Performance Analysis of Barnacles Mating Optimization Algorithm and Black Widow Optimization Algorithm on Improved Pi-Sigma Neural Network

T. Mathi Murugan[1], Dr. E. Baburaj[2]

Department of Computer Science and Engineering

Sathyabama Institute of Science and Technology, Chennai[1],

Marian Engineering College, Trivandrum, Kerala, India[2]

***E-mail:-** mnamashivayam11@gmail.com[1], alanchybabu@gmail.com[2]*

DOI: - https://doi.org/10.47531/MANTECH/ECC.2021.46

Abstract

The utilization of multilayer perception and feed forward network has limitations in the neural network, such as multi-layering and linear threshold unit for different engineering applications. Thus, higher-order neural networks were beneficial in performing non-linear mapping, which utilises the input units with the single layer for conquering the limitations of the conventional neural networks. The paper utilises a higher-order neural network known as the improved Pi-Sigma neural network coupled with the Barnacles Mating Optimization (BMO) algorithm and Black Widow Optimization (BWO) algorithm for developing an effective hybrid training algorithm for classification process with global and local searching abilities. For validating the performance of the proposed BMO-IPSNN and BWO-IPSNN algorithm, the algorithm was tested with different types of datasets obtained from UCI machine learning source, and then the algorithm was evaluated with the other existing algorithms such as PSO-PSNN, IPSNN, FFA-PSNN and PSNN. The result from the experiment concludes that the proposed algorithm gains superior performance for classification problems.

***Keywords:-** Bio-inspired algorithms, Optimization, Classification, Neural networks*

INTRODUCTION

Optimization was the method of acquiring the suitable variables that overcome the limitations in obtaining the suitable function for minimization or maximization process. The meta-heuristics algorithms were simple optimization algorithms that were inspired by nature. These meta-heuristics algorithms were mainly categorized into four groups which were human behaviour based, physics-based, swarm-based and evolutionary-based algorithms [1]. Here, the evolutionary algorithm inspires the development process taking place in nature in which the global optimum was attained by creating new offspring which inherits the behaviour of parents. The set of candidate solution was updated repeatedly till it satisfies the terminating process. Hence, during iterations or generations, the chance of attaining a better outcome nearby the optimum global increases, although it was not certain to achieve an exact estimate of the global optimum. The objective function was generally framed depending on the functions or complications needed to be resolved, which includes profits, efficiency, cost and more. For solving the optimization problem, many methods or algorithms were utilised, from hard computing to meta-heuristic tactics. The hard computing process depends on gradient-based data of the implicated functions to locate the optimum solution. Although these methods were utilised even now by various operators to solve optimization difficulties, they generally undergo local optimum setup. It was also unsuccessful for unidentified problems with computationally expensive derivation [2]. Instead, the meta-heuristic tactics turn out to be a more well-liked approach in optimization process because of their capability to escape from the local optimum. Also, it relies on simple conceptions which mimic nature, and it was used in various kinds of problems from different specialities.

Data classification was an important process in the data mining works. The classification process includes the steps such as allocating separate classes for the objects which were unknown and then recognizing the appropriate model for analysing them. The classification model depends on the training dataset, and a unique class label was assigned, which differentiates it from several other classes. Many classification models were developed from traditional neural networks, which include feed forward network, multilayer perceptron, backpropagation neural network and so on. These neural network models were widely used in various classification fields. The traditional neural networks have certain limitations such as late convergence, slow training rate and non-linear mapping capability. Thus, high order neural networks (HONN) were developed to overcome these drawbacks. These HONNs need terms with higher-order, which may affect the network complexity. To overcome this limitation, Pi-Sigma neural network was developed, which prevents the rise in a number of weight vectors with the processing units. During past years, different versions of PSNN were developed for the different classification process. Unlike other higher-order neural networks, the PSNN only need a lesser number of weights, and a very important number of weights was not required [21]. Many works were developed to improve the performance of the PSNN. In differential pulse code modulation systems, a continual PSNN were utilised as a predictor structure [22]. For resolving the problem in reference transformation, the pi-sigma neural network was trained with an online learning algorithm [23]. For resolving parity problems, chemical reaction optimization-based PSNN was used in which the population size was fixed [24]. In the proposed approach, we used an improved Pi-Sigma Neural Network, which requires two alterations. The modifications used in the proposed approach were Activation Function Extension and adding delayed units.

Currently, because of the large difficulty occurring in practical problems, the necessity for effective meta-heuristic approaches arises. The meta-heuristic approaches were very popular because of their easy implementation and high efficiency. These approaches were implemented for resolving problems, practical engineering problems and attaining the possible optimum solution for the required time [3]. The reputations of these algorithms were not restricted to a computer or other engineering fields. They were also implemented to other problems such as holiday planning and economics. The meta-heuristic algorithms were utilised in various areas of science and industries due to its capability to escape from local optimum. The very well-known meta-heuristic algorithm was the Particle Swarm Optimization (PSO) [4, 5] which mimics the behaviour of a school of fish and the swarm of birds. Here, the best location in the search space was guided by the movement of particles. The swarm-based algorithms other than particle swarm optimization includes Ant Colony Optimization (ACO) [13, 14], Satin Bower-bird Optimizer (SBO) [6], Grasshoppers Optimization Algorithm (GOA) [7], Dragonflies Algorithm (DA) [8], Grey Wolf Optimizer (GWO) [9], Firefly Algorithm (FA) [11], Artificial Bee Colony (ABC) [12] and Ant-Lion Optimizer (ALO) [10]. The other common algorithms include Genetic Algorithm [15, 16], Evolutionary Strategies [17], Genetic Programming [18], Evolutionary Programming [19] and Differential Evolution [20].

The nature-inspired algorithms were also known as evolutionary algorithms, and it was solving many engineering problems because it was the easiest and flexible algorithm. While applying the meta-heuristic algorithms to them, the structural modifications were not necessary. Thus, these algorithms were widely used for solving engineering problems these days. This paper compares two optimization algorithms that were derived from the behaviour of some birds or animals, which mimics the biological nature, such as fighting for food or mates. It helps to realize the biological behaviour and response in diverse situations in nature creates solutions to complex issues by inspiring analogical reasoning and thinking of these birds or animals. The improved pi-sigma neural network was trained with evolutionary optimization algorithms such as barnacles mating optimizer (BMO) and black widow optimizer (BWO). Then, the performance of each algorithm was tested with ten different datasets to analyse the efficiency of these algorithms. The performance was also compared with existing algorithms such as PSNN, PSO-PSNN, IPSNN and FFA-PSNN in association with convergence rate, mean square error, the best solution, worst solution, mean and standard deviation.

LITERATURE REVIEW

This section presents the background methods and state-of-the-art works on different machine learning-based neural network classification and bio-inspired algorithms [25] implemented for optimization purpose. The Black Widow Optimization (BWO) Algorithm mimics the mating behaviour of the black widow spiders. This algorithm involves the cannibalism stage in which the spiders with improper fitness were neglected from the circle, which leads to early convergence. Barnacles Mating Optimization (BMO) algorithm was one of the evolutionary algorithms which were inspired by the mating behaviour of barnacles. It paved a path for the employment of bio-inspired optimization methods in different data mining fields, including healthcare, finance, marketing, biomedicine, etc. The main advantage of these algorithms stands in its ability to handle different issues such as local optima, optima isolation, search space biasing, premature convergence and fast execution.

For the classification process, PSO-PSNN was presented in [26], in which the higher-order neural network known as PSNN was coupled with the PSO algorithm for developing an effective training algorithm for the classification process. For analysing the capability of the proposed approach, the algorithm was evaluated with different datasets, and the results were then compared with other existing algorithms such as PSNN and GA-PSNN. In this proposed model, the population section was done by random generation, and the best solutions were kept in an archive collection. The result from the experiment shows that the proposed approach attains improved performance in the classification process. The proposed algorithm has enhanced computational efficiency and classification accuracy. Also, it was noted for the given datasets that the proposed model gives better performance than the GA based algorithm. Although the PSO-PSNN shows better performance, it has drawbacks when the testing was conducted with a large number of datasets. The PSO experiences premature convergence due to rapid trailing in large datasets. The proposed PSO-PSNN cannot be effectively utilised for large datasets.

A novel FFA-PSNN was proposed [27], which have high classification accuracy. The proposed algorithm demonstrates the efficiency of the firefly algorithm in the data classification process. Also, it shows that the proposed model was reliable and better than the other proposed algorithms. The proposed algorithm was compared with other existing algorithms like particle swarm optimization algorithm, genetic algorithm and hybrid PSO-GA algorithm. Many statistical tests were conducted on the proposed algorithm, which includes the Post Hoo test, Dunnett test, Friedman test and ANOVA test. The tests were conducted on the proposed model to demonstrate that the algorithm was statistically valid, and it also gives the best results when evaluated with the other algorithms. Here, the behaviour of the firefly, which finds the better and brighter butterfly, was utilised for obtaining the best fitness in neural network function. The results from the experimental process of the proposed algorithm concluded that the proposed algorithm has a lesser error and better convergence rates when compared to the other algorithms. Barnacles mating optimization algorithm was proposed by Sulaiman et al. [1], which imitates the mating behaviour of micro-organism called as barnacles. The proposed algorithm was tested on 23 benchmark datasets and obtained reliable solutions for the complex problem. The algorithm was currently implemented for solving economic dispatch problem in power system applications. Many researches were conducted on the black widow optimization algorithm for effective optimization process [28, 29]. The mating behaviour of black widow spiders were utilised widely for creating successful solutions for optimization problems [30].

The higher-order neural networks were modified from the artificial neural networks to overcome the drawbacks in the conventional artificial neural networks such as Multi-Layer Perceptron [31]. Most of the neural networks were categorized as linear networks because they contain the combination of the linearity input neurons in HONNs. Here, the higher-order neuron combination gives excellent dynamics to the network and also efficiently performs non-linear mapping [32]. The higher-order neural networks have the benefit of having an excellent generalisation, and so it was widely focused on by many researchers. However, the computation process was very expensive and also the training process was more complex occasionally [33]. The higher-order neural networks were widely applied in many fields such as pattern recognition [34], non-linear time series [35], a non-linear classification depending on Chemical Reaction Optimization HONN [36], misclassification cost for various financial distress prediction models and financial time series prediction [37, 38]. The Pi-Sigma Neural Network was the most widely used High Order Neural Networks.

PRELIMINARIES

A. Improved Pi-Sigma Neural Network (PSNN)

Similar to the structure of a feed-forward neural network, the PSNN consists of a set of output, hidden and input layers. The PSNN uses only a lesser number of weights when compared with higher-order neural networks. But, they can indirectly include many of their abilities and strengths. The Pi-Sigma Neural Networks were effectively utilised for many difficult functions where the traditional FFNNs has complications which include polynomials function [39] and zeroing polynomials [40]. Since the PSNN has a lesser number of weights, the training process was faster in the network. Let the input $X = (X_0, X_1, \ldots, X_j \ldots X_n)^T$ be the $(n+1)$ dimensional input vectors wherex_j represents the jth element of X. The $(n+1)k$dimensional weights vectors while $W_{ij} = (W_{ij_o}, W_{ij_1}, W_{ij_2} \ldots W_{ij_n})^T, i = 1, 2 \ldots k$ were summed at the level of k summing units, where k was the consequent order of the network and Bj was taken as the bias unit. The output at the hidden level hj can be computed by Equation (11).

$$h_j = B_j + \sum w_{ji} x_i \quad (1)$$

where w_{ij} denotes the weight in input to summing unit. Here the hidden to output layer weight was taken as 1, and the output O can be calculated by using Equation (12).

$$O = f\left(\prod_{j=1}^{k} h_j\right) \quad (2)$$

where $f(.)$ was an appropriate activation function

1) Activation Function Extension

The first modification in the IPSNN involves activating the function extension. Here, the activation function was applied for entire hidden neurons. Also, the output of the network never contains the activation function. This initially modified network in the PSNN was given as Activation Function Extension Pi-Sigma (AFEPS). The equation formed by this network was given in equation (13).

$$\frac{\partial E}{\partial b_i(k)} = \left(y(k) - y_d(k)\right) \prod_{\substack{p=1 \\ p \neq i}}^{N} h_p(k) f^t\left(s_i(k)\right) \quad (3)$$

2) Addition of Delayed Units to Hidden Neurons

The second modification in the IPSNN involves adding the delayed units to the neurons which were hidden. From equation (13), every hidden neuron was added to the delayed units with a resultant weight $\propto_i (k)$ among hidden neurons and their units and $(1 - \propto_i (k))$ among the input units and their outputs. The output of the network was calculated by the sum of both unit delay output and hidden neuron output. This equation was known as the Delayed Pi-Sigma, which was shown in equation (14).

$$\frac{\partial E}{\partial \propto_i(k)} = (y(k) - y_d(k)) \prod_{\substack{p=1 \\ p \neq i}}^{N} v_p(k)(h_i(k) - v_i(k-1)) \quad (4)$$

B. Barnacles Mating Optimizer Algorithm

Barnacles were small micro-organisms that present since the times of Jurassic. They can be able to swim at birth and attach them to any material in the water to form shell at their adult age. The majority of the barnacles were hermaphroditic, which signifies that they undergo both males as well as female reproduction. One of the remarkable facts about barnacles was their long penises, in addition to several of the lengthiest in animals compared to their body, which was seven to eight times their body length to manage with varying tides and sedentary way of life. The mating group of a barnacle was comprised of all its neighbours within reach of the penis and all its possible contenders for mates. Any variation in the reach of the penis might have an essential role in evaluating the cluster size of mating and local competition on mating. The mating process of barnacles was inspired for developing an optimization algorithm defined as follows.

- ***Initialization***

The optimization problem being in need of a solution was assumed to be barnacles, where the population vector was defined as given in Equation (1).

$$Y = \begin{bmatrix} y_1^1 & \cdots & y_1^N \\ \vdots & \ddots & \vdots \\ y_n^1 & \cdots & y_n^N \end{bmatrix} \quad (5)$$

where N represents the total control variables and n depicts the total population or total barnacles. The control variables described in Equation (1) were directed to upper and lower parameters of the problem to be resolved as termed in Equations (2) and (3).

$$up = [up_1, \ldots\ldots, up_j] \quad (6)$$

$$lp = [lp_1, \ldots\ldots lp_j] \quad (7)$$

where up and lp denotes the upper and lower limits of the jth variable. The estimation of the vector Y was completed at the beginning, and the sorting procedure was undergone for identifying the best solution at the top of the vector Y.

- ***Selection***

The choice of two barnacle's selection for mating was dependent on the penis length pl. Moreover, the selection imitates some behavior of the barnacles. The mathematical form for random selection of barnacles wasgiven in Equations (4) and (5).

$$barn\_n = randperm(n) \quad (8)$$

$$barn\_o = randperm(n) \quad (9)$$

where $barn\_n$ and $barn\_o$ were the barnacles to be mated called parent barnacles and ▱ exhibits the number of barnacles.

- ***Reproduction***

Reproduction of offspring was based on the inheritance features of parent barnacles expressed in terms of Hardy–Weinberg principle. Equation (6) shows how to yield new offspring variables from parent barnacles.

$$y_j^{N\_new} = ay_{barn\_n}^{N} + by_{barn\_o}^{N} \quad (10)$$

where $u$was normal distributed pseudorandom value in the range [0,1], ▱ = 1 − [0,1], $b = 1 - a$, $y_{barn\_n}^{N}$ and $y_{barn_o}^{N}$were the male and female parent barnacles respectively, which has been nominated through Equations (10) and (11). It can be noted that u and v denote the rate of features of male and female barnacles respectively implanted in the formation of new offsprings. The offspring receives the behavior of the male and female based on the random probability within *0* to *1*. For instance, let us consider $a$was *0.7* (random selection), it states that *70%* of male features and *30%* of female features were implanted in the formation of new offsprings. If the barnacles selected for mating exceeds initially set yl value, then BMO uses sperm cast technique as given in Equation (7).

$$y_l^{n\_new} = rand() \times y_{barn\_o}^{n} \quad (11)$$

where$rand()$was the random number between [*0,1*]. According to Equation (7), the new offspring is produced with female barnacles since it gets sperm from water that has been released by other barnacles.

- ***Sorting***

On every iteration, the best solution was updated and stored at the top of the vector Y. The expansion of the matrix beyond the barnacles' size can be controlled by merging the new offspring with parent barnacles. Thereafter, sorting was performed to pick half of the topmost solutions that fit the initial barnacle size. The worst results in the bottom are supposed to be deceased and removed. The flowchart for the BMO algorithm was given below.

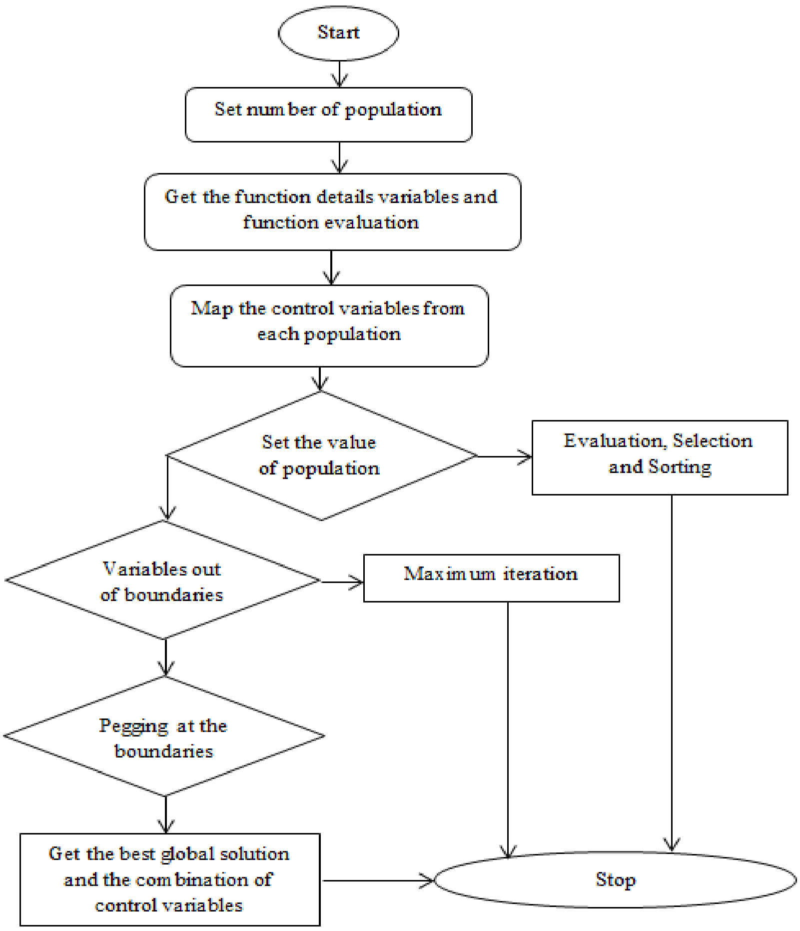

Figure:-1 Flowchart for barnacles mating optimization algorithm

Algorithm- 1: Pseudo Code for BMO-IPSNN

INPUT: Dataset with target vector 'tv', initial population of weight set 'PW', Bias 'Bi'.

OUTPUT: IPSNN with optimized weight set 'W'.

Step-1: Generate initial population of barnacles Xi

Step-2: Calculate the fitness of each barnacles where f(Wi) the objective junction to compute the fitness of weight set Wi.

Step-3: Sorting to assign the best solution at T where T = best solution.

Step-4: ***while****(i<Maximum iterations)*

Set the value of pl

Selection using following condition, (i.e. d and m means Dad and Mum respectively)

barn_n=randperm(n)

barn_o=randperm(n)

if selection of Dad and Mum = *pl*

for each variable

Off spring generation using,

$y_j^{n\_new} = ay_{barn\_n}^{n} + by_{barn\_o}^{n}\ for\ k < pl$

end for

else if selection of dad and mum >*pl*

for each variable

Off spring generation using,

$y_j^{n\_new} = rand0 \times y_{barn\_o}^{n} for\ k > pl$

end for

end if

Bring the current barnacle back if it goes outside the boundaries

Calculate the fitness of each barnacles.

Sorting and update if there is a better solution

i=i+1

end while

Step-5: find the barnacle fitness and best one.

Step-6: return best solution (T) or go to the step 3

A. Black widow optimizer algorithm

Spiders were the air-breathing insects which have eight-legs and venomous teeth. A subfamily of spiders known as Latrodectus includes a popular species called black widow spiders or red "hour-glass" due to a sign in their abdomen. This kind of spiders was generally nightly, and the female spider stands out of vision at daytime and spins her web during night. Each time the female black widow wants to mate, it remarks some spots of her net with the pheromone, a chemical secreted hormone to invite the males. The first male getting inside the web reduces web less attraction of female to competitors by reduction of web. The female widow eats the male spiders at the time or after mating and transfer the egg to her egg cell. After egg hatching, the new born involves in cannibalism of its sibling. Nevertheless, they live on their mother's web for a small time-period in which they might even eat the mother. This is a cyclic process which presents fittest survival and strong individuals.

- ***Initialization***

The potential solution needed for each complex problem was termed as a black widow spider and the population was considered in the form an array. For a K_u dimension optimization problem, the potential solutions of the widow were taken as an array of the form: $1 \times K_u$. It was generally defined as given in Equation (8).

$$BW = [q_1, q_2, \dots \dots . q_{k_u}] \quad (12)$$

where $(q_1, q_2, \dots \dots . q_{k_u})$were variables and each variable possess a floating point value. The fitness of each widow was evaluated by applying fitness factor f at a widow as given in Equation (9).

$$f(BW) = f(q_1, q_2, \dots \dots . q_{k_u}) \quad (13)$$

The optimization problem was initialized with a widow matrix of the dimension $K_{pop} \times K_u$ with an initial set of black widow spiders. Then, the parent-pairs are randomly selected to perform procreation through mating where the male black widow was swollen by the female on or after mating.

- ***Procreation***

Though the parent-pairs were independent of themselves, they start to mate for the reproduction of new generation. Subsequently but also in nature, every pair mate in its web, individually from others. About 1000 eggs were formed in every mating; however, only some of them will survive based on their fitness. In real-world optimization challenges, the reproduction stage was accomplished by the formation of an alpha array filled with random numbers. Therefore, the offspring generation is achieved by following Equation (10).

$$r_1 = \propto \times s_1 + (1 - \propto) \times s_2$$
$$r_2 = \propto \times s_2 + (1 - \propto) \times s_1 \quad (14)$$

This process was continued for $K_u/2$ times whereas random filled alpha arrays could not be duplicated. At last, the new generation and female widow were placed in an array and then sorted on the basis of their fitness value. Some of the fittest individuals were then selected as new generation based on their cannibalism rating. These procedures were performed with all the pairs.

- ***Cannibalism***

Cannibalism was the practice of eating another individual of the similar kind. The black widows encompass in three methods of cannibalism. (i) Sexual cannibalism: the female widow consumes the male widow spiders on or after mating. The female as well as male spiders are differentiated by their fitness score. (ii) Sibling cannibalism: after reproduction, the strongest baby widow spiders eat their weaker siblings. To achieve this, cannibalism rating was used which selects fittest individuals and eliminates the weakest ones. (iii) Parent cannibalism: baby spiders eat the female mother widow. The fitness value was again calculated for baby spiders to distinguish strong as well as weak survivors.

- ***Mutation***

A mutation rate was set to select a range of random individuals from fittest population for mutation. Mutation was done in such a way that each of the selected solutions arbitrarily exchanges two features of the array.

A. Convergence

Black widow optimization algorithm follows three different stopping conditions namely (i) a predefined count of iterations. (ii) Fitness value of best widows remains same after several iterations. (iii) Attaining the appropriate accuracy rate. The flowchart for black widow optimization algorithm was given below.

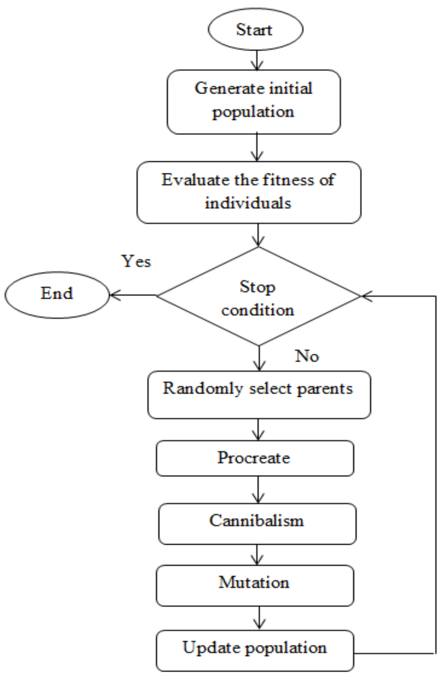

Fig. 2 Flowchart for black widow optimization algorithm

Algorithm- 2: Pseudo Code for BWO-IPSNN

INPUT: Dataset with target vector ‘tv’, initial population of weight set ‘PW’, Bias ‘Bi’.

OUTPUT: IPSNN with optimized weight set ‘W’.

Step-1: Initial population of black widow spiders

Each population is a D-dimensional array of chromosomes for D-Dimensional problem.

Step-2: Loop until the terminal condition

Calculate the number of reproduction “*nr*” based on procreating rate.

Step-3: Select best “nr” solution in pop and save them as pop1.

Step-4: Procreating and cannibalism

for*i=1 to nr do*

select two solution as parents from pop1 randomly,

generate D children by using,

$r_1 = \propto \times s_1 + (1-\propto)\times s_2$

$r_2 = \propto \times s_2 + (1-\propto)\times s_1$

destroy father and destroy some children based on the cannibalism rate

new achieved solutions and save the remain solutions into pop2.

end for

Step-5: Calculate number of mutation children “nm” based on mutation rate

for*i=1 to nm* do

select a solution from pop1

mutate randomly one chrome of the solution and generate a new solution.

save the new solution into pop3.

end for

Step-6: Update $pop = pop2 + pop3$ and Return the best solution from *pop*.

PROPOSED METHOD

The proposed BWO-IPSNN algorithm and BMO-IPSNN algorithm utilises the BP-GDL algorithm developed by Rumelhart et al. [41]. The approach begins with the initialization process in which the weights were initialised to a small randomly produced value. At starting iteration, the initial population was taken as the local best of the algorithm. The fitness of all individuals was calculated by the Fitness from training algorithm by employing the root mean square error (RMSE) given in Equation (15).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} O_i - \hat{O}_i}{n}} \quad (15)$$

The network was trained with the product of errors and then the overall predictable error function E was determined which was given in Equation (16).

$$E_j(t) = d_j(t) - O_j(t) \quad (16)$$

where $d_j(t)$ signifies the desired final output at time $(t-1)$. The output of each $O_j(t)$ was determined for each time $(t-1)$. During the fitness calculation, the network was trained with the BP-GDL. The weight change and the updating of weight value were calculated using Equation (17) and (18) respectively.

$$\Delta w_j = n\left(\prod_{j\neq 1}^{m} h_{ji}\right)x_k \quad (17)$$

where h_{ji} was the summing layer output and n was the learning rate.

$$w_i = w_i + \Delta w_i \quad (18)$$

For speeding up the error convergence, the term momentum $(\propto)$ was combined and the value of weight connection was calculated as shown in Equation (19).

$$w_i = w_i + \propto \Delta w_i \quad (19)$$

EXPERIMENTAL SETUP

The proposed approach was designed for properly classifying the data which have many feature sets and numerous class labels. The datasets were obtained from the University of California at Irvine (UCI) machine learning source [42]. A comparative evaluation was carried out between BWO-IPSNN and BMO-IPSNN with other existing algorithms such as PSNN, PSO-PSNN, FFA-PSNN and IPSNN. The techniques were implemented in MATLAB2018a software running on a Windows8.1 operating system. The datasets were set up by dividing them into 5-folds, among which 4 folds were employed for training process and 1 fold was employed for testing process.

Algorithm- 3: Fitness from Training Procedure

Function f= fitness from training (xi, W, tv, Bi)

for*i= 1 to n*, where n is the dataset length

Compute the output at the hidden layer at time t.

$$h_i(t) = \sum_{j=1}^{m} w_{ij}(t)x_j(t) + b_i(t)$$

Compute the output of the network at time t.

$$y(t) = f(pradh(t))$$

Activation function extended by using,

$$y(t) = \prod_{i=1}^{N} h_i(t)$$

Adding delayed units to hidden network by using,

$$y(t) = \prod_{i=1}^{N} h_i(t) + \prod_{i=1}^{N} v_i(t)$$

Calculate the error term at time by using,

$E_i(t) = f_i(t) - O_i(t)$ and find fitness= 1/ RMSE.

end for

Compute root mean square error (RMSE) from target value and output by using,

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Compute the weight changes by using BP-GDL algorithm,

$$\Delta W_{ij} = n\left(\prod_{j\neq 1}^{m} h_{ij}\right)x_t$$

Update weight and weight value can be calculated after adding the momentum term.

$$W_i = W_i + \Delta W_{ij}$$

if is a training error or maximum no. of epochs are satisfied, then stop.

else repeat the step 4.

end.

A. Dataset information

Ten different kinds of datasets were utilised for testing the proposed algorithms. The details regarding the datasets utilised for the analytical process was given as follows.

1) Balance Dataset

The balance dataset was utilised for modelling the psychological results from the experiment. The scale in the balance can change from right or left or may also be balanced. The dataset depends on the measurements from balance scale which has 2 classes, 4 attributes and 625 patterns. It does not have any missing value.

2) Ecoli Dataset

The Ecoli dataset was utilised for predicting the localization spot of proteins by utilising certain features regarding the cell including lipoprotein, cytoplasm and so on. It includes 3 classes, 9 attributes and 336 patterns. It does not have any missing value.

3) Hayes-Roth Dataset

The Hayes-Roth dataset includes 5 numerically assessed characteristics such as age, hobby, name and so on. The dataset comprises of 2 classes, 5 attributes and 161 patterns. It does not have any missing value.

4) Heart Dataset

The heart dataset was associated with human heart and its characteristics were resting blood pressure, chest pain type, age, sex and so on. The dataset consists of 2 classes, 14 attributes and 304 patterns. It does not have any missing value and its type was multivariate.

5) Hepatitis Dataset

The Hepatitis dataset was utilised from the details about the patients having Hepatitis. The dataset consists of 2 classes, 20 attributes and 143 patterns.

6) Parkinson Dataset

The Parkinson dataset consists of 2 classes, 23 attributes and 196 patterns.

7) Pima Dataset

The Pima dataset has a compilation of Pima Indian Heritage females with an age of 21 and above. The dataset comprises of 4 classes, 9 attributes and 768 patterns. It does not have any missing value.

8) Sonar Dataset

The Sonar dataset was utilised for predicting the signal of the recipients in various angles. The dataset consists of 2 classes, 60 attributes and 208 patterns. It does not have any missing value.

9) Vehicle Dataset

The Vehicle dataset was utilised for classifying the given profile as one of four kinds of vehicle. It uses a set of characteristics which were obtained from the profile. The dataset consist of 4 classes, 18 attributes and 846 patterns. It does not have any missing value.

10) WBC (D) Dataset

The dataset consists of 3 classes, 30 attributes and 569 patterns.

The details about the dataset used in the experimental process were given in table 1. The dataset name was given in the first column. Then the other respective columns give the details about the number of patterns, attributes and classes.

TABLE 1: DETAILS ABOUT DATASET

Dataset	Number of Pattern	Number of Attributes	Number of Classes
Balance	625	04	02
Ecoli	336	09	03
Hayes-Roth	161	05	02
Heart	304	14	02
Hepatitis	143	20	02
Parkinson	196	23	02
Pima	768	09	04
Sonar	208	60	02
Vehicle	846	18	04
WBC(D)	569	30	03

RESULTS AND DISCUSSION

The proposed BMO-IPSNN and BWO-IPSNN method was employed and experimented for ten datasets using the 5-fold cross validation process. For each single fold ten numbers of individual runs were performed. For computing the performance of the proposed algorithm, cross validation [43] technique was utilised. For cross validation process, the dataset was divided into two sets known as training and testing dataset. It was utilised for training and testing the proposed algorithm effectively. Here, the balance dataset was cross validated by employing the 5-fold cross validation and the other datasets were also organized in the same way. Table 2 gives the five-fold cross validated balance dataset.

The accuracy of classification was trained and tested for all algorithms and it was then validated for all the ten datasets. The results from the proposed algorithm were then compared with other existing algorithms such as PSNN, PSO-PSNN and IPSNN, FFA-PSNN. Table 3 gives the accuracy of classification for the 5-folded balance datasets.

A. Statistical analysis

The statistical analysis was performed for analysing the improvement in the performance of the proposed algorithms than the other existing algorithms. It analyses classification problems such as the error rates, classification accuracy and so on. Many tests were conducted for this statistical analysis which includes Post Hoc test, Dunnett test, Tukey test, Friedman test and ANOVA test. In this paper, ANOVA test was conducted to analyse the statistical correctness of the proposed algorithms over other existing algorithms.

The ANOVA test examines the null hypothesis for evaluating the variability in the performance of the algorithms. The sum variability was split into variability among the datasets and the error variability by ANOVA and the variability by the classifiers. The test was conducted on one way ANOVA with linear polynomial contrast, 0.05 significance level and 95% confidence level. The ANOVA test results were given in Table 4 and 5.

The proposed algorithms such as BMO-IPSNN, BWO-IPSNN, were evaluated with other existing algorithms like IPSNN, FFA-PSNN, PSO-PSNN and PSNN Algorithm for comparing the runtime and classification accuracy.

The optimal solutions obtained for the two optimization algorithms were evaluated with different metrics like best solution, worst solution, mean and standard deviation as shown in table 6. It can be seen that BMO-IPSNN and BWO-IPSNN has obtained good results for almost all datasets. The convergence curve comparison of different techniques for all the input datasets is shown in Figures 1-10.

Table 2: Five- Fold Cross Validation Balance Dataset

Dataset	Data Files	Number of Pattern	Task	Number of Pattern in Class-1	Number of Pattern in Class-2
Balance	Balance-5-1trn.data	525	Training	237	263
	Balance-5-1tst.data	100	Testing	48	52
	Balance-5-2trn.data	525	Training	237	263
	Balance-5-2tst.data	100	Testing	48	52
	Balance-5-3trn.data	525	Training	237	263
	Balance-5-3tst.data	100	Testing	48	52
	Balance-5-4trn.data	525	Training	237	263
	Balance-5-4tst.data	100	Testing	48	52
	Balance-5-5trn.data	525	Training	237	263
	Balance-5-5tst.data	100	Testing	48	52

Table 3: Classification Accuracy Performance on Balance Dataset

Dataset		Task	Accuracy of Classification in %					
Data	*5-folds*		*BMO-IPSNN*	*BWO-IPSNN*	*IPSNN*	*FFA-PSNN*	*PSO-PSNN*	*PSNN*
Balance	Balance-5-1trn.data	Training	1	99.000	98.007	97.009	96.002	90.358
	Balance-5-1tst.data	Testing	99.800	98.390	97.500	96.896	95.986	90.196
	Balance-5-2trn.data	Training	99.831	99.005	97.983	97.015	95.991	91.109
	Balance-5-2tst.data	Testing	99.452	98.915	97.530	96.665	95.032	89.205
	Balance-5-3trn.data	Training	1	98.991	98.032	96.991	96.026	90.000
	Balance-5-3tst.data	Testing	99.520	98.231	97.821	96.852	96.155	89.500
	Balance-5-4trn.data	Training	99.517	98.980	98.012	97.035	95.289	91.258
	Balance-5-4tst.data	Testing	99.231	98.607	97.955	97.000	95.302	89.864
	Balance-5-5trn.data	Training	99.877	99.110	97.998	97.000	96.126	91.119
	Balance-5-5tst.data	Testing	99.038	99.002	97.639	96.980	95.122	90.201

Table 4: Anova Test with 95% Confidence Interval

	N	Mean	Std. Dev	95% confidence interval for Mean		Minimum	Maximum
				Lower Bound	*Upper Bound*		
IPSNN-BMO	10	97.6149	1.6750	93.6254	97.7956	93.86	97.90
IPSNN- BWO	10	85.9679	1.9933	90.3569	96.2569	92.67	97.02
IPSNN	10	80.8616	5.0966	89.4652	94.5697	90.21	96.56
PSNN-FFA	10	80.1120	5.6671	88.5591	93.2586	89.59	95.00
PSNN- PSO	10	75.1949	5.7832	83.2659	93.1356	88.23	94.28
PSNN	10	62.7469	6.1952	81.1896	91.9001	87.56	93.33
Total	10	80.4163	4.4017	87.7436	94.4860	90.35	95.68

Table 5: Anova Results with Sum of Squares and Mean Square

ANOVA Table					
Source	*Sum of Squares*	*df*	*Mean Square*	*F*	*Prob>F*
Groups	43521.9	5	8704.37	362.26	0.0153e-29
Error	865	36	24.03		
Total	44389.9	41			

Table 6: Performance of all Dataset

Datasets		BMO-IPSNN	BWO-IPSNN	IPSNN	FFA-PSNN	PSO-PSNN	PSNN
Balance	Best	3.8815e-118	2.2525e-14	1.5099e-14	2.8696e-09	2.3594e-06	6.3384e-06
	Worst	1.6461e+04	1.14108e+03	18.0691	7.9262e+03	16.3482	2.7696e+03
	Avg	47.8728	3.1428	0.2048	76.8176	0.0345	3.6245
	Std	681.7605	63.1574	1.2699	439.3669	0.6205	89.1439
Ecoli	Best	1.3228e-39	1.0085e-20	8.5274e-15	1.9647e-06	5.9856	0.0013
	Worst	1.4602	19.5595	3.9478	3.5703	4.3706e+07	44.5172
	Avg	0.0097	0.0284	0.0155	0.1924	7.8010e+04	0.0873
	Std	0.0853	0.6599	0.1685	0.4743	1.5096e+06	1.6510
Hayes-Roth	Best	6.2982e-40	4.4409e-15	6.8603e-11	1.7018e-10	0.1698	4.4520
	Worst	0.7081	20.2395	646.5882	5.8220	90.5799	5.9262e+06
	Avg	0.0085	0.1871	1.1563	0.0196	1.4407	7.1138e+03
	Std	0.0423	1.5220	22.6863	0.2435	5.4022	1.8864e+05
Heart	Best	5.9812e-50	2.7073e-15	4.4409e-15	1.9010e-10	1.8079e-05	6.0642e-04
	Worst	1.4965	3.3191	20.2066	5.9602	3.4669	2.6498
	Avg	0.0051	0.0218	0.1812	0.0240	0.2050	0.0056
	Std	0.0335	0.1991	1.4992	0.2732	0.5113	0.0899
Hepatitis	Best	4.9878e-21	9.9703e-06	0.0150	0.0271	0.0443	0.1576
	Worst	17.8979	1.0864e+03	103.4464	12.8490	56.2312	50.8414
	Avg	0.0223	1.7494	0.3856	0.1667	0.1986	1.3338
	Std	0.5746	37.0658	4.8636	0.6324	2.1706	4.0356
Parkinson	Best	1.2660e-09	1.3498e-32	1.8087e-08	3.1553e-07	0.3055	6.2524
	Worst	9.3779e+03	1.1988e+07	6.5837e+03	6.8396e+03	57.9928	6.4656e+07
	Avg	85.5225	1.2057e+04	10.9693	8.1757	0.5540	1.1658e+05
	Std	496.4893	3.7909e+05	223.7171	218.2365	2.1558	2.4824e+06
Pima	Best	9.4830e-13	1.1973e-10	9.9455e-07	1.4875e-06	8.9187e-04	0.3222
	Worst	42.8906	6.5100	1.4631e+04	6.9225e+03	0.0907	53.1850
	Avg	0.4258	0.0260	33.6139	12.3780	0.0040	1.1453
	Std	2.2540	0.2922	562.5606	242.8617	0.0078	2.8793
Sonar	Best	4.4409e-15	8.0585e-21	3.8491e-13	6.9455e-04	0.0021	0.0197
	Worst	20.2536	11.8311	4.3135e+03	0.0942	1.6127e+03	11.2791
	Avg	0.1701	0.0195	8.3290	0.0012	2.6959	0.0775
	Std	1.4323	0.4081	152.8529	0.0053	55.6234	0.5253
Vehicle	Best	2.4042e-31	6.4461e-24	3.6331e-07	1.7185e-05	6.8350e-04	0.9880
	Worst	3.3724e+05	0.6211	7.8623e+06	5.2427e+03	6.1072	230.7765
	Avg	674.5799	0.0010	8.5419e+03	6.6947	0.0171	1.8752
	Std	1.5074e+04	0.0215	2.4945e+05	169.8584	0.2585	12.6412
WBC(D)	Best	5.6082e-45	8.4009e-32	1.3358e-10	3.5366e-10	4.1693e-08	4.9650e-07
	Worst	0.0985	32.6755	7.7526	7.3507e+06	2.4385e+07	9.5854e+07
	Avg	1.5506e-04	0.1701	0.0241	3.0523e+04	2.5961e+04	1.3405e+05
	Std	0.0033	1.1221	0.3149	3.4559e+05	7.7265e+05	3.1594e+06

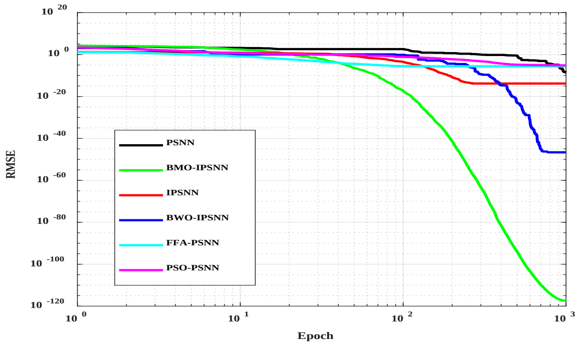

Fig. 3: Balance Dataset

Figure 1 shows the performance of the different comparative algorithms for balance dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs better next to the BMO-IPSNN.

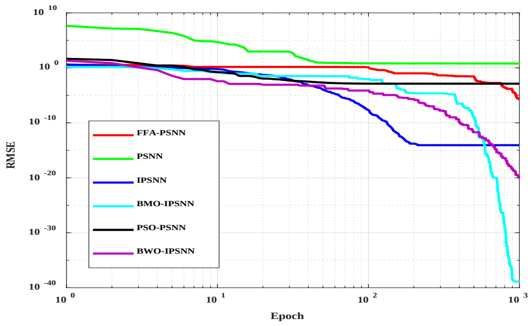

Fig. 4: Ecoli Dataset

Figure 2 shows the performance of the different comparative algorithms for ecoli dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs slightly better than the other comparative algorithms.

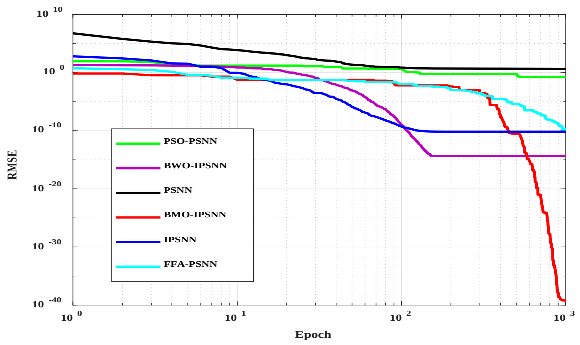

Fig. 5: Hayes-Roth Dataset

Figure 3 shows the performance of the different comparative algorithms for hayes-roth dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs slightly better than the other comparative algorithms.

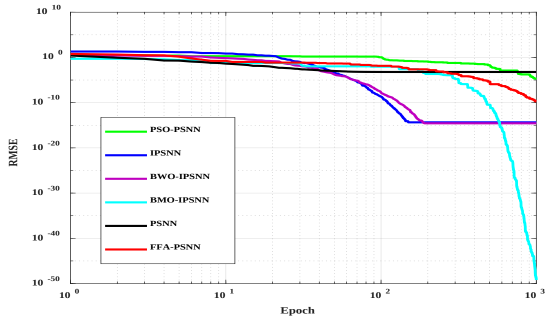

Fig. 6: Heart Dataset

Figure 4 shows the performance of the different comparative algorithms for heart dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs better next to the BMO-IPSNN. Here, the other comparative algorithm IPSNN performs almost equal to the proposed algorithm BWO-IPSNN.

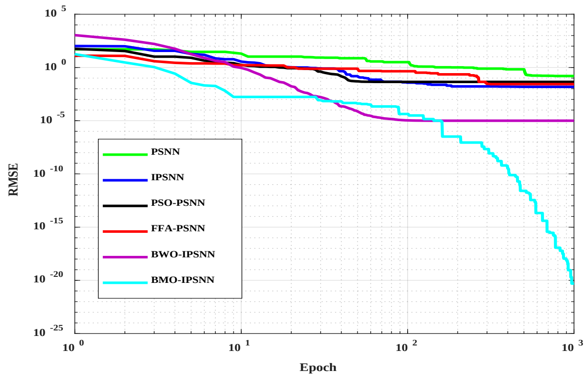

Fig.7: Hepatitis Dataset

Figure 5 shows the performance of the different comparative algorithms for hepatitis dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs slightly better than the other comparative algorithms.

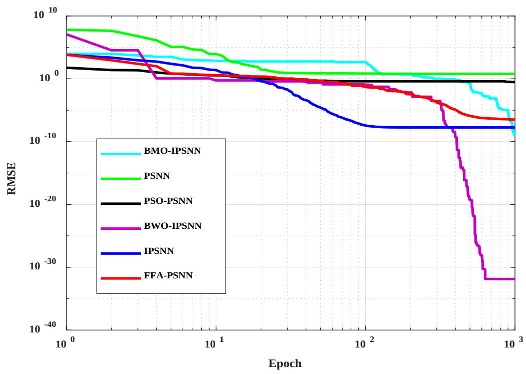

Fig. 8: Parkinson Dataset

Figure 6 shows the performance of the different comparative algorithms for parkinson dataset. From the figure it was observed that the proposed approach BWO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs slightly better than the other comparative algorithms.

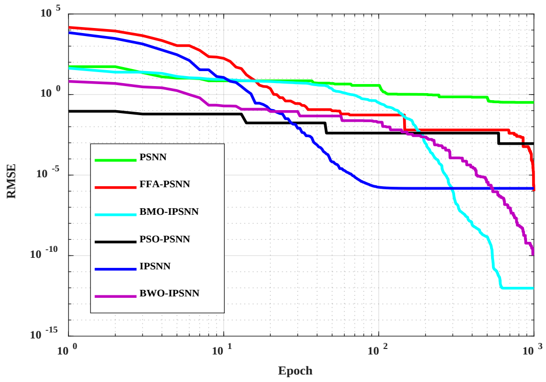

Fig. 9: Pima Dataset

Figure 7 shows the performance of the different comparative algorithms for Pima dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs better next to the BMO-IPSNN by outperforming other comparative algorithms.

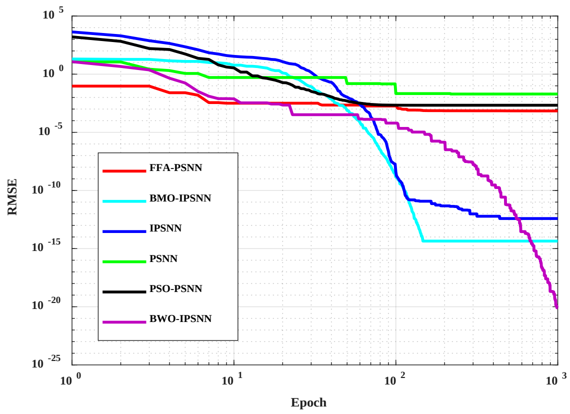

Fig. 10: Sonar Dataset

Figure 8 shows the performance of the different comparative algorithms for sonar dataset. From the figure it was observed that the proposed approach BWO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs better next to the BMO-IPSNN outperforming other comparative algorithms.

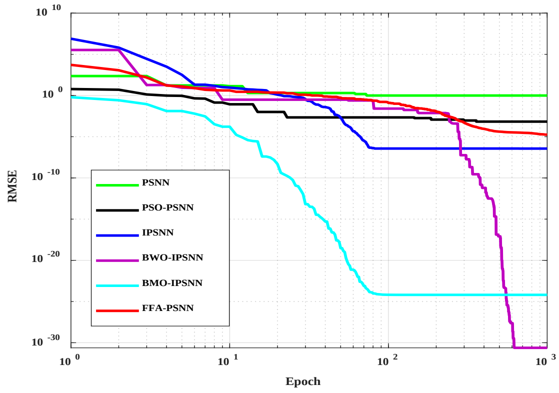

Fig. 11: Vehicle Dataset

Figure 9 shows the performance of the different comparative algorithms for vehicle dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs better next to the BMO-IPSNN by outperforming other comparative algorithms.

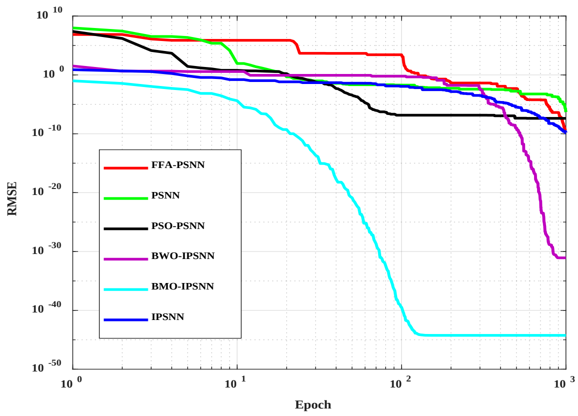

Fig. 12: WBC (D) Dataset

Figure 10 shows the performance of the different comparative algorithms for WBC (D) dataset. From the figure it was observed that the proposed approach BMO-IPSNN outperforms than all the other comparative algorithms. Also, the other proposed algorithm BWO-IPSNN also performs better next to the BMO-IPSNN by outperforming other comparative algorithms.

BMO-IPSNN

Barnacles mating optimizer was an efficient algorithm in training datasets. It has gained best classification rates for all the datasets. It also had fast runtimes and convergence curve than existing optimization algorithms. However, the runtime is comparatively higher for balance and heart dataset than other proposed algorithms. It has obtained best results in convergence which maintains good balancing between exploration and exploitation. In all datasets, the convergence was very fast and thus, it can be the best choice for data mining tasks.

BWO-IPSNN

The classification of BWO was attractive due to fast runtime and classification rates. The results were competitive for all datasets and all techniques. It too showed comparatively larger runtime only for ecoli dataset. The convergence was achieved faster with minimum number of iterations. Thus, it can be a good competitor for BMO-IPSNN in data mining field.

CONCLUSION

In this paper, a performance analysis was performed on Barnacles Mating Optimization (BMO) algorithm and Black Widow Optimization (BWO) algorithm which were coupled with higher order neural network known as improved Pi-Sigma neural network for developing effective classification algorithm with global and local searching abilities. The main aim of this paper was to validate the efficiency of the proposed algorithms in data classification and to demonstrate that the proposed algorithms were reliable and better when compared with the other algorithms. Depending on the experimental result attained from the proposed approaches, it was concluded that proposed algorithms were able to be effectively utilised in all types of classification process with good convergence and low error rates, while the other algorithms were capable for producing only the suboptimal solutions.

REFERENCES

1. M.H. Sulaiman, Z.Mustaffa, andO. Aliman,“An Application of Barnacles Mating Optimizer for Solving Economic Dispatch Problems”,In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) , pp. 835-839, April 2019.
2. S. Saremi, S. Mirjalili and A. Lewis,“Grasshopper optimisation algorithm: Theoryand application”, Adv. Eng. Softw. vol. 105, pp. 30–47, 2017.
3. V. Hayyolalam and A.A. Pourhaji Kazem, “Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems”, vol. 87, pp. 1-28, 2020.
4. J. Kennedy and R. Eberhart, “Particle swarm optimization. In: Neural Networks”, Proc. IEEE International Conference on, vol. 4,pp. 1942–1948, 1995.
5. J. Kennedy and R.C. Eberhart, “A discrete binary version of the particle swarm algorithm”,IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 5, pp. 4104–410, 1997.
6. S.H. SamarehMoosavi and V. KhatibiBardsiri, “Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation”,Eng. Appl. Artif. Intell, vol. 60, pp. 1–15, 2017.
7. S. Saremi, S. Mirjaliliand A. Lewis, “Grasshopper optimisation algorithm: Theory and application” Adv. Eng. Softw, vol. 105, pp. 30–47, 2017.
8. S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems”,NeuralComput. Appl, vol. 27, pp. 1053–1073, 2016a.
9. S. Mirjalili, S.M. Mirjalili, and A. Lewis,“Grey wolf optimizer”, Adv. Eng. Softw, vol. 69, pp. 46–61, 2014.
10. S. Mirjalili, “The ant lion optimizer”, Adv. Eng. Softw, vol. 83, pp. 80–98, 2015a.
11. X.S. Yang, “Firefly algorithm, Lévy Flights and global optimization” In: M. Bramer, R. Ellis, M. Petridis,(Eds.), Research and Development in Intelligent Systems XXVI, pp. 209–218, 2010.
12. D. Karaboga and B.Akay, “A comparative study of Artificial Bee Colony algorithm”, Appl. Math. Comput,vol. 214, pp. 108–132, 2009.
13. A. Colorni, M. Dorigo andV. Maniezzo,“Distributed optimization by ant colonies”, In: European Conference on Artificial Life, pp. 134–142, 1999.
14. M. Dorigo, M. Birattari and T. Stutzle, “Ant colony optimization”, IEEE Comput. Intell. Mag,vol. 1, pp. 28–39, 2006.
15. D.E. Goldberg and J.H. Holland, “Genetic algorithms and machine learning”, Mach. Learn, vol. 3, pp. 95–99, 1988.
16. J.H. Holland, “Genetic algorithms”, Sci. Am,vol. 267, pp. 66–73, 1992.
17. M. Amoretti, “Evolutionary strategies for ultra-large-scale autonomic systems”, Inf. Sci. (Ny) vol. 274, pp. 1–16, 2014.
18. J.R. Koza,“Genetically breeding populations of computer programs to solve problems in artificial intelligence”, In: Proc. of the 2nd International IEEE Conference on Tools for Artificial Intelligence,pp. 819–827, 1990.
19. Yao, Y. Liu andG. Lin, “Evolutionary programming made faster”, IEEE Trans. Evol. Comput, vol. 3, pp. 82–102, 1999.
20. R. Stornand K. Price,“Differential evolution–A simple and efficient heuristic for global

optimization over continuous spaces", J. Global Optim, vol. 11, pp. 341–359, 1997.

21. J. Ghosh and Y. Shin,"Efficient higher-order neural networks for classification and function approximation," International Journal of Neural Systems, vol. 3,no. 4, pp. 323-350, 1992.
22. A.J. Hussain andP.Liatsis ,"Recurrent Pi–Sigma networks for DPCM image coding,'' Neurocomputing, vol.55, pp. 363–82, 2002.
23. C.Weber, S.Wermter,"A self-organizing map of Sigma-Pi units,'' Neurocomputing, vol. 70, pp. 2552-2560, 2007.
24. K.K. Sahu, S. Panigrahi and H.S. Behera,"A novel chemical reaction optimization algorithm for higher order neural network training,'' Journal of Information Technology Theory and Application, vol. 53, pp. 402-409, 2013.
25. S.S. Gill and R. Buyya, "Bio-inspired algorithms for big data analytics: a survey, taxonomy, and open challenges. In Big Data Analytics for Intelligent Healthcare Management", pp. 1-17., inPress. 2019.
26. J. Nayak, et al.: "Particle swarm optimization based higher order neural network for classification",Computational Intelligence in Data Mining, Springer, India, vol. 1, pp. 401–414, 2015.
27. J. Nayak, B. Naik and H.S. Behera, "A novel nature inspired firefly algorithm with higher order neural network: performance analysis",In: Engineering Science and Technology, an International Journal ,2015.
28. D.R. Nelsen, W. Kelln andW.K. Hayes, "Poke but don't pinch: risk assessment and venom metering in the western black widow spider", Latrodectus hesperus. Animal Behaviour, vol. 89, pp. 107-114, 2014.
29. J.C. Johnson, J. Urcuyo, C. Moen and D. R. Stevens, "Urban heat island conditions experienced by the Western black widow spider (Latrodectus hesperus): Extreme heat slows development but results in behavioral accommodations", PloS one, 14(9), 2019.
30. [K.S. Sarath, and S.Sekar, "Black Widow Optimization Algorithm: Optimal Designing and modelling and of LLC Resonant Converter", 2020.
31. H. C. Tsai, "Modeling concrete strength with higher-order neural networks," Neural Computing and Applications vol. 27, pp. 2465–2473, 2016.
32. El-Nabarawy, A.M. Abdelbar and D.C. Wunsch,"Levenberg-Marquardt and Conjugate Gradient methods applied to a high-order neural network,'' International Joint Conference on Neural Networks (IJCNN), pp. 1-7, 2013.
33. M. A. Ashraf and M. S.Khalid,"Ant Colony Optimization Applied to the Regularization Process of a High Order Neural Network,'' IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2016.
34. T. Kaita, S. Tomita and J. Yamanaka, "On a higher order neural network for distortion invariant pattern recognition,'' Pattern Recognit Letters, vol. 23, pp. 977–984, 2002.
35. H. Tawfik and P. Liatsis," Prediction of non-linear time series using higher order neural networks,'' In Proceeding IWSSIP1997 conference, Poznan, Poland, 1997.
36. J.Nayak, B.Naik andH.S.Behera," A novel chemical reaction optimization based higher order neural network (CRO-HONN) for nonlinear classification,'' Ain Shams Engineering Journal, vol. 6, No. 3, pp.1069–1091, 2015.
37. R.Ghazali and D.al-Jumeily,"Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction,'' In Artificial Higher Order Neural Networks for Economics and Business, pp. 271-293, 2009.
38. J.Fulcher, M.Zhang and S.Xu,"The Application of Higher-Order Neural Networks to Financial Time Series,'' In J. Kamruzzaman (Ed),Artificial Neural Networks in Finance, Health and manufacturing : Potential and Challenges, pp. 80-108, 2006.
39. S. Perantonis, N. Ampazis, S. Varoufakis and G. Antoniou, "Constrained learning in neural networks", application to stable factorization of 2nd polynomials. Neural Process. Lett.,vol. 7(1), pp. 5–14, 1998.
40. D.S. Huang, H.H.S. Ip, K.C.K. Law, "Chi: Zeroing polynomials using modified constrained neural network approach",IEEE Trans. Neural Netw., vol. 16(3), pp. 721–732, 2005.
41. D.E. Rumelhart, G.E. Hinton andR.J. Williams, "Learning representations by back-propagating errors", Nature., vol. 323(9),pp. 533–536,1986.
42. K. Bache and M. Lichman, "UCI Machine Learning Repository", University of California, School of Information and Computer Science, Irvine, CA, 2013.
43. S. Larson, "The shrinkage of the coefficient of multiple correlation", J. Educat.Psychol. vol. 22 (1931) pp. 45–55.