

# A Deep Learning Breast Cancer Prediction Framework

Asmaa E. E. Ali\*, Mofreh Mohamed Salem, Mahmoud Badway and Ali I. EL Desouky

Department of Computer Science, Faculty of Engineering, Mansoura University, Mansoura, 35111, Egypt

\*Corresponding Author: Asmaa E. E. Ali. Email: EngAsmaAli@std.mans.edu.eg

Received: 07 August 2021; Accepted: 15 December 2021

**Abstract:** Breast cancer (BrC) is now the world's leading cause of death for women. Early detection and effective treatment of this disease are the only rescues to reduce BrC mortality. The prediction of BrC diseases is very difficult because it is not an individual disease but a mixture of various diseases. Many researchers have used different techniques such as classification, Machine Learning (ML), and Deep Learning (DL) of the prediction of the breast tumor into Benign and Malignant. However, still there is a scope to introduce appropriate techniques for developing and implementing a more effective diagnosis system. This paper proposes a DL prediction BrC framework that uses a selected Bidirectional Recurrent Neural Network (BRNN). An efficient fast and accurate optimizer is needed to train the neural network used. The more recent Dynamic Group-based Cooperative Optimization Group (DGCO) algorithm is modified MDGCO for this purpose. The Deep Learning Breast Cancer Prediction Framework (DLBCPF) includes four layers: preprocessing, feature selection, optimized Recurrent Neural Networks, and prediction. Four different Wisconsin BrC datasets are used to test the validity of the proposed framework and optimizer against others. The results obtained have shown the superiority of both the framework DLBCPF and the optimizer MDGCO when they are compared to others.

**Keywords:** BrC diagnosis; DGCO algorithm; deep recurrent neural networks; classification

## 1 Introduction

Worldwide, BrC is an important disease because it affects large numbers of people. BrC is the second only to lung cancer when it comes to mortality [1]. It is possible that the risk factors include being from one's family history, genetics, and that how the breast cells mutate into the disease are not properly known until now [2]. There is no documented procedure that prevents BrC, however, early detection is the only means to get a better prognosis and faster treatment is critical for a woman's chances of a successful outcome [3]. Cancer detected in early stages, before it has a chance to spread to other organs, is better for women's health [4–5].

Numerous methods of breast cancer diagnosis have been developed in the literature using the Wisconsin Breast Cancer Dataset (WBCD) and Wisconsin Diagnostic Breast Cancer Dataset (WDBC) [6]. While these methods are getting very accurate, e.g., they have about 95 to 100% reported classification rates [7–9].

A number of researches have recently been published to solve BrC classification, segmentation, and detection/grading problems by employing ML techniques such as support vector, logistic, naive Bayes, and decision tree or by using Artificial Neural Network (ANN)-based approaches [10] (e.g., shallow neural networks (SNNs) [11] and deep neural networks (DNNs) [12]).

In deep neural networks (DNNs), there are mostly two or more hidden layers between the input and output than in the way many shallow neural networks (SNNs) have only one. However, in the medical imaging modalities only a few researches [5,9,13] exist to summarize BrC classification. Thus, other studies



[14–17] have compared the advantages of using Hand-Engineered Features to those of ML approaches to images of BrC.

Moreover, the recently discussed utilization of H&E BrC prognosis images includes Hematoxylin and Eosin (H&E) in [14]. A new algorithm should be discussed to handle a specific type of image. Finally, ML methods have been quantified and tested against future approaches. An abundance of ultrasound imaging techniques has been used in [18] to look for (occasional forms of) cancer, including breast X-ray, ultrasound imaging (US), and computed tomography (CT). They compared how the various ML methods had been used for detecting and classifying the various cancers. Along with other imaging modalities, many imaging techniques have been analyzed and presented in the search for the diagnosis of cancer. Suggestions have been offered as to future researchers in the area of future endeavors. Previous research that has employed ML modalities has primarily focused on binary classifications by using imaging modalities. Some recent assessment researches [19–20] have concentrated on multimodal Artificial Neural Networks (ANNs) in BrC analysis. For instance, sophisticated breast imaging (involving multimodalities) using in [19]. To understand the components of artificial neural networks in BrC research, they employed them multimodalities like mammogram (Mg), Magnetic Resonance Imaging (MRI), and thermal imaging. Many of the newer methods of BrC diagnosis use various imaging modalities and ANN depend on computer-aided diagnosis (CAD) in [20]. They compared the imaging procedures of Mg, MRI, and thermography, then determined the advantages and disadvantages of each method. As mentioned above, the multimodal analyses have been used to describe the application of ANN models in various applications. Moreover, medical images were used to test all of the different CAD systems, including Histopathology (HP) images in [21–22].

Several studies on BrC have been reported. Five of them are given in Table 1 [23–27]. All these researches used all the datasets that were used in this research.

**Table 1:** An overview of papers on different methods and results of BrC

Reference	Year	Methods	Dataset	Accuracy (%)
Islam et al. [23]	2020	ANN	Wisconsin Breast Cancer	98.57
		SVM	Diagnosis (WBCD) dataset	97.14
Wu et al. [24]	2020	Acoustic radiation force impulse (ARFI)	acoustic radiation force impulse by virtual touch tissue imaging (ARFI-VTI)	87.78±2.17
Wu et al. [25]	2020	A multilayer machine vision classifier	ARFI-VTI	86.97
Benhammou et al. [26]	2019	ResNet	BreakHis	88.9
			WBC	97.4
Osman et al. [27]	2020	Radial Based Function Neural Network models (RBFNN)	BCD	98.4
			BCE	97.7
			WBCD UCI	97.0

This paper proposes a biomedical diagnosis DL framework for BrC prediction. A Modified Dynamic Group-based Cooperative Optimization Group (MDGCO) is also proposed to train the selected BRNN neural network used.

The rest of this paper is organized as follows:

- Section 2 discusses the related work and problem definition.
- Section 3 presents the framework (DLBCPF) and the proposed optimization algorithm (MDGCO) of this research.
- Section 4 displays the experimental MDGCO and DLBCPF tests.
- Section 5 reveals the conclusions and the future work.

## 2 Related Works

### 2.1 Recurrent Neural Networks (RNNs)

RNNs are a Neural Network type (NNs) concerned with serial input and output data applications. RNNs capture the temporal relation between input/output sequences with feedback to Feed Forward (FF) of the neural networks. Speech recognition of sequential data may actually benefit from RNNs.

#### 2.1.1 Unidirectional Recurrent Neural Network (URNN)

Let the input to a RNN by  $X = \{x_t\}$  where for each time step  $t$ ,  $x_t \in R^N$  is an input vector. Furthermore, consider the output as  $Y = \{y_t\}$ , where  $y_t \in R^M$  is the vector representing the output for each time step  $t$ . Our objective is to model the distribution  $P(Y|X)$ .

The output of a RNN  $y_t$  is determined by [28]:

$$P(y_t | \{x_i\}_{i=1}^t) = \sigma(W_y h_t + b_y) \quad (1)$$

where:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h) \quad (2)$$

The weight matrix that connects the hidden layer to the output layer is denoted by  $W_y$ . The weight matrix that connects the hidden layer to the hidden layer is denoted by  $W_h$ . The weight matrix that connects the input layer to the hidden layer is denoted by  $W_x$ . The bias vectors for the output layer is denoted by  $b_y$ .  $b_h$  is the hidden layer bias vectors [29].

Final nonlinearity  $\sigma$  in classification is represented by the sigmoid, tanh, and the relu activation functions. Depending on the information that propagates from the hidden layer, the recurrent network computes the output ( $y_t$ ) in any case of it depends directly or indirectly on the values  $\{x_i\}_{i=1}^t = \{x_1, \dots, x_t\}$  [29].

#### 2.1.2 Bidirectional Recurrent Neural Network (BRNN)

BRNNs have a major advantage of more input information being passed to the network [30]. Multilayer perceptron networks (MLPs) and time delay recurrent networks (TDNs) suffer from fixed input data and the absence of future information. While BRNNs is solving one of these two problems, however, no fixed input data and future information are both inaccessible.

BRNN introduces a second hidden layer where connections from the hidden to hidden layers are in opposite temporal order [30]. Thus, the framework is able to draw on data from both directions; past and future. The output  $y_t$  has been determined by [28].

$$P(y_t | \{x_i\}_{i \neq t}) = \sigma(W_y^f h_t^f + W_y^b h_t^b + b_y) \quad (3)$$

$\sigma$  is the activation function to generate the hidden state.

where:

$$h_t^f = \tanh(W_h^f h_{t-1}^f + W_x^f x_t + b_h^f) \quad (4)$$

$$h_t^b = \tanh(W_h^b h_{t+1}^b + W_x^b x_t + b_h^b) \quad (5)$$

where the weight matrix from the input layer to the hidden layer is  $W_x^f, W_x^b$ .  $W_h^f, W_h^b$  is the weight matrix between two consecutive hidden states ( $h_{t-1}^f$  and  $h_{t+1}^b$ ).  $b_h^f, b_h^b$  is the hidden layer's bias vector.

If the inputs to any neuron are  $x_1, x_2$ , and  $x_3$ , and the weights applied are  $W_1, W_2$ , and  $W_3$ , The output is:

$$y = f(x) = \sum_{j=1}^n x_j W_j \quad (6)$$

where the number of inputs is denoted by  $n$ . A matrix multiplication operation used to compute the weighted sum is denoted by  $y$ .

In a linear equation, the bias term is added at the point of interception. It may be use a variable to adapt the output in addition to the weighted totally of the inputs to the neuron [28]. A neuron's final output is as follows:

$$y = f(x) = \sum_{j=1}^n x_j W_j + b \quad (7)$$

where  $b$  is the bias.

## 2.2 Dynamic Group-Based Cooperative Optimization (DGCO)

In our world, there are various optimization applications. Actually problems with real-world optimization with a high-dimensional search area are also usually difficult. Techniques for heuristic optimization has been used in many areas like engineering, ML, business processes, mechanics, economics.

There are two sub-groups of the population of candidate individuals: the exploration group (expRG), and the exploitation group (expTG). The expRG is mainly responsible for exploring new regions within the search area in search of the optimal solution. In contrast, another second group focuses mainly on enhancing the best solution's performance depending on an objective function. DGCO can balance both exploration and exploitation, helping to prevent the steady regions of the search region [31]. The dynamic group-based optimization algorithm's expRG and expTG are depicted in Fig. 1 [31].

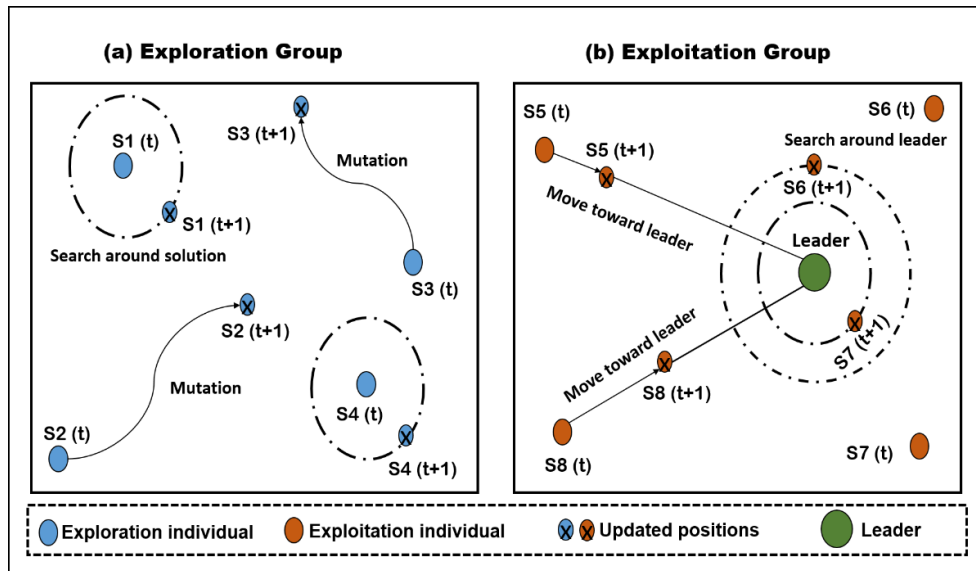


Figure 1: DGCO Algorithm expRG and expTG

### (1) Exploration Group

ExpRG can be accomplished in 2 ways: Explore the solution's search area and mutation [31].

#### (a) Exploring the search area's solution

Individual searches in this strategy for a prospective region in their search region was achieved through reduplicate looking for the best fitness value solution among their neighboring solutions. DGCO uses the following equations for this purpose [31]:

$$\vec{C} = v_1 \cdot (\vec{B}(i) - 1) \quad (8)$$

$$\vec{B}(i+1) = \vec{B}(i) + \vec{C} \cdot (2 \vec{v}_2 - 1) \quad (9)$$

where Coefficient vectors are  $\vec{v}_1$  and  $\vec{v}_2$ , respectively, in intervals  $[0, 2]$  and  $[0, 1]$ .  $i$  is the real iteration.  $\vec{B}$  is the real solution vector.  $\vec{C}$  Specifies the circle diameter in which the solution can search for a prospective region.

(b) Mutating

Another technique applicable to exploration is mutation. It is used to increase and preserve population diversity. It helps to avoid local optima.

(2) Exploitation Group (*expTG*)

DGCO employs two distinct techniques to accomplish exploitation:

(a) The movement to the best solution

The individual uses the following formulas for moving to the best solution [31]:

$$\vec{C} = \vec{v}_3 \cdot (\vec{X}(i) - \vec{B}(i)) \tag{10}$$

$$\vec{B}(i+1) = \vec{B}(i) + \vec{C} \tag{11}$$

where  $\vec{v}_3$  is a random vector at the interval [0, 2] which regulates the movement towards the solution leader.  $i$  is the real iteration.  $\vec{B}$  is the vector of the real solution.  $\vec{X}$  is the best solution vector.  $\vec{C}$  shows the vector distance.

(b) Searching around the best solution

The region immediately surrounding the leader is almost certainly the most promising. Additionally, some individuals are on the lookout for the best solution in the region, with the goal of obtaining an even more optimal solution. The following formulas are used by DGCO:

$$\vec{C} = \vec{X}(i) * (\vec{E} - \vec{v}_4) \tag{12}$$

$$\vec{B}(i+1) = \vec{B}(i) + \vec{C} \cdot (2 \vec{v}_5 - 1) \tag{13}$$

$$\vec{E} = 2 \left[ 1 - \left( \frac{i}{IC} \right)^2 \right] \tag{14}$$

where  $\vec{v}_4$  and  $\vec{v}_5$  are random interval vectors [0, 1].  $\vec{E}$  decreases from 2 to 0 exponentially during iterations.  $\vec{X}$  is the best solution vector.  $\vec{B}$  is the real solution vector.  $\vec{C}$  shows the diameter of the circle in which the solution seeks optimal solutions.

### 3 Methodology

Some notes are given below for BrC datasets.

#### 3.1 Breast Cancer Dataset

All experiments use the datasets for Wisconsin Breast Cancer in the UCI Machine Learning Repository to distinguish between malignant (cancerous) and benign (non-cancerous) samples [6] as shown in Table 2.

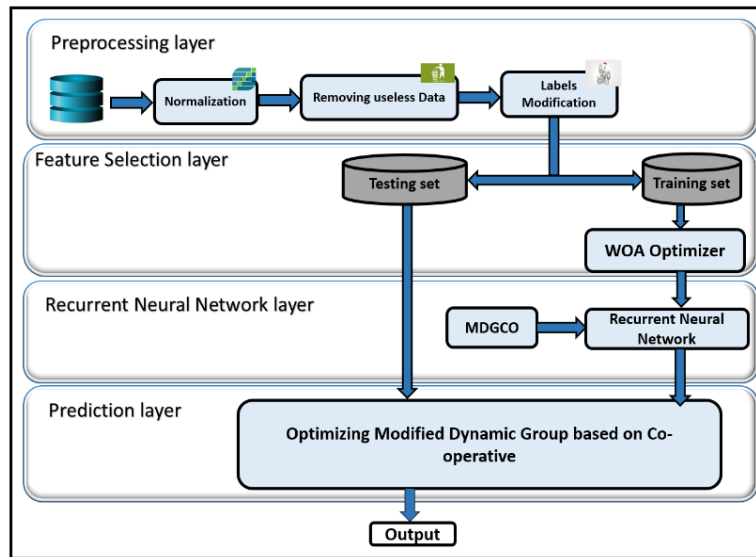
**Table 2:** The BrC datasets are described in detail [6]

Dataset	No. of Attributes	No. of Instances	No. of Classes
Wisconsin Breast Cancer (D1)	11	699	2
Wisconsin Diagnosis Breast Cancer (D2)	32	569	2
Breast Cancer Coimbra (D3)	116	9	2
Breast Cancer (D4)	286	9	2

#### 3.2 The Proposed Deep Learning Breast Cancer Prediction Framework (DLBCPF)

A four-layer DL prediction framework is proposed in this section is shown in Fig. 2 that consists of the preprocessing layer, feature selection layer, optimized recurrent neural networks layer, and prediction layer. The binary WOA is used as the more accurate optimizer between the four optimizers tested bPSO, bGWA, and bGA to extract the feature selection of the dataset used. The Bidirectional Recurrent Neural Network is

selected according to a comparative experiment between BRNN and URNN under the dataset used. To improve the learning process of the neural network and avoid local optima in the optimization problem, a high-quality optimizer is needed. A modified version of the DGCO is used (MDGCO) for that propose.



**Figure 2:** The proposed DLBCPF

### 3.3 Modified Dynamic Group Cooperative Optimization Algorithm (MDGCO)

Several trials for improving the performance of the Dynamic Group Cooperative Optimization algorithm (DGCO) are failed or resulted in a very small improvement in its performance. Another trial uses the Gaussian of Eq. (9) in the expRG.

To generate new particles using the Gaussian method of distribution, a random walk procedure is done in the process of Diffusion Limited Aggregation (DLA) growth. A series of created diffusion procedures can be calculated based on the better solution:

$$\vec{B}_i^{t*} = \text{Gaussian}(\mu_{\vec{B}^*}, \omega) + (\alpha \times \vec{B}^* - \acute{\alpha} \times \vec{b}_i) \quad (15)$$

where  $\vec{B}_i^{t*}$  is the best-modified diffusion process-based solution.  $\alpha$  and  $\acute{\alpha}$  are parameters of the random numbers in  $[0; 1]$ .  $\vec{B}^*$  and  $\vec{b}_i$  display the better location and the  $i$ -th point in the group around.  $\mu_{\vec{B}^*} = |\vec{B}^*|$  and  $\omega = |\vec{b}_i - \vec{B}^*|$  as the number of generations is decreased about a better solution. It may improve the exploration potential in the suggested DGCO by using the diffusion procedure to find an optimal solution [32].

Exploration performance is enhanced by using diffusion process rather than search space process in DGCO. This may require individuals to explore a prospective region in the searching region further and prevent local stagnation by substituting Eq. (15). Namely, to improve the performance of DGCO, Eq. (9) in DGCO has been replaced by Eq. (15) to reach a global minimum and avoid a local minimum. Finding the global minimum is made more likely, but being trapped in a local minimum is avoided.

---

#### Algorithm 1: Pseudocode of MDGCO Optimization Algorithm

---

- 1 Require: better solution  $\vec{B}^*$
  - 2 Define: population size = PS  
iteration\_count = IC  
mutation rate
  - 3 Starting: population  $\vec{B} = \{B_1, B_2, \dots, B_d\}$ , and  $fit_n$ (fitness function)
  - 4 Starting:  $v_1, v_2, v_3, v_4$ , PS (DGCO parameters)
-

---

```

5 Starting: SFS parameters
6 Starting: initialize counter (i) = 1
7 While i < IC
8   If (i is even)
9     evaluate the fitness of each solution
10    evaluate the better solution  $\vec{B}^*$ 
11    update in each group, the number of solutions
12    If it didn't improve the best fitness from the previous two iterations
13      increasing the number of possible solutions in the expRG
14    end
15    for each solution in the group of explorations
16      update  $v_1, v_2$ , and PS
17      choose the better solution
18      if (PS > = 0.5)
19        Mutation of the solution
20      else
21         $\vec{B}_i^* = \text{Gaussian}(\mu_{\vec{B}^*}, \omega) + (\alpha \times \vec{B}^* - \alpha \times \vec{b}_i)$ 
22      end
23    end
24    for each solution in the expTG
25      choose the better solution
26      update  $v_2, v_3, v_4$  and PS
27      if (PS > = 0.5)
28         $\vec{B}(i+1) = \vec{B}(i) + \vec{C}$ 
29      else
30         $\vec{S}(t+1) = \vec{S}(t) + \vec{C} \cdot (2 \vec{v}_5 - 1)$ 
31      end
32    end
33    evaluate  $fit_n$  for each  $\vec{B}_i$  from DGCO
34    else
35      evaluate  $fit_n$  for each  $\vec{B}_i$  from SFS
36    end
37    modify solutions that reach beyond the search space
38    update previous_fitness1, previous_fitness2
39    get  $\vec{B}^*$ 
40    i = i + 1
41  end
42  return  $\vec{B}^*$ 

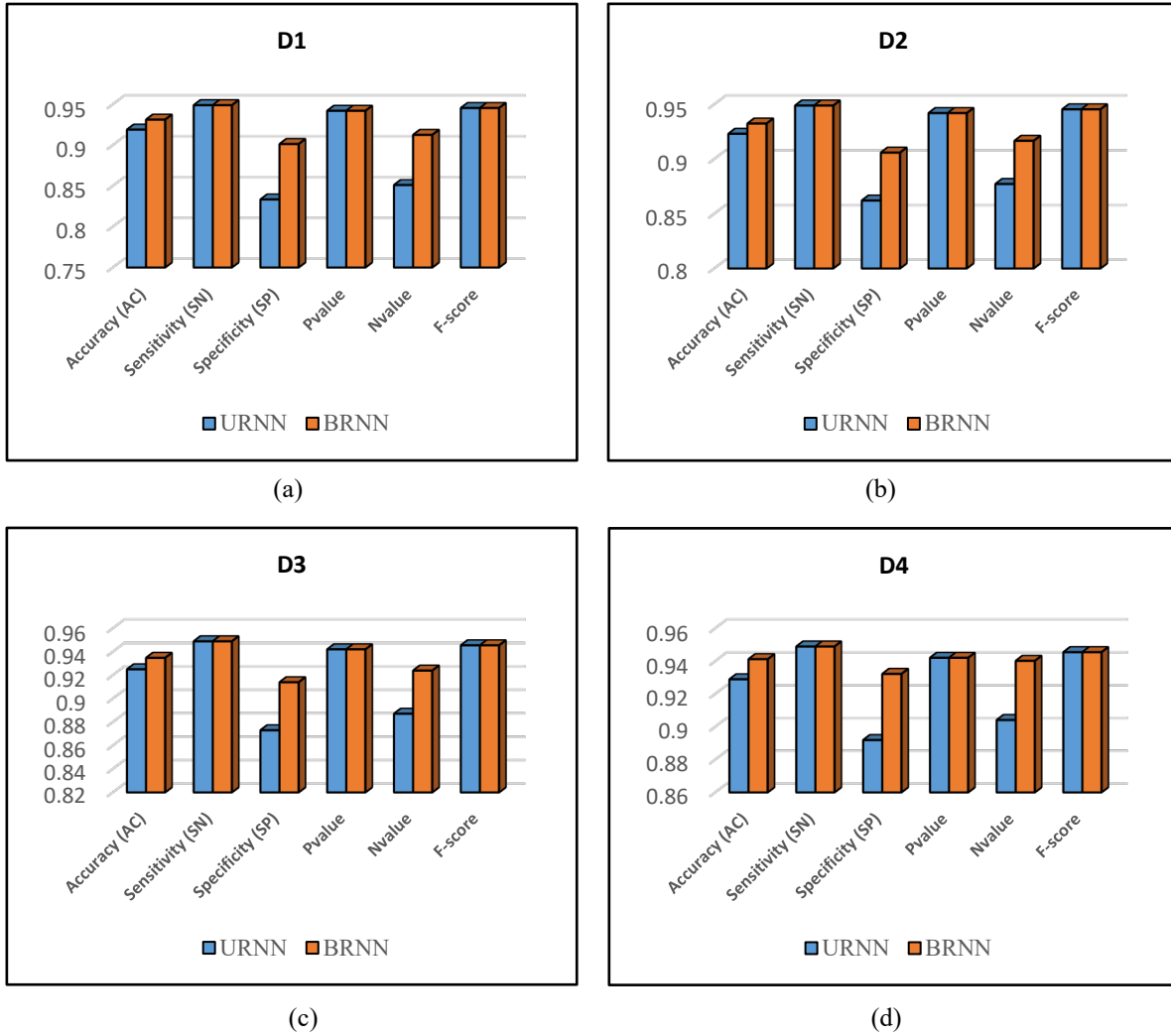
```

---

## 4 Experimental MDGCO

### 4.1 Experiment 1: Select the Type of the Recurrent Neural Networks

This experiment is made to select one of the URNN or BRNN for the proposed framework. MDGCO uses to predict the result. The results of applying MDGCO on BRNN and URNN are shown in Fig. 3. These figures show that BRNN is more suitable than URNN for all datasets used.



**Figure 3:** Accuracy comparative study for URNN and BRNN analysis

Metrics efficiency are accuracy, sensitivity, specificity, precision ( $P$ -value), negative predictive value ( $N$ -value), and  $F$ -score.

$P^+$ : true-positive value;  $N^+$ : true-negative value;  $N^-$ : false-negative value; and  $P^-$ : false-positive value. The metrics are described as follows:

- Accuracy: tests the potential of the model to correctly classify status, instead of the case whether it is positive or negative as following:

$$\text{Accuracy} = \frac{P^+ + N^+}{P^+ + N^+ + P^- + N^-} \quad (16)$$

- Sensitivity (Recall): the true positive rate is named. Sensitivity calculates the positive case's potential as following:

$$\text{Sensitivity} = \frac{P^+}{P^+ + N^-} \quad (17)$$

- Specificity (selectivity): the true negative rate is named. Specificity can find negative cases as following:

$$\text{Specificity} = \frac{N^+}{N^+ + P^-} \quad (18)$$

- $P$ -value: named positive predictive value. The rate of true positive values is guided between all



positive values as following:

$$P\text{-value} = \frac{P^+}{P^+ + P^-} \tag{19}$$

- Negative predictive value (*N*-value): The rate of true negative values is calculated between all negative values as following:

$$N\text{-value} = \frac{N^+}{N^+ + N^-} \tag{20}$$

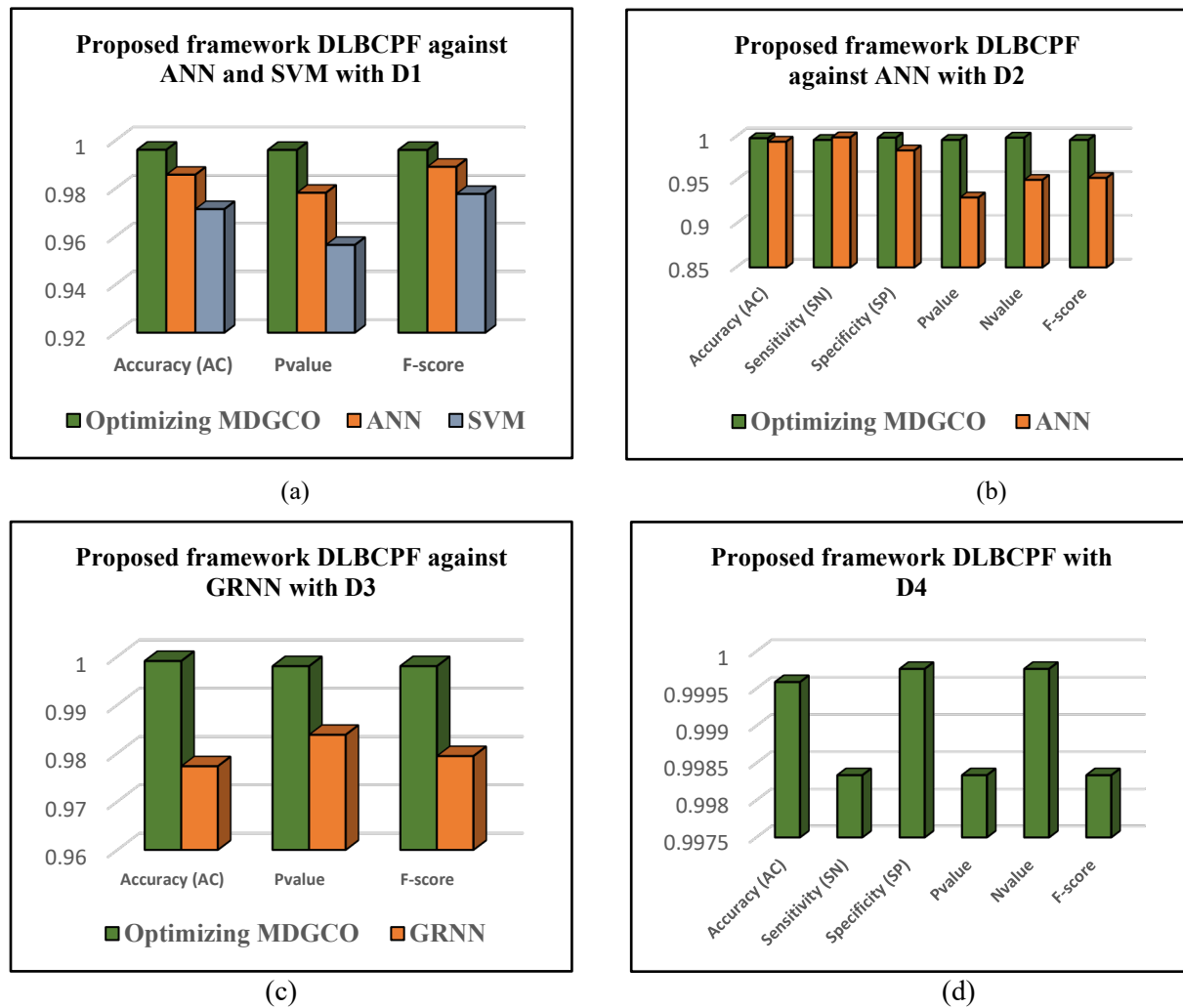
- *F*-score: Tests the harmonic mean of *P*-value and Sensitivity as following:

$$F\text{-score} = 2 \times \frac{P^P \times SN}{P^P + SN} \tag{21}$$

#### 4.2 Experiment 2: Comparative Study Between DLBCPF, ANN, SVM, and GRNN

Four comparative experiments were conducted to assess the proposed DLBCPF’s performance in comparison to ANN, SVM, and GRNN. The results obtained are shown.

ANN and SVM in (Fig. 4a) D1, ANN [23] in (Fig. 4b) D2, and Generalized Regression Neural Network (GRNN) [33] in (Fig. 4c) D3.



**Figure 4:** Comparative study of the BrC datasets of the DLBCPF against others ANN, SVM, and GRNN for all datasets under study

As illustrated in Fig. 4, the proposed framework (DLBCPF) outperforms ANN, SVM, and GRNN.

## 5 Comparative Study between DLBCPF Against ANN, SVM, and GRNN

Comparative studies have been conducted to evaluate the proposed MDGCO optimizer's performance in terms of identifying the optimal points for each function and its convergence behavior. The results obtained are shown in Table 3.

**Table 3:** Comparative study between DLBCPF against ANN, SVM, and GRNN

D	Case	Accuracy	Sensitivity	Specificity	P-value	N-value	F-score
	<b>DLBCPF</b>	0.99609375	0.996015936	0.996168582	0.996015936	0.996168582	0.996016
D1	<b>ANN</b>	0.9857	_____	_____	0.9782	_____	0.989
	<b>SVM</b>	0.9714	_____	_____	0.9565	_____	0.9777
D2	<b>DLBCPF</b>	0.998003992	0.996015936	0.998668442	0.996015936	0.998668442	0.996016
	<b>ANN</b>	0.994	0.999	0.984	0.93005	0.950256	0.95256
D3	<b>DLBCPF</b>	0.999070632	0.998003992	0.999394306	0.998003992	0.999394306	0.998004
	<b>GRNN</b>	0.9773	_____	_____	0.9838	_____	0.9794
D4	<b>DLBCPF</b>	0.999587799	0.998336106	0.999764761	0.998336106	0.999764761	0.998336

## 6 DLBCPF Tests

### 6.1 BenchMark Functions (BenFN)

All test functions F1 to F23 are carried out on the proposed MDGCO algorithm and compared to other recent algorithms. Twenty-three BenFNs have been applied to search space to determine the performance of the proposed optimization algorithm. These BenFNs are frequently used in literature to evaluate different approaches of optimization algorithms [34]. These functions are classified as unimodal, multimodal, and multimodal with fixed dimensions. Tables 4 and 5 list the BenFNs, where D denotes the function's dimension and  $f_{min}$  denotes the function's optimum value, which is the smallest in this case [34–35].

**Table 4:** The unimodal BenFNs used in our experiments are described

BenFNs	D	Range	$f_{min}$
$f_{01}(x) = \sum_{i=1}^n x^2$	30	[-100, 100]	0
$f_{02}(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	[-10, 10]	0
$f_{03}(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100, 100]	0
$f_{04}(x) = \max_i\{ x_i , 1 \leq i \leq D\}$	30	[-100, 100]	0
$f_{05}(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
$f_{06}(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	30	[-100, 100]	0
$f_{07}(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1]$	30	[-1.28, 1.28]	0

**Table 5:** Multimodal BenFNs

BenFN	D	Range	$f_{min}$
$f_{08}(x) = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	30	[-500, 500]	-12569.487
$f_{09}(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	30	[-32, 32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
$f_{12}(x) = \frac{\pi}{D} \{10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1) + (yD - 1)^2 + \sum_{i=1}^D u(x_i, 10, 100, 4)]\}$ $y_i = 1 + \frac{x_i+1}{4}, \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50, 50]	0
$f_{13}(x) = 0.1 \{10 \sin^2(3\pi y_i) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + 10 \sin^2(3\pi y_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50, 50]	0
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5, 5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	[-5, 5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 x (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2)$	3	[1, 3]	-3.86
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_j^6 a_{ij} (x_j - p_{ij})^2)$	6	[0, 1]	-3.32
$f_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0, 10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0, 10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + C_i]^{-1}$	4	[0, 10]	-10.5363

**6.2 Experiment 1: Test the Validity of MDGCO**

Five tests have been made to test the performance and the analysis of the proposed MDGCO in the following subsections. It should be noted that all experiments have run 20 times on each BenFNs. Each run has 500 iterations MDGCO is compared to other well-known cooperative and competitive algorithms such as Dynamic Group Co-operative Optimization (DGCO) [31], Particle Swarm Optimization (PSO) [36], Whale Optimization Algorithm (WOA) [37], Grey Wolf Optimizer (GWO) [38], and Genetic Algorithm (GA) [39]. The results are given in Tables 6 and 7.

**6.2.1 Performance Evaluation of Exploitation**

To BenFN exploitation optimization algorithms, only unimodal functions (1 to 7) have one global optimum. Table 6 shows the superiority of the MDGCO optimizer where it has the most efficient results for most functions for exploitation Performance Analysis.

**Table 6:** Results of statistical (mean, standard deviation) for performing the evaluation of exploitation (F1–F7)

Functions	Algorithm	MDGCO	DGCO[31]	PSO [36]	WOA[37]	GWO[38]	GA[39]
F1	<b>Mean</b>	0	0	0.000136	1.41E-30	6.59E-28	4.5616E-172
	<b>StDev</b>	0	0	0.000202	4.91E-30	0.0000634	0
F2	<b>Mean</b>	0	0	0.042144	1.06E-21	7.18E-17	3.43918E-90
	<b>StDev</b>	0	0	0.045421	2.39E-21	0.029014	6.12647E-90
F3	<b>Mean</b>	0	0	70.12562	5.39E-07	0.00000329	1.6584E-127
	<b>StDev</b>	0	0	22.11924	2.93E-06	79.14958	8.6363E-127
F4	<b>Mean</b>	0	0	1.086481	0.072581	0.000000561	1.15478E-75
	<b>StDev</b>	0	0	0.317039	0.39747	1.315088	2.45326E-75
F5	<b>Mean</b>	0.000281449	0.001824	96.71832	27.86558	26.81258	28.372867
	<b>StDev</b>	0.005421514	0.006422	60.11559	0.763626	69.90499	0.582802101
F6	<b>Mean</b>	0.000537247	0.000647	0.000102	3.116266	0.816579	3.932625965
	<b>StDev</b>	0.000770354	0.00088	0.0000828	0.532429	0.000126	0.431754883
F7	<b>Mean</b>	0.00036727	0.000473	0.122854	0.001425	0.002213	0.022991503
	<b>StDev</b>	0.00040865	0.000509	0.044957	0.001149	0.100286	0.021966199

### 6.2.2 Performance Evaluation of Exploration

The multimodal BenFNs (8 to 23) contain a large number of local minima that grow exponentially in size with the problem. As a result, they are well-suited for assessing exploration performance and the capacity to avoid local optima. Based on Table 7, it can be calculated that MDGCO is on the top among other optimization algorithms. This table yields these reported findings, which show MDGCO exploration's robustness.

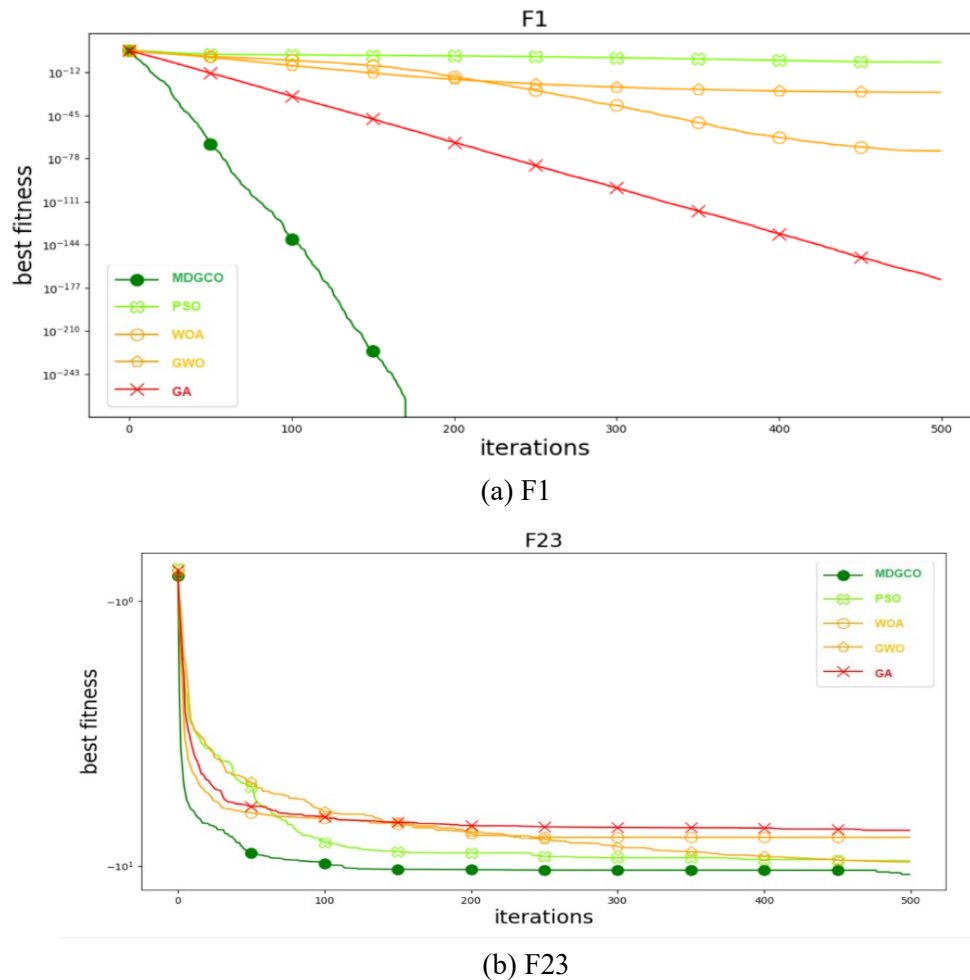
**Table 7:** Results of statistical (mean, standard deviation between MDGCO and other optimizers) for performance evaluation of exploration

Functions	Algorithm	MDGCO	DGCO[31]	PSO[36]	WOA[37]	GWO[38]	GA[39]
F8	<b>Mean</b>	-8212.65252	-7349.753	-4841.29	-5080.76	-6123.1	-4080.18242
	<b>StDev</b>	694.504763	782.604763	1152.814	695.7968	-4087.44	551.6504246
F9	<b>Mean</b>	0	0	46.70423	0	0.310521	0
	<b>StDev</b>	0	0	11.62938	0	47.35612	0
F10	<b>Mean</b>	4.22E-17	4.44E-16	0.276015	7.4043	1.06E-13	7.99361E-16
	<b>StDev</b>	0	0	0.50901	9.897572	0.077835	1.06581E-15
F11	<b>Mean</b>	0	0	0.009215	0.000289	0.004485	0
	<b>StDev</b>	0	0	0.007724	0.000289	0.006659	0
F12	<b>Mean</b>	5.66281E-05	5.88E-05	0.006917	0.339676	0.053438	0.556173028

	<b>StDev</b>	5.90261E-05	6.02E-05	0.026301	0.214864	0.020734	0.063582238
	<b>Mean</b>	2.11417E-05	2.23E-05	0.006675	1.889015	0.654464	2.132496784
F13	<b>StDev</b>	5.00199E-05	5.41E-05	0.008907	0.266088	0.004474	0.174792075
	<b>Mean</b>	0.978003838	0.998004	3.627168	2.111973	4.042493	0.998003839
F14	<b>StDev</b>	9.11E-13	9.83E-13	2.560828	2.498594	4.252799	1.36835E-09
	<b>Mean</b>	0.000256867	0.000367	0.000577	0.000572	0.000337	0.002317517
F15	<b>StDev</b>	2.319E-05	4.72E-05	0.000222	0.000324	0.000625	0.010072146
	<b>Mean</b>	-1.01331628	-1.031628	-1.03163	-1.03163	-1.03163	-1.03162685
F16	<b>StDev</b>	1.00217E-06	1.42E-06	6.25E-16	4.2E-07	-1.03163	4.4372E-06
	<b>Mean</b>	0.397897123	0.397897	0.397887	0.397914	0.397889	0.398222669
F17	<b>StDev</b>	1.6693E-05	1.60E-05	0	2.7E-05	0.397887	0.001394927
	<b>Mean</b>	3	3.000014	3	3	3.000028	3.000028828
F18	<b>StDev</b>	1.69077E-05	1.79E-05	1.33E-15	4.22E-15	3	4.22314E-05
	<b>Mean</b>	-3.86273744	-3.862737	-3.86278	-3.85616	-3.86263	-3.86272389
F19	<b>StDev</b>	8.60091E-05	8.8E-05	2.58E-15	0.002706	-3.86278	9.0175E-05
	<b>Mean</b>	-3.21119131	-3.277191	-3.26634	-2.98105	-3.28654	-3.25066404
F20	<b>StDev</b>	0.059339409	0.058339	0.060516	0.376653	-3.25056	0.081811358
	<b>Mean</b>	-11.0668054	-10.098054	-6.8651	-7.04918	-10.1514	-6.03721489
F21	<b>StDev</b>	0.008885108	0.008885	3.019644	3.629551	-9.14015	1.99897306
	<b>Mean</b>	-10.1772871	-10.164287	-8.45653	-8.18178	-10.4015	-6.76809171
F22	<b>StDev</b>	0.006845479	0.006955	3.087094	3.829202	-8.58441	2.628445545
	<b>Mean</b>	-10.77896	-10.47606	-9.95291	-9.34238	-10.5343	-5.79459095
F23	<b>StDev</b>	0.000654086	0.009741	1.782786	2.414737	-8.55899	2.643454405

### 6.3 Convergence Behavior

The convergence behavior of MDGCO against DGCO, GWO, PSO, WOA, and GA for some BenFNs (F1, and F23) are shown in Fig. 5. The figure proves that MDGCO converges is faster than other algorithms. MDGCO possesses the ability to balance exploration and exploitation, besides initiating the exploitation in the early iterations. For two figures, the convergence accuracy is superior to that of other algorithms.



**Figure 5:** MDGCO, PSO, WOA, GWO, and GA convergence curves for three BenFNs

## 7 Conclusion

The analysis of BrC diagnosis is an extremely complex process as it is a mixture of various diseases rather than one. Experiments are used to select each component of the four layers framework such as neural network (BRNN), modified MDGCO to adapt the parameters of the BRNN and to use in the prediction layer. Twenty-three BenFNs have been applied to search space to determine the performance of the proposed optimization algorithm. MDGCO exhibits a high rate of convergence as a result of its extensive exploration and exploitation capabilities and ability to avoid local optima. The performance of the proposed DLBCPF framework was tested against others.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] D. R. Ericeira, A. C. Silva, A. C. de Paiva and M. Gattass, "Detection of masses based on asymmetric regions of digital bilateral mammograms using spatial description with variogram and cross-variogram functions," *Computers in Biology and Medicine*, vol. 43, no. 8, pp. 987–999, 2013.

- [2] J. M. Jerez-Aragonés, J. A. Gómez-Ruiz, G. Ramos-Jiménez, J. Muñoz-Pérez and E. Alba-Conejo, “A combined neural network and decision trees model for prognosis of breast cancer relapse,” *Artificial Intelligence in Medicine*, vol. 27, no. 1, pp. 45–63, 2003.
- [3] N. M. Abdel Razek, W. A. Yousef and W. A. Mustafa, “Microcalcification detection with and without CAD system (LIBCAD): A comparative study,” *The Egyptian Journal of Radiology and Nuclear Medicine*, vol. 44, no. 2, pp. 397–404, 2013.
- [4] S. M. A. Beheshti, H. AhmadiNoubari, E. Fatemizadeh and M. Khalili, “An efficient fractal method for detection and diagnosis of breast masses in mammograms,” *Journal of Digital Imaging*, vol. 27, no. 5, pp. 661–669, 2014.
- [5] S. Dalwinder, S. Birmohan and K. Manpreet, “Simultaneous feature weighting and parameter determination of neural networks using ant lion optimization for the classification of breast cancer,” *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 337–351, 2020.
- [6] *UCI Machine Learning Repository: Breast Cancer Coimbra Data Set*. University of California, 2018. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer>.
- [7] N. Liu, E. S. Qi, M. Xu, B. Gao and G. Q. Liu, “A novel intelligent classification model for breast cancer diagnosis,” *Information Processing & Management*, vol. 56, no. 3, pp. 609–623, 2019.
- [8] R. Jafari-Marandi, “An optimum ANN-based breast cancer diagnosis: bridging gaps between ANN learning and decision-making goals,” *Applied Soft Computing*, vol. 56, no. 3, pp. 609–623, 2018.
- [9] N. Liu, “An effective approach for breast cancer diagnosis based on routine blood analysis features,” *Medical & Biological Engineering & Computing*, vol. 56, no. 7, pp. 1583–1601, 2020.
- [10] V. Sze, Y. H. Chen, T. J. Yang and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.
- [11] A. P. Piotrowski, J. J. Napiorkowski and A. E. Piotrowska, “Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modelling,” *Earth-Science Reviews*, vol. 201, no. 11, 103076, 2019.
- [12] F. Altaf, S. M. S. Islam, N. Akhtar and N. K. Janjua, “Going deep in medical image analysis: concepts, methods, challenges and future directions,” *IEEE Access*, vol. 7, pp. 99540–99572, 2019.
- [13] X. Zhang, “Deep learning based analysis of breast cancer using advanced ensemble classifier and linear discriminant analysis,” *IEEE Access*, vol. 8, pp. 120208–120217, 2020.
- [14] J. M. Chen, “Computer-aided prognosis on breast cancer with hematoxylin and eosin histopathology images: A review,” *Tumor Biology*, vol. 39, no. 3, 2017.
- [15] S. Z. Ramadan, “Methods used in computer-aided diagnosis for breast cancer detection using mammograms: A review,” *Journal of Healthcare Engineering*, vol. 2020, 2020.
- [16] M. A. Al-antari, S. M. Han and T. S. Kim, “Evaluation of deep learning detection and classification towards computer-aided diagnosis of breast lesions in digital X-ray mammograms,” *Computer Methods and Programs in Biomedicine*, vol. 196, 105584, 2020.
- [17] M. M. Rahman, Y. Ghasemi, E. Suley, Y. Zhou, S. Wang *et al.*, “Machine learning based computer aided diagnosis of breast cancer utilizing anthropometric and clinical features,” *IRBM*, vol. 42, pp. 215–226, 2021.
- [18] L. Heck and J. Herzen, “Recent advances in X-ray imaging of breast tissue: From two- to three-dimensional imaging,” *Physica Medica*, vol. 79, no. 11, pp. 69–79, 2020.
- [19] S. Bharati, P. Podder and M. R. H. Mondal, “Artificial neural network based breast cancer screening: A comprehensive review,” *arXiv:2006.01767*, pp. 1–13, 2020.
- [20] V. Mrzljak, E. Markova-Car and Z. Car, “Dataset size-based approach in design of artificial neural network for breast cancer diagnosis,” *World of Health*, no. 9, 2020.
- [21] G. Murtaza, “Deep learning-based breast cancer classification through medical imaging modalities: State of the art and research challenges,” *Artificial Intelligence Review*, vol. 53, no. 3, pp. 1655–1720, 2020.
- [22] V. Chaurasia and S. Pal, “Applications of machine learning techniques to predict diagnostic breast cancer,” *SN Computer Science*, vol. 1, no. 5, 2020.
- [23] M. M. Islam, M. R. Haque, H. Iqbal, M. M. Hasan, M. Hasan and M. N. Kabir, “Breast cancer prediction: A comparative study using machine learning techniques,” *SN Computer Science*, vol. 1, no. 5, pp. 1–14, 2020.

- [24] J. X. Wu, H. C. Liu, P. Y. Chen, C. H. Lin, Y. H. Chou *et al.*, “Enhancement of ARFI-VTI elastography images in order to preliminary rapid screening of benign and malignant breast tumors using multilayer fractional-order machine vision classifier,” *IEEE Access*, vol. 8, pp. 164222–164237, 2020.
- [25] J. X. Wu, P. Y. Chen, C. H. Lin, S. Chen and K. K. Shung, “Breast benign and malignant tumors rapidly screening by ARFI-VTI elastography and random decision forests based classifier,” *IEEE Access*, vol. 8, pp. 54019–54034, 2020.
- [26] Y. Benhammou, B. Achchab, F. Herrera and S. Tabik, “BreakHis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights,” *Neurocomputing*, vol. 375, pp. 9–24, 2020.
- [27] A. H. Osman and H. M. A. Aljahdali, “An effective of ensemble boosting learning method for breast cancer virtual screening using neural network model,” *IEEE Access*, vol. 8, pp. 39165–39174, 2020.
- [28] E. M. Hassib, A. I. El-Desouky, L. M. Labib and E. S. M. El-kenawy, “WOA+BRNN: An imbalanced big data classification framework using whale optimization and deep neural network,” *Soft Computing*, vol. 24, no. 8, pp. 5573–5592, 2020.
- [29] Y. Guan, Z. Yuan, G. Sun and J. Cong, “FPGA-based accelerator for long short-term memory recurrent neural networks,” in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Chiba, Japan, pp. 629–634, 2017.
- [30] Z. Cui, R. Ke, Z. Pu and Y. Wang, “Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values,” *Transportation Research Part C: Emerging Technologies*, vol. 118, no. 5, 102674, 2020.
- [31] M. M. Fouad, A. I. El-Desouky, R. Al-Hajj and E. S. M. El-Kenawy, “Dynamic group-based cooperative optimization algorithm,” *IEEE Access*, vol. 8, pp. 148378–148403, 2020.
- [32] H. Salimi, “Stochastic fractal search: A powerful metaheuristic algorithm,” *Knowledge-Based Systems*, vol. 75, pp. 1–18, 2015.
- [33] E. Yavuz and C. Eyupoglu, “An effective approach for breast cancer diagnosis based on routine blood analysis features,” *Medical & Biological Engineering & Computing*, vol. 58, no. 7, pp. 1583–1601, 2020.
- [34] J. G. Digalakis and K. G. Margaritis, “An experimental study of benchmarking functions for genetic algorithms,” *International Journal of Computer Mathematics*, vol. 79, no. 4, pp. 403–416, 2002.
- [35] M. H. Sulaiman, Z. Mustaffa, M. M. Saari and H. Daniyal, “Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems,” *Engineering Applications of Artificial Intelligence*, vol. 87, 103330, 2020.
- [36] R. Poli, J. Kennedy and T. Blackwell, “Particle swarm optimization: an overview,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [37] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [38] S. Mirjalili, S. M. Mirjalili and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [39] J. H. Holland, “Genetic algorithms,” *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.