



How to cite this article:

Kumar, N., & Kumar, D. (2021). An improved grey wolf optimization-based learning of artificial neural network for medical data classification. *Journal of Information and Communication Technology*, 20(2), 213-248. <https://doi.org/10.32890/jict2021.20.2.4>

An Improved Grey Wolf Optimization-based Learning of Artificial Neural Network for Medical Data Classification

Narender Kumar & Dharmender Kumar
Department of Computer Science and Engineering
Guru Jambheshwar University of Science and
Technology, India

narenderster@gmail.com
dharmindia24@gmail.com

Received: 21/7/2020 Revised: 8/9/2020 Accepted: 1/10/2020 Published: 23/2/2021

ABSTRACT

Grey wolf optimization (GWO) is a recent and popular swarm-based metaheuristic approach. It has been used in numerous fields such as numerical optimization, engineering problems, and machine learning. The different variants of GWO have been developed in the last five years for solving optimization problems in diverse fields. Like other metaheuristic algorithms, GWO also suffers from local optima and slow convergence problems, which result in degraded performance. An adequate equilibrium among exploration and exploitation is a key factor to the success of metaheuristic algorithms, especially for optimization tasks. In this paper, a new variant of GWO, called inertia

motivated GWO (IMGWO), was proposed. The aim of IMGWO was to establish better balance between exploration and exploitation. Traditionally, artificial neural network (ANN) with backpropagation (BP) depends on initial values and in turn, attains poor convergence. The metaheuristic approaches are better alternatives instead of BP. The proposed IMGWO was used to train ANN to prove its competency in terms of prediction. The proposed IMGWO-ANN was applied for medical diagnosis tasks. Several benchmark medical datasets including heart disease, breast cancer, hepatitis, and Parkinson's diseases were used for assessing the performance of IMGWO-ANN. The performance measures were described in terms of mean squared errors, classification accuracies, sensitivities, specificities, the area under the curve, and receiver operating characteristic curve. It was found that IMGWO outperformed three popular metaheuristic approaches including GWO, genetic algorithm, and particle swarm optimization. Results confirmed the potency of IMGWO as a viable learning technique for an ANN.

Keywords: Metaheuristic, medical diagnosis, grey wolf optimization, artificial neural network, multilayer perceptron, particle swarm optimization.

INTRODUCTION

The process of medical diagnosis becomes easier and faster if a decision support system assists the doctors because machines do not suffer from fatigue or boredom. Numerous tests are involved in a disease diagnosis that can make the process complicated. High performance is desirable in the medical diagnosis process as a little difference in accuracy may lead to a substantial change in prediction (Li et al., 2017). Continuous efforts are put in this direction to improve the performance of the diagnosis process through machine learning methods like support vector machine (SVM) by Akay (2009) and Maglogiannis et al. (2009). Das et al. (2009), Lin and Chuang (2010), and Yan et al. (2006) developed artificial neural network (ANN)-based models for medical diagnosis. Such other methods are also reported in the literature of medical data mining. Among all of these methods, ANNs are assumed as universal methods for approximation and generic classifiers. In 1943, McCulloch-Pitts

developed a model called ANN and gained much popularity in the field of artificial intelligence and machine learning. ANN effectively modeled the different problems related to computational intelligence, optimization, function approximation, and complex predictive system. Three fundamental architectures for ANN are reported including single-layer feedforward network, multilayered feedforward network, and recurrent networks. Furthermore, neural network identifies six learning tasks such as pattern association, pattern recognition, function approximation, control, filtering, and beam forcing (Haykin, 1994). ANN is further described by patterns of connections between the neurons, methods for determining the weights of communications, and its activation functions (Amirsadri et al., 2018).

Apart from the abovementioned merits, the training process of ANN through backpropagation has been criticized in many studies, due to its slow convergence and poor performance (Brent, 1991; Gori & Tesi, 1992). The poor performance is due to dependence on initial values, whereas trapped in local minima is responsible for causing slow convergence. Metaheuristic algorithm-based training of ANN is a possible substitution of backpropagation learning and existing studies support the aforementioned claim (Ojha et al., 2017). There are various ways to integrate the metaheuristic techniques with ANN, such as: (i) setting the architecture of the network: a network structure is defined by the number of layers and number of neurons per layer; (ii) managing weight and bias by a metaheuristic technique: any proposed metaheuristic technique is used to adjust the weight and bias of the connections established between neurons of different layers, either called the learning of weight and bias or the training of an ANN; and (iii) tuning different learning parameters: it is kind of a hybridization in which a metaheuristic approach is used to adjust some important parameters such as learning rate and momentum in a gradient descent-based learning method (Mirjalili et al., 2012).

The performance of multilayered perceptron (MLP) was deeply investigated in Seiffert (2001). A study adopted artificial bee colony (ABC) algorithm to train ANN and it performed better than genetic algorithm (GA) as well as backpropagation (BP) (Karaboga et al., 2007). Gudise and Venayagamoorthy (2003) found that particle swarm optimization (PSO) is a fast-learning algorithm with non-linear function for feedforward neural network and supersede

backpropagation in terms of speed. Blum and Socha (2005) applied ant colony optimization (ACO) algorithm to train a feedforward neural network, and improved classification results were obtained using benchmark medical datasets including breast cancer, diabetes, and heart disease.

Grey wolf optimization (GWO) is a recent swarm-based metaheuristic technique developed by Mirjalili et al. (2014), inspired through social hierarchy and the hunting process of grey wolves. GWO proved its competence in terms of fast convergence and better global search capabilities and provides competitive results for a wide variety of optimization problems (Long et al., 2017; Qais et al., 2018; Saremi et al., 2015). Zhang et al. (2019) found that GWO is comparatively a new and more capable technique among other swarm-based optimization algorithms for solving numerical optimization. A binary version of GWO has also been used for feature selection in various medical datasets downloaded from University of California Irvine (UCI) machine learning repository (Emary et al., 2016). GWO is also used for prediction of medical data by Sahoo and Chandra (2017). Furthermore, the classification of cervical cancer is predicted through GWO and a significant discrimination is produced between benign and malignant lesions. Khairuzzaman and Chaudhury (2017) used GWO for multilevel thresholding in image segmentation. Some recent studies also applied GWO with ANN for prediction tasks (Djema et al., 2019; Mirjalili, 2015; Nosratabadi et al., 2020; Turabieh, 2016). Due to the diverse applicability of GWO, several drawbacks are also associated with the GWO algorithm like other population-based metaheuristic algorithms. These drawbacks include local optima entrapment at the end of the optimization process and slower convergence during the later part of the evolution process. Moreover, it is easily entrapped in local optima with complex multimodal tasks (Long et al., 2018; Zhang et al., 2019). Long et al. (2018) stated that a proper balance between exploration and exploitation is needed to achieve global optima solution in case of population-based stochastic methods. They proposed a new non-linear control variable \vec{a} and modification in the position updating equation inspired by PSO.

In this paper, a new variant of GWO called inertia motivated GWO (IMGWO) is proposed to act as a training algorithm for multilayered perceptron (MLP), a kind of ANN. The new training technique

IMGWO is a noticeable and significant extension of the study carried out by Long et al. (2018). The current study extends the work of Long et al. (2018) by introducing a new equation for non-linear adjustment of the control parameter \vec{a} . However, the position updating equation remained the same as suggested by Long et al. (2018). Significant improvements are obtained in simulation results by the proposed study while the training of MLP is considerably improved and found to be better than GA, PSO, and GWO. The four well-known real-world medical datasets are taken from the UCI machine learning repository. The evaluation measures like MSE, convergence curve, accuracies, sensitivities, specificities, and receiver operating characteristic (ROC) are considered for evaluating the results of IMGWO-ANN and compared with three well-known metaheuristic algorithms, i.e., GWO, GA, and PSO.

The structure of the paper is described as follows: related works including metaheuristic algorithms and diagnosis systems are reviewed in the second section. The third section discusses the proposed methodology and also includes basic concepts of MLP, GWO, PSO, exploration and exploitation, and hybridization of metaheuristic with ANN. The fourth section explains the proposed study in depth. The experimental results are presented in the fifth section. Section six presents the conclusion and future scope of the work.

RELATED WORKS

The literature on metaheuristic algorithm-based training and optimizing of ANN is very rich and profound base. The metaheuristic algorithms can be categorized into: (i) single-solution-based, (ii) multiple-solutions-based, (iii) evolutionary algorithms (EA), (iv) nature-inspired algorithms (NIA), and (v) swarm intelligence (SI)-based optimization. Many of the EA and NIA algorithms are considered as swarm-based optimization like GA and GWO. Metaheuristic algorithms such as EA, NIA, and SI are multiple-solutions-based methods. Simulated annealing, a single-solution-based metaheuristic was used to optimize ANN in a work by Sarkar and Modak (2003) and it performed better than traditional methods. The metaheuristic methods that are based on multiple solutions can be found to be more capable of avoiding local optima problems (Mirjalili, 2015). Recent

literature also confirmed that SI, NIA, and EA have better exploration capabilities than single-solution-based metaheuristic algorithms to train ANN (Ojha et al., 2017). Nevertheless, according to the No Free Lunch theorem, there is no universal metaheuristic-based training algorithm for ANN. Therefore, various metaheuristic-based algorithms have been applied to train ANN (Amirsadri et al., 2018).

Recent studies showed that GA outperforms conventional backpropagation to train ANN for some real-world applications (Ding et al., 2011; Tong & Mintram, 2010). Slowik (2011) incorporated an advanced differential evolution technique to train neural networks, and claimed that simulation results proposed that the technique was better than EA and traditional backpropagation methods. Several studies explored the capabilities of PSO for weight optimization of ANN (Mendes et al., 2002; Green et al., 2012; Zhang et al., 2007; Gudise & Venayagamoorthy, 2003). A recent study optimized the weight, structure, and activation function of ANN using PSO (Das et al., 2015). ABC with backpropagation was used to optimize the weight of ANN and it was observed that ABC and BP integration improved the slow convergence rate issues (Sarangi et al., 2014). ABC was successfully applied to optimize the weight and structure of ANN in Garro et al. (2011). ACO was effectively used for the training purpose of ANN by Blum and Socha (2007), and some other studies also applied ACO to train the ANN model to solve prediction, scheduling, and image recognition problems (Irani & Nasimi, 2011; Shariati et al., 2019). Several studies adopted other metaheuristics methods to train ANN for different applications. Teaching-learning based optimization was applied to train ANN for estimating the energy consumption problem (Uzlu et al., 2014). Harmony search-based neural network was used to classify UCI datasets including breast cancer and thyroid disease (Kulluk et al., 2012). The biogeography-based optimizer (BBO) was also used to train MLP and further applied to classify breast cancer and heart disease datasets (Mirjalili et al., 2014). A social-spider optimization algorithm was employed to train MLP for medical datasets (Pereira et al., 2014).

Metaheuristic optimization techniques have been in practice for more than a decade especially for medical data classification. These techniques improved the results of different machine learning methods used for medical diagnosis. Some of these are discussed in Pham and Triantaphyllou (2009) and Muhaideb and Menai (2014).

Breast cancer is one of the leading causes of death in the world and many studies are reported on breast cancer literature. However, earlier symptoms of breast cancer are not easily identified (Das et al., 2020; Pham & Triantaphyllou, 2009). If a woman lives for 85 years and does not have a family history of breast cancer, then there is also a 0.12 probability of being a breast cancer patient (Bhardwaj & Tiwari, 2015). Metaheuristic based algorithms like fruit fly optimization, homogeneity-based algorithm, genetic programming, and PSO are extensively adopted for breast cancer prediction and it is observed that good results were achieved by the aforementioned algorithms (Brameier & Banzhaf, 2001; Chen et al., 2011; Huang et al., 2019; Pham & Triantaphyllou, 2009; Shen et al., 2016).

Heart disease is also considered in this work, as it is also a life-threatening disease and common in old and middle-aged people. Nevertheless, young people may also suffer from heart disease especially in developing nations like India. A study showed that heart disease is responsible for 24 percent of the total deaths that occur due to non-communicable diseases in India. Heart disease is most common in Asia Pacific and almost 17 million people die every year worldwide due to heart-related problems (Latha & Jeeva, 2019). In the United States (US), a person with Myocardial Infarction, a type of heart disease, is detected in every 34 seconds and the probability of death is 0.15 (Tay et al., 2014). Turabieh (2016) adopted the GWO algorithm to train ANN for accurate prediction of heart disease and observed that the root mean squared error (RMSE) was significantly reduced as compared to standard ANN.

Parkinson's disease (PD) and Alzheimer's disease (AD) are two common neurodegenerative diseases and millions of people are suffering from these diseases throughout the world (Oliva & Hinojosa, 2020). PD is the second major neurodegenerative disease in the world (Cai et al., 2018). A detailed analysis of PD using association rule mining and metaheuristic algorithms was discussed in Altay and Alatas (2020).

Liver is also an important organ of living beings and hepatitis is one of major diseases associated with it. It has been found that the diagnosis of hepatitis is a very difficult process as an expert does many comparisons with previously identified cases. The hepatitis virus may

be of different kinds such as HAV, HBV, HCV, HDV, HEV, HGV, and over 1.5 million deaths worldwide occur due to this viral disease per year (Sartakhti et al., 2012). The medical diagnosis of hepatitis using SVM has been discussed in detail (Chen et al., 2011). Bascil and Temurtas (2011) adopted ANN for the same purpose. It is observed that ANN is widely adopted in the field of data mining for prediction tasks. It is also noticed that various metaheuristic algorithms are integrated with ANN for improving prediction results. Apart from medical diagnosis, ANN is extensively applied for optimizing various applications using metaheuristic methods (Mirjalili, 2015; Mirjalili et al., 2012).

As per the extensive literature review, it is seen that GWO outperforms other SI and EA techniques. Nevertheless, this algorithm suffers from local optima entrapment and slow convergence rate. Long et al. (2018) studied the reasons behind the aforementioned issues and highlighted several reasons:

- (i) In GWO, the control variable \vec{a} is linearly decreased while it must decrease in a non-linear fashion so that better exploration can be achieved during later optimization, and also to maintain rich exploitation.
- (ii) To achieve global optima position, the update rule of classic GWO needs some improvements because the best positions achieved in the previous steps cannot be retained as GWO does not have a memory concept. Therefore, GWO must remember the previous best solution and attain the global optima.

Long et al. (2018) suggested the following solutions to the abovementioned problems:

- (i) Develop a non-linear control parameter \vec{a} that can manage exploration and exploitation in a better way.
- (ii) A memory concept is introduced in GWO that must remember the personal best component inspired through PSO. Furthermore, the position of alpha wolf represents the global best position, and in every iteration, the previous personal best must be stored to achieve the global best position.

The current study proposed a new expression (Equation [26]) for the control variable \vec{a} (to provide an adequate balance of exploration and exploitation) so that GWO could escape from local optima and

slow convergence. Nonetheless, the study kept the position updating rule (Equation [27]) unchanged from the study of Long et al. (2018). The proposed optimization technique is termed as IMGWO since the non-linear control variable \vec{a} resembles the inertia of PSO in behavior (Chatterjee & Siarry, 2006). IMGWO used MSE as fitness function during the training of MLP and obtained a significant improvement in the results as compared to other contemporary metaheuristic methods (GA, PSO, and GWO).

BACKGROUND CONCEPTS

Multilayer Perceptron

ANN consists of three structures, such as single-layer feedforward, multilayer feedforward neural network (FNN), and recurrent neural network (RNN). Several other variants of the three structures are also presented in the literature, viz., convolution neural network, radial basis function, Hopfield network, Boltzmann machine, liquid state machine, Kohonen networks, extreme learning machines, and modular neural networks. However, this study considered the multilayer perceptron (MLP) model. MLP is a kind of FNN with one or more hidden layers between the input and output layers and each node is associated with an activation function. It is assumed that the activation function should be different for the hidden layer and output layer. Nevertheless, this study considered the same activation function for both layers. The activation function used in this study is highlighted in Equation 1.

$$Sig(x) = \frac{1}{1-e^{-x}} \quad (1)$$

Assume that an input layer consists of n number of input neurons, a hidden layer with h number of neurons, and an output layer with o number of neurons. The weighted sum of inputs for MLP can be calculated using Equation 2.

$$S_j = \sum_{i=1}^n (W_{ij} \cdot X_i) - \theta_j, \quad j = 1, 2, \dots, h \quad (2)$$

W_{ij} is the weight of connection from a neuron i in the input layer to a neuron j in the hidden layer and θ_j is the bias associated with

j^{th} neuron in the hidden layer. X_i denotes the input for i^{th} neuron. It is assumed that input layer neurons are passive; they simply pass the information without being processed through the activation function and bias unlike in the hidden and output layers. Neurons at the hidden layer produce output using Equation 3.

$$Sum_j = Sig(S_j) = \frac{1}{(1+e^{(-S_j)})}, \quad j = 1, 2, \dots, h \quad (3)$$

The output of the neurons can be described using Equation 4.

$$O_k = \sum_{j=1}^h W_{jk} \cdot Sum_j - \theta_k, \quad k = 1, 2, \dots, c \quad (4)$$

Where W_{jk} is the weight of the connection between j^{th} hidden neuron to k^{th} output node and θ_k is the bias at k^{th} output neuron.

$$Out_k = Sig(O_k) = \frac{1}{(1+e^{(-O_k)})}, \quad k = 1, 2, \dots, c \quad (5)$$

The output is further passed to the sigmoid function as in Equation 5. A sigmoid function is a kind of squashing function that converts input into an output range (0, 1). It is assumed that denotes distinct classes.

Grey Wolf Optimizer

Grey wolf optimization is inspired through the searching and hunting behaviors of a special family of wolves, named as *Canis lupus*. The searching mechanism of grey wolves follows a hierarchical system that consists of alpha, beta, delta, and gamma wolves. All wolves are found in a group of 5 to 12. However, for the implementation point of view, wolves' population can be defined in the range of 50 to 250. Furthermore, the alpha wolf acts as the leader of the group and has the highest priority. The priority is computed using some fitness functions. The alpha wolf consists of the best fitness value, while beta and delta wolves contain the second best and third best fitness values, respectively. The rest of wolves are known as omega (ω) and follow their superior when searching for the prey and encircling it. The prey is the target solution (most optimal) and it can be explored during the searching process. When the prey is discovered, then the search begins and the iteration variable is set as t . Then, the top three wolves (α , β , and δ) lead the other wolves (ω) to reach the target solution,

i.e., the position of the prey. Each wolf must change its position after every iteration to meet the target. At last, when the alpha attacks the prey, GWO stops and returns the position of the alpha or prey as the possible solution. The encircling mechanism is described through the coefficients as given in Equations 8 to 10 and generic Equations 6 and 7 are used to update their positions around the top best three solutions, i.e., alpha, beta, and delta.

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (6)$$

$$\vec{X}(t) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (7)$$

The two important variables \vec{A} and \vec{C} are responsible for exploration and exploitation, and change as per the expressions $\vec{A} = 2\vec{a}r_1 - \vec{a}$ and $\vec{C} = 2r_2$, where, r_1 and r_2 are the random values responsible for the movement of wolves in multidimensional space. Equations 8 to 10 define the step size toward the best three solutions (Mirjalili et al., 2014).

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (8)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (9)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (10)$$

Equations 11 to 13 describe the three best solutions and the best position is achieved through Equation 14. Thus, it is assumed that the position of the prey is obtained through Equation 14.

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (11)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \quad (12)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (13)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (14)$$

t represents the current iteration in the aforementioned equations (Amirsadri et al., 2018). GWO is described in Algorithm 1.

Algorithm 1: GWO

Declaration 1:

pop is the population of the search agents

t is the iteration variable

$max_iteration$ is the total number of iterations

A and C are necessary variables responsible for exploration and exploitation

\vec{a} is the control variable that changes from 2 to 0 during the course of iterations

\vec{a} is calculated as $\vec{a}(t) = 2 - \frac{2t}{max_iteration}$

Initialize pop , A , C and \vec{a}

Initialize iteration variable $t = 0$

Input: pop , $max_iteration$,

Calculate the fitness for each of the search agents

α wolf = search agent with best fitness

β wolf = search agent with second best fitness

δ wolf = search agent with third best fitness

Assume all, other search agents as ω

while $t < max_iteration$

for $i = 1$ to $1:pop$

 Calculate and update the position of the current search agent using Equation 14

end for

 Update A , C , and \vec{a}

 Calculate the fitness for all of the grey wolves

 Update positions of α , β and δ wolves

$t = t + 1$

end while

Output: return position of α as optimal solution

Particle Swarm Optimization

Like GWO, PSO is also an NIA optimization and it is one of the popular optimization methods presented in the literature among different SI techniques. The basic working of PSO is described through personal thinking factor and collaborative thinking factors as shown in Equation 15. The concept of inertia weight has gained wide popularity especially in the case of PSO, which also proved its importance for exploring optimum solution. Furthermore, PSO

reported the best performance when inertia is in the range [0.9, 1.2] (Shi & Eberhart, 1998).

$$v_{id} = w * v_{id} + \underbrace{c_1 * \text{rand}_1() (p_{id} - x_{id})}_{\text{PeronalThinking}} + \underbrace{c_2 * \text{rand}_2() (p_{gd} - x_{id})}_{\text{CollaberativeThinking}} \quad (15)$$

$$x_{id} = x_{id} + v_{id} \quad (16)$$

v_{id} is the velocity of i^{th} participating particles, c_1, c_2 are the positive coefficients, while $\text{rand}_1()$ and $\text{rand}_2()$ functions generate random numbers in the range [0,1]. The variables p_{id} and p_{gd} denotes the personal best and global best positions of particles, respectively. Chatterjee and Siarry (2006) suggested that a better balance between exploration and exploitation can be achieved through Equation 17.

$$v_{idnew} = w_{iteration}(v_{idold}) + \psi_i(p_{id} - x_{id}) + \psi_g(p_{gd} - x_{id}) \quad (17)$$

v_{id} is the velocity vector for the iteration i , p_{id} and p_{gd} are the personal best and global positions as per classic PSO definitions. ψ_i and ψ_g are the weighting variables of stochastic nature for balancing private thinking component and global thinking component.

The update in inertia $w_{iteration}$ is done through the updating rule according to Equation 18 (Chatterjee & Siarry, 2006).

$$w_{iteration} = \varphi(\text{iteration}) = \left\{ \frac{\{\text{iteration}_{max} - \text{iteration}\}^n}{\{\text{iteration}_{max}\}^n} \right\} (w_{initial} - w_{final}) + w_{final} \quad (18)$$

Where iteration_{max} denotes total iterations and iteration denotes current iteration. It is assumed that inertia weight maintains a good balance between exploitation and exploration using non-linear modulation index n . It becomes linear when it equals to 1. The inertia weights are changed from $w_{initial}$ to w_{final} during successive iterations. The integration of inertia weight concept into the PSO algorithm reflects the non-linear behavior and also improves the simulation results in a significant manner (Chatterjee & Siarry, 2006).

Exploration and Exploitation

All non-deterministic metaheuristic algorithms are described as population-based algorithms. These algorithms consist of a natural tendency to maintain the balance between exploration and exploitation

to avoid local optima. Without the proper balancing between these two processes, i.e., exploration and exploitation, the optimal solution cannot be achieved for optimization problems. Exploration or diversification generates the solution in a distant area with respect to the current solution so that the generated solution proceeds to the global solution. Whereas exploitation or intensification searches the solution near to the current solution (Emery et al., 2018). For all stochastic population-based metaheuristic methods, exploration means to use the previously attained knowledge for finding better and probably optimal solutions. Classic GWO clearly stated that candidate solutions diverge (exploration) with respect to the target when $|\vec{A}| > 1$ and converge (exploitation) toward the target when $|\vec{A}| < 1$ and in turn, control parameter is decreased from 2 to 0 to emphasize divergence and convergence during successive iterations (Mirjalili et al., 2014). However, it is practically observed that a linear decrement of does not reflect the actual search process due to complex and non-linear nature of exploration and exploitation. Long et al. (2018) developed a new variant of GWO that integrated PSO and GWO. The exploration and exploitation processes were improved by mimicking the control variable \vec{a} as the inertia w concept (a weight alike parameter) of PSO. The aim of inertia factor is to make PSO more effective in terms of performance. Therefore, Long et al. (2018) developed a non-linear control variable \vec{a} that maintains the balance between exploration and exploitation as well as an adjustment strategy that is adjusted through a decay function (Long et al., 2018).

Learning of the MLP Using a Metaheuristic Technique

There are certain encoding schemes available for learning the weight and bias parameters through metaheuristics (Zhang et al., 2007).

- (i) **Vector Encoding Scheme:** In this scheme, each search agent is encoded in the form of a vector and this vector represents the weights and biases of MLP as shown in Equation 19:

$$\vec{V}_{\text{Search_Agent}} = \{\vec{wt} | wt: \text{connection weights}, \vec{b} | b: \text{biases}\} =$$

$$[wt_{i,p_{11}}, wt_{i,p_{12}}, \dots, wt_{i,p_{1h}}, \dots, wt_{i,p_{21}}, wt_{i,p_{22}}, \dots, wt_{i,p_{2h}}, \dots, \dots, wt_{i,p_{nh}},$$

$$wt_{h,o_{11}}, wt_{h,o_{12}}, \dots, wt_{h,o_{1o}}, \dots, wt_{h,o_{21}}, wt_{h,o_{22}}, \dots, wt_{h,o_{2o}}, \dots, \dots, wt_{h,o_{ho}},$$

$$b_{h1}, b_{h2}, \dots, b_{hh}, \dots, \dots, b_{o1}, b_{o2}, \dots, \dots, \dots, \dots, b_{oo}] \quad (19)$$

Where \overrightarrow{wt} are the weights of different connections between the input layer and hidden layer; and the weights of different connections between the hidden layer and the output layer. For example, $wt_{i_{p11}}$ is the weight of the connection from node 1 in the input layer to node 1 in the hidden layer, and $wt_{h_{o21}}$ is the weight of the connection from node 2 in the hidden layer to node 1 in the output layer. Bias is associated with the neurons of the hidden and output layers only. Bias b_{h1} is associated with the first node in the hidden layer and b_{hh} is the bias associated with the last node in the hidden layer. Similarly, b_{1o} and b_{oo} are the biases for the first and last nodes in the output layer. The architecture of MLP is $n - h - o$, where n is the number of neurons in the input layer, h is the number of neurons in the hidden layer, and o is the number of neurons in the output layer.

The search agent matrix for vector encoding scheme can be described using Equation 20:

$$Search_Agent_Matrix = [Search_Agent(1); Search_Agent(2) \dots Search_Agent(M)] \quad (20)$$

Where M describes the population of search agents or size of the swarm.

(ii) **Matrix Encoding Scheme:** In this encoding, each search agent is encoded in the form of matrices. For a given MLP with structure , the matrices can be represented as:

$$wt_1 = \begin{bmatrix} wt_{i_{h11}} & wt_{i_{h21}} & \dots & wt_{i_{hn1}} \\ \vdots & \vdots & & \vdots \\ wt_{i_{h1h}} & wt_{i_{h2h}} & \dots & wt_{i_{hnh}} \end{bmatrix} \quad (21)$$

$$wt'_2 = \begin{bmatrix} wt_{h_{o11}} & wt_{h_{o12}} & \dots & wt_{h_{o1o}} \\ \vdots & \vdots & & \vdots \\ wt_{h_{oh1}} & wt_{h_{oh2}} & & wt_{h_{oho}} \end{bmatrix} \quad (22)$$

$$\theta_{hidden} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_h \end{bmatrix} \quad (23)$$

$$\theta_{output} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_o \end{bmatrix} \quad (24)$$

The structure of the search agent using the matrix encoding scheme can be represented as:

$$Search\ agent\ (:, :, i) = [wt_1, \theta_{hidden}, wt'_2, \theta_{output}] \quad (25)$$

The matrices wt_1 and wt_2' are the weight matrices for the hidden layer and the transpose of the output layer, whereas θ_{hidden} and θ_{output} are the bias matrices for the hidden layer and output layer, respectively. Variable i can vary from 1 to any population size M . Typically, population size can vary in the range of 50 to 500 depending on the type of problems being solved.

- (iii) **Binary Encoding Scheme:** It is a kind of encoding scheme, where each search agent can be either 0 or 1 and the weight for MLP training can be described through a series of zeros and ones. The cost of the encoding scheme is comparatively high for training purposes; however, this encoding is good for feature selection tasks.

The abovementioned encoding schemes are used to represent the population of metaheuristic algorithms with ANN, specifically for MLP. Equations 19 and 20 are vector representations of weights and biases, while Equations 21 to 25 represent the weights and biases in matrix form.

INERTIA MOTIVATED GREY WOLF OPTIMIZATION

In this section, the new swarm optimization algorithm named IMGWO is discussed and further, IMGWO is used to train an MLP.

Inertia Motivated Grey Wolf Optimization

Through the literature, it is observed that the exploration and exploitation processes of GWO depend on the behavior of control variable \vec{a} , since it determines the vectors \vec{A} and \vec{C} (using Equations 6 and 7) that can affect the final solution. The current study focuses on the non-linear behavior of control variable and enhances the non-linearity of by proposing Equation 26.

$$\vec{a}(t) = \vec{a}_{ini} - (\vec{a}_{ini} - \vec{a}_{fin}) \times \left(\left(\frac{1}{e-1} \right) \times \left(e^{\frac{t}{max_iteration}} - 1 \right) \right) \quad (26)$$

Consider that t denotes the current iteration, $max_iteration$ denotes the total number of iterations, \vec{a}_{ini} and \vec{a}_{fin} denote the initial and final values of control variable \vec{a} , respectively. The non-linear decay of the control variable is expressed through Equation 26 and also manages the balance between exploitation and exploration in an effective manner. This study uses the rule for updating the position of grey wolf that is mentioned in Equation 27 (Long et al., 2018).

$$X(t + 1) = w \cdot \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} + c_1 \cdot r_3 \cdot (\vec{X}_{pbest} - \vec{X}) + c_2 \cdot r_4 \cdot (\vec{X}_1 - \vec{X}) \quad (27)$$

Where t is the current iteration, r_3, r_4 are some random variables in the range of 0 to 1. Further, c_1 and c_2 represent the individual memory coefficient and population communication coefficient in the range of [0, 1], respectively. denotes the personal best position during successive iteration. is considered as the global best solution of position of α (alpha wolf) and is the weight similar to inertia weight in PSO. The value of variable changes from an initial value to as per Equation 28 with respect to iteration variable (Long et al., 2018).

$$w = f(t) = \frac{t_{max_iteration} - t}{t_{max_iteration}} x (w_f - w_0) + w_f \quad (28)$$

Where $t_{max_iteration}$ is the total number of iterations specified in the algorithm.

Several findings can be highlighted as follows: (i) The variable w can increase the speed for exploitation; (ii) random variables r_3, r_4 are responsible for improving the degree of randomness and exploration; and (iii) coefficients c_1 and c_2 are responsible for better balancing among exploration and exploitation. The aim of the proposed study is to enhance the balance between exploration and exploitation by introducing a non-linear control variable \vec{a} as expressed in Equation 26. The study used the position updating rule as expressed in Equation 27 that determines the global best position. Finally, the abovementioned improvements are incorporate into the GWO algorithm and proposed to train MLP, as during the training process, MLP is often trapped in local optima. The summary of IMGWO is described in Algorithm 2.

Algorithm 2: IMGWO

Declaration 2:

pop is the population of the search agents

t is the iteration variable

max_iteration is the total number of iterations

A and *C* are necessary variables responsible for exploration and exploitation

\vec{a} is the control variable that changes according to Equation 26;

Initialize *pop*, *A*, *C* and \vec{a}

Initialize iteration variable *t* = 0

Input: *pop*, *max_iteration*

Calculate the fitness for each of the search agents using Equations 30 and 31

α wolf = search agent with best fitness

β wolf = search agent with second best fitness

δ wolf = search agent with third best fitness

Assume all, other search agents as omega ω

while *t* < *max_iteration*

for *i* = 1 to 1: *pop*

 Calculate *was* per Equation 28

 Update the position of the current search agent using Equation 27

end for

 Update *A*, *C*, and \vec{a}

 Recalculate the fitness for all of the grey wolves using Equations 30 and 31

 Update positions of α , β and δ wolves

t = *t* + 1

end while

Output: return position of α as optimal solution

Training of MLP using IMGWO

This section discusses the IMGWO-based training algorithm for MLP. IMGWO attained promising results during the experimental setup and attracted the attention for solving the weight optimization problem of ANN. The algorithm managed better trade-off between the divergence and convergence processes, which is one of the essential steps to obtain competitive results using metaheuristic techniques. The current study focuses on optimizing the weights and biases during the training of MLP using IMGWO. Furthermore, the vector encoding scheme (already discussed in the background concepts) is adopted to represent the weights and biases of MLP for search agents (wolves) (α , β , δ , and ω). The weight and bias vectors (as shown in Equation 19) represent the dimensions or variables for IMGWO. The variables should be optimized to obtain maximum classification accuracy. Classification or prediction accuracy also depends on

another important measure, i.e., MSE. MSE can be considered as an objective function for evaluating classifier performance.

The training of MLP is also considered as a challenging problem due to large search space and gradient descent nature of weight and bias. To meet the aforementioned challenges of MLP, the proposed IMGWO-MLP framework can be described in Algorithm 3.

Algorithm 3: IMGWO-MLP

Declaration 3:

i-h-o is the structure of neural network where

i is the number of neurons in input layer

h is the number of neurons in hidden layer

o is the number of neurons in output layer

Dimensions are the number of weights and biases expressed in Equation 19

MSE is the mean squared error also known as average training error expressed in Equation 30

Calculate the number of dimensions as per formulate given in Equation 29

Initialize all the dimensions in a suitable range

Input: *training samples of a medical dataset, and set of weights and biases (dimensions)*

for *each training sample*

Apply Algorithm 2 (IMGWO) to each of the dimensions to obtain MSE and optimum values of weights and biases

Use the optimum weights and biases to classify the sample

end for

Output: *MSE, and set of optimum weights and biases*

$$\text{Dimensions} = \text{Neurons}_{\text{hidden}}(\text{Neurons}_{\text{input}} + \text{Neurons}_{\text{output}}) + \text{Biases}_{\text{hidden}} + \text{Biases}_{\text{output}} \quad (29)$$

In the IMGWO algorithm, the fitness of search agents is computed using Equations 30 and 31. For simplicity, an MLP with one hidden layer is considered in the proposed study. The input layer consists of n neurons, whereas number of neurons in the hidden layer is h . It is assumed that the input dataset contains q number of training patterns and each pattern could be classified in m number of classes. Therefore, the total neurons in the output layer are set to m . The output can be described in terms of average training error ($Av g_{Err}$) for a given input unit i and pattern p^{th} pattern as follows:

$$Av g_{Err} = \sum_{p=1}^q \frac{\sum_{i=1}^m (ob_i^p - exp_i^p)^2}{q} \quad (30)$$

Where ob_i^p and exp_i^p are the observed and expected outputs of the input unit i with respect to p^{th} training pattern.

Finally, the IMGWO-based training of MLP can be formulated as a minimization problem for the variables defined in Equation 19. The average training error Avg_{Err} is described using Equation 31, which acts as a fitness function for all search agents.

$$\text{Min. (F}(\vec{V}_{\text{Seach_Agent}} = Avg_{Err})) \quad (31)$$

EXPERIMENTAL STUDY

The four benchmark medical datasets were taken from the UCI machine learning repository and a description of these datasets is given in Table 1. All datasets had two classes; therefore, the medical data classification problem could be interpreted as a two-class problem or binary classification.

Table 1

Database Descriptions

Database	#input	#Instances	Missing values	Structure of MLP (I-H-O)
Heart disease	13	303	No	13-27-1
Breast cancer	9	699	Yes	9-19-1
Hepatitis	19	155	Yes	19-39-1
Parkinson's	22	195	No	22-45-1

The second column denotes the total number of attributes, whereas the third column indicates the total number of samples. The fourth column provides the information regarding the presence of missing values in the datasets. The last column describes the structure of MLP. In the MLP structure, 'I' denotes the total number of input nodes in the input layer, 'H' denotes the total number of nodes in the hidden layer, and 'O' indicates the total number of nodes in the output layer. The MLP structure contains a single hidden layer for all datasets. For every MLP structure, the total number of neurons in the hidden layer is calculated as $2*n+1$, where n is the number of neurons in the input layer. The missing values are replaced with mean values of the corresponding attributes and normalization can be performed through the *mapminmax()* function.

The heart disease dataset consists of 303 data instances, 13 attributes, and one class attribute. Originally, it has 76 attributes, however, most of the irrelevant attributes (e.g., ID, social security number, etc.) were removed during the pre-processing task. All attributes are numeric and the name of class attribute is *num*, which is either 0 or more than 0. However, it will always be less than 1.0 that stands for the absence of heart disease, whereas a value near to 1 has the worst situation. Breast cancer data is a medium-size data, named Breast Cancer Wisconsin (original) (BCW) dataset and it contains a total of 9 attributes (ID number is omitted) and 699 data instances. The tenth attribute is the class attribute that is represented through either benign (non-cancerous) or malignant (cancerous). The hepatitis dataset contains a small number of data instances (total 155) as given in Table 1, despite having a large number of attributes (total 19). Out of 19 attributes, 13 attributes are of binary type while other attributes contain numeric values. The class attribute has two distinct values, i.e., 'die' and 'live'. The Parkinson's disease (PD) dataset is a medium-size data and has 22 attributes and one class attribute. The PD dataset consists of 195 subjects. The class attribute discriminates the patients as either PD-affected or the healthier one. The disease identification in PD is associated with the differences observed when pronouncing the vowels.

Experimental Setup

IMGWO is a modified metaheuristic computational approach that was applied to refine the classification results of ANN. MSE could be considered as a fitness function for IMGWO and it should be minimized in successive iteration for attaining optimal solution. The simulation results of IMGWO -MLP classifier was compared with ANN-based classifier using three well-known metaheuristic techniques, namely GA, PSO, and GWO.

Each dataset was partitioned into training and testing sets, whereby the partitioning ratio was 70:30, i.e., 70 percent of the data were used for training the network and the rest of the data was used to obtain the classification results using IMGWO-MLP. Three other competent metaheuristics-based classification (GWO, PSO, and GA) techniques were compared with IMGWO for performance evaluation. The implementation details of the proposed IMGWO-MLP and other three techniques (GWO, PSO, and GA) are given in Table 2.

Table 2

Implementation Details

Parameters	IMGWO	GWO	PSO	GA
Population	250	250	250	250
Max. Iteration	500	500	500	500
Other	\vec{a} =decreased linearly	\vec{a} =decreased	$c_1=1$	$Probability_{crossover} = 1$
Technique-	from 2 to 0 over	linearly from 2 to	$c_2=1$	$Probability_{mutation}$
Specific	course of the iteration,	0 over course of	$w=0.3$	
Parameters	$w=2, c_1=1, c_2=1$	the iteration		

The experiments were implemented in MATLAB software tool (version 2018) and all user-defined parameters were set prior to executing the experiments. The total number of iterations was set to 500 and the population of search agents was set to 250. In GWO, \vec{a} was initially set to 2, and it changed linearly during the successive iterations and reached 0. In IMGWO, the initial value of \vec{a} was 2, and reached 0 during the successive iterations; however, it decreased non-linearly. In IMGWO, a new parameter w was proposed and initially, the value of w was set to 2. In PSO, w was set to 0.3, and the personnel learning coefficient c_1 and social learning coefficients c_2 were set to 1. In GA, the single point probability of crossover was set as 1, while the mutation uniform probability was set as 0.01.

Result and Discussion

This section presents the discussion on the simulation results obtained through the MLP classification model trained by IMGWO and the other techniques, namely GWO, GA, and PSO. The study included four benchmark medical datasets, i.e., heart disease, breast cancer, hepatitis, and Parkinson's disease for implementation tasks. All datasets were partitioned into two sets as training and testing sets in the ratio of 70:30. The first phase constructed a model using the training set for each medical dataset and further, optimal weights and biases were computed using the IMGWO technique. In the second phase, model usage was described through the testing set for each medical dataset. The proposed models (IMGWO-MLP, GWO-MLP, PSO-MLP, and GA-MLP) executed ten different runs for each medical dataset. Tables 3–7 present the simulation results of all MLP models using all medical datasets. The results are presented as arithmetic

mean of ten different independent runs. The convergence rates of the training task using IMGWO, GWO, GA, and PSO are shown in Figures 1–4.

Figure 1

Convergence Curve for Heart Disease.

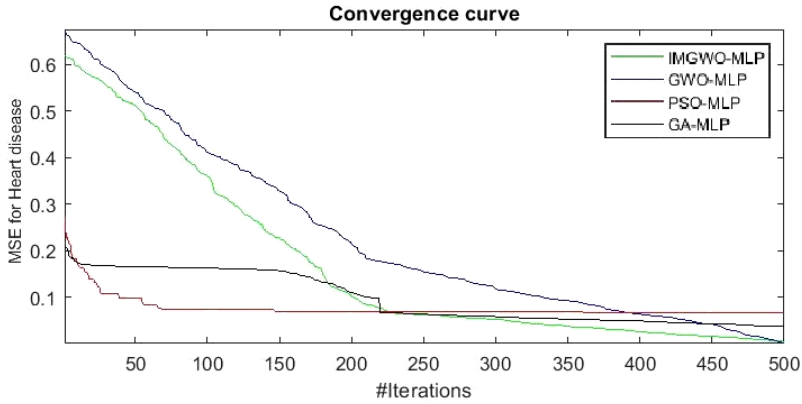
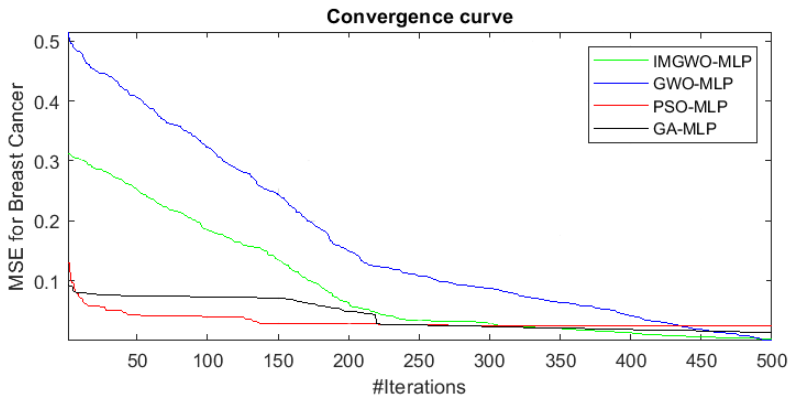


Figure 2

Convergence Curve for Breast Cancer.



It was observed that the convergence rate of IMGWO was better than all other metaheuristic techniques (GWO, GA, and PSO) used for comparisons. It could be achieved by balancing the exploration and exploitation processes of GWO through control variable \vec{a} and position updates of wolves.

Figure 3

Convergence Curve for Hepatitis.

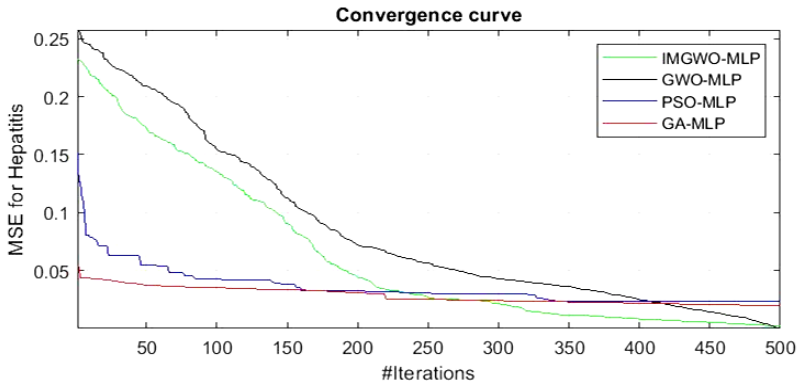
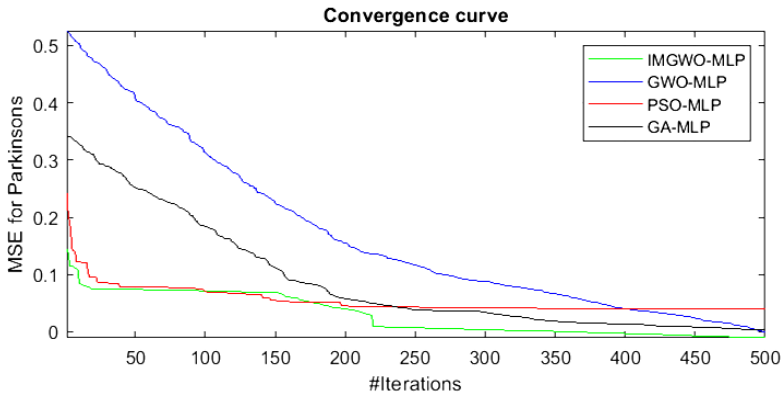


Figure 4

Convergence Curve for Parkinson's Disease.



The framework of IMGWO used a new expression for Equation 26 and a position updating rule Equation 27. These two improvements enhanced the convergence rate of IMGWO in a significant manner. The other significant improvement, i.e., avoidance of entrapment in local optima, could not be directly seen in the recorded parameter, but was also improved as IMGWO obtained better convergence rate and reduced MSE.

Table 3 illustrates the simulation results using MSE using all four models. It showed that IMGWO-MLP achieved a low MSE rate

except for GWO (minimum MSE for hepatitis disease in the testing phase).

Table 3

Mses Over Ten Runs for Different Datasets

Medical Dataset	Training MSE (Avg.)				Testing MSE (Avg.)			
	IMGWO	GWO	GA	PSO	IMGWO	GWO	GA	PSO
Heart disease	0.037	0.038	0.041	0.155	0.028	0.032	0.033	0.135
Breast cancer	0.030	0.037	0.039	0.039	0.030	0.038	0.037	0.053
Hepatitis	0.031	0.052	0.032	0.035	0.031	0.030	0.032	0.040
Parkinson's disease	0.037	0.104	0.126	0.137	0.030	0.099	0.119	0.112

Table 4

Accuracies Over Ten Runs for Different Datasets

Medical Dataset	Training Accuracies (Avg.)				Testing Accuracies (Avg.)			
	IMGWO	GWO	GA	PSO	IMGWO	GWO	GA	PSO
Heart disease	84.91	83.962	83.02	81.13	90.11	79.12	85.71	81.32
Breast cancer	95.92	94.49	93.67	91.84	92.82	79.43	79.9	78.47
Hepatitis	89.91	87.156	84.4	81.65	84.78	80.43	82.61	73.91
Parkinson's disease	80.29	77.372	75.91	76.64	82.76	75.86	77.59	68.97

The average accuracy of all models is presented in Table 4. The results claimed that IMGWO-MLP had a better accuracy rate in comparison to GWO, GA, and PSO. IMGWO-MLP also achieved better accuracies during the training phase. The best mean accuracies achieved by the IMGWO-MLP model were 90.11 percent, 92.82 percent, 84.78 percent, and 82.76 percent with all medical datasets.

Table 5

Sensitivities Over Ten Runs for Different Datasets

Medical Dataset	Training Sensitivities (Avg.)				Testing Sensitivities (Avg.)			
	IMGWO	GWO	GA	PSO	IMGWO	GWO	GA	PSO
Heart disease	84.21	83.16	82.11	78.95	95.45	70.45	90.91	79.55
Breast cancer	94.44	93.33	92.22	88.89	95.08	81.97	80.33	77.05
Hepatitis	78.26	73.91	69.57	65.22	77.78	66.67	88.89	55.56
Parkinson's disease	67.57	62.16	59.46	67.57	81.82	63.64	72.73	54.55

Table 6

Specificities Over Ten Runs for Different Datasets

Medical dataset	Training Specificities (Avg.)				Testing Specificities (Avg.)			
	IMGWO	GWO	GA	PSO	IMGWO	GWO	GA	PSO
Heart disease	85.47	84.62	83.76	82.91	85.11	87.23	80.85	82.98
Breast cancer	96.77	95.16	94.52	93.55	91.89	78.38	79.73	79.05
Hepatitis	93.02	90.7	88.37	86.05	86.49	83.78	81.08	78.38
Parkinson's disease	85.00	83.00	82.00	80.00	82.98	78.72	76.6	72.34

The simulation results of sensitivities and specificities parameters are also presented in Tables 5–6 and the best values are highlighted in bold. For designing an expert system for medical diagnosis, it is necessary to compare true-positive rate (sensitivity) and true-negative rate (specificity).

ROC curves were also computed as shown Figures 5–6. It is also considered as an important measure in the field of medical data mining. The ROC curve gave true positive classification rate at the cost of false-positive classification rate. The area under the curves (AUC) is also illustrated in Table 7. It was found that better AUC

values were obtained by IMGWO for all datasets and the AUC values were also near to 1, which showed the significance of the proposed IMGWO-MLP model.

Table 7

AUC for Different Datasets

Medical dataset	Training AUC				Testing AUC			
	IMGWO	GWO	GA	PSO	IMGWO	GWO	GA	PSO
Heart disease	0.89	0.867	0.863	0.842	0.881	0.782	0.845	0.786
Breast cancer	0.936	0.927	0.915	0.899	0.921	0.81	0.788	0.779
Hepatitis	0.888	0.836	0.811	0.805	0.847	0.798	0.819	0.734
Parkinson's disease	0.802	0.782	0.779	0.783	0.789	0.756	0.752	0.699

Figure 5

ROC Curve for IMGWO for Heart Disease .

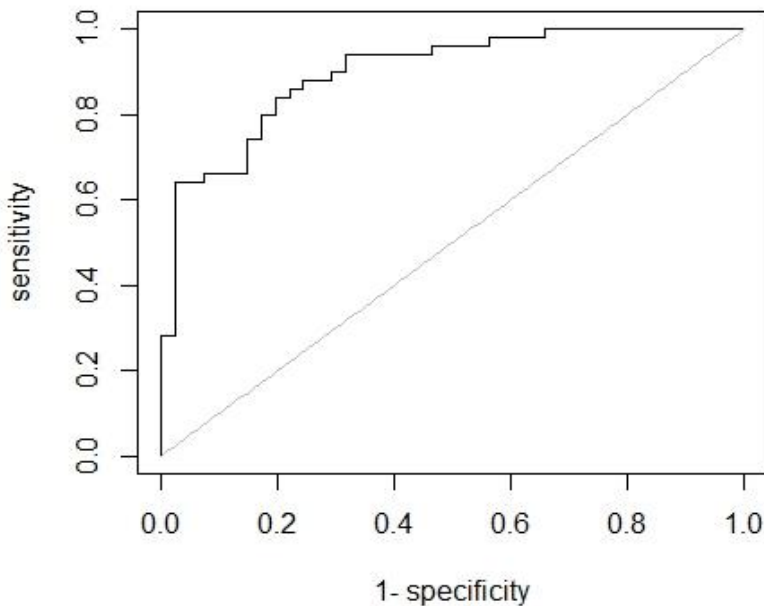
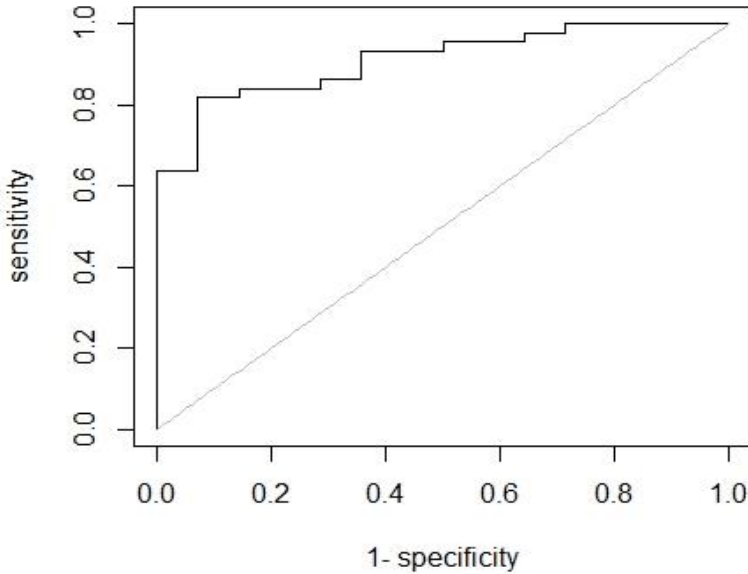


Figure 6

ROC Curve for IMGWO for Parkinson's Disease.



Conclusion and Future Scope

In this study, a new metaheuristic algorithm, IMGWO, was developed for training the MLP model. Furthermore, the searching (exploration and exploitation) capability of GWO was enhanced with the concept of time-variant inertia. The proposed method proved its performance in terms of convergence. The four benchmark medical datasets were considered to evaluate the performance of the IMGWO-MLP model and results were compared with three popular MLP models. IMGWO can effectively optimize weight and bias, and in turn, improves the performance in terms of convergence rate as well reduces MSE. Some other performance measures like accuracy, sensitivity, specificity, ROC, and AUC were also computed to signify the performance of IMGWO-MLP using all datasets. It is found that the proposed method supersedes most of the datasets as compared to the rest of the models. The key points of the IMGWO-based training algorithm are highlighted as follows: (i) IMGWO has better management of exploration and exploitation; (ii) It is a well-defined model inspired through mathematical formulations that can make it

easy to comprehend and implement; and (iii) No technique-specific parameters adjustment is done, only some common parameters of population-based metaheuristic methods are adjusted.

Finally, it can be concluded that the proposed IMGWO technique is a better alternative solution with respect to many contemporary metaheuristic methods including GA, PSO, and GWO to train the neural network, especially for medical data classification. Future work may incorporate other neural network techniques such as deep neural networks or RNN and investigate the performance of IMGWO.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or non-profit sectors.

REFERENCES

- Akay, M. F. (2009). Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2), (Part 2), 3240–3247. <https://doi.org/10.1016/j.eswa.2008.01.009>
- AlMuhaideb, S., & Menai, M. E. B. (2014). HColonies: A new hybrid metaheuristic for medical data classification. *Applied Intelligence*, 41(1), 282–298. <https://doi.org/10.1007/s10489-014-0519-z>
- Altay, E. V., & Alatas, B. (2020). Association analysis of Parkinson disease with vocal change characteristics using multi-objective metaheuristic optimization. *Medical Hypotheses*, 141, 109722. <https://doi.org/10.1016/j.mehy.2020.109722>
- Amirsadri, S., Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2018). A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. *Neural Computing and Applications*, 30(12), 3707–3720. <https://doi.org/10.1007/s00521-017-2952-5>
- Bascil, M. S., & Temurtas, F. (2011). A study on hepatitis disease diagnosis using multilayer neural network with Levenberg Marquardt training algorithm. *Journal of Medical Systems*, 35(3), 433–436. <https://doi.org/10.1007/s10916-009-9378-2>

- Bhardwaj, A., & Tiwari, A. (2015). Breast cancer diagnosis using genetically optimized neural network model. *Expert Systems with Applications*, 42(10), 4611–4620. <https://doi.org/10.1016/j.eswa.2015.01.065>
- Blum, C., & Socha, K. (2005). Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)* (pp. 6-pp.). <https://doi.org/10.1109/ichis.2005.104>
- Brameier, M., & Banzhaf, W. (2001). Evolving teams of predictors with linear genetic programming. *Genetic Programming and Evolvable Machines*, 2(4), 381–407. <https://doi.org/10.1023/A:1012978805372>
- Brent, R. P. (1991). Fast training algorithms for multilayer neural nets. *IEEE Transactions on Neural Networks*, 2(3), 346–354. <https://doi.org/10.1109/72.97911>
- Cai, Z., Gu, J., Wen, C., Zhao, D., Huang, C., Huang, H., Tong, C., Li, J., & Chen, H. (2018). An intelligent Parkinson's disease diagnostic system based on a chaotic bacterial foraging optimization enhanced fuzzy KNN approach. *Computational and Mathematical Methods in Medicine*, 2018. <https://doi.org/10.1155/2018/2396952>
- Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research*, 33(3), 859–871. <https://doi.org/10.1016/j.cor.2004.08.012>
- Chen, H.-L., Liu, D.-Y., Yang, B., Liu, J., & Wang, G. (2011). A new hybrid method based on local fisher discriminant analysis and support vector machines for hepatitis disease diagnosis. *Expert Systems with Applications*, 38(9), 11796–11803. <https://doi.org/10.1016/j.eswa.2011.03.066>
- Das, G., Pattnaik, P. K., & Padhy, S. K. (2014). Artificial neural network trained by particle swarm optimization for non-linear channel equalization. *Expert Systems with Applications*, 41(7), 3491–3496. <https://doi.org/10.1016/j.eswa.2013.10.053>
- Das, R., Turkoglu, I., & Sengur, A. (2009). Effective diagnosis of heart disease through neural networks ensembles. *Expert Systems with Applications*, 36(4), 7675–7680. <https://doi.org/10.1016/j.eswa.2008.09.013>
- Das, S., Mishra, S., & Senapati, M. R. (2020). New approaches in metaheuristic to classify medical data using artificial neural

- network. *Arabian Journal for Science and Engineering*, 45(4), 2459–2471. <https://doi.org/10.1007/s13369-019-04026-y>
- Ding, S., Su, C., & Yu, J. (2011). An optimizing BP neural network algorithm based on genetic algorithm. *Artificial Intelligence Review*, 36(2), 153–162. <https://doi.org/10.1007/s10462-011-9208-z>
- Djema, M. A., Boudour, M., Agbossou, K., Cardenas, A., & Doumbia, M. L. (2019). Adaptive direct power control based on ANN-GWO for grid interactive renewable energy systems with an improved synchronization technique. *International Transactions on Electrical Energy Systems*, 29(3), e2766. <https://doi.org/10.1002/etep.2766>
- Emary, E., Zawbaa, H. M., & Grosan, C. (2018). Experienced gray wolf optimization through reinforcement learning and neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(3), 681–694. <https://doi.org/10.1109/TNNLS.2016.2634548>
- Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172, 371–381. <https://doi.org/10.1016/j.neucom.2015.06.083>
- Garro, B. A., Sossa, H., & Vazquez, R. A. (2011). Artificial neural network synthesis by means of artificial bee colony (ABC) algorithm. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 331–338). <https://doi.org/10.1109/CEC.2011.5949637>
- Gori, M., & Tesi, A. (1992). On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1), 76–86. <https://doi.org/10.1109/34.107014>
- Green, R. C., Wang, L., & Alam, M. (2012). Training neural networks using central force optimization and particle swarm optimization: Insights and comparisons. *Expert Systems with Applications*, 39(1), 555–563. <https://doi.org/10.1016/j.eswa.2011.07.046>
- Gudise, V. G., & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)* (pp. 110–117). IEEE. <https://doi.org/10.1109/m/SIS.2003.1202255>

- Haykin, S. (1994). *Neural networks: A comprehensive foundation* (1st ed.). Prentice Hall PTR.
- Huang, H., Feng, X., Zhou, S., Jiang, J., Chen, H., Li, Y., & Li, C. (2019). A new fruit fly optimization algorithm enhanced support vector machine for diagnosis of breast cancer based on high-level features. *BMC Bioinformatics*, 20(8), 290. <https://doi.org/10.1186/s12859-019-2771-z>
- Irani, R., & Nasimi, R. (2011). Evolving neural network using real coded genetic algorithm for permeability estimation of the reservoir. *Expert Systems with Applications*, 38(8), 9862–9866. <https://doi.org/10.1016/j.eswa.2011.02.046>
- Karaboga, D., Akay, B., & Ozturk, C. (2007). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In V. Torra, Y. Narukawa, & Y. Yoshida (Eds.), *Modeling Decisions for Artificial Intelligence* (Vol. 4617, pp. 318–329). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73729-2_30
- Khairuzzaman, A. K. M., & Chaudhury, S. (2017). Multilevel thresholding using grey wolf optimizer for image segmentation. *Expert Systems with Applications*, 86, 64–76. <https://doi.org/10.1016/j.eswa.2017.04.029>
- Kulluk, S., Ozbakir, L., & Baykasoglu, A. (2012). Training neural networks with harmony search algorithms for classification problems. *Engineering Applications of Artificial Intelligence*, 25(1), 11–19. <https://doi.org/10.1016/j.engappai.2011.07.006>
- Latha, C. B. C., & Jeeva, S. C. (2019). Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked*, 16, 100203. <https://doi.org/10.1016/j.imu.2019.100203>
- Li, Q., Chen, H., Huang, H., Zhao, X., Cai, Z., Tong, C., Liu, W., & Tian, X. (2017). An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis. *Computational and Mathematical Methods in Medicine*, 2017, 1–15. <https://doi.org/10.1155/2017/9512741>
- Lin, R.-H., & Chuang, C.-L. (2010). A hybrid diagnosis model for determining the types of the liver disease. *Computers in Biology and Medicine*, 40(7), 665–670. <https://doi.org/10.1016/j.compbiomed.2010.06.002>
- Long, W., Jiao, J., Liang, X., & Tang, M. (2018). Inspired grey wolf optimizer for solving large-scale function optimization problems. *Applied Mathematical Modelling*, 60, 112–126. <https://doi.org/10.1016/j.apm.2018.03.005>

- Long, W., Liang, X., Cai, S., Jiao, J., & Zhang, W. (2017). A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. *Neural Computing and Applications*, 28(S1), 421–438. <https://doi.org/10.1007/s00521-016-2357-x>
- Maglogiannis, I., Loukis, E., Zafiropoulos, E., & Stasis, A. (2009). Support vectors machine-based identification of heart valve diseases using heart sounds. *Computer Methods and Programs in Biomedicine*, 95(1), 47–61. <https://doi.org/10.1016/j.cmpb.2009.01.003>
- Mendes, R., Cortez, P., Rocha, M., & Neves, J. (2002). Particle swarms for feedforward neural network training. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)* (Vol. 2, pp. 1895–1899). IEEE. <https://doi.org/10.1109/ijcnn.2002.1007808>
- Mirjalili, Seyedali. (2015). How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43(1), 150–161. <https://doi.org/10.1007/s10489-014-0645-7>
- Mirjalili, Seyedali, Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mirjalili, Seyedali, Mirjalili, S. M., & Lewis, A. (2014). Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences*, 269, 188–209. <https://doi.org/10.1016/j.ins.2014.01.038>
- Mirjalili, SeyedAli, MohdHashim, S. Z., & MoradianSardroudi, H. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, 218(22), 11125–11137. <https://doi.org/10.1016/j.amc.2012.04.069>
- Nosratabadi, S., Szell, K., & Beszedes, B. (2020). *Comparative analysis of ANN-ICA and ANN-GWO for crop yield prediction*. 5.
- Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60, 97–116. <https://doi.org/10.1016/j.engappai.2017.01.013>
- Oliva, D., & Hinojosa, S. (Eds.). (2020). *Applications of hybrid metaheuristic algorithms for image processing* (Vol. 890). Springer International Publishing. <https://doi.org/10.1007/978-3-030-40977-7>

- Pereira, L. A. M., Rodrigues, D., Ribeiro, P. B., Papa, J. P., & Weber, S. A. T. (2014). Social-spider optimization-based artificial neural networks training and its applications for Parkinson's disease identification. In *2014 IEEE 27th International Symposium on Computer-Based Medical Systems* (pp. 14–17). <https://doi.org/10.1109/CBMS.2014.25>
- Pham, H. N. A., & Triantaphyllou, E. (2009). An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets. *Expert Systems with Applications*, 36(5), 9240–9249. <https://doi.org/10.1016/j.eswa.2008.12.007>
- Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2018). Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems. *Applied Soft Computing*, 69, 504–515. <https://doi.org/10.1016/j.asoc.2018.05.006>
- Sahoo, A., & Chandra, S. (2017). Multi-objective grey wolf optimizer for improved cervix lesion classification. *Applied Soft Computing*, 52, 64–80. <https://doi.org/10.1016/j.asoc.2016.12.022>
- Sarangi, P. P., Sahu, A., & Panda, M. (2014). Training a feed-forward neural network using artificial bee colony with back-propagation algorithm. In D. P. Mohapatra & S. Patnaik (Eds.), *Intelligent Computing, Networking, and Informatics* (Vol. 243, pp. 511–519). Springer India. https://doi.org/10.1007/978-81-322-1665-0_49
- Saremi, S., Mirjalili, S. Z., & Mirjalili, S. M. (2015). Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications*, 26(5), 1257–1263. <https://doi.org/10.1007/s00521-014-1806-7>
- Sarkar, D., & Modak, J. M. (2003). ANNSA: A hybrid artificial neural network/simulated annealing algorithm for optimal control problems. *Chemical Engineering Science*, 58(14), 3131–3142. [https://doi.org/10.1016/S0009-2509\(03\)00168-4](https://doi.org/10.1016/S0009-2509(03)00168-4)
- Sartakhti, J. S., Zangoeei, M. H., & Mozafari, K. (2012). Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA). *Computer Methods and Programs in Biomedicine*, 108(2), 570–579. <https://doi.org/10.1016/j.cmpb.2011.08.003>
- Seiffert, U. (2001). Multiple layer perceptron training using genetic algorithms. In *European Symposium on Artificial Neural Networks (ESANN)* (pp. 159 – 164).

- Shariati, M., Mafipour, M. S., Mehrabi, P., Bahadori, A., Zandi, Y., Salih, M. N. A., Nguyen, H., Dou, J., Song, X., & Poi-Ngian, S. (2019). Application of a hybrid artificial neural network-particle swarm optimization (ANN-PSO) model in behavior prediction of channel shear connectors embedded in normal and high-strength concrete. *Applied Sciences*, 9(24), 5534. <https://doi.org/10.3390/app9245534>
- Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., Yang, B., & Liu, D. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems*, 96, 61–75. <https://doi.org/10.1016/j.knosys.2016.01.002>
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 69–73). <https://doi.org/10.1109/icec.1998.699146>
- Slowik, A. (2011). Application of an adaptive differential evolution algorithm with multiple trial vectors to artificial neural network training. *IEEE Transactions on Industrial Electronics*, 58(8), 3160–3167. <https://doi.org/10.1109/tie.2010.2062474>
- Tay, D., Poh, C. L., Goh, C., & Kitney, R. I. (2014). A biological continuum based approach for efficient clinical classification. *Journal of Biomedical Informatics*, 47, 28–38. <https://doi.org/10.1016/j.jbi.2013.09.002>
- Tong, D. L., & Mintram, R. (2010). Genetic algorithm-neural network (GANN): A study of neural network activation functions and depth of genetic algorithm search applied to feature selection. *International Journal of Machine Learning and Cybernetics*, 1(1), 75–87. <https://doi.org/10.1007/s13042-010-0004-x>
- Turabieh, H. (2016). A hybrid ANN-GWO algorithm for prediction of heart disease. *American Journal of Operations Research*, 06(02), 136–146. <https://doi.org/10.4236/ajor.2016.62016>
- Uzlu, E., Kankal, M., Akpinar, A., & Dede, T. (2014). Estimates of energy consumption in Turkey using neural networks with the teaching–learning-based optimization algorithm. *Energy*, 75, 295–303. <https://doi.org/10.1016/j.energy.2014.07.078>
- Yan, H., Jiang, Y., Zheng, J., Peng, C., & Li, Q. (2006). A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications*, 30(2), 272–281. <https://doi.org/10.1016/j.eswa.2005.07.022>

- Zhang, J.-R., Zhang, J., Lok, T.-M., & Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, 185(2), 1026–1037. <https://doi.org/10.1016/j.amc.2006.07.025>
- Zhang, Y., Jin, Z., & Chen, Y. (2019). Hybridizing grey wolf optimization with neural network algorithm for global numerical optimization problems. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-019-04580-4>