

A Hybrid Swarm Intelligence Based Feature Selection Algorithm for High Dimensional Datasets

Jomana Yousef Khaseeb^a, Anas Youssef^b, Arabi Keshk^b

^a Applied Computing Dept., Palestine Technical University, Palestine

^b Computer Science Dept., Faculty of Computers and Information, Menoufia University, Egypt
{jumanah.khaseeb@ptuk.edu.ps , anas.youssef@ci.menofia.edu.eg , arabikeshk@yahoo.com }

Abstract — High dimensional datasets expose a critical obstacle in machine learning. Feature selection overcomes this obstacle by eliminating duplicated and unimportant features from the dataset to increase the robustness of learning algorithms. This paper introduces a binary version of a hybrid swarm intelligence approach as a wrapper method for feature selection that gathers between the strengths of both the grey wolf and particle swarm optimizers. This approach is named Improved Binary Grey Wolf Optimization (IBGWO). The original version of this hybrid approach was proposed in the literature with a continuous search space as a high-level hybrid form, which runs the optimizers one after the other. Two different types of transfer functions, named S-Shaped and V-Shaped, are applied in this work to turn continuous data into binary. Nine of high-dimensional small-instance medical datasets are employed to assess the proposed approach. The experimental results demonstrate that IBGWO based on S-Shaped (IBGWO-S) outperforms the binary particle swarm and the binary grey wolf optimizers on six out of nine datasets according to the classification accuracy and fitness values. IBGWO-S selects the fewest features on 100% of the datasets. The results show IBGWO based on V-Shaped (IBGWO-V) outperforms the binary particle swarm and binary grey wolf optimizers on five datasets based on the classification accuracy and fitness values. The results indicate that IBGWO-V outperforms IBGWO-S in terms of all studied evaluation metrics. The results also show that IBGWO-S and IBGWO-V outperform eight meta-heuristics known in the literature in selecting the relevant features with acceptable classification accuracy.

Keywords — *Hybrid Algorithm, Feature Selection, Particle Swarm Optimization, Transfer Function, Grey Wolf Optimization.*

I. INTRODUCTION

Feature Selection (FS) is an essential task for machine learning algorithms. It decreases the dimensionality by removing irrelevant information. FS can be defined as a kind of optimization and search problem as it searches for the most related features to reduce the size of the dataset without any loss of information. There are three searching techniques with FS [1]; random search, exhaustive search, and heuristic search. Random search means creates and selects the features randomly then evaluates each selected subset until finding the best. An exhaustive search assesses all candidate features to detect the best subset. Finally, heuristic search generates the initial feature subset randomly then heuristic searching techniques guide the search strategies toward the best feature subset. The heuristic search has been widely used nowadays for FS to determine the best feature subset in an acceptable time.

From an evaluation point of view, three general models are employed to examine the selected features; filters [2, 3], wrappers [4], and embedded [5]. A feature subset in the filter models is assessed based on the interconnections between the features rather than utilizing any classifier. Otherwise, the wrapper models use the classifiers as assessment indicators. The embedded models are a classifier dependent on which the selected feature subset in the training process is based on a specific classifier.

Meta-heuristic (MH) algorithms have been successfully utilized as a search algorithm with high-dimensional datasets. Swarm Intelligence (SI) algorithms that imitate animals' social behaviors in nature, such as birds, fish, whales, and wolves, are a kind of MHs [6]. They have been used to solve optimization problems and they have also proved their capabilities in the literature with the FS problem [7]. Some examples of SI algorithms, Particle Swarm Optimization (PSO) [8], Ant Colony Optimization (ACO) [9], Gravitational Search Algorithm (GSA) [10], Harris Hawks Optimization (HHO) [11], Grey Wolf Optimizer (GWO) [12], Whale Optimization Algorithm (WOA) [13], and Salp Swarm Algorithm (SSA) [14]. SI algorithms suffer from trapping on local optima when searching in a large search space. So, these algorithms ought to adjust the local and the global search process to obtain the global optima.

Particle Swarm Optimization (PSO) is an example of SI that imitates creatures' social behaviors when they search for their food, such as birds and fish [15, 16]. The movement of the particles in the search area depends on the best value attained by the particle and the fittest value attained by the swarm. Grey Wolf Optimization (GWO) is another example of SI that emulates the hunting technique of the grey wolves [12, 17]. Each wolf follows the leaders until finding the prey.

MHs are adequate for both continuous and binary problems. Two groups of Transfer Functions (TFs) called S-Shaped and V-Shaped are used to transform continuous values into binaries according to [18]. FS is considered as a binary optimization issue since the features can be displayed as a string of zeros and ones [19, 20].

This paper presents a binary form of the hybrid SI approach as a wrapper method for the FS. SI ought to prevent the local optima by balancing between two phases during the searching process; exploration and exploitation. The binary form of a hybrid SI approach introduced in this paper merges the great expertise of both PSO and GWO. In addition to the binary representation that aims to maintain a stability between exploration and exploitation efforts. This approach is termed the Improved Binary Grey Wolf Optimization (IBGWO). The IBGWO benefits from the strong exploration ability of binary PSO (BPSO) to enhance the exploratory behavior of binary GWO (BGWO). BPSO can refine the solutions that are generated by BGWO in each iteration until finding the best solution. This approach is compared with BPSO and BGWO and with eight well-known MHs. The results indicate that IBGWO outperforms other optimizers in classification accuracy, the number of detected features, and fitness values.

This paper is organized into the following sections. The related works are presented in Section 2. Section 3 briefly describes GWO, PSO, and the binary version of optimization algorithms. The proposed approach is described in Section 4. The datasets and the experimental setup are presented in Section 5. Section 6 presents and discusses the experimental results. Finally, the conclusion is summarized in Section 7.

II. Related Works

MHs have been suggested to handle optimization problems such as the FS problem [22]. These algorithms seek to find the global optima by balancing between searching locally, i.e., exploitation, and searching globally, i.e., exploration, to avoid the local optima. Examples of MHs include but not limited to Genetic Algorithm (GA) [23], Honey Bee Mating Optimization (HBMO) [24], Particle Swarm Optimization (PSO) [8, 19], Whale Optimization Algorithm (WOA) [25], Ant Colony Optimization (ACO) [26, 27], Dragonfly Algorithm (DA) [28], Harris Hawks Optimizer (HHO) [11], Ant Lion Optimizer (ALO) [29, 30, 31], Grey Wolf Optimizer (GWO) [12, 20, 32, 33, 34], Bacterial Foraging Optimization (BFO) [35] and many others.

Hybrid MHs have been utilized in optimization problems to improve the searching techniques to avoid local optima. According to [22], hybrid algorithms were employed in two forms; low level and high level. In the first form, one algorithm is mixed and embedded inside the other algorithm. In the second form, one algorithm is employed and run after another algorithm. The first hybrid MH algorithm in the FS area was proposed in [36]. The searching technique was by using local search strategies with the GA algorithm for the FS domain. Jinjie Huang et al. introduced in [37] a hybrid wrapper MH approach based on GA and Mutual Information (MI) [38] to detect the optimal features from the dataset. Mafarja and Abdullah in [3] discussed a hybrid approach of GA and Simulated Annealing (SA) inside the rough set theory. A hybrid algorithm for the FS domain relying on ACO and Artificial Neural Networks (ANNs) was mentioned in [39].

A hybrid algorithm based on PSO with a spiral-shaped technique for the FS domain was presented in [40]. The spiral-shaped mechanism [13] was used in the position's update formula of PSO to enrich the accuracy of the feasible solution that can be selected to avoid a local optimum. Alper Unler et al. presented in [41] a hybrid approach for FS called maximum relevance minimum redundancy PSO (mr2PSO). The hybridization merged the MI as a filter approach with the PSO as a wrapper approach. Mafarja et al. in [42] introduced a hybrid approach that enhanced the exploitation and exploration of WOA by combining it with SA and tournament selection [43] algorithms. A hybrid algorithm that merged the Artificial Bee Colony (ABC) and Differential Evolution (DE) was presented in [44]. A hybrid approach called SSAPSO was proposed in [45]. It utilized the SSA and PSO to enhance the stability between local search and global search.

Rough Set (RS) was hybridized with improved Harmony Search (HS) to resolve FS problem in [46]. HS algorithm imitates the process of musical orchestration, where each musician usually provides the sound of his device by seeking an ideal state of tune [47]. RS was merged with GA in [48] as a hybrid method for medical datasets classification. A hybrid model that mixed GWO with the strong search ability of DE to avoid local optima was adopted in [49]. Shahrzad Saremi in [50] used a combination of Evolutionary Population Dynamics (EPD) with GWO to solve the optimization problem. Also, the Cuckoo Search (CS) was used with GWO to improve the searching mechanism in [51]. A hybrid method of ACO and GA was presented in [52] for text FS. In [53], ACO was also merged with CS for FS in digital mammographic sets of data.

Hybrid optimization approaches that combine GWO and PSO to resolve the FS problem are mentioned in [78, 79]. El-Hasnony et al. in [78] utilized GWO to achieve the three best solutions, then the individuals update their position using PSO equation which is interested in the individual's best location in addition to the three best solutions of a whole swarm. This approach also used the idea of inertia weight to give each best solution a certain inertia weight to increase the control of exploration and exploitation. In [79], the approach proposed a hybrid technique that

combines GWO and PSO. It divided the population into two groups. The first group followed the GWO procedures, while the second group followed the PSO procedures. This hybridization is intended to achieve a balance between exploitation and exploration.

The hybrid algorithms in [21, 54] also benefit from the potentials of both GWO and PSO. These optimization algorithms used hybridization to resolve the problems with continuous values. In [21], the hybrid algorithm was an example of a high-level hybrid form utilized to resolve the economic load dispatch problem. The results indicated a high ability of this algorithm to resolve this kind of problem. The same hybrid approach that uses GWO and PSO was introduced in [55] to reach optimal reactive power dispatch problem solution in the field of electric power networks. A hybrid algorithm was presented in [54] called HPSOGWO. It combined PSO and GWO in a low-level hybrid form to enhance exploitation and exploration ability during the search process. Some test functions had been used to determine the effectiveness of HPSOGWO. The outcomes showed that the hybrid approach surpassed the PSO and GWO in obtaining the optimal solution. The binary version of the HPSOGWO was proposed in [56] called BPSOGWO. It was used as a wrapper method for FS. It exceeded the binary versions of GWO, PSO, GA, and the WOA with SA regarded to classification accuracy, number of selected features, and the computing time.

This work proposes a binary form of a hybrid MH algorithm that depends on the hybrid approach defined in [21]. This binary approach is called the IBGWO algorithm. According to [18], the binary version of the MHs balances between exploitation and exploration to avoid the local optima. Two types of TFs were used in this study to convert the solutions into binary format, namely S2 and V2, as will show later. The experimental results show that IBGWO selects the most relevant features with acceptable classification accuracy. The suggested approach also demonstrates a good performance compared with several related algorithms mentioned in the literary works in terms of different evaluation metrics used in this paper.

According to the No-Free-Lunch (NFL) theorem [57], no single algorithm is ideal to handle all optimization problems. That means, not only one optimization algorithm can solve FS problems for all datasets. For this reason, there is a demand to find new optimization algorithms to solve these problems.

III. BACKGROUND

A. Particle Swarm Optimization PSO

It was created by Kennedy and Eberhart [58]. It simulates the social interactions between the birds when searching for their food [58, 59]. PSO employs a group of searching individuals called particles to search for the best solution within the search area. Each particle has its position that represents a candidate solution, and its velocity that used to adjust the speed and the direction of the particle. PSO initializes the searching process by generating random solutions, then in each iteration, each particle modifies its position under the orientation of the particle's best position namely, $pbest$, and the swarm's best position namely, $gbest$. Such steps are repeated until the stopping criteria are attained, such as detecting the best solution or reaching the maximum iteration [59].

The new velocity of the particle, vel^{new} depending on $pbest$ and $gbest$ solutions as shown in equation (1).

$$vel^{new} = w * vel^{old} + c_1 * rd_1 * (pbest - pos^{old}) + c_2 * rd_2 * (gbest - pos^{old}) \quad (1)$$

where vel^{old} is an old velocity, pos^{old} is a particle's old position. The Numbers rd_1 and rd_2 are random numbers between 0 and 1. The acceleration factors, c_1 and c_2 , refer to a particle's confidence on itself and its neighbors, respectively. These parameters lead each particle towards the positions of $pbest$ and $gbest$. In the literary works, c_1 and c_2 are equal to 2 as a generally accepted setting for most of the problems [15, 41, 60, 61, 62]. Inertia weight w is utilized to control the exploitation and exploration aspects of the particles. The inertia weight decreased linearly or non-linearly according to the equations presented in [63, 64, 65].

The new position, pos^{new} , is calculated using the following equation:

$$pos^{new} = vel^{new} + pos^{old} \quad (2)$$

B. Grey Wolf Optimization GWO

It is SI algorithm which was created by Mirjalilia et al. to resolve optimization problems [12]. It simulates the grey wolves' hunting process, from the searching strategy until surrounding the prey, then attacking it. The hunting process of the grey wolves depends on the social hierarchy structure of these creatures, where all wolves follow the

leaders who have the strength and good hunting ability. As shown in Figure 1, the social hierarchy structure is categorized into four groups of wolves: alpha, beta, delta, and omega.

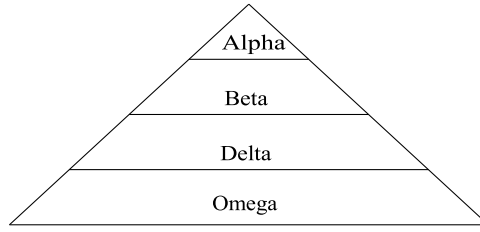


Fig. 1. Hierarchical Structure of Grey Wolves

Alpha α wolf is a leader and a decision-maker. Beta β is alpha's assistant in leading the group. Delta δ obeys the orders from alpha and beta. The remaining wolves are called omega ω which follow other wolves in their movement.

The grey wolves that represent the solutions in a search region are initialized randomly. The solutions are assessed, and the three best wolves are determined. The remaining wolves, ω , update their locations depending on the best three solutions. The process, from solutions evaluation to select the three optimal solutions until updating positions, are iterated until α has been obtained. The descriptions and the mathematical models that represent the hunting process are discussed in detail as follows:

i. Surrounding the prey

The wolves surround the prey at the beginning of the hunting mechanism. The equations that inspire the encircling activity are as follow:

$$Dis = |C \cdot Pos_P(i) - Pos(i)| \tag{3}$$

$$Pos_P(i+1) = Pos_P(i) - A \cdot Dis \tag{4}$$

where Dis is the distance from each wolf's location to the prey, i and $i+1$ refer the current and next iterations, respectively. Pos refers to the position of the wolf. Pos_P refers to the prey's position. A and C are calculated by the following equations.

$$A = 2a \cdot rd1 - a \tag{5}$$

$$C = 2 \cdot rd2 \tag{6}$$

where $rd1$ and $rd2$ are random values within the interval $[0,1]$. Vector a decreases linearly during the iterations from 2 to 0 based on the equation (14).

ii. Hunting prey

Each wolf relies on the location of the three optimal wolves to change its position. This process is formulated by the following equations:

$$Pos_1 = |Pos_\alpha - A_1 \cdot Dis_\alpha| \tag{7}$$

$$Pos_2 = |Pos_\beta - A_2 \cdot Dis_\beta| \tag{8}$$

$$Pos_3 = |Pos_\delta - A_3 \cdot Dis_\delta| \tag{9}$$

$$Pos_{(i+1)} = \frac{Pos_1 + Pos_2 + Pos_3}{3} \tag{10}$$

$Pos_{(i+1)}$ is the new position of ω wolves. The new position depends on Pos_1 , Pos_2 , and Pos_3 based on the distances Dis between each wolf's position and the three optimal solutions' position Pos_α , Pos_β , and Pos_δ . The distances are calculated by the following equations:

$$Dis_\alpha = |C1 \cdot Pos_\alpha - Pos| \tag{11}$$

$$Dis_\beta = |C2 \cdot Pos_\beta - Pos| \tag{12}$$

$$Dis_\delta = |C3 \cdot Pos_\delta - Pos| \tag{13}$$

$C1$, $C2$, and $C3$ are calculated based on equation (6).

By performing the previous equations, the new position of each wolf would be in any location around the prey that is simulated by α , β , and δ .

iii. Attacking prey

Attacking the prey can be simulated by the following equation:

$$a = 2 \left(1 - \frac{n}{N} \right) \quad (14)$$

where a is a linear decreased vector that reduces from 2 to 0 during the iterations to tune the exploitation and the exploration aspects of GWO. Variables n and N are the current and the overall iterations of the algorithm, respectively.

C. Binary Version of Optimization Algorithms

FS can indeed be defined as a binary issue since the solutions can be converted to a binary set of 0's and 1's, where 0 indicates the not selected feature while 1 indicates the selected one. According to Mirjalili and Lewis [18], there are two types of TFs that are used to convert continuous solutions into binaries. These functions called S-Shaped and V-Shaped TFs. TFs produce a probability to transform the binary solution from 0 to 1 and conversely. Table 1 illustrates the mathematical formulas of TFs.

Table 1: V-Shaped and S-Shaped TFs [18]

V-Shaped TFs		
1	V1	$T(x) = \left \operatorname{erf} \left(\frac{\sqrt{\pi}}{2} x \right) \right $
2	V2	$T(x) = \tanh(x) $
3	V3	$T(x) = \left \frac{x}{\sqrt{1+x^2}} \right $
4	V4	$T(x) = \left \frac{2}{\pi} \arctan \left(\frac{\pi}{2} x \right) \right $
S-Shaped TFs		
1	S1	$T(x) = 1/(1 + e^{-2x})$
2	S2	$T(x) = 1/(1 + e^{-x})$
3	S3	$T(x) = 1/\left(1 + e^{-\frac{x}{2}}\right)$
4	S4	$T(x) = 1/\left(1 + e^{-\frac{x}{3}}\right)$

The binary solution can be updated using equation (15) that was presented by Kennedy and Eberhart [66]. They concentrated on S2-TF that called the sigmoid function.

$$x_{(i+1)} = \begin{cases} 1, & TF(v_{(i+1)}) > rd \\ 0, & \text{Otherwise} \end{cases} \quad (15)$$

where $x_{(i+1)}$ is the binary value at iteration $i+1$. $TF(v_{(i+1)})$ is the probability of any value $v_{(i+1)}$ that can be retrieved by TF mathematical formula. The number rd is any random value between 0 and 1.

According to Rashedi et al. [67], V2-TF which is called hyperbolic tangent is utilized to update the binary solution according to the following equation:

$$x_{(i+1)} = \begin{cases} \operatorname{comp}(x_{(i)}), & TF(v_{(i+1)}) > rd \\ x_{(i)}, & \text{Otherwise} \end{cases} \quad (16)$$

where $x_{(i)}$ and $x_{(i+1)}$ are the binary solutions at iterations i and $i+1$, respectively. The function $\operatorname{comp}(\cdot)$ is the complement of any binary solution. $TF(v_{(i+1)})$ is the probability of any value $v_{(i+1)}$ that can be retrieved by TF mathematical formula. The number rd is any random value between 0 and 1.

IV. The proposed approach

Through this paper, a binary version of a hybrid MH algorithm, IBGWO, is utilized as a wrapper method for the FS. This approach aims to look for the appropriate features from the dataset that convenient for the classification task. MH algorithms have the challenge of achieving a global optimum solution. The proposed approach would have to improve the adjustment between the exploitation and exploration aspects to overcome this challenge, thus prevents the local optima. A binary form of both PSO and GWO algorithms are used to design a high-level hybrid approach that interests from the exploration and exploitation ability of both PSO and GWO, respectively. This work utilizes S2 and V2 TFs to modify continuous solutions into binary ones.

The objective function that assesses the solutions is measured using equation (17). This function depends on the size of the picked attributes and the classification error value. The fittest solution is the one that has the lowest fitness value, which means the smallest number of selected attributes and the minimum classification error value.

$$Fitness\ Value = m\ Error + n\ \frac{|y|}{|Y|} \quad (17)$$

where *Error* is a classification error. $|y|$ is the size of selected attributes and $|Y|$ is the overall set of attributes. The two parameters m and n indicate the importance of the classifier performance and the number of picked attributes. m is any value within the interval $[0,1]$ and $n = (1-m)$ as mentioned in [34, 42].

According to the GWO algorithm, the wolves modify their positions depending on the optimal solutions α , β , and δ . These solutions may not be the best, especially in the high dimensional set of data. So, the algorithm may trap in a local optimum [51]. IBGWO combines both BPSO and BGWO to enhance the global search capability when solving the FS problem in high dimensional datasets. This approach is proposed in more detail in the following subsection.

A. Improved Binary Grey Wolf Optimization Algorithm (IBGWO)

In this approach, the BPSO algorithm is used to enhance the exploration aspect of BGWO. As mentioned earlier, PSO has a strong exploration ability while GWO has a strong exploitation ability. The binary form of a hybrid approach can protect BGWO from dropping into a local optimum by depending on the BPSO's superior exploration ability. This approach is called Improved Binary Grey Wolf Optimization Algorithm (IBGWO).

The original model of the hybrid algorithm that combined GWO and PSO, called GWO-PSO [21]. The individuals moved continuously in the search region. In this approach, the individuals move in a binary search space. In BGWO [34], the wolves' steps toward α , β , and δ are calculated by equation (7), equation (8), equation (9), and the new position of each wolf is measured using equation (10). The new position is binarized using equation (15). According to BPSO in [66], the particle's new velocity that is calculated by equation (1) is substituted in equation (15) to binarize the new position.

Figure 2 demonstrates the proposed approach. The initial population for the BGWO algorithm is generated randomly. BGWO improves the solutions then passes them to the BPSO algorithm. The BPSO also improves the solutions again and returns them to the BGWO algorithm. These processes will be iterated until stopping criteria are fulfilled. Figure 3 shows the pseudo-code of the hybrid algorithm. In lines 1-6, the population size is determined, the solutions are generated randomly, and all the parameters are initialized. Then the fitness value of each solution is calculated, and the three best solutions are determined as shown in lines 9 and 10. Each wolf updates its position according to the BGWO equations as illustrated during lines 12-17. The new fitness values are calculated in line 18 and *gbest* is identified in line 19. The new velocities and locations are calculated according to the BPSO equations as shown in lines 20-24. These steps are repeated until the best solution is found or the maximum number of iterations is reached which is determined by the main while loop between lines 8 and 26.

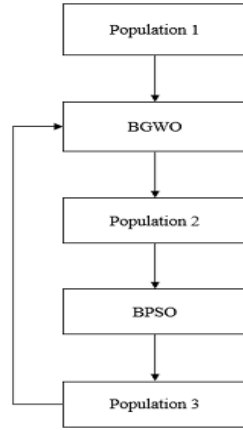


Fig. 2. Improved Binary Grey Wolf Optimization Algorithm (IBGWO)

```

1: Start
2: Set number of iterations  $T$ 
3: Set population size  $P$ 
4: Initialize the random positions and velocities of the grey wolves
5: Initialize BGWO parameters  $a$ ,  $A$ ,  $C$  based on Eqs. (5), (6) and (14)
6: Initialize BPSO parameters  $c1$ ,  $c2$ , and  $w$ 
7:  $t \leftarrow 1$ 
8: while  $t < T$  do
9:   Calculate the fitness of each wolf
10:  Find three best solutions  $Pos_{\alpha}$ ,  $Pos_{\beta}$  and  $Pos_{\delta}$ 
11:   $p \leftarrow 1$ 
12:  for  $p < P$  do
13:    Find  $Pos_{s1}$ ,  $Pos_{s2}$ , and  $Pos_{s3}$  according to Eqs. (7), (8) and (9)
14:    Calculate the new position of each wolf based on Eq. (10)
15:    Convert the new position into binary using S2 or V2 TFs according to Eq. (15).
16:    Update  $A$ ,  $C$ ,  $a$  according to Eqs. (5),(6) and (14)
17:  end for
18:  Calculate the new fitness of each wolf
19:  Find global best  $gbest$ 
20:  for  $p < P$  do
21:    Find personal best  $pbest$ 
22:    Calculate the new velocity for each particle based on PSO using Eq. (1)
23:    Find the new binary position based on Eq. (15) after substituting the new velocity in
    S2 or V2 TF
24:  end for
25:   $t++$ 
26: end while
27: Return the best solution
28: End
  
```

Fig. 3. Pseudocode of IBGWO Algorithm

V. The Datasets and the Experimental Setup

A. The Datasets

The experiments are evaluated using nine of the high-dimensional medical datasets that were used in [68]. Table 2 provides a list of the datasets and their characteristics, such as number of instances, number of features, and number of classes. These sets of data are a kind of high-dimensional small instance datasets that contain thousands or millions of attributes with a small set of instances. As mentioned in [69], dealing with these types of datasets is a big obstacle since a small set of observations is not sufficient to train the learning model. Besides, the large set of features increases the search area and the computational complexity.

Table 2: High-dimensional small instance datasets

Data set	No. of instances	No. of features	No. of classes
11_Tumors	174	12533	11
14_Tumors	308	15009	26
Brain_Tumor1	90	5920	5
Brain_Tumor2	50	10367	4
DLBCL	77	5469	2
Leukemia1	72	5327	3
Leukemia2	72	11225	3
Prostate_Tumor	102	10509	2
SRBCT	83	2308	4

B. Experimental Setup

According to a cross-validation manner, samples in the datasets are divided randomly into 80% training and 20% testing subsets. The implementations were done using MATLAB 2013 and the algorithms were run 20 times on Intel Core i5, 2.2 GHz CPU and 4GB RAM. The results are recorded upon 100 iterations with population size equal 10. In the fitness equation, parameters m and n are assigned to 0.99 and 0.01, respectively, to give the classification accuracy more importance than the number of selected features. These settings are tuned after several experiments, in addition to some earlier empirical studies.

C. Evaluation Metrics

The evaluation metrics that are used to assess the optimizers are: average classification accuracy, average FS size, average fitness value, average running time, the Statistical standard deviation (Std), in addition to the F-Test of each algorithm to determine its average rank.

The average classification accuracy is defined by the following equation

$$Avg\ Classification\ Accuracy = \frac{1}{R} \sum_{j=1}^R \frac{1}{T} \sum_{i=1}^T (P_i == A_i) \quad (18)$$

where R is the overall runs, T is the instances number in the dataset, P_i and A_i are the predictive and actual class, respectively.

The average FS size is defined by the following equation

$$Avg\ Feature\ Selection\ Size = \frac{1}{R} \sum_{i=1}^R \frac{f_i}{F} \quad (19)$$

where R is the overall runs, f_i is the best number of features at i^{th} iteration, and F is the overall number of features.

The average fitness value is by the following equation

$$Avg\ Fitness\ Value = \frac{1}{R} \sum_{i=1}^R Fit_i \quad (20)$$

where R is the overall runs, Fit_i is the best fitness value at i^{th} iteration.

The, average running time is defined by the following equation

$$Avg\ Running\ Time = \frac{1}{R} \sum_{i=1}^R RunTime_i \quad (21)$$

where R is the overall runs, $RunTime_i$ is the run time in millisecond at i^{th} iteration.

Finally, Std is utilized to reflect the stability and robustness of each optimizer. A low standard deviation indicates that the values tend to be close to the mean. The Std is defined by the following equation

$$Std = \sqrt{\frac{1}{R} \sum (optimal_i - avg)^2} \quad (22)$$

where R is the number of times to run the optimization algorithm, $optimal_i$ is the optimal solution resulted from iteration i , and avg is the average of solutions acquired from running an optimization algorithm.

VI. Experimental Results and Discussion

Several comparisons are conducted to achieve an overall view of the positive and negative aspects of the binary hybrid approach (IBGWO). The S-Shaped and V-Shaped TFs are employed to transform continuous values into binaries. First, IBGWO approach is compared with the native version of BGWO and BPSO based on the S-Shaped TF. Then, IBGWO approach is compared with original version of BGWO and BPSO based on the V-Shaped TF. After that, the comparisons between the best approaches among S-Shaped and V-shaped are done. Finally, a comparative study between the best approaches among S-Shaped and V-Shaped with eight well-known MH algorithms is done.

The results compared the novel hybrid approach with the original version of BPSO and BGWO in addition to the comparisons with eight well-known MH algorithms used in FS domain, such as Binary Gravitational Search Algorithm (BGSA) [67], Binary Ant Lion Optimization (BALO) [30], Binary Bat Algorithm (BBA) [70], Binary Salp Swarm Algorithm (BSSA) [71], Binary Dragonfly Algorithm (BDA) [72], Binary Whale Optimization Algorithm (BWOA) [73], Binary Harris Hawk Optimization (BHHO) [74] and Binary Teaching Learning Based Optimization (TLBO) [75]. The classification performance was measured by using K-Nearest Neighbors (KNN) classifier [76, 77]. The best results in each table are written in bold font.

A. Comparisons between IBGWO, BGWO and BPSO based on S-Shaped TF

Tables 3, 4, 5, and 6 illustrate the results of the comparisons between IBGWO, BGWO, and BPSO based on S-Shaped TF according to the classification accuracy results, number of picked features, fitness results, and running time. Table 3 shows that IBGWO-S attained the best accuracy on six datasets, while BPSO-S achieved the highest accuracy on three datasets. BGWO-S was unable to compete other approaches with classification accuracy. According to F-Test, IBGWO-S attained the best accuracy rank.

By analyzing the results on table 4, IBGWO-S obtained the best results based on selecting the fewest features on all datasets. BGWO-S and BPSO-S couldn't compete the proposed approach on picking the relevant features. As shown by F-Test, IBGWO-S achieved the best rank and surpassed BGWO-S and BPSO-S by attaining the minimum set of features.

The outcomes according to the fitness values are provided in Table 5. The findings showed IBGWO-S obtained the best optimizer on six datasets. In the case of BPSO, it gained the best findings on three datasets. As F-Test showed, IBGWO-S was the best approach followed by BPSO.

Table 6 illustrates the outcomes based on the average running time. BPSO-S recorded the best running time followed by BGWO-S then IBGWO-S. The results conclude that IBGWO-S was the best according to the fitness values that integrate the minimum features with the best classification accuracy. In other words, IBGWO-S is the best S-shaped TF approach.

Table 3: Classification accuracy results obtained from various approaches based on S-Shaped TF

Dataset	Metric	BGWO-S	BPSO-S	IBGWO-S
11_Tumors	Avg	0.7714	0.8055	0.8257
	Std	0.0000	0.0114	0.0137
14_Tumors	Avg	0.5999	0.5996	0.6842
	Std	0.0076	0.0069	0.0160
Brain_Tumor1	Avg	0.7627	0.9046	0.7741
	Std	0.0107	0.0244	0.0096
Brain_Tumor2	Avg	0.6067	0.5067	0.8000
	Std	0.0254	0.0254	0.0000
DLBCL	Avg	0.8750	1.0000	0.9396
	Std	0.0000	0.0000	0.0114
Leukemia1	Avg	0.8644	0.9244	0.9400
	Std	0.0122	0.0230	0.0203
Leukemia2	Avg	0.8000	0.7333	0.8644
	Std	0.0000	0.0000	0.0410
Prostate_Tumor	Avg	0.8984	0.9524	0.9349
	Std	0.0165	0.0000	0.0233
SRBCT	Avg	0.9412	0.8235	0.9980
	Std	0.0000	0.0000	0.0107
Mean Rank	F-Test	4.89	4.22	3.11
Overall Rank		3	2	1

Table 4: Number of selected features obtained from various approaches based on S-Shaped TF

Dataset	Metric	BGWO-S	BPSO-S	IBGWO-S
11_Tumors	Avg	7736.30	6206.57	2037.10
	Std	486.93	49.91	198.06
14_Tumors	Avg	9650.33	7460.10	3338.60
	Std	414.84	51.47	421.20
Brain_Tumor1	Avg	3435.23	2825.03	801.90
	Std	320.02	39.74	140.27
Brain_Tumor2	Avg	5192.90	4927.73	1293.30
	Std	316.01	42.69	77.41
DLBCL	Avg	2674.70	2543.70	650.73
	Std	24.28	15.19	31.95
Leukemia1	Avg	3271.20	2559.83	750.50
	Std	235.28	35.38	38.15
Leukemia2	Avg	5619.67	5391.97	1525.97
	Std	235.90	27.83	144.39
Prostate_Tumor	Avg	6408.63	5105.10	1540.33
	Std	483.60	32.65	199.12
SRBCT	Avg	1262.37	1042.73	357.50
	Std	134.60	9.65	67.72
Mean Rank	F-Test	6.00	5.00	2.00
Overall Rank		3	2	1

Table 5: Fitness results obtained from various approaches based on S-Shaped TF

Dataset	Metric	BGWO-S	BPSO-S	IBGWO-S
11_Tumors	Avg	0.2325	0.1975	0.1742
	Std	0.0004	0.0113	0.0136
14_Tumors	Avg	0.4026	0.4014	0.3149
	Std	0.0075	0.0068	0.0159
Brain_Tumor1	Avg	0.2407	0.0992	0.2250
	Std	0.0105	0.0241	0.0094
Brain_Tumor2	Avg	0.3944	0.4932	0.1992
	Std	0.0248	0.0251	0.0001
DLBCL	Avg	0.1286	0.0047	0.0610
	Std	0.0000	0.0000	0.0113
Leukemia1	Avg	0.1403	0.0796	0.0608
	Std	0.0121	0.0228	0.0201
Leukemia2	Avg	0.2030	0.2688	0.1356
	Std	0.0002	0.0000	0.0406
Prostate_Tumor	Avg	0.1067	0.0520	0.0659
	Std	0.0158	0.0000	0.0231
SRBCT	Avg	0.0637	0.1792	0.0035
	Std	0.0006	0.0000	0.0106
Mean Rank	F-Test	5.00	4.22	3.11
Overall Rank		3	2	1

Table 6: Running time in millisecond obtained from various approaches based on S-Shaped TF

Dataset	Metric	BGWO-S	BPSO-S	IBGWO-S
11_Tumors	Avg	177.3157	136.3651	347.4332
14_Tumors	Avg	747.0982	440.8359	1338.7236
Brain_Tumor1	Avg	25.4853	19.7345	51.2097
Brain_Tumor2	Avg	22.8867	18.1753	51.3258
DLBCL	Avg	19.6718	15.7296	47.191
Leukemia1	Avg	18.5404	14.4966	40.8158
Leukemia2	Avg	36.1515	27.6931	97.8216
Prostate_Tumor	Avg	57.893	40.6904	134.5022
SRBCT	Avg	9.2276	7.9226	17.877

B. Comparisons between IBGWO, BGWO and BPSO based on V-Shaped TF

This section Clarifies the outcomes of the comparisons based on V-Shaped TF based on the assessment metrics. According to table 7, IBGWO-V got the most accurate results on five datasets followed by BGWO-V on four datasets. According to the F-Test, IBGWO-V yielded the best rank amongst all approaches.

Table 8 illustrates that BGWO-V attained the best performance on eight datasets followed by IBGWO-V which obtained the best results only on one dataset. However, BGWO-V has rated the best approach based on the F-Test. Table 9 reports that IBGWO-V outperformed other approaches on five datasets based on the fitness values, while BGWO-V obtained the best fitness values on four datasets. These results are compatible with the overall ranks where IBGWO-V achieved the highest rank followed by BGWO-V.

Table 10 includes the average running time. It is observed that BGWO-V had the best running time followed by BPSO-V, and IBGWO-V, respectively. Although IBGWO-V is a hybrid algorithm, there is no big difference between it and the native BPSO and BGWO in the running time that is calculated in milliseconds.

Overall, IBGWO-S is best S-Shaped approach, while IBGWO-V is the best V-Shaped approach. The next section presents the comparisons between IBGWO-V and IBGWO-S.

Table 7: Classification accuracy results obtained from various approaches based on V-Shaped TF

Dataset	Metric	BGWO-V	BPSO-V	IBGWO-V
11_Tumors	Avg	0.9018	0.8007	0.9208
	Std	0.0215	0.0164	0.0229
14_Tumors	Avg	0.7587	0.5972	0.7018
	Std	0.0349	0.0133	0.0165
Brain_Tumor1	Avg	0.9043	0.8941	0.9537
	Std	0.0510	0.0216	0.0329
Brain_Tumor2	Avg	0.9867	0.5067	0.9700
	Std	0.0346	0.0254	0.0466
DLBCL	Avg	0.9958	1.0000	1.0000
	Std	0.0159	0.0000	0.0000
Leukemia1	Avg	1.0000	0.8956	0.9956
	Std	0.0000	0.0336	0.0169
Leukemia2	Avg	0.9978	0.7356	1.0000
	Std	0.0122	0.0213	0.0000
Prostate_Tumor	Avg	0.9968	0.9429	0.9984
	Std	0.0121	0.0194	0.0087
SRBCT	Avg	0.9980	0.8235	0.9961
	Std	0.0107	0.0000	0.0149
Mean Rank	F-Test	2.89	5.11	2.61
Overall Rank		2	3	1

Table 8: Number of selected features obtained from various approaches based on V-Shaped TF

Dataset	Metric	BGWO-V	BPSO-V	IBGWO-V
11_Tumors	Avg	341.40	6127.23	1891.53
	Std	123.34	74.03	539.54
14_Tumors	Avg	1032.70	7393.97	4004.87
	Std	358.13	75.31	742.30
Brain_Tumor1	Avg	49.33	2771.63	125.30
	Std	17.25	40.03	58.79
Brain_Tumor2	Avg	81.47	4879.90	219.10
	Std	37.36	50.19	99.82
DLBCL	Avg	52.90	2514.03	113.00
	Std	17.59	34.95	57.84
Leukemia1	Avg	90.43	2490.97	252.53
	Std	23.03	39.64	122.14
Leukemia2	Avg	92.17	5337.63	5.27
	Std	24.52	38.60	2.69
Prostate_Tumor	Avg	138.60	5042.40	977.10
	Std	38.92	57.76	487.49
SRBCT	Avg	38.47	1019.10	222.73
	Std	12.20	14.34	101.91
Mean Rank	F-Test	2.00	6.00	2.78
Overall Rank		1	3	2

Table 9: Fitness results obtained from various approaches based on V-Shaped TF

Dataset	Metric	BGWO-V	BPSO-V	IBGWO-V
11_Tumors	Avg	0.0975	0.2022	0.0799
	Std	0.0213	0.0162	0.0228
14_Tumors	Avg	0.2395	0.4037	0.2979
	Std	0.0346	0.0132	0.0166
Brain_Tumor1	Avg	0.0948	0.1095	0.0460
	Std	0.0505	0.0214	0.0326
Brain_Tumor2	Avg	0.0133	0.4931	0.0299
	Std	0.0342	0.0251	0.0462
DLBCL	Avg	0.0042	0.0046	0.0002
	Std	0.0157	0.0001	0.0001
Leukemia1	Avg	0.0002	0.1081	0.0049
	Std	0.0000	0.0332	0.0167
Leukemia2	Avg	0.0023	0.2666	0.0000
	Std	0.0120	0.0211	0.0000
Prostate_Tumor	Avg	0.0033	0.0614	0.0025
	Std	0.0120	0.0191	0.0087
SRBCT	Avg	0.0021	0.1791	0.0048
	Std	0.0106	0.0001	0.0149
Mean Rank	F-Test	2.56	5.33	2.50
Overall Rank		2	3	1

Table 10: Running time in millisecond obtained from various approaches based on V-Shaped TF

Dataset	Metric	BGWO-V	BPSO-V	IBGWO-V
11_Tumors	Avg	57.1855	296.4533	242.8638
14_Tumors	Avg	225.7733	869.2334	1001.608
Brain_Tumor1	Avg	10.8095	26.4363	38.8603
Brain_Tumor2	Avg	14.7791	29.7486	45.6683
DLBCL	Avg	17.1194	22.9108	39.3595
Leukemia1	Avg	17.1668	22.5012	42.504
Leukemia2	Avg	25.9756	40.5877	50.8046
Prostate_Tumor	Avg	30.0724	63.9304	125.5382
SRBCT	Avg	14.0818	12.4427	31.7024

C. Computational complexity analysis

The computational complexity is an essential indicator to evaluate the algorithm performance. When using MH algorithms for solving the FS problem, the computational complexity depends on the dimension of the problem D , the population size P , and the process of updating the individuals' positions that continue until the maximum number of iterations T is reached. As shown in table 11, the three algorithms have the same total computational complexity. The main steps of these three algorithms are initializing the population, evaluating the fitness value of each individual, and updating the positions according to the algorithm's equations. The computational complexity is commonly expressed using the *Big-O* notation. After summarizing all the complexity discussed in the table below, the total computational complexity for all algorithms is the same which equals $O(P \times D \times T)$, where T is the main while loop which indicates the maximum number of iterations to run the algorithms.

Table 11: Computational complexity analysis for PSO, GWO, and IBGWO.

Main step for BPSO	Complexity	Main step for BGWO	Complexity	Main step for IBGWO	Complexity
initializing the particles	$O(P \times D)$	initializing the wolf pack	$O(P \times D)$	initializing the wolf pack	$O(P \times D)$
Evaluate objectives	$O(P)$	Evaluate objectives	$O(P)$	Evaluate objectives/ GWO	$O(P)$
Update velocities	$O(P \times D)$	Compute new positions	$O(P \times D)$	Compute new positions (GWO)	$O(P \times D)$
Compute new positions	$O(P \times D)$			Evaluate objectives/ PSO	$O(P)$
				Update velocities	$O(P \times D)$
				Compute new positions (PSO)	$O(P \times D)$
total computational complexity	$O(P \times D \times T)$	total computational complexity	$O(P \times D \times T)$	total computational complexity	$O(P \times D \times T)$

D. The comparison between IBGWO-S and IBGWO-V

As shown in the previous two sections, IBGWO-S and IBGWO-V outperformed other approaches. In this section, the comparisons between these two best approaches shown in table 12. According to the classification accuracy values, IBGWO-V achieved the best on 89% of the datasets. Based on reduction rates, it is observed that IBGWO-V outperformed IBGWO-S on 89% of the datasets. According to fitness values, IBGWO-V also outperformed IBGWO-S on 89% of the datasets. The results detected that the outweigh of IBGWO-V over IBGWO-S on selecting the optimal features with minimal classification error. The performance of the IBGWO-V optimizer also demonstrated that V-Shaped TF enhance IBGWO optimizer by balancing exploration and exploitation. Next subsection, the comparisons between IBGWO-S, IBGWO-V and eight well-known MH algorithms are conducted.

Table 12: Comparison of IBGWO-S with IBGWO-V based on classification accuracy, number of selected features, and fitness values

Dataset	Metric	Accuracy		No. Features		Fitness	
		IBGWO-S	IBGWO-V	IBGWO-S	IBGWO-V	IBGWO-S	IBGWO-V
11_Tumors	Avg	0.8275	0.9208	2037.10	1891.53	0.1742	0.0799
	Std	0.0137	0.0229	198.06	539.54	0.0136	0.0228
14_Tumors	Avg	0.6842	0.7018	3338.60	4004.87	0.3149	0.2979
	Std	0.0160	0.0165	421.20	742.30	0.0159	0.0166
Brain_Tumor1	Avg	0.7741	0.9537	801.90	125.30	0.2250	0.0460
	Std	0.0096	0.0329	140.27	58.79	0.0094	0.0326
Brain_Tumor2	Avg	0.8000	0.9700	1293.30	219.10	0.1992	0.0299
	Std	0.0000	0.0466	77.41	99.82	0.0001	0.0462
DLBCL	Avg	0.9396	1.0000	650.73	113.00	0.0610	0.0002
	Std	0.0114	0.0000	31.95	57.84	0.0113	0.0001
Leukemia1	Avg	0.9400	0.9956	750.50	252.53	0.0608	0.0049
	Std	0.0203	0.0169	38.15	122.14	0.0201	0.0167
Leukemia2	Avg	0.8644	1.0000	1525.97	5.27	0.1356	0.0000
	Std	0.0410	0.0000	144.39	2.69	0.0406	0.0000
Prostate_Tumor	Avg	0.9349	0.9984	1540.33	977.10	0.0659	0.0025
	Std	0.0233	0.0087	199.12	487.49	0.0231	0.0087
SRBCT	Avg	0.9980	0.9961	357.50	222.73	0.0035	0.0048
	Std	0.0107	0.0149	67.72	101.91	0.0106	0.0149
Ranking	W T L	1 0 8	8 0 1	1 0 8	8 0 1	1 0 8	8 0 1

E. The comparison between IBGWO-S, IBGWO-V, and well-known MH algorithms

The comparisons are done between IBGWO-S and IBGWO-V with eight well-known metaheuristic algorithms such as BGSA, BALO, BBA, BSSA, BDA, BWOA, BHHO, and BTLBO. Table 13 reveals that IBGWO-V performed highest accuracy result on four datasets. F-Test showed the excellence of IBGWO-V over other optimizers followed by IBGWO-S. Table 14 observes that IBGWO-V surpassed other algorithms on 89% of datasets. Table 15 shows the superiority of IBGWO-V on 6 out of 9 datasets referred to fitness values. F-Test supports the superiority of IBGWO-V followed by IBGWO-S over other competitors. These results prove that the proposed binary hybrid approach based on V-Shaped TF is effective and has an excellent performance on FS optimization problem.

As we have noted, various optimizers produced various results for the same dataset, this confirms the NFL theorem mentioned previously which concludes that the permanent demand for new optimizers to tackle FS problem.

Table 13: Comparison of IBGWO-S, IBGWO-V with other well-known algorithms in terms of classification accuracy results

Dataset	IBGWO-S	IBGWO-V	BGSA	BALO	BBA	BSSA	BDA	BWOA	BHHO	BTLBO
11_Tumors	0.8275	0.9208	0.9114	0.9219	0.7657	0.8399	0.7733	0.7324	0.7257	0.8781
14_Tumors	0.6842	0.7018	0.5209	0.5738	0.5874	0.5038	0.6281	0.4893	0.5532	0.5968
Brain_Tumor1	0.7741	0.9537	0.8130	1.0000	0.8278	1.0000	0.8333	1.0000	0.8889	0.7222
Brain_Tumor2	0.8000	0.9700	0.6833	0.8000	0.4733	0.5300	0.8967	0.6000	0.6033	0.6933
DLBCL	0.9396	1.0000	0.8208	0.9688	0.8563	0.8750	0.8813	0.9979	0.9146	0.8750
Leukemia1	0.9400	0.9956	0.9911	0.9800	0.9356	0.9333	0.9711	0.9178	1.0000	0.8689
Leukemia2	0.8644	1.0000	0.8022	0.8133	0.9578	0.9333	0.9333	1.0000	1.0000	0.9333
Prostate_Tumor	0.9349	0.9984	0.9302	0.8286	0.6254	0.9937	0.9651	0.8921	1.0000	0.9048
SRBCT	0.9980	0.9961	0.9882	0.9451	0.8765	1.0000	0.9157	1.0000	0.9922	1.0000
F-Test	3.11	2.61	7.00	5.33	7.78	6.06	5.50	5.89	5.00	6.28
Overall Rank	2	1	9	4	10	7	5	6	3	8

Table 14: Comparison of IBGWO-S, IBGWO-V with other well-known algorithms in terms of number of selected features

Dataset	IBGWO-S	IBGWO-V	BGSA	BALO	BBA	BSSA	BDA	BWOA	BHHO	BTLBO
11_Tumors	2037.10	1891.53	6252.87	8764.90	5035.77	7407.87	6218.33	6248.80	4741.13	5818.70
14_Tumors	3338.60	4004.87	7570.20	12283.43	6130.67	9068.40	7473.97	8534.73	7398.67	7077.67
Brain_Tumor1	801.90	125.30	2927.17	3098.20	2432.43	2901.83	2781.40	1944.17	1637.10	2526.77
Brain_Tumor2	1293.30	219.10	5166.03	5106.63	4120.63	5632.33	5018.47	3830.00	2755.37	4767.57
DLBCL	650.73	113.00	2681.93	2939.43	2148.10	3070.67	2554.63	2362.60	1665.57	2466.57
Leukemia1	750.50	252.53	2635.70	3337.93	2098.73	2756.63	2534.37	2151.83	1693.43	2281.33
Leukemia2	1525.97	5.27	5543.83	5764.73	4460.70	5564.43	5370.17	4498.87	3304.43	4863.43
Prostate_Tumor	1540.33	977.10	5215.00	7486.43	4080.80	5953.00	5145.57	4829.07	3555.10	4549.90
SRBCT	357.50	222.73	1134.33	1318.23	936.83	1290.23	1070.90	955.03	921.60	979.90
F-Test	2.00	2.78	8.11	9.67	4.11	9.11	6.78	5.44	3.22	5.56
Overall Rank	1	2	8	10	4	9	7	5	3	6

Table 15: Comparison of IBGWO-S, IBGWO-V with other well-known algorithms in terms of fitness values

Dataset	IBGWO-S	IBGWO-V	BGSA	BALO	BBA	BSSA	BDA	BWOA	BHHO	BTLBO
11_Tumors	0.1742	0.0799	0.0927	0.0843	0.2019	0.1645	0.2294	0.2699	0.2753	0.1253
14_Tumors	0.3149	0.2979	0.4794	0.4302	0.3778	0.4973	0.3732	0.5113	0.4473	0.4039
Brain_Tumor1	0.2250	0.0460	0.1901	0.0052	0.1395	0.0049	0.1697	0.0033	0.1128	0.2793
Brain_Tumor2	0.1992	0.0299	0.3185	0.2029	0.4429	0.4707	0.1071	0.3997	0.3954	0.3082
DLBCL	0.0610	0.0002	0.1823	0.0363	0.1253	0.1294	0.1222	0.0064	0.0876	0.1283
Leukemia1	0.0608	0.0049	0.0137	0.0261	0.0650	0.0712	0.0334	0.0854	0.0032	0.1341
Leukemia2	0.1356	0.0000	0.2007	0.1899	0.0104	0.0710	0.0708	0.0040	0.0029	0.0703
Prostate_Tumor	0.0659	0.0025	0.0741	0.1768	0.2975	0.0120	0.0395	0.1115	0.0034	0.0986
SRBCT	0.0035	0.0048	0.0166	0.0601	0.0873	0.0056	0.0881	0.0041	0.0118	0.0042
F-Test	3.11	2.50	7.00	5.44	6.89	6.67	5.89	5.89	5.22	6.44
Overall Rank	2	1	9	4	8	7	5	5	3	6

VII. CONCLUSION

This paper introduces a binary hybrid MH algorithm called IBGWO used to resolve the FS problem. This hybrid approach benefits from the strong ability of both BGWO exploitation and BPSO exploration. The evaluation metrics were applied on nine of high-dimensional small-instance datasets to assess the hybrid approach. The results of the comparisons between IBGWO, BPSO, and BGWO based on the S-shaped TFs proved that IBGWO-S is the best approach according to all evaluation metrics. The outcomes of the comparisons between IBGWO, BPSO, and BGWO according to V-Shaped TFs showed that IBGWO-V is the ideal approach according to the classification accuracy and fitness values. The comparisons between the two best approaches IBGWO-S and IBGWO-V showed that IBGWO-V outperformed IBGWO-S on selecting the minimal features with high classification accuracy. The results of the comparisons between IBGWO-S, IBGWO-V, and eight well-known MHs revealed that IBGWO-V outperformed other optimizers in the searching capabilities for the best solution followed by IBGWO-S. The remarkable results of the IBGWO approach revealed its ability to adjust the behavior of exploitation and exploration among the iterations.

The recommendation for future studies is to hybridize the GWO algorithm with other MH algorithms. Additionally, it would be interesting to employ the IBGWO approach to solve other types of datasets not only medical ones.

REFERENCES

- [1] H. Liu, and H. Motoda, *Feature selection for knowledge discovery and data mining*. Kluwer Academic Publishers, 1998.
- [2] M. Mafarja and S. Abdullah, "Modified great deluge for attribute reduction in rough set theory," 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2011, pp. 1464 -1469.
- [3] M. Mafarja and S. Abdullah, "Investigating memetic algorithm in solving rough set attribute reduction," *International Journal of Computer Applications in Technology*, vol. 48, no. 3, Oct., pp. 195-202, 2013.
- [4] H. Faris, A. M. Al-Zoubi, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, and H. Fujita, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks," *Information Fusion*, vol. 48, Aug., pp. 67-83, 2019.
- [5] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, Apr., pp. 141-158, 2017.
- [6] A. Y. Zomaya, Eds., *Handbook of nature-inspired and innovative computing: integrating classical models with emerging technologies*. Springer Science + Business Media, 2006.
- [7] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, pp. 1521-1552, 2018.
- [8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39-43.
- [9] M. Dorigo, V. Maniezzo, and A. Colomi, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, Feb., pp. 29-41, 1996.
- [10] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information sciences*, vol. 179, no. 13, June, pp. 2232-2248, 2009.
- [11] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, Aug., pp. 849-872, 2019.
- [12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, Mar., pp. 46-61, 2014.
- [13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, May, pp. 51-67, 2016.
- [14] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, Dec., pp. 163-191, 2017.
- [15] Y. Shi, R. Eberhart, "A modified particle swarm optimizer," Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, 1998, pp. 69-73.
- [16] J. Kennedy, "The particle swarm: social adaptation of knowledge," Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), 1997, pp. 303-308.
- [17] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural computing and applications*, vol. 30, no. 2, Nov., pp. 413-435, 2017.
- [18] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, Apr., pp. 1-14, 2013.

- [19] M. Mafarja, R. Jarrar, S. Ahmad, and A. A. Abusnaina, "Feature selection using binary particle swarm optimization with time varying inertia weight strategies," in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, 2018, pp. 1-9.
- [20] J. Too, A. R. Abdullah, N. M. Saad, N. M. Ali, and W. H. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," *Computers*, vol. 7, no. 4, Nov., pp. 1-18, 2018.
- [21] N. Chopra, G. Kumar, and S. Mehta, "Hybrid GWO-PSO algorithm for solving convex economic load dispatch problem," *International Journal of Research in Advent Technology*, vol. 4, no. 6, June, pp. 37-41, 2016.
- [22] E.-G. Talbi, *Metaheuristics: from design to implementation*, John Wiley & Sons, 2009.
- [23] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [24] O. B. Haddad, A. Afshar, and M. A. Mariño, "Honey-Bees Mating Optimization (HBMO) algorithm: a new heuristic approach for water resources optimization," *water resources management*, vol. 20, no. 5, July, pp. 661-680, 2006.
- [25] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Applied Soft Computing*, vol. 62, Jan., pp. 441-453, 2018.
- [26] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," in *Handbook of metaheuristics*, M. Gendreau and JY.Potvin, Eds. Springer, Boston, 2010, pp. 227-263.
- [27] S. Kashef and H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, Jan., pp. 271-279, 2015.
- [28] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053-1073, 2016.
- [29] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, May, pp. 80-98, 2015.
- [30] M. Mafarja, D. Eleyan, S. Abdullah, and S. Mirjalili, "S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem," in *Proceedings of the international conference on future networks and distributed systems*, 2017, pp. 1-7.
- [31] M. M. Mafarja and S. Mirjalili, "Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection," *Soft Computing*, vol. 23, no. 15, pp. 6249-6265, 2019.
- [32] A. A. Heidari and R. A. Abbaspour, "Enhanced chaotic grey wolf optimizer for real-world optimization problems: A comparative study," in *Handbook of Research on Emergent Applications of Optimization Algorithms*, P. Vasant, S. Z. Alparslan-Gok, G. Weber, Eds., IGI Global, 2018, pp. 693-727.
- [33] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with lévy flight for optimization tasks," *Applied Soft Computing*, vol. 60, Nov., pp. 115-134, 2017.
- [34] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, Jan., pp. 371-381, 2016.
- [35] Y.-P. Chen, Y. Li, G. Wang, Y.-F. Zheng, Q. Xu, J.-H. Fan, and X.-T. Cui, "A novel bacterial foraging optimization algorithm for feature selection," *Expert Systems with Applications*, vol. 83, Oct., pp. 1-17, 2017.
- [36] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, Nov., pp. 1424-1437, 2004.

- [37] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol. 28, no. 13, Oct., pp. 1825-1844, 2007.
- [38] H. Li, X.-Z. Wang, and Y. Li, "Using mutual information for selecting continuous-valued attribute in decision tree learning," in *Proceedings IEEE International Conference on Machine Learning and Cybernetics*, 2003, pp. 1496-1499.
- [39] R. K. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," *Expert systems with applications*, vol. 33, no. 1, July, pp. 49-60, 2007.
- [40] K. Chen, F.-Y. Zhou, and X.-F. Yuan, "Hybrid particle swarm optimization with spiral shaped mechanism for feature selection," *Expert Systems with Applications*, vol. 128, Aug. pp. 140-156, 2019.
- [41] A. Unler, A. Murat, and R. B. Chinnam, "mr2ps0: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification," *Information Sciences*, vol. 181, no. 20, Oct., pp. 4625-4641, 2011.
- [42] M. M. Mafarja, and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, Oct., pp. 302-312, 2017.
- [43] T. Blicke, "Tournament selection," in *Evolutionary Computation I*, T. Baeck, D.B. Fogel, Z. Michalewicz, Ed. New York, NY: Taylor & Francis Group, 2000, pp. 181-186.
- [44] E. Zorarpacı and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Systems with Applications*, vol. 62, Nov., pp. 91-103, 2016.
- [45] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. A. Elaziz, and S. Lu, "Improved salp swarm algorithm based on particle swarm optimization for feature selection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, Aug., pp. 3155-3169, 2019.
- [46] H. H. Inbarani, M. Bagyamathi, and A. T. Azar, "A novel hybrid feature selection method based on rough set and improved harmony search," *Neural Computing and Applications*, vol. 26, no. 8, Nov., pp. 1859-1880, 2015.
- [47] X.-S. Yang, "Harmony Search as a Metaheuristic Algorithm," in *Music-Inspired Harmony Search Algorithm*, Z. W. Geem, Ed. Berlin, Germany: Springer-Verlag Berlin and Heidelberg, 2009, pp. 1-14.
- [48] H. I. Elshazly, A. T. Azar, A. E. Hassanien, and A. M. Elkorany, "Hybrid system based on rough sets and genetic algorithms for medical data classifications," *International Journal of Fuzzy System Applications (IJFSA)*, vol. 3, no. 4, Oct., pp. 31-46, 2013.
- [49] A. Zhu, C. Xu, Z. Li, J. Wu, and Z. Liu, "Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3d stacked SoC," *Journal of Systems Engineering and Electronics*, vol. 26, no. 2, Apr., pp. 317-328, 2015.
- [50] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, July, pp. 1257-1263, 2015.
- [51] H. Xu, X. Liu, and J. Su, "An improved grey wolf optimizer algorithm integrated with cuckoo search," in *IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, 2017, pp. 490-493.
- [52] M. E. Basiri and S. Nemati, "A novel hybrid ACO-GA algorithm for text feature selection," in *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation*, 2009, pp. 2561-2568.
- [53] J. B. Jona and N. Nagaveni, "Ant-cuckoo colony optimization for feature selection in digital mammogram," *Pakistan journal of biological sciences*, vol. 17, no. 2, pp. 266-271, 2014.
- [54] N. Singh and S. B. Singh, "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance," *Journal of Applied Mathematics*, vol. 2017, Nov., pp. 1-15, 2017.

- [55] M. A. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Engineering Journal*, vol. 12, no. 1, Mar., pp. 621-630, 2021.
- [56] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary Optimization using Hybrid Grey Wolf Optimization for Feature Selection," *IEEE Access*, vol. 7, pp. 39496-39508, 2019.
- [57] D. H. Wolpert, W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, Apr., pp. 67-82, 1997.
- [58] R. Eberhart and J. Kennedy, "Particle swarm optimization," Proceedings of the IEEE international conference on neural networks, 1995, pp. 1942-1948.
- [59] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, June, pp. 33-57, 2007.
- [60] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, Dec., pp. 1362-1381, 2009.
- [61] M. N. Alam, "Particle swarm optimization: Algorithm and its codes in MATLAB," *ResearchGate*, Mar. 7, 2016. [Online]. Available: <https://www.researchgate.net/publication/297245624>.
- [62] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization*, vol. 39, no. 3, Apr. pp. 459-471, 2007.
- [63] S.-K. S. Fan and Y.-Y. Chiu, "A decreasing inertia weight particle swarm optimizer," *Engineering Optimization*, vol. 39, no. 2, Mar., pp. 203-228, 2007.
- [64] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in Proceedings of the Congress on Evolutionary Computation, 1999, pp. 1945-1950.
- [65] C. Yang, W. Gao, N. Liu, and C. Song, "Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight," *Applied Soft Computing*, vol. 29, Apr., pp. 386-394, 2015.
- [66] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in IEEE International conference on systems, man, and cybernetics. 1997, pp. 4104-4108.
- [67] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, Sep., pp. 727-745, 2010.
- [68] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," *Bioinformatics*, vol. 21, no. 5, Sep., pp. 631-643, 2004.
- [69] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, "Feature selection for high-dimensional data," *Progress in Artificial Intelligence*, vol. 5, no. 2, Feb., pp. 65-75, 2016.
- [70] R. Y. Nakamura, L. A. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA: a binary bat algorithm for feature selection," in SIBGRAPI conference on graphics, Patterns and Images, 2012, pp. 291-297.
- [71] R. M. Rizk-Allah, A. E. Hassanien, M. Elhoseny, and M. Gunasekaran, "A new binary salp swarm algorithm: development and application for optimization tasks," *Neural Computing and Applications*, vol. 31, no. 5, May, pp. 1641-1663, 2019.
- [72] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary dragonfly algorithm for feature selection," in International Conference on New Trends in Computing Sciences (ICTCS), 2017, pp. 12-17.

- [73] A. G. Hussien, E. H. Houssein, and A. E. Hassanien, "A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection," in *IEEE International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2017, pp. 166-172.
- [74] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection," in *Evolutionary Machine Learning Techniques*, S. Mirjalili, H. Faris, and I. Aljarah, Eds. Springer Singapore, 2020, pp. 251–272.
- [75] M. Allam and M. Nandhini, "Optimal feature selection using binary teaching learning based optimization algorithm," *Journal of King Saud University-Computer and Information Sciences*, Dec., pp. 1-13, 2018.
- [76] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, Aug., pp. 175-185, 1992.
- [77] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient KNN classification with different numbers of nearest neighbors," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, May, pp. 1774-1785, 2018.
- [78] I. M. El-Hasnony, S. I. Barakat, M. Elhoseny, and R. R. Mostafa, "Improved feature selection model for big data analytics," *IEEE Access*, vol. 8, pp. 66989-67004, 2020.
- [79] E.-S. El-Kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," *International Journal of Innovative Computing Information and Control*, vol. 16, no. 3, June, pp. 831-844, 2020.