Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033042
Article

Tech Science Press

Hybrid Dipper Throated and Grey Wolf Optimization for Feature Selection Applied to Life Benchmark Datasets

Doaa Sami Khafaga[1], El-Sayed M. El-kenawy[2,3], Faten Khalid Karim[1,*], Mostafa Abotaleb[4], Abdelhameed Ibrahim[5], Abdelaziz A. Abdelhamid[6,7] and D. L. Elsheweikh[8]

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia
[2]Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, 35111, Egypt
[3]Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, 35712, Egypt
[4]Department of System Programming, South Ural State University, Chelyabinsk, 454080, Russia
[5]Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt
[6]Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, 11566, Egypt
[7]Department of Computer Science, College of Computing and Information Technology, Shaqra University, 11961, Saudi Arabia
[8]Department of Computer Science, Faculty of Specific Education, Mansoura University, Egypt
*Corresponding Author: Faten Khalid Karim. Email: fkdiaaldin@pnu.edu.sa

Received: 05 June 2022; Accepted: 12 July 2022

Abstract: Selecting the most relevant subset of features from a dataset is a vital step in data mining and machine learning. Each feature in a dataset has 2^n possible subsets, making it challenging to select the optimum collection of features using typical methods. As a result, a new metaheuristics-based feature selection method based on the dipper-throated and grey-wolf optimization (DTO-GW) algorithms has been developed in this research. Instability can result when the selection of features is subject to metaheuristics, which can lead to a wide range of results. Thus, we adopted hybrid optimization in our method of optimizing, which allowed us to better balance exploration and harvesting chores more equitably. We propose utilizing the binary DTO-GW search approach we previously devised for selecting the optimal subset of attributes. In the proposed method, the number of features selected is minimized, while classification accuracy is increased. To test the proposed method's performance against eleven other state-of-the-art approaches, eight datasets from the UCI repository were used, such as binary grey wolf search (bGWO), binary hybrid grey wolf, and particle swarm optimization (bGWO-PSO), bPSO, binary stochastic fractal search (bSFS), binary whale optimization algorithm (bWOA), binary modified grey wolf optimization (bMGWO), binary multiverse optimization (bMVO), binary bowerbird optimization (bSBO), binary hysteresis optimization (bHy), and binary hysteresis optimization (bHWO). The suggested method is superior and successful in handling the problem of feature selection, according to the results of the experiments.

Keywords: Metaheuristics; dipper throated optimization; grey wolf optimization; binary optimizer; feature selection

1 Introduction

One of the first Artificial Intelligence (AI) strategies that have been utilized in solving complex real-world research problems was search algorithms. Uninformed and informed search methods are two types of search strategies [1]. Uninformed search approaches conduct a blind search of the search space without any prior knowledge of how close or far away the optimal answer is from the current solution [2]. Brute-force search, breadth-first search, depth-first search, and bidirectional search are examples of uninformed search algorithms [3,4]. Such search algorithms are comprehensive and infeasible for computationally expensive issues because they are time-consuming. The informed search technique, sometimes referred to as heuristics, employs a heuristics assistant, which is typically a function, to estimate how the final result is near to the optimal solution. The use of such heuristic algorithms can yield an increased speed and efficiency in terms of computing and storage requirements. However, the blind search strategies have become more feasible because of recent considerable developments in computing and data storage systems; there are still a considerable number of problems that they can't solve, specifically those having exponential growth in the search space. Hill climbing and simulated annealing are two examples of early methods of heuristic search algorithms. The main shortage of such algorithms is being application dependent and requiring an adjustment to the heuristic function to be employed in solving different problems. Metaheuristics search algorithms are well-known methods that handle such drawbacks of the heuristic methods. In these methods, the search problem is considered a black box. The heuristic info is deduced using the problem's outcome, which is typically referred to as the fitness or cost function. The algorithm is iterated until it yields results for the objective function to find the best solution. Recently metaheuristics have become one of the leading research areas in the field of AI and have been widely used in science and industrial research problems. The main categories of the metaheuristics include Swarm intelligence, and Evolutionary Algorithms, EA. The Swarm-based search algorithms simulate the local interactions among entities in nature to solve problems collaboratively. Ant Colony is one of the early Swarm-based search approaches in which each ant makes use of her pheromone to help other ants to follow her path in discovering the sources of food. The second category, EA, such as Genetic Algorithm, GA, simulates natural evolutionary processes.

Machine learning, ML, is an area of artificial intelligence that focuses on developing an algorithm that can learn from existing examples. However, building effective learning algorithms is challenging. Metaheuristic optimization techniques are frequently used to overcome such issues. Mainly the construction of an efficient ML algorithm has four challenges. Firstly, the identification of the structure of the learning algorithms and the best learning parameters to achieve the best classification/forecasting accuracy is an optimization problem. Gradient-based approaches have been popular; however, due to the enormous number of parameters involved in the training process, they suffer from local optima stagnation. Consequently, metaheuristic techniques are becoming more reliable. Secondly, the training of the learning algorithm has a number of regulating parameters that should be tuned and optimized during the training phase. The backpropagation learning algorithm, for example, has two parameters that should be tuned during the training phase, including the momentum and learning rate. Estimating optimal values for learning parameters is necessary as it can yield great results in terms of the prediction system's accuracy. This challenge can be handled by applying the metaheuristics search methods in the tuning of the learning parameters. The third challenge includes the formulation of a multi-objective function in which the ML application has more than one objective. The metaheuristics search methods can be utilized in finding the optimal solutions for such multiple objective applications. The fourth challenge is concerned with handling the under/overfitting problems. The over-fitting problem occurs when the learning algorithm is excessively adaptable and learns the noise in the data in addition to the fundamental data pattern. However, the under-fitting happens if the learning algorithm cannot learn the underlying pattern in the data. Discovering the correct balance between the number of samples and data records is another issue that can be handled using the metaheuristics AI search methods. Finally, the fifth challenge is the selection of input features that will be used in the training/testing of the learning algorithm. We may not need to include all of the features in the training/testing phases to achieve high forecasting/classification accuracy. Redundant and correlated features should be filtered or aggregated. Finding an optimal set of features can help in achieving the high accuracy of the prediction system. The goal of this research is to use metaheuristics paradigms to find the best set of relevant features that will result in a high-accuracy prediction system.

2 Literature Review

Generally, the nature-inspired optimizers employ new mathematical components and demonstrate greater performance. Consequently, they attract a lot of attention from researchers to create or enhance the existing optimizers for different ML problems. They can be basically classified according to the underlying mechanics into four categories, including swarm intelligence-based, evolution-based, human-related, and physics-based algorithms [5]. Natural evolution is the inspiration for evolution-based algorithms, which begin with a random population of recommended solutions. The solutions that yield good fitness with the objective function are combined in these algorithms to produce different individuals. Crossover and Mutation are used to create a new population. GA [6] is the most prominent algorithm in this category. Other algorithms include evolution strategy [7], tabu search [8], and genetic programming [9]. Only a few algorithms in the evolution-based category have been developed recently, such as the work done in [10], in which an optimized technique for selecting the significant features for the classification of images of galaxies has been introduced. The biological image matching methods can be thought of as an optimization process that aims to identify the highest degree of similarity between a pattern and a template. Therefore, many optimization algorithms were utilized in that era, such as the work done in [11] in which the chaotic differential search technique was employed to discover the presence of a pattern inside a given image. In [12], and optimized ML model has been established in wind speed predictions in a wind-power producing system. The backtracking search methods have been utilized in optimizing the learning parameters of the learning algorithm. As well as an optimized search paradigm has also been utilized in [13] in implementing a computer-aided diagnosing system for blood cancer. A stochastic search algorithm, Fractal search optimizer, has been employed along with the K-means clustering in the segmentation of blood images for Leukemia classification.

Swarm intelligence algorithms are based on the social behavior of birds, insects, fish, and animals, among other things. Particle Swarm Optimization, PSO, [14], is a prominent approach and is based on the behavior of a flock of birds that flies over the search space to find the ideal site for them. Other algorithms include ant colony optimization [15], monkey optimization [16], honey-bee swarm optimization algorithm [17], and others. Algorithms based on physics are developed by the utilization of the laws of physics, and they include Harmony search [18], simulated annealing [19], etc. Algorithms based on human behavior are mainly based on simulating human behavior. Every human being has a unique method of carrying out tasks that has an impact on their overall performance. League Championship algorithm [20], and Teaching learning-based optimization algorithm [21], others are popular algorithms.

3 Methodology

The proposed methodology is based on a hybrid dipper throated optimization and grey wolf optimization. In this section, the basics of the proposed approach, along with a presentation and discussion of the proposed approach, are explained.

3.1 Dipper Throated Optimization

Dipper throated optimization (DTO) is based on a simulation of the real process of looking for food by tracking the positions and speeds of swimming and flying birds. The following equations are used to update the position and speed of the swimming birds.

$$BL_{nd}\,(t+1) = BL_{best}\,(t) - C_1\,.\,|\,C_2\,.\,BL_{best}\,(t) - BL_{nd}\,(t)| \tag{1}$$

where t is the iteration number, and $BL_{nd}(t)$ and $BL_{best}\,(t)$ are the normal location and best location of the bird, and C_1 and C_2 are adaptive values whose values are changed during the optimization process based on the iteration number and random values. On the other hand, the update of the flying bird's location is performed using the following equation.

$$BL_{nd}\,(t+1) = BL_{nd}\,(t) + BS\,(t+1) \tag{2}$$

$$BS(t+1) = C_3BS(t) + C_4r_1(BL_{best}(t) - BL_{nd}(t)) + C_5r_1(BL_{Gbest} - BL_{nd}(t)) \tag{3}$$

where $BS(t+1)$ is the updated speed of each bird, r_1 is a random number in [0; 1], BL_{Gbest} is the global best location, and C_3 is a weight value, C_4 and C_5 are constants.

3.2 Grey Wolf Optimization

One of the world's apex predators, grey wolves, may be found around the world. It is common for grey wolves to live in packs. On average, the group size is between 5 and 12. Grey wolves have a fascinating social structure with a well-defined hierarchy of authority. Male and female alphas rule the pack. When it comes to key group choices like hunting, resting, and so on, the alpha is in charge. The alpha's commands should be followed by everyone in the group. It has also been noticed that the alpha wolf follows the rest of the pack, which may be seen as a form of democracy. For this reason, they're also known as alphas and leaders because the entire pack must obey them. Interestingly, the alpha doesn't have to be the strongest member but rather the one who is best suited to taking the group's leadership position. As a result, the group's discipline and structure is more crucial than its strength [22–28].

Grey wolves have a two-tiered social structure, with Beta being the lowest. Beta wolves are subordinate wolves tasked with assisting the alpha wolf in making sound judgments and participating in group activities. While the betas are expected to show respect to the alpha, they are also in charge of the pack's lesser ranks. They're both a counselor to the alpha and a group enforcer at the same time. The alpha's henchmen, the betas, are in charge of enforcing his orders across the group and providing feedback to them. Omega is the lowest-ranked grey wolf in the pack. Scapegoat: Omega performs this role. Omega wolves are constantly forced to give in to the more powerful wolves in the pack. They are also the last of the wolves to be permitted to consume meat. There is a possibility that the omegas are not considered significant members of the group, yet losing one of them might lead to internal strife and troubles for the entire group. All of the wolves in the pack are angry and frustrated, and the omegas are letting it all out. Since everyone needs to be satisfied, the omegas play a crucial function in helping to keep things in their proper order.

A wolf is considered inferior if it is not an alpha, beta, or omega wolf (or, according to some references, delta). Alphas and betas must submit to delta wolves, who rule over the omega. This group includes scouts, elders, and hunters. The alphas and betas rely on the hunters for assistance in hunting animals, and the hunters, in turn, provide food for the group. Sentinels are in charge of ensuring and defending the group's security. A scout's primary responsibility is to keep an eye out for any threats to the group's territory and alert the rest of the troop. The Elders of the pack are the wolves who have previously served as alpha or beta. Grey wolves' social structure is fascinating, but their collective hunting strategy is perhaps more fascinating. The grey wolves are surrounding their victim, as we previously stated. The following equations are suggested for mathematical modeling of encircling behavior:

$$\vec{D} = \left| \vec{C} \, . \, \vec{F}_p \left(t \right) - \vec{F} \left(t \right) \right| \tag{4}$$

$$\vec{F} \left(t + 1 \right) = \vec{F}_p \left(t \right) - \vec{A} \, . \, \vec{D} \tag{5}$$

where $\vec{A}$ and $\vec{C}$ are vectors of coefficients, t represents the current iteration, $\vec{F}$ is the grey wolf's position vector, and $\vec{F}_p$ indicates the position vector of the prey. If there is a better solution in each iteration, $\vec{F}$ is updated to the best solution.

$$\vec{a} = 2 - t \left(\frac{2}{Max_{iter}} \right) \tag{6}$$

$$\vec{A} = 2\vec{a} \, . \, \vec{r}_1 - \vec{a} \tag{7}$$

$$\vec{C} = 2\vec{r}_2 \tag{8}$$

Loop counter t, Max_{iter} is the maximum number of iterations, $\vec{r}_1$ and $\vec{r}_2$ are random vectors in [0,1], and $\vec{a}$ is linearly decreasing from 2 to 0 throughout the length of iterations. Consider a two-dimensional position vector and some of the potential neighbors depicted in Fig. 2 as a starting point for exploring the impact of Eqs. (1) and (2). In this illustration, a grey wolf in position (X, Y) can adjust its position in response to the prey's movement (X^*, Y^*). It is possible to go to other locations surrounding the optimal agent by modifying EA and EC vectors' values with respect to their present location. Setting $\vec{A} = (1,0)$ and $\vec{C} = (1, 1)$, for example, will result in (X^* $-X$, Y^*). Grey wolves can travel between the places depicted in Fig. 1 using the random vectors r1 and r2. Eqs. (1) and (2) allow a grey wolf to change its posture in the vicinity of its prey at will. The same method may be used in other dimensions, and the grey wolves will move in around the best answer so far (hyper-spheres/cubes).

Grey wolves have the ability to locate prey and then encircle it. The alpha is generally in charge of the entire hunt. The beta and delta occasionally join in on the action during a hunting expedition. We, on the other hand, have no notion where the prey is located in our simulated 2D search area (optimum). The alpha, beta, and delta wolves all know where to look for prey; therefore, we'll use this assumption to model grey wolf hunting behavior statistically. As a consequence, we'll keep the top three search results and compel the rest of the grey wolves (including the omegas) to recalculate their locations based on them. We came up with the following formulae to do this:

$$\vec{D}_\alpha = \left|\vec{D}_1 * \vec{F}_\alpha - \vec{F}\right|, \vec{D}_\beta = \left|\vec{D}_2 * \vec{F}_\beta - \vec{F}\right|, \vec{D}_\delta = \left|\vec{D}_3 * \vec{F}_\delta - \vec{F}\right| \tag{9}$$

$$\vec{F}_1 = \vec{F}_\alpha - \vec{A}_1 * \vec{D}_\alpha, \vec{F}_2 = \vec{F}_\beta - \vec{A}_2 * \vec{D}_\beta, \vec{F}_3 = \vec{F}_\delta - \vec{A}_3 * \vec{D}_\delta \tag{10}$$

$$\vec{F}(t+1) = \frac{\vec{F}_1 + \vec{F}_2 + \vec{F}_3}{3} \tag{11}$$

A grey wolf (a search agent) is seen in Fig. 1 moving around in a 2D search space according to alpha, beta, and delta. Fig. 1 shows that the ultimate location of a grey wolf (search agent) will be in a random location inside the search area given by the coordinates of alpha, beta, and delta. Prey's location is estimated by alpha, beta and delta wolves; other wolves follow this guess and update their locations around the prey at random.

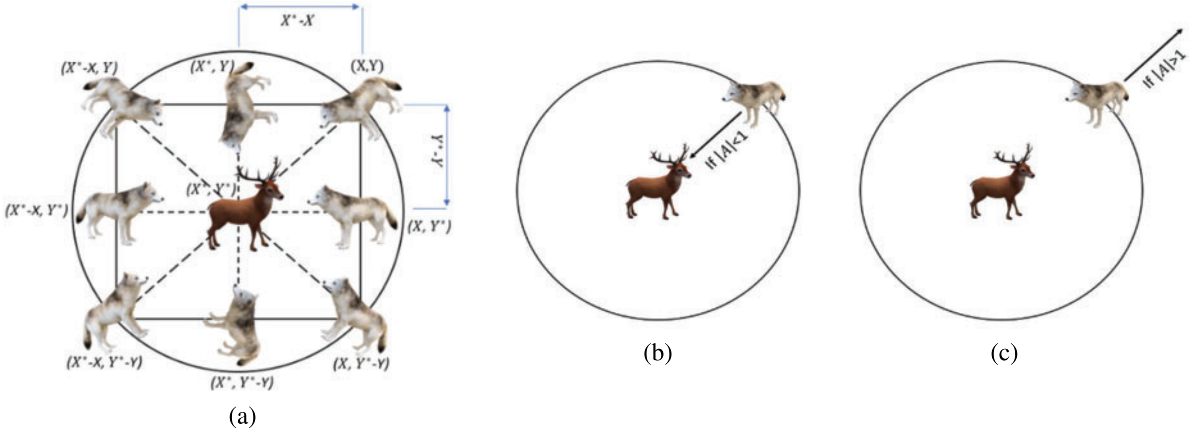

Figure 1: Hunting process: (a) potential locations of grey wolves (b) convergence (attacking the prey), (c) divergence (searching for prey)

3.3 The Proposed DTO-GW Algorithm

The proposed bDTO-GW employs a KNN classifier to make sure that only decency-preserving features are used. Due to its primary goal of maximizing classification accuracy while minimizing the.

Algorithm 1: The proposed DTO-GW algorithm

```
Initialize birds locations $BL_i (i = 1, 2, 3, \ldots, n)$ with size $n$, $BS_i (i = 1, 2, 3, \ldots, n)$,
Fitness function $Fn, f_n, r_1, r_2, r_3, R, C_1, C_2, C_3, C_4, C_5, t = 1$, and max iterations $iter\_max$
Evaluate fitness function $Fn$ for each $BL_i$
Find best bird $BL_{best}$
While t < $iter\_max$ do
    for ($i = 1; i \leq n$) do
        If (R < 0.5) then
            Update Location of the gey wolf agents using:
```

(Continued)

Algorithm 1: Continued

```
            $\vec{D}_\alpha = \left|\vec{D}_1 * \vec{F}_\alpha - \vec{F}\right|, \vec{D}_\beta = \left|\vec{D}_2 * \vec{F}_\beta - \vec{F}\right|, \vec{D}_\delta = \left|\vec{D}_3 * \vec{F}_\delta - \vec{F}\right|$
        else
            Update Speed of the flying bird using:
            $BS(t+1) = C_3 BS(t) + C_4 r_1 (BL_{best}(t) - BL_{nd}(t))$
                $+ C_5 r_1 (BL_{Gbest} - BL_{nd}(t))$
            Update Location of the swimming bird using:
            $BL_{nd}(t+1) = r_1 + z * r_2 + (1-z) * r_3 + BS(t+1)$
        end for
    end for
    Evaluate fitness function $Fn$ for each $\vec{BL}_i$
    Update $R, r_1, r_2, r_3, c, C_1, C_2$
    Find best bird $BL_{best}$
    Set $BL_{Gbest} = BL_{best}$
    Set $t = t + 1$
end while
return $BL_{Gbest}$
```

3.4 The Proposed Feature Selection Method

The proposed bDTO-GW employs a KNN classifier to make sure that only decency-preserving features are used. Due to its primary goal of maximizing classification accuracy while minimizing the number of features picked and the error rate, feature selection methods aim to decrease both these factors [29–33]. As a result, Eq. (1) is employed in the proposed feature selection technique. To maximize the feature assessment norm, the proposed bDTO-GW approach proposes an adaptive search space exploration presented in Eq. (1).

$$Fitness = \alpha \gamma_R (D) + \beta \frac{|C - R|}{|C|} \tag{12}$$

In which the condition attribute set R in relation to the decision D has a classification quality denoted by $\gamma_R (D)$. The total number of features is denoted by C and the number of selected features is referred to as R. Additional parameters that affect the classification accuracy are $\alpha \in [0,1]$ and $\beta = 1 - \alpha$. Based on the error rate and selected feature ratio, Eq. (1) can be converted into a minimization problem. Therefore, the minimization of the error rate can be performed using the following fitness function, where $E_R(D)$ is the error rate.

$$Fitness = \alpha E_R (D) + \beta \frac{|R|}{|C|} \tag{13}$$

To select the best set of features, the resulting best solution is converted into binary 0 or 1. To perform this conversion, the sigmoid function is usually employed as represented by the following equation where S_{best} refers to the best position at iteration t.

$$S_d (t+1) = \begin{cases} 0, & if\, Sigmoid\, (S_{best}) < 0.5 \\ 1, & otherwise \end{cases} \tag{14}$$

4 Experimental Results

The proposed binary feature selection approach was put to the test on eight benchmark datasets from the UCI repository (UCI Feature selection dataset 2017). Various databases have different numbers of attributes and occurrences. There is a complete breakdown of all datasets in Table 1. Because of this, it was utilized to choose the most significant features. The KNN classifier's accuracy is used to choose the best features to use in this approach. K is set to 5 by the KNN classifier based on empirical data. Training, validation, and testing datasets are randomly separated for cross-validation. K-fold cross-validation, which uses K-1 folds for training, validation, and testing, was used to verify the performance of the proposed bDTO-GW method.

Table 1: Life datasets included in the conducted experiments

Dataset	#Features	#Samples	#Classes	Notation
Meta all	62	71	6	D1
Zoo	17	101	7	D2
Parkinsons	22	195	2	D3
HeartEW	13	270	5	D4
Vowel	10	990	2	D5
WDBC	30	569	2	D6
Vehicle	18	846	4	D7
Segment	19	2310	7	D8

4.1 Evaluation Criteria

Table 2 shows the performance evaluation metrics used to assess the results of the proposed approach. These metrics are composed of six criteria, including best fitness, worst fitness, average error, average select size, average fitness size, and standard deviation. The following section presents the evaluation of each of these metrics based on the achieved results using the proposed approach.

Table 2: Performance metrics used in measuring the performance of the achieved results

Metric	Value
Best fitness	$bmin_{i=1}^{M} g_*^i$
Worst fitness	$Max_{i=1}^{M} g_*^i$
Average error	$\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} mse\left(C_i, L_i\right)$
Average select size	$\frac{1}{M}\sum_{M}^{1} g_*^i$
Average fitness size	$\frac{1}{M}\sum_{i=1}^{M}\frac{size\left(g_*^i\right)}{D}$
Standard deviation	$\sqrt{\frac{1}{M-1}\sum\left(g_*^i - Mean\right)^2}$

4.2 Algorithm Configurations

Table 3 shows the suggested approach's setup settings. There are 50 iterations and 10 search agents in this table. Also included is a table that lists the values of the optimization equation variables.

Table 3: The configuration parameters of the proposed bDTO-GW algorithm

Parameter	Value
Num of iterations	50
Num of agents	20
α	[0,2]
R	[0,1]
a, r, b, p	[0,1]

4.3 Evaluation Results

For comparison, there are eleven existing state-of-the-art approaches, including bGWO, PSO, bPSO and SFS, bMGWO and SBO, and the proposed bDTO-GW technique bDTO-GW. We were able to derive the average values for each strategy after 30 iterations of each method for the purposes of comparison. The six performance criteria given above were used to assess the proposed bDTO-GW method.

A strategy that does not remove any data from its source was used to examine the effects of data transformation and feature selection methods. The suggested bDTO-GW and the other techniques tested have an average inaccuracy shown in Table 4. Overall, bDTO-average GW's error of 0.577 is lower than the other techniques' average errors, according to a table. We may investigate the influence of the recommended feature selection approach on the average select size of selected features, as shown in Table 5, through data transformation. As can be seen from this graph, bDTO-GW yields the greatest possible select size.

Table 4: Average error

Dataset	D1	D2	D3	D4	D5	D6	D7	D8	Average
bDTO-GW	**0.600**	**0.537**	**0.609**	**0.690**	**0.538**	**0.553**	**0.643**	**0.446**	**0.577**
bGWO	0.614	0.550	0.622	0.700	0.556	0.568	0.652	0.451	0.589
bGWO_PSO	0.607	0.551	0.620	0.653	0.557	0.556	0.658	0.453	0.582
bPSO	0.604	0.578	0.628	0.684	0.554	0.560	0.659	0.448	0.589
bSFS	0.613	0.557	0.624	0.653	0.543	0.553	0.670	0.450	0.583
bWAO	0.601	0.563	0.619	0.695	0.545	0.562	0.653	0.446	0.586
bMGWO	0.610	0.539	0.611	0.660	0.554	0.585	0.652	0.453	0.583
bMVO	0.601	0.572	0.635	0.682	0.557	0.564	0.651	0.447	0.589
bSBO	0.619	0.558	0.642	0.661	0.532	0.567	0.652	0.450	0.585
bGWO_GA	0.623	0.598	0.628	0.661	0.566	0.560	0.641	0.452	0.591
bFA	0.608	0.569	0.634	0.693	0.544	0.571	0.646	0.449	0.589
bGA	0.601	0.558	0.623	0.683	0.550	0.558	0.658	0.450	0.585

Table 5: Average select size

Dataset	D1	D2	D3	D4	D5	D6	D7	D8
bDTO-GW	**0.744**	**0.539**	**0.809**	**0.931**	**0.659**	**0.675**	**1.045**	**0.879**
bGWO	0.808	0.670	0.903	0.910	0.814	0.831	1.103	0.928
bGWO_PSO	0.803	0.694	0.903	0.945	0.794	0.860	1.003	0.903
bPSO	0.938	0.894	0.911	1.103	0.871	0.935	1.053	0.997
bSFS	0.763	0.717	0.906	0.917	0.695	0.889	1.014	0.928
bWAO	1.033	0.735	0.911	1.089	0.887	1.081	1.178	1.040
bMGWO	0.854	0.657	0.892	0.964	0.712	0.774	0.947	0.924
bMVO	0.908	0.874	0.920	0.931	0.939	0.917	1.140	0.997
bSBO	0.923	0.833	0.836	1.003	0.893	0.974	1.053	1.053
bGWO_GA	0.883	0.827	1.103	0.917	0.903	0.931	1.053	1.053
bFA	0.943	0.891	0.911	0.989	0.873	0.928	1.165	1.040
bGA	0.903	0.794	0.911	0.953	0.851	0.910	1.178	0.984

As an alternative, Table 6 shows an average of the quality of the specified features' quality. As seen in this table, the proposed technique has an overall average fitness of 0.745, which is the best among the various alternatives. Tables 7–9 show the best, worst, and standard deviation of each feature's fitness, respectively, depending on the selected features. These tables show that the suggested feature selection algorithm is superior and more successful than any other method that has been tried before.

Table 6: Average fitness

Dataset	D1	D2	D3	D4	D5	D6	D7	D8	Average
bDTO-GW	**0.628**	**0.549**	**0.745**	**0.767**	**0.573**	**0.716**	**1.255**	**0.726**	**0.745**
bGWO	0.664	0.584	0.793	0.797	0.566	0.731	1.275	0.742	0.769
bGWO_PSO	0.646	0.551	0.620	0.857	0.556	0.556	0.658	0.753	0.649
bPSO	0.650	0.612	0.799	0.781	0.582	0.723	1.282	0.738	0.771
bSFS	0.664	0.557	0.647	0.854	0.566	0.543	0.671	0.750	0.656
bWAO	0.651	0.597	0.791	0.792	0.573	0.725	1.275	0.737	0.768
bMGWO	0.629	0.581	0.626	0.791	0.574	0.585	0.652	0.750	0.648
bMVO	0.651	0.606	0.806	0.779	0.585	0.726	1.274	0.738	0.771
bSBO	0.658	0.558	0.741	0.831	0.592	0.567	0.652	0.950	0.694
bGWO_GA	0.622	0.597	0.728	0.851	0.566	0.560	0.641	0.752	0.665
bFA	0.658	0.603	0.805	0.790	0.573	0.733	1.267	0.740	0.771
bGA	0.651	0.592	0.794	0.780	0.578	0.721	1.281	0.740	0.767

Table 7: Best fitness

Dataset	D1	D2	D3	D4	D5	D6	D7	D8	Average
bDTO-GW	**0.533**	**0.498**	**0.711**	**0.503**	**0.463**	**0.679**	**1.235**	**0.712**	**0.667**
bGWO	0.552	0.515	0.730	0.588	0.478	0.692	1.235	0.715	0.688
bGWO_PSO	0.630	0.540	0.778	0.644	0.509	0.701	1.239	0.732	0.722
bPSO	0.552	0.548	0.720	0.559	0.524	0.688	1.243	0.720	0.694
bSFS	0.613	0.504	0.768	0.638	0.478	0.704	1.271	0.720	0.712
bWAO	0.591	0.523	0.720	0.559	0.493	0.683	1.235	0.711	0.690
bMGWO	0.580	0.548	0.719	0.570	0.570	0.696	1.260	0.718	0.708
bMVO	0.610	0.506	0.740	0.630	0.493	0.701	1.243	0.715	0.705
bSBO	0.610	0.531	0.778	0.630	0.493	0.709	1.241	0.728	0.715
bGWO_GA	0.610	0.582	0.749	0.673	0.554	0.709	1.231	0.732	0.730
bFA	0.591	0.498	0.740	0.644	0.478	0.696	1.231	0.707	0.698
bGA	0.552	0.531	0.720	0.588	0.493	0.692	1.231	0.720	0.691

Table 8: Worst fitness

Dataset	D1	D2	D3	D4	D5	D6	D7	D8	Average
bDTO-GW	**0.727**	**0.649**	**0.853**	**0.913**	**0.644**	**0.750**	**1.338**	**0.763**	**0.830**
bGWO	0.746	0.658	0.836	1.054	0.630	0.800	1.338	0.779	0.855
bGWO_PSO	0.707	0.608	0.797	0.842	0.646	0.744	1.322	0.762	0.804
bPSO	0.765	0.667	0.865	1.012	0.676	0.782	1.338	0.758	0.858
bSFS	0.688	0.691	0.827	0.842	0.661	0.722	1.366	0.762	0.820
bWAO	0.765	0.684	0.865	1.012	0.661	0.765	1.318	0.754	0.853
bMGWO	0.746	0.667	0.874	0.822	0.661	0.838	1.322	0.771	0.838
bMVO	0.727	0.650	0.903	0.941	0.661	0.748	1.338	0.758	0.841
bSBO	0.727	0.633	0.903	0.842	0.646	0.739	1.322	0.758	0.821
bGWO_GA	0.727	0.692	0.845	0.842	0.676	0.739	1.322	0.749	0.824
bFA	0.785	0.701	1.009	0.955	0.646	0.899	1.298	0.766	0.882
bGA	0.727	0.709	0.903	1.026	0.646	0.757	1.338	0.771	0.859

Table 9: Standard deviation fitness

Dataset	D1	D2	D3	D4	D5	D6	D7	D8	Average
bDTO-GW	**0.436**	**0.436**	**0.429**	**0.496**	**0.435**	**0.418**	**0.423**	**0.416**	**0.436**
bGWO	0.455	0.445	0.432	0.519	0.444	0.435	0.428	0.417	0.447
bGWO_PSO	0.438	0.441	0.431	0.484	0.455	0.422	0.437	0.416	0.441
bPSO	0.462	0.436	0.443	0.507	0.446	0.425	0.431	0.412	0.445
bSFS	0.455	0.478	0.451	0.492	0.479	0.430	0.447	0.436	0.459
bWAO	0.449	0.445	0.441	0.510	0.455	0.427	0.428	0.414	0.446

(Continued)

Table 9: Continued

Dataset	D1	D2	D3	D4	D5	D6	D7	D8	Average
bMGWO	0.445	0.441	0.440	0.498	0.438	0.437	0.426	0.414	0.442
bMVO	0.443	0.443	0.443	0.482	0.456	0.418	0.428	0.414	0.441
bSBO	0.450	0.440	0.454	0.502	0.460	0.420	0.443	0.415	0.448
bGWO-GA	0.447	0.449	0.438	0.489	0.451	0.422	0.439	0.409	0.443
bFA	0.453	0.451	0.467	0.486	0.447	0.448	0.424	0.416	0.449
bGA	0.453	0.450	0.448	0.509	0.440	0.421	0.434	0.416	0.446

On the other hand, to clearly show the effectiveness and superiority of the proposed approach, Figs. 2 and 3 visualize the achieved average error and standard deviation of the fitness, respectively. As shown in these figures, the best results are achieved by the proposed approach, which confirms its superiority when compared to the other approaches.

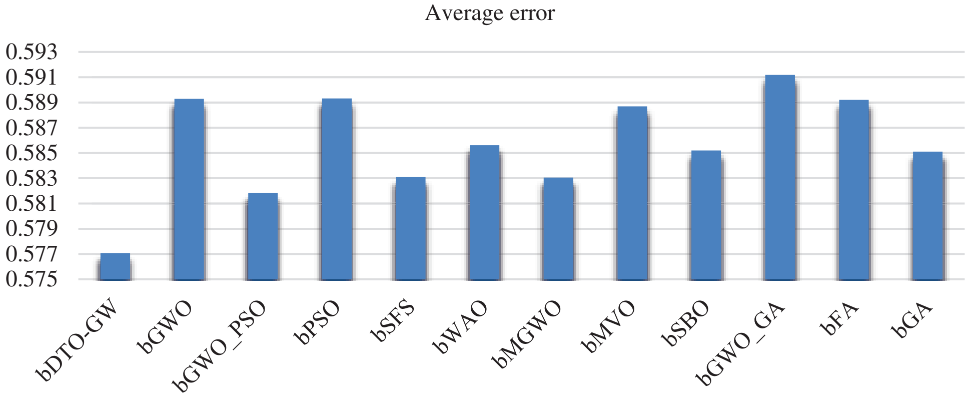

Figure 2: Visualizing the achieved average error using the proposed and competing approaches

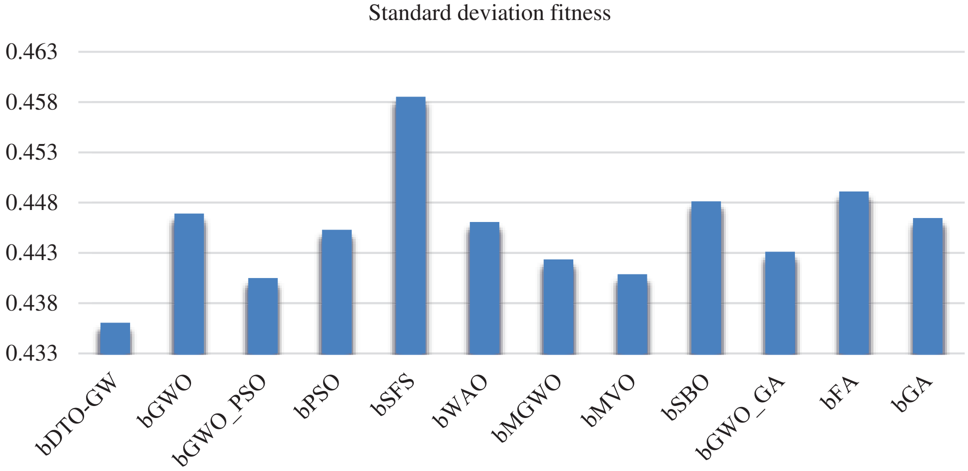

Figure 3: Visualizing the achieved standard deviation of the fitness using the proposed and other competing approaches

5 Conclusions

The binary optimization is introduced in this study by employing dipper-toothed and grey wolf optimizers (bDTO-GW). The bDTO-GW method was used to choose the most essential and non-redundant features from the large datasets. It is necessary to make changes to the standard dataset before selecting the best possible collection of features from the new dataset. In order to test the proposed binary feature selection method, eight benchmark datasets were used. bDTO-GW has been evaluated against eleven other techniques in the literature using six assessment criteria, which include bGWO, bGWO PSO, and other bGWO-related approaches as well as bPSO, bSFS, bWOA, bMGWO, bMVO, and bSBO methods as well as bFA and bGA. The suggested bDTO-GW technique outperforms the previous feature selection methods with an average error of 0.577. A standard deviation fitness of 0.436 was found for the bDTO-GW technique, which beat the competition. Future studies will investigate the possibilities of enhancing accuracy by altering various performance criteria. An extra real-world dataset is an option for testing the technique.

Acknowledgement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

[1] W. Hidayat, F. Susanti and D. Wijaya, "A comparative study of informed and uninformed search algorithm to solve eight-puzzle problem," *Journal of Computer Science*, vol. 17, no. 11, pp. 1147–1156, 2021.

[2] R. Chiong, J. H. Sutanto and W. J. Jap, "A comparative study on informed and uninformed search for intelligent travel planning in borneo island," in *Proc. of the Int. Symp. on Information Technology*, Kuala Lumpur, Malaysia, pp. 1–5, 2008.

[3] S. Pooja, S. Chethan and C. Arjun, "Analyzing uninformed search strategy algorithms in state space search," in *Proc. of the Int. Conf. on Global Trends in Signal Processing, Information Computing and Communication*, Jalgaon, India, pp. 97–102, 2016.

[4] H. Khaled, "Enhancing recursive brute force algorithm with static memory allocation: Solving motif finding problem as a case study," in *Proc. of the 14th Int. Conf. on Computer Engineering and Systems*, Cairo, Egypt, pp. 66–70, 2019.

[5] M. Fouad, A. El-Desouky, R. Al-Hajj and E. -S. M. El-Kenawy, "Dynamic group-based cooperative optimization algorithm," *IEEE Access*, vol. 8, no. 1, pp. 148378–148403, 2020.

[6] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[7] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," in *Adaptation in Natural and Artificial Systems*. Bradford Books, MIT Press, 1992.

[8] K. M. Hosny, M. A. Elaziz, I. M. Selim and M. M. Darwish, "Classification of galaxy color images using quaternion polar complex exponential transform and binary Stochastic Fractal Search," *Astronomy and Computing*, vol. 31, no. 1, pp. 100383, 2020.

[9] M. M. Eid, E. -S. M. El-Kenawy and A. Ibrahim, "A binary sine cosine-modified whale optimization algorithm for feature selection," in *4th National Computing Colleges Conf. (NCCC 2021)*, Taif, Saudi Arabia, IEEE, pp. 1–6, 2021.

[10] L. Gan and H. Duan, “Biological image processing via Chaotic Differential Search and lateral inhibition,” *Optik (Stuttg)*, vol. 125, no. 9, pp. 2070–2075, 2014.

[11] K. G. Dhal, J. Gálvez, S. Ray, A. Das and S. Das, “Acute lymphoblastic leukemia image segmentation driven by stochastic fractal search,” *Multimedia Tools and Applications*, vol. 79, no. 17–18, pp. 12227–12255, 2020.

[12] M. Dorigo, V. Maniezzo and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Managment, and Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[13] R. Zhao and W. Tang, “Monkey algorithm for global numerical optimization,” *Journal of Uncertain Systems*, vol. 2, pp. 165–176, 2008.

[14] D. Karaboga, “An idea based on honey bee swarm for numerical optimization: Technical report - TR06,” in *Technical Report*, Erciyes University, 2005.

[15] S. S. M. Ghoneim, T. A. Farrag, A. A. Rashed, E. -S. M. El-Kenawy and A. Ibrahim, “Adaptive dynamic meta-heuristics for feature selection and classification in diagnostic accuracy of transformer faults,” *IEEE Access*, vol. 9, pp. 78324–78340, 2021.

[16] H. Hassan, A. I. El-Desouky, A. Ibrahim, E. -S. M. El-Kenawy and R. Arnous, “Enhanced QoS-Based model for trust assessment in cloud computing environment,” *IEEE Access*, vol. 8, no. 1, pp. 43752–43763, 2020.

[17] Z. Geem, J. Kim and G. Loganathan, “A new heuristic optimization algorithm: Harmony search,” *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[18] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of Statistical Physics*, vol. 34, no. 5–6, pp. 975–986, 1984.

[19] A. H. Kashan, “League championship algorithm: A new algorithm for numerical function optimization,” in *Int. Conf. of Soft Computing and Pattern Recognition (SoCPaR)*, Malacca, Malaysia, pp. 43–48, 2009.

[20] E. -S. M. El-Kenawy, S. Mirjalili, S. S. M. Ghoneim, M. M. Eid, M. El-Said *et al.,* “Advanced ensemble model for solar radiation forecasting using sine cosine algorithm and Newton’s laws,” *IEEE Access*, vol. 9, pp. 115750–115765, 2021.

[21] A. Salamai, E. -S. M. El-kenawy and A. Ibrahim, “Dynamic voting classifier for risk identification in supply chain 4.0,” *CMC-Computers, Materials & Continua*, vol. 69, no. 3, pp. 3749–3766, 2021.

[22] A. Ibrahim, S. Mirjalili, M. El-Said, S. S. M. Ghoneim, M. Al-Harthi *et al.,* “Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm,” *IEEE Access*, vol. 9, pp. 125787–125804, 2021.

[23] E. -S. M. El-Kenawy, S. Mirjalili, A. Ibrahim, M. Alrahmawy, M. El-Said *et al.,* “Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient COVID-19 X-ray chest image classification,” *IEEE Access*, vol. 9, no. 1, pp. 36019–36037, 2021.

[24] A. Abdelhamid and S. Alotaibi, “Optimized two-level ensemble model for predicting the parameters of metamaterial antenna,” *Computers, Materials & Continua*, vol. 73, no. 1, pp. 917–933, 2022.

[25] A. Abdelhamid and S. R. Alotaibi, “Robust prediction of the bandwidth of metamaterial antenna using deep learning,” *Computers, Materials & Continua*, vol. 72, no. 2, pp. 2305–2321, 2022.

[26] E. -S. M. El-kenawy and M. Eid, “Hybrid gray wolf and particle swarm optimization for feature selection,” *International Journal of Innovative Computing, Information & Control*, vol. 16, no. 1, pp. 831–844, 2020.

[27] D. Sami Khafaga, A. Ali Alhussan, E. M. El-kenawy, A. Ibrahim, H. S. Abd Elkhalik *et al.,* “Improved prediction of metamaterial antenna bandwidth using adaptive optimization of LSTM,” *Computers, Materials & Continua*, vol. 73, no. 1, pp. 865–881, 2022.

[28] E. -S. M. El-Kenawy, S. Mirjalili, F. Alassery, Y. Zhang, M. Eid *et al.,* “Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems,” *IEEE Access*, vol. 10, pp. 40536–40555, 2022.

[29] A. Abdelhamid, E. -S. M. El-kenawy, B. Alotaibi, M. Abdelkader, A. Ibrahim *et al.,* “Robust speech emotion recognition using CNN+LSTM based on stochastic fractal search optimization algorithm,” *IEEE Access*, vol. 10, pp. 49265–49284, 2022.

[30] E. -S. M. El-Kenawy, M. M. Eid, M. Saber and A. Ibrahim, "MbGWO-SFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection," *IEEE Access*, vol. 8, no. 1, pp. 107635–107649, 2020.

[31] D. Sami Khafaga, A. Ali Alhussan, E. M. El-kenawy, A. E. Takieldeen, T. M. Hassan *et al.,* "Meta-heuristics for feature selection and classification in diagnostic breast cancer," *Computers, Materials & Continua*, vol. 73, no. 1, pp. 749–765, 2022.

[32] N. Abdel Samee, E. M. El-Kenawy, G. Atteia, M. M. Jamjoom, A. Ibrahim *et al.,* "Metaheuristic optimization through deep learning classification of COVID-19 in chest x-ray images," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 4193–4210, 2022.

[33] H. Nasser AlEisa, E. M. El-kenawy, A. Ali Alhussan, M. Saber, A. A. Abdelhamid *et al.,* "Transfer learning for chest x-rays diagnosis using dipper throated algorithm," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 2371–2387, 2022.