GSA-MLP shows a lower accuracy among them.

In addition, to evaluate the classification abilities of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP, a comparison of the algorithms in the terms of Sensitivity, Specificity and AUC has been performed as shown in Tables 4, 5 and 6.

In terms of Sensitivity and Specificity, the proposed method has offered better results than the others algorithms. In other words, CAPSO-MLP presents better Sensitivity and Specificity on testing data (unseen data).

As shown in Table 6, the proposed method provides the best AUC both in training and testing sets, except in Liver Disorders dataset. Among the algorithms, GSA-MLP has the lowest AUC, both in training and testing data. The better results of CAPSO-MLP are also observable in Fig. 10. As the ROC of training and testing Heart Disease illustrate, CAPSO-MLP has yielded the best AUC on the dataset.

Further, in order to determine whether the results achieved by the algorithms are statistically different from each other, t-test (Box et al. 2005) and Wilcoxon's signed ranks test (Wilcoxon 1945) are conducted between the results obtained by the proposed algorithm, CAPSO-MLP, with PSO-MLP, GSA-MLP and ICA-MLP. The tests are performed on datasets for training and testing accuracy. In these tests, a *p-value* is computed and as shown in Table 7, if *p-value* is greater than the significance level of $\alpha = 0.05$, there are no significant differences between algorithms; otherwise, there are significant differences. In the table, the symbols '>' and '=' mean the algorithm on the left side is significantly and no significantly better than the algorithm on the right side respectively. As illustrated, there are significant differences between CAPSO-MLP and the other algorithms.

5 Conclusions

In this paper, an improved scheme of particle swarm algorithm (PSO) and Newtonian's motion laws, called centripetal accelerated particle swarm optimization (CAPSO) has been proposed to accelerate the learning and convergence procedure of classifiers. CAPSO has been adapted on MLP network to optimize the accuracy and the weights of the network. The obtained results of nine medical disease diagnosis benchmarks indicate that the proposed method provides bet-

Table 7 Results of *t*-test and Wilcoxon's signed-ranks test taking into account accuracy rate

CAPSO-MLP VS.	PSO-MLP			GSA-MLP			ICA-MLP		
Dataset	p-value *t*-test	p-value signed rank	Significance	p-value *t*-test	p-value signed rank	Significance	p-value *t*-test	p-value signed rank	Significance
Train									
Hepatitis	0.0406	0.0430	>	0.0002	0.0020	>	0.0003	0.0039	>
Heart Disease	2.03e−05	0.0020	>	1.47e−08	0.0020	>	5.12e−07	0.0020	>
Diabetes	0.8363	1.00	=	7.20e−14	0.0020	>	2.05e−09	0.0020	>
Breast Cancer	0.0091	0.0234	>	6.81e−07	0.0020	>	9.08e−05	0.0020	>
Parkinson's	0.8114	0.7617	=	6.06e−11	0.0020	>	0.0032	0.0078	>
Echocardiogram	8.60e−09	0.0020	>	6.75e−06	0.0020	>	5.52e−08	0.0020	>
Liver Disorders	0.3205	0.4453	=	4.89e−06	0.0020	>	0.0036	0.0078	>
Laryngeal 1	0.0445	0.1094	>	5.30e−11	0.0020	>	0.1405	0.2500	=
Acute Inflammations	1.00	1.00	=	9.40e−17	0.0020	>	5.49e−10	0.0020	>
Test									
Hepatitis	0.0368	0.0781	>	0.0068	0.0156	>	9.08e−05	0.0020	>
Heart Disease	8.93e−05	0.0039	>	7.97e−09	0.0020	>	4.40e−06	0.0020	>
Diabetes	0.1934	0.2813	=	5.35e−13	0.0020	>	4.38e−10	0.0020	>
Breast Cancer	0.0031	0.0156	>	0.2695	0.2500	=	0.0002	0.0078	>
Parkinson's	1.00	1.00	=	2.81e−09	0.0020	>	0.5911	0.5000	=
Echocardiogram	5.24e−143	0.0020	>	5.54e−06	0.0020	>	5.24e−143	0.0020	>
Liver Disorders	0.7976	0.8828	=	0.0003	0.0020	>	1.76e−05	0.0020	>
Laryngeal 1	0.0128	0.0156	>	1.81e−07	0.0020	>	0.0006	0.0078	>
Acute Inflammations	1.00	1.00	=	1.38e−09	0.0020	>	0.0001	0.0020	>

ter results than PSO-MLP, GSA-MLP and ICA-MLP in terms of good convergence rate and classification accuracy. The superiority of CAPSO-MLP against other methods is considerable especially in unseen data (testing data) and datasets with high missing data values.

The main advantages of CAPSO for neural network learning are the simple concept, the ease of implementation and the needless to tune to any algorithm-specific parameters. The algorithm requires only common controlling parameters such as the number of generation and population size. Therefore, it can be concluded that the proposed method is a viable tool to evolve the neural network learning and accuracy.

Acknowledgments This research is supported by Research University Grant (03H72), Universiti Teknologi Malaysia. The authors would like to thank *Soft Computing Research Group (SCRG)*, Universiti Teknologi Malaysia (UTM), Johor Bahru Malaysia, for supporting this study. Also, they give kind respect and special thanks to Dr. Zhaleh Beheshti, for proof reading the manuscript and providing valuable comments.

References

Abbod MF, Cheng KY, Cui XR, Huang SJ, Han YY, Shieh JS (2011) Ensembled neural networks for brain death prediction for patients with severe head injury. Biomed Signal Process Control 6:414–421

Ahmadi MA, Ahmadi MR, Shadizadeh SR (2013) Evolving artificial neural network and imperialist competitive algorithm for prediction permeability of the reservoir. Neural Comput Appl 23:563

Anderson JA (2006) An introduction to neural networks. Prentice-Hall of India Private Limited

Ari S, Saha G (2009) In search of an optimization technique for artificial neural network to classify abnormal heart sounds. Appl Soft Comput 9:330–340

Ashizawa K, Ishida T, MacMahon H, Vyborny CJ, Katsuragawa S, Doi K (1999) Artificial neural networks in chest radiography: application to the differential diagnosis of interstitial lung disease. Acad Radiol 6:2–9

Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: Proceedings of IEEE congress on evolutionary computation, pp 4661–4667

Awad M, Motai Y (2008) Dynamic classification for video stream using support vector machine. Appl Soft Comput 8:1314–1325

Bache K, Lichman M (2013) UCI machine learning repository. School of Information and Computer Science, University of California, Irvine. http://archive.ics.uci.edu/ml

Beheshti Z, Shamsuddin SM (2013a) CAPSO: centripetal accelerated particle swarm optimization. Inf Sci. doi:10.1016/j.ins.2013.08.015

Beheshti Z, Shamsuddin SM (2013b) A review of population-based meta-heuristic algorithm. Int J Adv Soft Comput Appl 5(1):1–35

Bennett KP, Mangasarian OL (1992) Robust linear programming discrimination of two linearly inseparable sets. Optim Method Softw 1:23–34

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Breiman L, Friedman JH, Olshen RA, Stone C (1984) Classification and regression trees. Wadsworth

Brent RP (1991) Fast training algorithms for multi-layer neural nets. IEEE Trans Neural Netw 2:346–354

Box GE-P, Hunter JS, Hunter WG (2005) Statistics for experiments: design, innovation, and discovery, 2nd edn. Wiley, New York

Chau KW (2007) Application of a PSO-based neural network in analysis of outcomes of construction claims. Autom Constr 16:642–646

Chen MY, Chen KK, Chiang HK, Huang HS, Huang MJ (2007) Comparing extended classifier system and genetic programming for financial forecasting: an empirical study. Soft Comput. 11:1173–1183

Coppini G, Miniati M, Paterni M, Monti S, Ferdeghini EM (2007) Computer-aided diagnosis of emphysema in COPD patients: neural-network-based analysis of lung shape in digital chest radiographs. Med Eng Phys 29:76–86

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Delen D, Walker G, Kadam A (2005) Predicting breast cancer survivability: a comparison of three data mining methods. Artif Intell Med 34:113–127

Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part C Appl Rev 26:29–41

Ekici S (2012) Support vector machines for classification and locating faults on transmission lines. Appl Soft Comput 12:1650–1658

Elveren E, Yumuşak N (2011) Tuberculosis disease diagnosis using artificial neural network trained with genetic algorithm. J Med Syst 35:329–332

Er O, Sertkaya C, Temurtas F, Tanrikulu AC (2009) A comparative study on chronic obstructive pulmonary and pneumonia diseases diagnosis using neural networks and artificial immune system. J Med Syst 33:485–492

Fana CY, Changb PC, Linb JJ, Hsiehb JC (2011) A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification. Appl Soft Comput 11:632–644

Farmer JD, Packard NH, Perelson AS (1986) The immune system, adaptation and machine learning. Phys D 2:187–204

Fausett LV (1994) Fundamentals of neural networks: architectures, algorithms, and applications. Prentice-Hall, New Jersey

Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27:861–874

Folland R, Hines E, Dutta R, Boilot P, Morgan D (2004) Comparison of neural network predictors in the classification of tracheal-bronchial breath sounds by respiratory auscultation. Artif Intell Med 31:211–220

Gao WF, Liu SY, Huang LL (2012) Particle swarm optimization with chaotic opposition based population initialization and stochastic search technique. Commun Nonlinear Sci Numer Simul 17:4316–4327

García S, Fernández A, Herrera F (2009) Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. Appl Soft Comput 9:1304–1314

Gori M, Tesi A (1992) On the problem of local minima in backpropagation. IEEE Trans Pattern Anal Mach Intell 14:76–85

Gurney KN (1997) An introduction to neural networks. Routledge

Handels H, Roß TH, Kreusch J, Wolff HH, Poppl SJ (1999) Feature selection for optimized skin tumor recognition using genetic algorithms. Artif Intell Med 16:283–297

Hecht-Nielsen R (1987) Kolmogorov's mapping neural network existence theorem. In: Proceedings of IEEE international conference on neural networks, pp 11–14

Heckerling PS, Gerber BS, Tape TG, Wigton RS (2004) Use of genetic algorithms for neural networks to predict community-acquired pneumonia. Artif Intell Med 30:71–84

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2:359–366

Hrstka O, Kučerová A (2004) Improvements of real coded genetic algorithms based on differential operators preventing premature convergence. Adv Eng Softw 35:237–246

Huang ML, Hung YH, Chen WY (2010) Neural network classifier with entropy based feature selection on breast cancer diagnosis. J Med Syst 34:865–873

Jerez JM, Molina I, García-Laencina PJ, Alba E, Ribelles N, Martín M, Franco L (2010) Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artif Intell Med 50:105–115

Karabatak M, Ince MC (2009) An expert system for detection of breast cancer based on association rules and neural network. Expert Syst Appl 36:3465–3469

Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report, TR06, Erciyes University

Kayaer K, Yildirim T (2003) Medical diagnosis on Pima Indian diabetes using general regression neural networks. In: Proceedings of international conference on artificial neural networks and neural information, pp 181–184

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948

Kisi O, Ozturk O (2007) Adaptive neuro-fuzzy computing technique for evapotranspiration estimation. J Irrig Drainage Eng ASCE 133:368–379

Kuncheva LI (2005) Real medical data sets. Tech. Rep. School of Informatics: University of Wales, Bangor. http://www.bangor.ac.uk/~mas00a/activities/real_data.htm

Leung Y, Gao Y, Xu ZB (1997) Degree of population diversity—a perspective on premature convergence in genetic algorithms and its Markov chain analysis. IEEE Trans Neural Netw 8:1165–1176

Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10:281–295

Mitra V, Wang CJ (2008) Content based audio classification: a neural network approach. Soft Comput 12:639–646

Moslemipour G, Lee TS, Rilling D (2012) A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems. Int J Adv Manuf Technol 60:11–27

Oreski S, Oreski D, Oreski G (2012) Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. Expert Syst Appl 39:12605–12617

Ozkan C, Kisi O, Akay B (2011) Neural networks with artificial bee colony algorithm for modeling daily reference evapotranspiration. Irrig Sci 29:431–441

Park J, Sandberg I (1991) Universal approximation using radial-basis function networks. Neural Comput 3:246–257

Prechelt L (1995) Some notes on neural learning algorithm benchmarking. Neurocomputing 9:343–347

Qasem SN, Shamsuddin SM (2011) Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis. Appl Soft Comput 11:1427–1438

Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248

Reynolds AP, Iglesia B (2009) A multi-objective GRASP for partial classification. Soft Comput 13:227–243

Rumelhart DE, McClelland JL (1986) Parallel distributed processing: explorations in the microstructure of cognition. MIT, Cambridge

Samanta B, Al-Balushi KR, Al-Araimi SA (2003) Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. Eng Appl Artif Intell 16:657–665

Saxena A, Saad A (2007) Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems. Appl Soft Comput 7:441–454

Schutz B (2003) Gravity from the ground up. Cambridge University Press, Cambridge

Setiono R, Hui LCK (1995) Use of a quasinewton method in a feedforward neural network construction algorithm. IEEE Trans Neural Netw 6:740–747

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings of IEEE international conference on evolutionary computation, pp 69–73

Sivagaminathan RK, Ramakrishnan S (2007) A hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert Syst Appl 33:49–60

Socha K, Blum C (2007) An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Comput Appl 16:235–247

Sulaiman SI, Abdul-Rahman TK, Musirin I, Shaari S (2012) An artificial immune-based hybrid multi-layer feedforward neural network for predicting grid-connected photovoltaic system output. Energy Proc 14:260–264

Tang KS, Man KF, Kwong S, He Q (1996) Genetic algorithms and their applications. IEEE Signal Process Mag 13:22–37

Tayefeh-Mahmoudi M, Taghiyareh F, Forouzideh N, Lucas C (2013) Evolving artificial neural network structure using grammar encoding and colonial competitive algorithm. Neural Comput Appl 22:1–16

Temurtas F (2007) A comparative study on thyroid disease diagnosis using neural networks. Expert Syst Appl 36:944–949

Tsoulos I, Gavrilis D, Glavas E (2008) Neural network construction and training using grammatical evolution. Neurocomputing 72:269–277

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bull 1:80–83

Yao X (1999) Evolving artificial neural networks. Proc IEEE 87:1423–1447

Zhang C, Shao H, Li Y (2000) Particle swarm optimization for evolving artificial neural network. In: Proceedings of IEEE international conference on systems, man, and sybernetics, pp 2487–2490

Ziver AK, Pain CC, Carter JN, de Oliveira CRE, Goddard AJ, Overton RS (2004) Genetic algorithms and artificial neural networks for loading pattern optimization of advanced gas-cooled reactors. Ann Nucl Energy 13:431–457