International Journal of Innovative Computing, Information and Control
Volume **16**, Number **3**, June **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 831–844

HYBRID GRAY WOLF AND PARTICLE SWARM OPTIMIZATION FOR FEATURE SELECTION

El-Sayed El-Kenawy* and Marwa Eid

Department of Communications and Electronics
Delta Higher Institute for Engineering and Technology
Mansoura 35111, Egypt
*Corresponding author: skenawy@ieee.org

Received November 2019; revised March 2020

Abstract. *The big data comprises a relatively developing area of study which is due to numerous facts gathered daily and the wishes to be helpful information for use in our day by day life. One of the most crucial pre-processing of data is the feature selection. This paper proposes a hybrid technique that combines two algorithms namely Gray Wolf Optimization (GWO) and Particle Swarm Optimization (PSO), the manner that lets in the critical functions to be recognized and lets the insignificant ones and the complexity to be erased. This enables the obligations of the gadget learning classification while making use of training to the classifier with the data set. A hybrid approach is primarily based on metaheuristics swarm intelligence algorithms, which simulate the gray wolf's management and hunting manner in nature and PSO which people are moving impacted by their local best positions and by the global best position. This hybridization is to acquire the balance between exploitation and exploration. We used seventeen datasets from UCI machine gaining knowledge of repository within the experiments and comparisons results to assess the effectiveness and quality of the GWOPSO.*
Keywords: Feature selection, Hybrid optimization, Grey wolf optimizer, Metaheuristics, Particle swarm optimizer

1. **Introduction.** The many facts are one of the quickest growing fields resulting from a lot of data gathered everyday and the need for information to be used in our everyday life [1]. Numerous fields are making use of massive information like machine learning and data mining. One of the most important parts in pre-processing of data is feature selection. It presents a way of detecting the essential features and disposing of irrelevant ones from the dataset [2]. The goals of feature selection are the reduction of data dimensions, improvement of prediction quality, and a good understanding of data for various machine learning applications [3]. Data representations frequently use many redundant features in real applications, which means that certain features can take on the role of others and delete unimportant features. Also, the interdependence features have an impact on output and contain important information that will be unknown if any of them are removed [4]. In the classification of machine learning, knowing the optimal subset of features that reduce a large number of features that are usually introduced in data sets is a very important step before classification processing. The following are categorizations of feature selection techniques: filter, wrapper, and hybrid-based [5]. The filter-based selection techniques or conventional feature selection techniques have the merit of being fast and having the ability to scale up to large datasets [6]. For situations where wrappers can overfit such as Information Gain (IG), the process of selecting features is often more useful [7]. Information Gain (IG) calculates how much "information" a feature gives us about the category

DOI: 10.24507/ijicic.16.03.831

and is useful in reducing the number of features in the classification model that give us more accuracy [8,9]. Wrapper feature selection strategies minimize search space for selecting features. As part of their select feature, the wrappers include the learning algorithm. Wrappers have greater precision, but it takes much longer. Exploitation and exportation are two concepts of the algorithm for optimization. Exportation is a technique to explore the promising points of search space. On the other hand, exploitation is the method of using a promising point in search space to locate a more promising point. For quality search algorithms, balancing between exploration and exploitation is essential. A hybrid strategy is used where it blends two or more algorithms to achieve the balance between exploitation and exportation.

In this paper, we suggest a hybrid approach combining two algorithms namely Gray Wolf Optimization (GWO) and Particle Swarm Optimization (PSO). The Gray Wolf Optimizer (GWO) is stimulated by the hierarchy of leadership and the hunting behavior of gray wolves in nature, with gray wolves preferring to live in a pack [10,11]. They are among themselves divided into more than one leader alpha, beta, omega, and delta. The alphas have the highest level in the hierarchy and are considered to be the most dominating among the group hence responsible for making decisions about hunting, sleeping, wake-up time and so on. The next level in the leadership hierarchy is beta. The betas are subordinate wolves who support the alpha in making decisions or other pack activities. The beta wolf is the group's best member who, if one of the alpha wolves passes away or becomes very old, can become the alpha. Next comes the omega that performs the role of scapegoat. Omega wolves should always submit to all the other dominant wolves. They are the last wolves which are allowed to eat. In some instances if a wolf is not alpha, beta, or omega, it is called delta. Alphas and betas must be submitted to by delta wolves, but they dominate the omega. The Particle Swarm Optimization (PSO) is an evolutionary computing technique that Kennedy and Eberhart suggested in 1995. PSO is motivated by social behaviors such as flocking birds and fish schooling; it works by having a (swarm) population of candidate solutions (particles) [12,13]. In the search space, these particles are moved around. The particle movements are controlled by their best-known space location and the best-known position of the whole swarm. These will come to direct the swarm's movement as improved locations are discovered. The process is repeated several times to achieve a satisfactory solution (hopefully). For example, one can determine the value of a subset of attributes after performing the search task by considering the feature's individual predictive ability along with the degree of redundancy between them.

To take advantage of the social behavior of PSO along with the hunting behavior of GWO, we combine both the powerful optimizers proposed in our hybrid approach. The suggested hybrid approach divides the population into two groups. The first group follows the GWO procedures while the second group follows the PSO procedures.

The rest of the paper is organized as follows. Section 2 presents related work for optimization algorithms GWO and PSO. The proposed hybrid algorithm GWOPSO describes in Section 3. Section 4 presents hybrid algorithm for feature selection. The experimental results are discussed in Section 5. Finally, conclusions are stated in Section 6.

2. **Related Work.** The idea of optimization occurs in several areas of research such as engineering, health, agriculture, computer science, and feature selection. In optimization, according to the problem description, the main goal is to settle the optimal solution to a particular problem from the available solutions. Also, there is a goal in optimization algorithms for maximizing the classification performance (minimizing classification error rate) and minimizing the number of features. In [14] framework for big data analyzing and extracting has been introduced, and in this framework for training multilayer perception

GWO has been used. Recently, researchers have gained further interest in the field of hybrid metaheuristics [15]. GWO is a new optimization algorithm that simulates the gray wolf's leadership and hunting manner in nature [16]. GWO is characterized by flexibility, deprivation-free mechanism, simplicity, and local optima avoidance [17]. Due to that, it has been used in many fields in the last years such as DC motors control [18], feature subset selection, economic emission dispatch problems [19], Radial Basis Function (RBF) networks training, and image registration [20]. In this paper data set has been hashed into buckets, then used the GWO+BRNN for finding the top 5 best candidates in each bucket and escalate them to the next level in our tournament [21]. GWO algorithm is also used to train the MLP network [22]. In the field of optimization, PSO has also gained a lot of popularity. To solve various optimization issues: artificial neural network training, fuzzy system control, and others, PSO has been successfully applied [23]. In the mechanical domain, PSO has been applied successfully to optimizing various solutions like optimal weight design of a gear train, process parameter optimization in casting, simultaneous optimization of design and machining tolerances, and machine scheduling problem [24].

3. GWOPSO.

3.1. Gray wolf optimization.

GWO is an algorithm based on the population. GWO begins with an initial random population and they are modified during iteration. GWO maintains a balance between exploitation and exportation. Exportation is a technique to explore the search space for promising points in search space. Exploitation is the technique of using the promising point to find the best promising point in search space. Every person is considered a solution to the problem being solved. Then, for all solutions, the fitness function is calculated. Therefore, alpha, beta, and gamma can be identified. The following equations are used to update the position of each wolf.

$$\begin{aligned} \overrightarrow{D} &= \left|\overrightarrow{C} \cdot \overrightarrow{X_p}(t) - \overrightarrow{X}(t)\right| \\ \overrightarrow{X}(t+1) &= \overrightarrow{X_p}(t) - \overrightarrow{A} \cdot \overrightarrow{D} \end{aligned} \tag{1}$$

where t refers to the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors, $\overrightarrow{X_p}$ is the preposition, and $\overrightarrow{X}$ is the position of the gray wolf. The vectors are calculated using the following equation:

$$\begin{aligned} \overrightarrow{A} &= 2\overrightarrow{a} \cdot \overrightarrow{r_1} - \overrightarrow{a} \\ \overrightarrow{C} &= 2 \cdot \overrightarrow{r_2} \end{aligned} \tag{2}$$

where a is linearly decreased from 2 to 0 over the course of iterations and r_1 and r_2 are a random number in the range $[0, 1]$. The whole pack reaches the prey and attack by updating the position based on the best locations of the alpha, beta, and delta using the following equations:

$$\begin{aligned} \overrightarrow{D_\alpha} &= \left|\overrightarrow{C_1} * \overrightarrow{X_\alpha} - \overrightarrow{X}\right| \\ \overrightarrow{D_\beta} &= \left|\overrightarrow{C_2} * \overrightarrow{X_\beta} - \overrightarrow{X}\right| \\ \overrightarrow{D_\delta} &= \left|\overrightarrow{C_3} * \overrightarrow{X_\delta} - \overrightarrow{X}\right| \\ \overrightarrow{X_1} &= \overrightarrow{X_\alpha} - \overrightarrow{A_1} * \overrightarrow{D_\alpha} \\ \overrightarrow{X_2} &= \overrightarrow{X_\beta} - \overrightarrow{A_2} * \overrightarrow{D_\beta} \end{aligned} \tag{3}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} * \overrightarrow{D_\delta} \tag{4}$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{5}$$

3.2. **Particle swarm optimization.** In the PSO algorithm, it mimics the intelligence of bird swarms in nature that the potential solutions called particles are flown in the search space for problems. The change of position of a particle is known as velocity. With time, the particles change their position. During the flight, the velocity of a particle is stochastically accelerated to its previous best position and the following equations are used to update it to a neighborhood best solution. A particle i is defined by its position vector, x_i, and its velocity vector, v_i

$$v_i^{k+1} = v_i^k + c_1 r_1 \left(Pbest_i^k + x_i^k\right) + c_2 r_2 \left(g_{best} - x_i^k\right) \tag{6}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{7}$$

3.3. **Hybrid GWOPSO.** The procedure for finding global optima is a challenging task. Two efficient algorithms are given for the proposed approach. The first algorithm is the PSO in which individuals move guided by their global best positions and by their local best position. The global best position is the best position found by the population whereas the local best position refers to the best position found so far by an individual. This social behavior allows individuals in PSO to converge to their global target. This behavior is influenced by nature's flock of birds and fish school. Due to its strength, reliability, simplicity, we selected PSO for our proposed hybrid optimizer. Gray wolf optimizers is the second optimizer in our proposed hybrid approach. GWO is a swarm-based metaheuristic optimizer that imitates the gray wolves' social hierarchy and foraging behavior. Individuals moved in the GWO are influenced by the position of the three leader's alpha, beta, and delta.

The optimization process begins with a random group of individuals in our hybrid optimizer. Such people have been repressing candidate solutions to the problem being solved. The fitness function is then determined for all individuals for each iteration and the first three leaders are labeled as alpha, beta, and delta. As mentioned above, the population is then divided equally into two classes where the first group follows the GWO procedures while the second group follows the PSO procedures. By doing so, the search space is highly searched for promising points and these points are exploited using the powerful PSO and GWO.

The pseudo-code of the GWOPSO algorithm is presented in Figure 1.

The flow-chart for proposed GWOPSO algorithm is presented in Figure 2.

4. **GWOPSO for Feature Selection.** The problem with feature selection is so unique because the search space is limited to two binary values 0 and 1. It is, therefore, we used sigmoid equations which are necessary to modify the traditional optimizer to work properly for this problem. In this section, a hybrid approach based on metaheuristics swarm intelligence algorithms (GWOPO) begins with an initial random vector population to select features than fitness function, KNN has been used for training instances and clarify how it has been used to features that are at pick.

4.1. **Binary modified hybrid GWOPSO optimizer.** To convert the values of standard optimizer from continuous to binary values, we modified equation (GWOPSO) to the following equations.

Initialization
Initialize A, a, C
Randomly initialize an agent of n wolf's positions 2 $[1, 0]$.
Based on the fitness function the α; β; δ solutions.
While ($t <$ Max_iter)
    if t % 2 == 0:
    Calculate fitness using (GWO)
  Else
    Calculate fitness using (PSO)
  Update A, a, C
  Evaluate all particles using the objective function
  Update the positions of the three best agents α; β; δ
  $t = t + 1$
end while

FIGURE 1. Pseudo-code of the GWOPSO algorithm

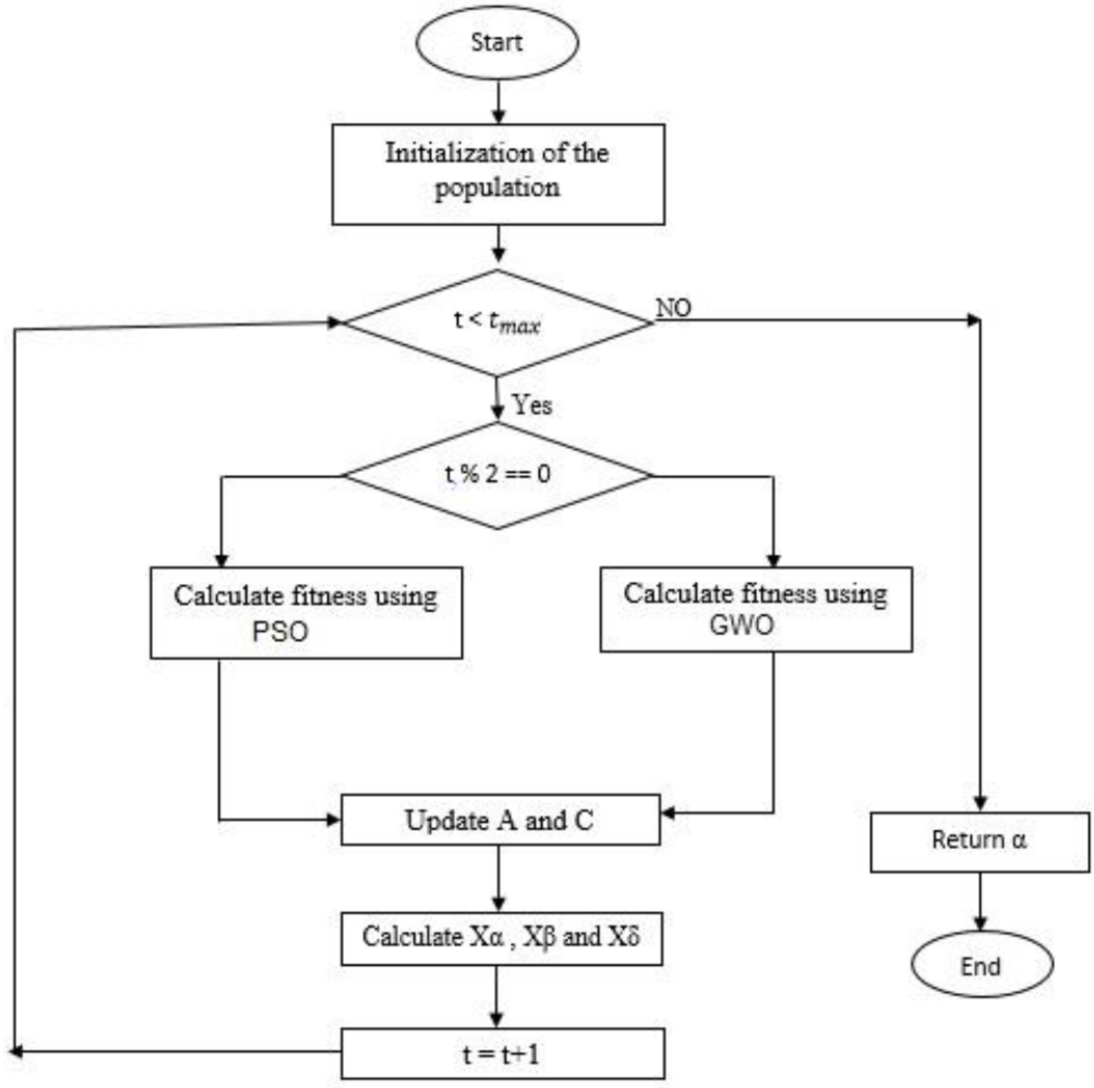

FIGURE 2. Flow-chart for proposed GWOPSO algorithm

$$x_d^{t+1} = \begin{cases} 1 & \text{if } sigmoid\left(x_d^{t+1}\right) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where x_d^{t+1} is the updated binary position of the dimension d at iteration t and x_d^{t+1} is the position for current dimension.

$$sigmoid = \frac{1}{1 + e^{-10(x-0.5)}} \tag{9}$$

The role of the sigmoid is to scale the continuous values of the optimizer between 0 and 1. Then using the conditions we decide whether the value of the dimension will be zero or one.

4.2. **Solution representation.** For the problem of feature selection, the solution may be defined as a vector of features with size h where h is the number of features. Each element in that vector is a binary value (0 or 1) where 0 means the feature is not included and 1 means that feature is included. Thus, the hybrid GWOPSO begins with an initial random vector population holding randomly selected features. After all, using high exploitation and exploration capabilities of the hybrid GWOPSO will be able to find the feature's most optimal subset.

4.3. **Fitness function.** The functions of fitness are used to measure the quality of each solution of the hybrid GWOPSO. The fitness function depends on two factors: the classification error rate and the number of selected features. The solutions are considered to be good if it selected a subset of features that give a lower classification error rate and a lower number of selected features. To evaluate the quality of each solution, the following equation is used:

$$Fitness = h_1 E(D) + h_2 \frac{|s|}{|f|} \tag{10}$$

where $E(D)$ is the classification error rate for each dimension, s is the number of selected features, f is the number of features and $h \in [0, 1]$, $h_2 = 1 - h_1$ are constants that manage the importance of classification error rate and the number of the selected features.

4.4. **K-nearest neighbor.** Wrapped feature selection approaches include a learning algorithm for determining the performance of the selected function sub-set. In this work, we use the K-Nearest Neighbor (KNN) [25]. KNN is a supervised learning algorithm. It is commonly used because it is simple and easy to implement. To decide the class of the unknown instance, KNN uses training instances instead of building models. In our experiments, KNN is used for classification tasks to measure the quality of the selected subset of features.

5. **Experimental Results and Discussion.** To evaluate our proposed algorithm, the hybrid GWOPSO tested against seventeen datasets from the UCI machine learning repository in the experiments and comparisons results. The datasets were selected to have various numbers of attributes and instances as representatives of various kinds of issues that the proposed technique will be tested on.

5.1. **Configurations.** Each dataset is divided into three randomly equal-size parts: training, validation, and test. Training is used to train KNN classifier during the learning phase. Validation is used to test. When calculating the fitness function for a specific solution, KNN configuration parameters are shown in Table 2.

TABLE 1. Datasets description

Dataset name	**Number of rows**	**Number of columns (Features)**
Hepatitis	155	10
Ionosphere	351	33
Vertebral	310	6
Seeds	210	7
Parkinsons	195	22
Australian	690	14
Blood	744	4
Breast_Cancer	699	8
Diabetes	768	8
Lymphography	148	18
Parkinson	195	22
Ring	7400	20
Titanic	2201	3
Towonorm	7400	20
WaveformEW	5000	21
Tic-Tac-Toe	985	9
M-of-n	1324	10

TABLE 2. Experiments configuration

Parameter	Value
No of search agents	10
No of iterations	80
Problem dimension	Number of features in the data
Search domain	$[0, 1]$
No. repetitions of runs	20
Inertia factor of PSO	0.1
α parameter in the fitness function	0.99
β parameter in the fitness function	0.01

5.2. **Evaluation metrics.**

- **Classification Average Error** shows the accuracy of the classifier given the set of selected functions. The classification average error can be calculated in Equation (11).

$$AvgPref = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} Match\left(C_i, L_i\right) \tag{11}$$

where M is the number of the runs to the algorithm, N number of test points, C_i the classifier output label for the point i, L_i is the class label for data point i, and $Match$ is the function that calculates if two inputs are matched or not.

- **Best Fitness** is the minimum fitness function obtained for a certain optimizer at the distinct M operations of an optimization algorithm. It represents the most optimistic solution acquired and can be formulated in Equation (12).

$$best = \min_{i=1}^{M} g_*^i \tag{12}$$

where M is the number of running the optimization algorithm to select the feature subset, and g_*^i is the optimal solution resulted from a run number i.

- **Worst Fitness** is the worst solution to run an optimization algorithm for M times among the best solutions found. Worst is the negative solution that can be formulated in Equation (13).
$$worst = Max_{i=1}^{M} g_*^i \tag{13}$$
where M is the number of running the optimization algorithm to select the feature subset and g_*^i is the optimal solution resulted from a run number i.
- **Average Fitness size** represents the average size of the selected features to the total number of features. This measure can be formulated as in Equation (14).
$$AVG\ Selection\ Size = \frac{1}{M} \sum_{i=1}^{M} \frac{size\left(g_*^i\right)}{D} \tag{14}$$
where M is the number of times to run the optimization algorithm to select the feature subset, g_*^i is the optimal solution resulted from run number i, $size(x)$ is the number of on values for the vector x, and D is the number of features in the original dataset.
- **Mean** is the average of solutions acquired from running an optimization algorithm for different M running. *Mean* represents the average value that can be formulated in a given Equation (15).
$$Mean = \frac{1}{M} \sum_{M}^{1} g_*^i \tag{15}$$
where M is the number of times to run the optimization algorithm to select the feature subset, and g_*^i is the optimal solution resulted from a run number i.
- **Std (Standard Deviation)** reflects the variance of the best solutions obtained to run a stochastic optimizer for different runs. *Std* can be used as an indication of optimizer stability and robustness, whereas *Std* being smaller this means that the optimizer always converges to the same solution; while larger values for *Std* mean many random results are formulated as in Equation (16).
$$Std = \sqrt{\frac{1}{M-1} \sum \left(g_1^i - Mean\right)^2} \tag{16}$$
where M is the number of times to run the optimization algorithm to select the feature subset, g_1^i is the optimal solution resulted from run number i, and *Mean* is the average defined in Equation (15).

5.3. **Experimental results and analysis.** Seven tests were conducted to determine the efficiency of the GWOPSO hybrid optimizer as shown below. Experimental 1, Table 3 shows the average error of various algorithms. The lower error shows that the optimizer has selected the correct set of features that can train the classifier and make a lower error on the hidden test data. Table 3 and Figure 3 reveal that the lowest error is achieved by the proposed hybrid GWOPSO, which proved to be highly exploring the search space, and this is due to GWO's high exploration and very good exploitation achieved by PSO. The proposed hybrid GWOPSO then uses PSO intensity to push towards the optimal solution which includes the optimal subset of features that can mitigate the error.

Table 4 shows the average features selected by different optimizers in Experimental 2. While choosing a lower number of features indicates that the selection of features performed by the optimizer is more important to maintain a lower error. For this reason, for the classification error, the fitness function assigns a higher weight and still encourages optimizer to select the lower number of features. The proposed hybrid GWOPSO, as shown in the table, has been able to find the least number of channels for twelve datasets

TABLE 3. Average error result from different optimizers in our experiments

Dataset	GWOPSO	GWO	PSO	WAO	MVO	FFA	GA
Hepatitis	**0.192**	0.203	0.219	0.211	0.215	0.227	0.211
Ionosphere	0.165	**0.138**	0.194	0.189	0.155	0.188	0.155
Vertebral	0.223	0.217	0.225	0.215	0.238	0.223	0.221
Seeds	**0.202**	0.254	0.248	0.248	0.228	0.237	0.251
Parkinsons	0.144	0.153	0.163	0.160	0.129	0.181	0.141
Australian	0.160	0.15334	0.157	0.161	0.164	0.181	**0.149**
Blood	0.254	0.254	**0.238**	0.252	0.248	0.259	0.268
Breast_Cancer	0.0489	0.0497	0.0489	**0.0463**	0.0472	0.0498	0.047
Diabetes	0.2535	0.256	0.258	0.270	0.25	0.275	0.266
Lymphography	**3.889**	4.387	5.248	5.0204	4.738	4.4897	4.6775
Parkinson	0.16	**0.138**	0.1692	0.163	0.156	0.166	0.1538
Ring	0.1635	0.1635	0.1673	0.1751	0.1656	**0.1602**	0.1675
Titanic	**0.2117**	**0.2117**	0.2212	0.2316	0.2147	0.2319	0.239
Towonorm	0.0488	0.0428	0.0695	**0.0344**	0.0523	0.0482	0.0610
WaveformEW	0.4139	0.4285	0.4364	**0.3963**	0.4147	0.4366	0.4352
Tic-Tac-Toe	**0.247**	0.274	0.289	0.287	0.3003	0.2927	0.2677
M-of-n	0.1619	**0.03764**	0.1324	0.0798	0.1079	0.1070	0.1383

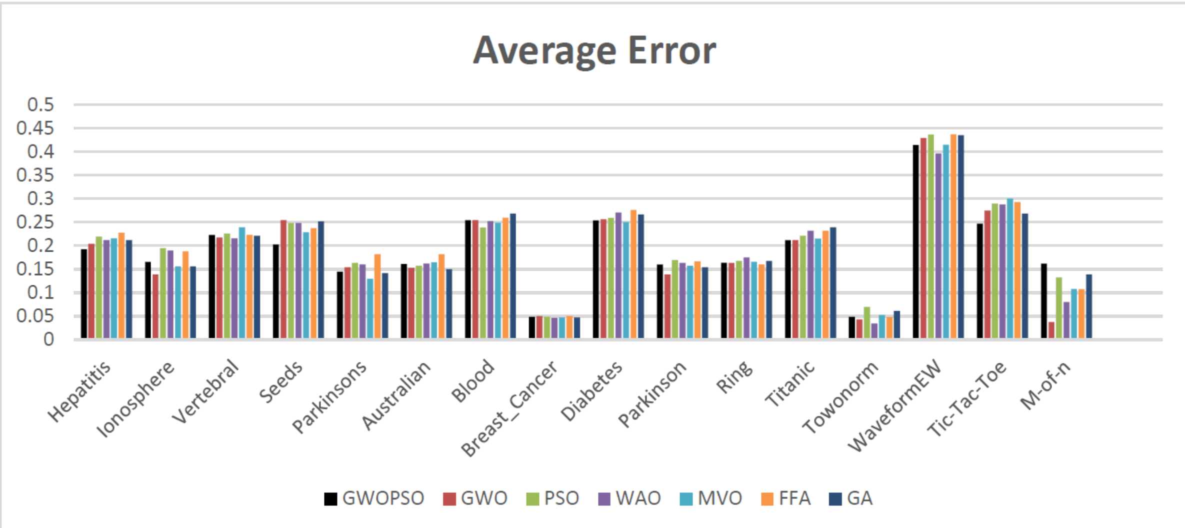

FIGURE 3. (color online) Average error for different optimizers

and has given them the lower classification. Nonetheless, for the remaining datasets, hybrid GWOPSO selects a higher number of features, retaining the smallest error for that dataset.

The numerical findings were listed in Tables 5, 6, 7, and 8 for Experimental 3, Experimental 4, Experimental 5, and Experimental 6 (Average, Best, Worst, and Standard Deviation). The proposed hybrid GWOPSO has been able to find the lowest fitness quality for all datasets as shown in Table 5, which means it can pick the optimum subset of features that offer the lowest classification error. The reason for this high performance is the cooperative nature of the individuals of the hybrid GWOPSO which utilized the proposed hybrid GWOPSO in highly exploration of the search space for different solutions. Moreover, the proposed hybrid GWOPSO enhanced the exploitation process due

Table 4. Average select size result from different optimizers in our experiments

Dataset	**GWOPSO**	**GWO**	**PSO**	**WAO**	**MVO**	**FFA**	**GA**
Hepatitis	0.3800	0.4000	0.4800	0.5000	0.5200	0.4600	**0.3600**
Ionosphere	**0.0909**	0.2909	0.4242	0.2909	0.4303	0.4540	0.4000
Vertebral	**0.4000**	0.5000	0.7000	0.4660	0.4330	0.5000	0.5330
Seeds	**0.4280**	0.5420	0.5140	0.6280	0.6000	0.5710	0.5140
Parkinsons	**0.2090**	0.3909	0.5000	0.3900	0.4900	0.4090	0.4720
Australian	**0.3143**	0.4571	0.5286	0.7143	0.5714	0.5714	0.4857
Blood	**0.5000**	0.6000	0.6500	0.6000	0.6500	0.5500	0.6000
Breast_Cancer	0.5250	**0.5000**	0.6500	0.6750	0.6500	0.5750	0.5250
Diabetes	**0.4000**	0.5250	0.5250	0.6500	0.5250	0.7250	0.5750
Lymphography	**0.1889**	0.2778	0.5333	0.4667	0.5333	0.5222	0.5333
Parkinson	**0.2818**	0.3818	0.4818	0.3909	0.4909	0.4909	0.4545
Ring	**0.3100**	0.3400	0.3200	**0.3100**	0.3400	0.3400	0.3400
Titanic	**0.7333**	0.8667	0.8000	0.8000	0.8000	**0.7333**	0.8667
Towonorm	0.6800	0.8500	**0.6700**	0.9700	0.7900	0.8300	0.8100
WaveformEW	**0.4381**	0.5143	0.6857	0.8762	0.6000	0.6000	0.6095
Tic-Tac-Toe	**0.2470**	0.2746	0.2897	0.2878	0.3003	0.2928	0.2677
M-of-n	0.1619	**0.0376**	0.1324	0.0798	0.1079	0.1070	0.1383

Table 5. Average fitness result from different optimizers in our experiments

Dataset	**GWOPSO**	**GWO**	**PSO**	**WAO**	**MVO**	**FFA**	**GA**
Hepatitis	**0.1920**	0.2030	0.2190	0.2110	0.2150	0.2270	0.2110
Ionosphere	0.1650	0.1380	0.1940	0.1890	0.1550	0.1880	0.1555
Vertebral	0.2230	0.2170	0.2250	0.2150	0.2380	0.2230	0.2210
Seeds	**0.2020**	0.2540	0.2480	0.2480	0.2280	0.2370	0.2510
Parkinsons	0.1440	0.1530	0.1630	0.1600	0.1290	0.1810	0.1410
Australian	0.1609	0.1530	0.1574	0.1617	0.1643	0.1817	**0.1496**
Blood	0.2546	0.2546	**0.2386**	0.2522	0.2490	0.2594	0.2683
Breast_Cancer	0.0489	0.0498	0.0489	0.0464	0.0472	0.0498	0.0472
Diabetes	0.2539	0.2563	0.2586	0.2703	0.2500	0.2758	0.2664
Lymphography	**3.8898**	4.3878	5.2490	5.0204	4.7388	4.4898	4.6776
Parkinson	0.1600	0.1385	0.1692	0.1631	0.1569	0.1662	0.1538
Ring	0.1636	0.1635	0.1673	0.1752	0.1657	**0.1603**	0.1676
Titanic	**0.2117**	0.2117	0.2213	0.2317	0.2147	0.2319	0.2390
Towonorm	0.0488	0.0428	0.0696	**0.0345**	0.0524	0.0483	0.0611
WaveformEW	0.4139	0.4286	0.4365	**0.3964**	0.4148	0.4366	0.4353
Tic-Tac-Toe	**0.2470**	0.2790	0.2897	0.2878	0.3003	0.2928	0.2677
M-of-n	0.1619	0.1079	0.1324	0.0798	0.1079	0.1070	0.1383

to the strength of PSO. As per Table 6, hybrid GWOPSO was able to find the best fitness compared to other optimizers throughout runs. On the other hand, in Table 7, hybrid GWOPSO did not find the worst fitness compared to other optimizers which prove the capability of the proposed hybrid GWOPSO to find the optimal subset of features. The standard deviation of the statistical results is illustrated in Table 8. Relative to other optimizers, the proposed hybrid GWOPSO has the lowest standard deviation as shown

Table 6. Best fitness result from different optimizers in our experiments

Dataset	GWOPSO	GWO	PSO	WAO	MVO	FFA	GA
Hepatitis	0.2073	0.2267	0.2073	0.2267	0.2073	0.2073	0.2073
Ionosphere	0.1708	0.1370	0.1793	0.1962	0.1285	0.1878	**0.1031**
Vertebral	0.3752	0.3752	0.3463	0.3656	0.3752	0.3463	0.3656
Seeds	0.2414	0.2414	0.2697	0.2839	**0.2273**	0.2697	0.2414
Parkinsons	0.1057	0.1057	0.1514	0.1057	0.0905	0.1666	**0.0752**
Australian	0.2977	0.2977	0.3063	0.3020	0.3063	0.2934	0.3020
Blood	0.8357	0.8357	**0.8278**	**0.8278**	**0.8278**	0.8596	0.8556
Breast_Cancer	**0.3210**	0.3295	0.3295	0.3252	0.3252	**0.3210**	0.3252
Diabetes	**0.5404**	0.5636	0.5520	0.5598	0.5482	0.5559	0.5559
Lymphography	2.5331	2.8563	3.8868	3.7453	3.8261	**1.4219**	3.4221
Parkinson	0.1362	0.1209	0.1514	0.1209	0.1057	0.1362	0.1514
Ring	**1.3812**	1.3849	1.3909	1.4009	1.3844	1.3849	1.3917
Titanic	**2.6358**	**2.6358**	**2.6358**	**2.6358**	**2.6358**	2.6371	2.6520
Towonorm	1.2765	1.2672	1.2813	**1.2624**	1.2805	1.2668	1.2849
WaveformEW	1.1759	1.2015	1.1890	**1.1622**	1.1640	1.2139	1.1818
Tic-Tac-Toe	0.5821	0.5821	0.5821	0.5945	0.5852	0.6287	0.5852
M-of-n	**0.4622**	0.4779	0.5632	0.4802	**0.4622**	0.4779	0.5565

Table 7. Worst fitness result from different optimizers in our experiments

Dataset	GWOPSO	GWO	PSO	WAO	MVO	FFA	GA
Hepatitis	0.2655	0.3044	0.3238	0.3238	0.3238	**0.3432**	0.2849
Ionosphere	0.2216	0.2047	0.2893	0.2639	0.2301	0.2639	**0.2978**
Vertebral	0.4232	0.3944	0.4425	0.4040	**0.5001**	0.4713	0.4136
Seeds	0.3404	0.4394	0.4394	0.4111	0.4394	0.4394	0.4394
Parkinsons	0.2428	0.2428	**0.2732**	**0.2732**	0.2428	0.2580	0.2580
Australian	0.3451	0.3408	0.3365	0.3365	0.3365	**0.4355**	0.3192
Blood	0.9192	0.9192	0.9192	0.9192	0.9192	0.9192	**0.9630**
Breast_Cancer	**0.3677**	0.3592	0.3465	0.3465	0.3550	**0.3677**	0.3592
Diabetes	0.6023	0.5791	0.5946	0.6178	0.5868	**0.6255**	0.5984
Lymphography	6.2506	5.0384	7.3012	7.1598	6.6547	5.9274	6.2304
Parkinson	0.2580	0.2275	0.2885	0.2580	0.2580	0.2580	0.2428
Ring	1.4061	1.4105	1.4057	1.4142	1.4130	1.3989	**1.4158**
Titanic	2.6776	2.6776	2.6898	2.7519	2.6871	2.7519	**2.7614**
Towonorm	1.2897	1.2829	1.3130	1.2736	1.2925	1.2957	1.3118
WaveformEW	1.2288	1.2502	1.2448	1.2116	1.2597	1.2448	1.2769
Tic-Tac-Toe	0.6318	0.6845	0.6907	0.7031	**0.7062**	0.6876	0.6597
M-of-n	0.5902	0.5722	0.5812	0.5767	0.5834	**0.6508**	0.6328

in the table. The lowest standard deviation shows the robustness and reliability of the proposed GWOPSO model.

As shown by the experiments resulting in previous tables, we can see that, in twelve datasets, the hybrid GWOPSO can achieve data compactness, this is due to high exploration and exploitation of hybrid GWOPSO that enables it to find the best subset of features, ensuring its robustness and reliability in classification tasks in different datasets in finding the optimal subset of features.

Table 8. Standard deviation fitness result from different optimizers in our experiments

Dataset	GWOPSO	GWO	PSO	WAO	MVO	FFA	GA
Hepatitis	0.0213	0.0325	0.0443	0.0398	0.0475	0.0524	0.0319
Ionosphere	**0.0236**	0.0283	0.0465	0.0277	0.0376	0.0299	0.0726
Vertebral	0.0192	**0.0086**	0.0356	0.0172	0.0511	0.0476	0.0185
Seeds	0.0392	0.0815	0.0759	0.0613	0.0794	0.0660	0.0791
Parkinsons	0.0501	0.0517	0.0477	0.0743	0.0576	**0.0394**	0.0765
Australian	0.0180	0.0189	0.0123	0.0141	0.0137	0.0541	**0.0078**
Blood	0.0338	0.0338	0.0361	0.0368	0.0403	**0.0276**	0.0446
Breast_Cancer	0.0172	0.0126	**0.0064**	0.0076	0.0116	0.0187	0.0131
Diabetes	0.0238	**0.0075**	0.0190	0.0241	0.0145	0.0316	0.0167
Lymphography	1.5429	0.8804	1.3669	1.4496	1.1322	1.7625	1.0943
Parkinson	0.0477	0.0444	0.0560	0.0501	0.0632	0.0486	**0.0388**
Ring	0.0093	0.0099	0.0059	**0.0051**	0.0108	0.0061	0.0102
Titanic	0.0154	0.0154	0.0229	0.0451	0.0187	0.0448	0.0446
Towonorm	0.0067	0.0065	0.0126	0.0043	0.0044	0.0107	0.0108
WaveformEW	0.0248	0.0206	0.0230	0.0191	**0.0425**	0.0114	0.0418
Tic-Tac-Toe	0.0213	0.0396	0.0419	0.0460	0.0434	0.0241	0.0275
M-of-n	0.0541	0.0408	**0.0068**	0.0437	0.0491	0.0721	0.0311

Table 9. Processing time (by second) result from different optimizers in our experiments

	Hepatitis	Ionosphere	Vertebral	Seeds	Parkinsons
GWOPSO	**2.834**	**3.252**	4.584	2.905	3.408
GWO	3.378	5.083	3.6	3.071	3.351
PSO	3.06	4.655	3.679	3.916	3.13
WAO	2.867	3.988	**2.415**	**2.507**	3.083
MVO	3.321	4.595	3.372	3.684	3.254
FFA	4.088	4.672	3.928	3.008	3.342
GA	3.334	4.608	3.079	3.166	**2.602**

	Australian	Blood	Breast Cancer	M-of-n	Diabetes
GWOPSO	**4.473**	4.483	**4.287**	8.151	6.448
GWO	5.742	4.206	5.594	8.161	**4.769**
PSO	5.656	**2.684**	5.906	7.643	5.454
WAO	5.828	4.602	5.278	7.782	6.327
MVO	5.64	5.738	5.95	8.132	6.657
FFA	5.847	6.052	5.49	7.717	7.312
GA	5.436	5.618	5.728	**7.169**	5.79

	Lymphographic	Parkinson	Ring	Titanic	Towonorm	WaveformEW	Tic-Tac-Toe
GWOPSO	**3.067**	3.798	77.112	**7.047**	155.321	48.402	**5.454**
GWO	3.612	4.344	101.007	11.821	127.76	48.051	6.946
PSO	4.759	**3.145**	75.948	10.383	137.247	54.455	6.934
WAO	3.269	4.253	68.716	11.236	722.644	64.347	6.437
MVO	3.785	3.601	76.058	13.286	157.921	49.838	6.7
FFA	4.335	3.599	76.924	12.849	164.873	52.057	6.967
GA	3.599	3.709	**49.956**	7.558	**95.268**	**32.995**	6.642

6. **Conclusion and Future Work.** In conclusion, this paper had a proposal issued concerning hybrid GWOPSO optimizer that is used with the KNN classifier to pick the optimum subset of features for the problem of choice of features. To achieve the balance between exploitation and exportation, we hybridize two optimizers by using the GWO, increase and achieve higher quest space exploration for more iterations. Exploration for a greater number of iterations is accomplished in the exponential method and PSO is used to increase population diversity and maximize production efficiency. To assess the quality and efficacy of the proposed solution, seventeen UCI machine learning database datasets are used to calculate the consistency of the proposed optimizer and to ensure that the proposed solution is reliable and stable. For prospects, we plan to test the proposed solution on more complex datasets. More so, a parallel variant of the GWOPSO model will be availed.

REFERENCES

[1] J. Han, J. Pei and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.

[2] B. Chizi, L. Rokach and O. Maimon, A survey of feature selection techniques, in *Encyclopedia of Data Warehousing and Mining*, 2nd Edition, IGI Global, 2009.

[3] G. Chandrashekar and F. Sahin, A survey on feature selection methods, *Computers & Electrical Engineering*, vol.40, no.1, pp.16-28, 2014.

[4] D. A. Bell and H. Wang, A formalism for relevance and its application in feature subset selection, *Machine Learning*, vol.41, no.2, pp.175-195, 2000.

[5] M. Tubishat, M. A. Abushariah, N. Idris and I. Aljarah, Improved whale optimization algorithm for feature selection in Arabic sentiment analysis, *Applied Intelligence*, vol.49, no.5, pp.1688-1707, 2019.

[6] S. Das, Filters, wrappers and a boosting-based hybrid for feature selection, *ICML*, vol.1, pp.74-81, 2001.

[7] C. H. Yang, L. Y. Chuang and C. H. Yang, IG-GA: A hybrid filter/wrapper method for feature selection of microarray data, *Journal of Medical and Biological Engineering*, vol.30, no.1, pp.23-28, 2010.

[8] M. A. Hall and L. A. Smith, Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, *FLAIRS Conference*, vol.1999, pp.235-239, 1999.

[9] B. Raman and T. R. Ioerger, Instance-based filter for feature selection, *Journal of Machine Learning Research*, vol.1, no.3, pp.1-23, 2002.

[10] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Engineering Software*, vol.69, pp.46-61, 2014.

[11] H. Hassan, A. I. El-Desouky, A. Ibrahim, E. M. El-Kenawy and R. Arnous, Enhanced QoS-based model for trust assessment in cloud computing environment, *IEEE Access*, DOI: 10.1109/ACCESS.2020.2978452, 2020.

[12] J. Kennedy and R. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, vol.4, pp.1942-1948, 1995.

[13] Y. Shi and R. Eberhart, A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation (CEC'98)*, pp.69-73, 1998.

[14] S. Mirjalili and A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.*, vol.95, pp.51-67, 2016.

[15] E. M. Hassib, A. I. El-Desouky, L. M. Labib and E. S. M. El-Kenawy, WOA + BRNN: An imbalanced big data classification framework using whale optimization and deep neural network, *Soft Computing*, pp.1-20, 2019.

[16] E. Emary, H. M. Zawbaa, C. Grosan and A. E. Hassenian, Feature subset selection approach by gray-wolf optimization, in *Advances in Intelligent Systems and Computing*, Cham, Springer, 2015.

[17] T. Ahmad and M. N. Aziz, Data preprocessing and feature selection for machine learning intrusion detection systems, *ICIC Express Letters*, vol.13, no.2, pp.93-101, 2019.

[18] A. Madadi and M. M. Motlagh, Optimal control of DC motor using grey wolf optimizer algorithm, *Technical Journal of Engineering and Applied Sciences*, vol.4, no.4, pp.373-379, 2014.

[19] H. M. Song, M. H. Sulaiman and M. R. Mohamed, An application of grey wolf optimizer for solving combined economic emission dispatch problems, *International Review on Modelling and Simulation*, vol.7, no.5, pp.838-844, 2014.

[20] P. Rathee, R. Garg and S. Meena, Using grey wolf optimizer for image registration, *International Journal of Advance Research in Science and Engineering*, vol.4, no.4, pp.360-364, 2015.

[21] E. M. Hassib, A. I. El-Desouky, E. S. M. El-Kenawy and S. Elghamrawy, An imbalanced big data mining framework for improving optimization algorithms performance, *IEEE Access*, 2019.

[22] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed and O. Aliman, Using the gray wolf optimizer for solving optimal reactive power dispatch problem, *Applied Soft Computing*, vol.32, pp.286-292, 2015.

[23] M. R. Mosavi, M. Khishe and A. Ghamgosar, Classification of sonar data set using neural network trained by gray wolf optimization, *Neural Network World*, vol.26, no.4, p.393, 2016.

[24] M. N. K. Kulkarni, M. S. Patekar, M. T. Bhoskar, M. O. Kulkarni, G. M. Kakandikar and V. M. Nandedkar, Particle swarm optimization applications to mechanical engineering – A review, *Materials Today: Proceedings*, vol.2, nos.4-5, pp.2631-2639, 2015.

[25] G. Guo, H. Wang, D. Bell, Y. Bi and K. Greer, KNN model-based approach in classification, in *Lecture Notes in Computer Science*, Berlin, Heidelberg, Springer, 2003.