IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 12, 2021, accepted February 15, 2021. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.3060096

Hybrid Binary Grey Wolf With Harris Hawks Optimizer for Feature Selection

RANYA AL-WAJIH[1,2], SAID JADID ABDULKADIR[1,2], (Senior Member, IEEE), NORSHAKIRAH AZIZ[1,2], QASEM AL-TASHI[1,3], AND NOUREEN TALPUR[1,2]

[1]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32160, Malaysia
[2]Centre for Research in Data Science, Universiti Teknologi PETRONAS, Seri Iskandar 32160, Malaysia
[3]Faculty of Administrative and Computer Sciences, University of Albaydha, Albaydha CV46+6X, Yemen

Corresponding author: Said Jadid Abdulkadir (saidjadid.a@utp.edu.my)

This work was supported by the Universiti Teknologi PETRONAS, under the Yayasan Universiti Teknologi PETRONAS (YUTP) Fundamental Research Grant Scheme under Grant YUTP-FRG/015LC0-119.

ABSTRACT Despite Grey Wolf Optimizer's (GWO) superior performance in many areas, stagnation in local optima areas may still be a concern. Several significant GWO factors can be explored to enhance the performance of selection in classification, with two conflicting concepts to be considered in using or modeling a metaheuristic method, exploring a search field, and exploiting optimal solutions. Balancing exploration and exploitation in a good manner will improve the search algorithm's performance. To achieve a good balance, this paper proposes a binary hybrid GWO and Harris Hawks Optimization (HHO) to form a memetic approach called HBGWOHHO. The sigmoid transfer function is used to transfer the continuous search space into a binary one to meet the feature selection nature requirement. A wrapper-based k-Nearest neighbor is used to evaluate the goodness of the selected features. To validate the performance of the proposed method, 18 standard UCI benchmark datasets were used. The performance of the proposed hybrid method was compared with Binary Grey Wolf Optimizer (BGWO), Binary Particle Swarm Optimization (BPSO), Binary Harris Hawks Optimizer (BHHO), Binary Genetic Algorithm (BGA) and Binary Hybrid BWOPSO. The findings revealed that the proposed method was effective in improving the performance of the BGWO algorithm. The proposed hybrid method outperforms the BGWO algorithm in terms of accuracy, selected feature size, and computational time. Similarly, compared with BPSO and BGA feature selection algorithms, the proposed HBGWOHHO surpassed them yield better accuracy, the smaller size of selected features in much lower computational time.

INDEX TERMS Classification, feature selection, Grey Wolf Optimizer, hybrid feature optimization, Harris Hawks Optimizer.

I. INTRODUCTION

Nowadays, feature selection or variable selection plays a vital role in machine learning and data mining. The considerable growth in the size of collected data and the massive amount of information is often associated with noisy data or irrelevant information that may affect the systems' performance and accuracy [1]. Feature selection is a preprocessing stage aiming to improve the relevancy of gathered data by reducing unnecessary features and selecting only useful or relevant variables. In some applications, such as physiological application, some other transforms are used for removing noisy data, such as multiscale principal component analysis (MSPCA) as mentioned in [2] and [3]. However, the exponential increase in the search space remains one of the biggest challenges. Some methods, such as the empirical wavelet transform (EWT) based signal decomposition methods, improve the classification accuracy of motor imagery-based electroencephalography signals. Such a transform was reported in [4] to analyze non-stationary and nonlinear signal behaviors successfully. In a dataset of n variables, the entire number of possible solutions is 2^n [1], [4]. The feature selection process generally involves assessing the feature subsets, finding the optimum feature subset by applying some search techniques, evaluating the selected features, stopping criteria, and subsets validation [5]. Feature selection

The associate editor coordinating the review of this manuscript and approving it for publication was Isaac Triguero.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

evaluation measures such as filter approach and wrapper approach are used to assess the generated subsets efficiency.

Filter methods evaluate the goodness of the features based on the properties of the training data. Such methods do not involve any machine learning algorithms. Filter techniques mostly consider all features' score, and then features with the highest score are selected. However, some filter techniques tend to choose the highest score of features per iteration [6]. Some other popular approaches, such as correlation-based feature selection strategy in [7] and dimension reduction techniques and neural networks as in [8] are also known to be robust in reducing the system complexity and computational load. Despite their fast and less expensive computation, filter methods ignore the selected features' performance [9].

Wrapper methods use an evaluating algorithm to measure the quality of selected features [4]. Common wrappers in the context of feature selection include decision tree (DT) [10], support vector machines (SVMs) [11], Naïve Bayes (NB) [10], K-nearest neighbor (KNN) [12], artificial neural networks (ANNs) [13], linear discriminant analysis (LDA) [14], local geometric structure Fisher analysis (LGSFA) [15] and local neighborhood structure preserving embedding (LNSPE) [16]. The results obtained by wrapper techniques are commonly better than those of filter methods.

The search techniques in feature selection have been developed over time to obtain optimum solutions. The traditional search methods, including sequential search (SFS) [17] and (SBS) [18] have been proposed in the early stage of feature selection [19]. These methods have several drawbacks such as nesting effect, stagnating in local optima, and expensive computation [4]. Floating search such as sequential forward floating search (SFFS) and sequential backward floating search (SBFS) emerged in an attempt to overcome the nesting effect, where restoring deleted features in advance stages is not possible [20]. However, floating search methods often fail in high dimensional search space [21].

The revolution in feature selection gave rise to more sophisticated search algorithms called metaheuristic algorithms. For example, there is continuous efforts to improve performance of Evolutionary algorithms (EA) [22]–[24] including Genetic algorithm (GA) [25], and Swarm Intelligence (SI) such Particles Swarm Optimization (PSO) [26], Ant Colony Optimization (ACO) [27] and Artificial Bee Colony (ABC) [28]. Furthermore, newer algorithms were developed such as Grey Wolf Optimizer (GWO) [29], Grasshopper Optimization algorithm (GOA) [30], Butterfly Optimization Algorithm (BOA) [31], Ant Lion Optimization (ALO) [32], Whale Optimization Algorithm (WOA) [33] and Harris Hawks Optimization (HHO) [34].

Depending on the exploration phase and exploitation phase, the metaheuristic algorithms are categorized into a single solution based (i.e., Microcanonic Annealing (MA) [35], Simulated annealing (SA) [36] and Tabu Search (TS) [37]) or population size based (i.e., GA [25], PSO [26] and ACO [27]). Single solution-based type (S- metaheuristics), processes only one solution at a time in the optimization phase. In contrast, the population size-based metaheuristics algorithms can process several solutions at a time. S-metaheuristics generate a set of solutions in the induced phase, replacing the current solution in the replacement phase with a solution set [38]. P-metaheuristics begin the searching with initial population size, and then the generated solutions start replacing the existed population iteratively [38]. This procedure continues until the stopping criterion is met. Generally, P-metaheuristics are further categorized into four main types [33], [34], [39]; Evolutionary Algorithms (EAs), Swarm Intelligence, (SI) Physics-based, and Human-based algorithms.

GWO [29] is a P-metaheuristic algorithm that mimics grey wolves' behavior in finding and hunting the prey. GWO stores the three best solutions, then update the new solution according to stored locations. The algorithm is simple (has few adjustable operators), but GWO still suffers from stagnating in local solutions [29]. Two binarization methods were proposed in [40] to allow GWO to solve binary problems. Moreover, in [41] multi-strategy ensemble, GWO (MEGWO) defines more search operators to update the search agents' locations. In [42], GWO was converted into binary then include two-phase mutation to compute the most informative features. Binary GWO (BGWO) was applied in many areas such as oil and gas [43], software defect problems [44], and the medical domain [45], [46]. A brief review for GWO for feature selection can be found in [46].

HHO [34] was developed in 2019 to imitate the Harris hawks hunting style called. The exploration phase of HHO starts with initializing a value for the hawk to detect the perching location then observing the prey. When the exploitation phase begins, the hawks performed four attacking styles according to the prey's energy and its possibility of escaping. The superior performance of HHO [34] brought attention to this algorithm. In [47], hybrid HHO-SVM and HHO-KNN were applied to obtain the optimum subsets for chemical drugs. Another study [48] aimed to improve the HHO algorithm to detect high-quality features by embedding the SSA algorithm in the native HHO called IHHO. In [49], a new Quadratic BHHO was proposed to solve classification tasks (feature selection).

Despite the superior performance of Grey Wolf Optimizer (GWO) in many fields, it still may face the risk of stagnation in local optima. Balancing exploration and exploitation in a good manner could improve the search algorithm's performance. To achieve a good balance, this paper proposes a binary hybrid GWO and Harris Hawks Optimization (HHO) to form a memetic approach called HBGWOHHO. The sigmoid transfer function is used to transfer the continuous search space into a binary one to meet the feature selection nature requirement. A wrapper-based k-Nearest neighbor is used to evaluate the goodness of the selected features. To validate the performance of the proposed method, 18 standard UCI benchmark datasets were used. The algorithm removed the redundant features, and only representative features were selected. The proposed hybrid method's performance was compared with BGWO and Binary Particle Swarm Optimization (BPSO) and Binary Genetic Algorithm (BGA).

Despite the success of Grey Wolf Optimizer (GWO), its exploration phase still suffers from a local stagnation problem or premature convergence, which can result in a wrong solution [8]. In this study, a more efficient exploration phase for GWO is proposed, aiming to resolve the immature convergence issue. The proposed approach represents a hybridization between GWO and HHO algorithms, where the exploration phase from HHO is utilized, while the exploitation phase is adopted from native GWO. This work's main contribution is to improve the exploration phase of GWO to enhance feature selection accuracy and reduce computational time. The adopted approach of hybridizing GWO and HHO algorithms is mainly motivated by the simplicity and effectiveness of GWO and the high accuracy results produced by HHO. Combining these two algorithms is anticipated to provide a robust and simple feature selection algorithm.

The other sections of the paper are as follows: section 2 discusses the related works. Section 3 presents a review of the GWO algorithm and HHO algorithm basics. In Section 3 the details of the proposed algorithm are given. The analysis of the experimental results is presented and discussed in section 4. Finally, section 5 offers the conclusion.

II. RELATED WORK

Several researchers adopt hybrid algorithms to solve many problems in recent years. Hybrid metaheuristics algorithms were reported to have high performance as indicated in [38] and [50]. As shown in [38], Hybrid algorithms are classified into high-level and low-level. There are two hybridization schemes in high-level hybrid algorithms: high-level rely on hybridization (HRH) and High-level teamwork hybridization (HTH). In HRH, the self-contained metaheuristics are implemented in sequence, while in HTH one algorithm assists another algorithm by providing information using cooperative search. Similarly, low-level hybridization is categorized into low-level rely on hybrid (LRH) and low-level teamwork hybrid (LTH), where an algorithm is contained inside another metaheuristic algorithm.

In feature selection, the performance of some hybrid algorithms was reported to be better than that of the native algorithms [51]. The first use of hybrid metaheuristics algorithm in feature selection is dated back to 2004 [52], where the search process was controlled by developing a GA algorithm with local search techniques. A combination of with EGA filter was introduced in a wrapper approach for text categorization. Many studies applied hybrid models on GA, such as [53], which employed the CS and GA to enhance the classification accuracy. Another study [54] hybridized the GA and ACO to improve the search capability. In [54], [55], and [56] the GA algorithm has been developed and hybridized using the simulated annealing technique.

Moreover, PSO also gained attention in hybridization. For example, in [57] PSO-SVM was proposed to increase the classification accuracy. In [58], PSO based Relative Reduct (PSO-RR) and PSO based Quick Reduct (PSO-QR) were introduced to improve the efficiency of the predictive accuracy. Also, in [59], the PSO algorithm combined with GSA in PSO-GSA was proposed to balance exploration and exploitation capability. In [60], it was hybridized with SA local search to improve PSO convergence. Similarly, there are studies on GWO hybridization, such as GWO-GOA, that minimized the computational time in text feature selection and text clustering [61]. In [62], GWO and GA were hybridized and used to solve the SCOS problem. Hybrid GWO-SVM proposed in [63] to enhance the accuracy in (PSLF) for (RSEDs). Also, GWO has been hybridized with many other methods such as hybrid GWO-SCA for optimization problems [64], GWO-ABC to enhance the premature convergence, and hybrid DE-GWO [65]. Besides, a hybrid fuzzy feature selection algorithm has been introduced to the Fuzzy Rule-Based Systems (FRBSs) to discover the relevant features [65].

Recently, many metaheuristic algorithms were developed using a hybrid method in feature selection. In [50], the WOA algorithm was hybridized with SA to find better search regions and enhance the algorithm's exploitation. The GWO and PSO wrapper-based algorithm's hybridization in [66] showed an enhancement in the selected features' accuracy. Hybrid MIMAGA in [67] has been applied to improve the selection of the gene expression data. In [68], IGWO-KELM was developed to find the optimal solution in medical diagnoses. Furthermore, a hybrid filter-based V-WSP with wrapper based PSO was proposed in [69] to improve selecting features' efficiency. The same study also introduced the Information Gain binary Butterfly Optimization Algorithm (IG-bBOA), another hybrid feature selection method that aims to eliminate the irrelevant and redundant features.

Despite the tremendous noticeable improvement of the feature selection algorithms, but as it is known in the No Free Lunch (NFL) theorem in the optimization field, there is no optimum algorithm for solving all types of problems, and there will not be. Even with the enhancement that the wrapper feature selection algorithm showed, there is still a chance for more development to find better solutions. A new hybrid wrapper-based feature selection will be discussed in the next section.

III. METHODS

A. GREY WOLF OPTIMIZER

GWO [29] is a metaheuristic algorithm proposed to mimic rey wolves ppacks' social behavior The leadership hierarchy of the grey wolf's pack has four different levels Alpha (α), Beta (β), Delta (δ), and Omega (ω). Alpha (α) dominance is the highest, while Omega (ω) has the lowest dominance in the hierarchy. Alpha (α), which could be either a male or female, is the pack leader and considered the decision-maker. Beta (β) follows and assists α but dominates Delta (δ). The pack scouts, sentinels, caretakers, and retired α are included in δ hierarchy level. The least dominant, ω, is referred to as the ''scapegoat'' of the pack and must follow all other levels (α, β, and δ) of the grey wolf's pack. In the mathematical implementation of GWO, there are three main steps, namely encircling the prey, hunting the prey, and attaching the prey. The distance from the prey and the search agents' position (grey wolves pack) is determined in the encircling step. The mathematical modelling of the encircling step in GWO is presented in Eq (1) through Eq (4):

$$\vec{D} = \left|\vec{C}.\vec{X}_P(t) - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A} \tag{2}$$

where $\vec{X}_P$ represent the position of the prey, t is the current iteration number, $\vec{A}$, and $\vec{C}$ are coefficient vectors and modeled as:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r}_2 \tag{4}$$

where $\vec{r}_1$, $\vec{r}_2$ are random vectors used to give randomness to the algorithm and the value of the parameter $\vec{a}$ linearly decreased from 2 to 0 and controlled the transition from the exploration phase into the exploitation phase throughout iterations.

In the hunting step of GWO, the position of alpha is the optimum position. Beta and Delta are expected to know the prey's position [66]. The best solutions obtained are saved and other grey wolves (represented by Omega in the hierarchy) are obliged to change their position accordingly. The exploitation phase starts when the absolute value of $\vec{A}$ is less than 1 ($|\vec{A}| < 1$) and the mathematical representation of updating positions of the search agent is expressed in the following equations:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{5}$$

where $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ are the positions of the three best search agents alpha beta and delta and it is modelled as:

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.\left(\overrightarrow{D_\alpha}\right) \tag{6}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.\left(\overrightarrow{D_\beta}\right) \tag{7}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.\left(\overrightarrow{D_\delta}\right) \tag{8}$$

where $\vec{D}_\alpha$, $\vec{D}_\beta$ and $\vec{D}_\delta$ represent the distance used to update the position of the three best search agents.

$$\vec{D}_{\boldsymbol{\alpha}} = |\vec{C}_1.\vec{X}_{\boldsymbol{\alpha}} - \vec{X}| \tag{9}$$

$$\vec{D}_{\boldsymbol{\beta}} = |\vec{C}_2.\vec{X}_{\boldsymbol{\beta}} - \vec{X}| \tag{10}$$

$$\vec{D}_{\boldsymbol{\delta}} = |\vec{C}_3.\vec{X}_{\boldsymbol{\delta}} - \vec{X}|. \tag{11}$$

B. HARRIS HAWKS OPTIMIZATION ALGORITHM

HHO is another metaheuristic algorithm [34] that was inspired by the hunting behavior of the Harris Hawks, often called ''surprise pounce'' or ''the seven kills''. Depending on the nature of the prey's escaping patterns, the hunting process could be completed in a few seconds or might take several hours. The modeling algorithm of HHO consists of two main stages (exploration and exploitation). The exploration stage refers to the process of perching and detecting prey. HHO algorithm considers Harris' hawks as the candidate solutions, where, in each step, the intended or optimum prey is represented by the best candidate solution [34]. The algorithm simulates the Harris' hawks perching strategies considering two possible conditions. The first possibility considers that Harris' hawk would perch on random locations within their group home range. This condition is modeled by $q \geq 0.5$ in Eq (12).

$$\overrightarrow{X_1}(t+1) = \begin{cases} X_{rand}(t) - r_1 \left|X_{rand}(t) - 2r_2X(t)\right|, & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) & \\ \quad - r_3(LB + r_4(UB - LB)), & q < 0.5. \end{cases} \tag{12}$$

The second possibility is that Harris' hawks would perch on the positions near other family members and the prey. Such condition is modelled in Eq (12) for $q < 0.5$:

where $\vec{X}_1(t+1)$ is the position vector of hawks, t is the next iteration, $X_{rand}(t)$ is a randomly selected hawk from the current population, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4, and q are random numbers inside (0,1), X(t) is the current position vector of hawks $X_{rabbit}(t)$ is the position of rabbit, X_m The average position of the current population of hawks and LB and UB are the upper and lower of variables.

In the exploitation phase, Harris' hawks attack the prey that was identified in the previous step. There are four possibilities in the HHO algorithm to model the different attacking styles followed by Harris' hawks. Assuming r represents the chance of prey to escape, successful escape is donated by $r < 0.5$, while $r \geq 0.5$ represents a failure to escape. The hawks will either perform soft besiege or hard besiege to catch the prey, depending on the prey's escaping chance (r). The parameter E in the HHO algorithm is utilized to determine the type of attacking besiege. If prey would possibly fail to escape ($r \geq 0.5$), soft besiege is adopted if $|E| \geq 0.5$, while hard besiege happens when $|E| < 0.5$. The mathematical representation of the soft besiege is shown in Eq (13) through Eq (14), and the hard besiege is expressed in Eq (15):

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \tag{13}$$

$$\Delta(t) = X_{rabbit}(t) - X(t) \tag{14}$$

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)|. \tag{15}$$

In case of possible successful escape ($r < 0.5$), soft besiege with progressive rapid dive occurs when $|E| \geq 0.5$ Eq.(16), (18), (19) and (26), whereas hard besiege with progressive rapid dive is applied when $|E| < 0.5$ Eq. (17), (18), (19) and (26).

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \tag{16}$$

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \tag{17}$$

$$Z = Y + S \times LF(D) \tag{18}$$

$$X(t+1) = \begin{cases} Y, & \text{if } f(Y) < F(X(t)) \\ Z, & \text{if } f(Z) < F(X(t)) \end{cases} \tag{19}$$

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}},$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \tag{20}$$

IV. THE PROPOSED HYBRID ALGORITHM (BGWOHHO)

The boundaries of feature selection search space are zero, and one or, in other words, feature selection is a binary problem. The native GWO algorithm has a continuous search space. Therefore, it is not possible to use native GWO for solving the feature selection problem. A modified version (binary version) of the algorithm should be developed. The following sigmoid function was included to update locations of search agents for the GWO algorithm in a binary search space:

$$\boldsymbol{X_{binary}(t+1) = f(x) = \begin{cases} 1, & sigmoid(\frac{X_1+X_2+X_3}{3}) \geq r \\ 0, & otherwise \end{cases}} \tag{21}$$

where $\boldsymbol{X_{binary}(t+1)}$ is the binary updating location mechanism, sigmoid function $\boldsymbol{S(x)}$ represented in eq(22) and r is a random value obtained from a uniform distribution $\in [\mathbf{0}, \mathbf{1}]$

$$\boldsymbol{S(x) = \frac{1}{1+e^{(-10*(x-0.5))}}} \tag{22}$$

Grey Wolf Optimizer (BGWO) provides better results than other well–known heuristics methods such as PSO, GSA, DE, EP, and ES [70]. However, the native BGWO algorithm's exploration phase still suffers from a local stagnation problem or premature convergence that can result in a wrong solution [8]. In the present study, a more efficient exploration phase for BGWO is proposed, aiming to resolve the immature convergence issue. This approach represents a hybridization between BGWO and HHO algorithms, where the exploration phase from HHO is utilized using Eq (12), while the exploitation phase of BGWO is adopted Eq(5). The developed hybrid algorithm (BGWOHHO) has a higher chance of skipping out of local optimum solutions, hence increasing solution accuracy.

Figure 1 provides a schematic of the proposed hybridization approach that was followed to improve the GWO search technique. In the proposed hybrid model (HBGWOHHO), the search starts with initializing the coefficient vector A value.

If $|A| \geq 1$, the exploration phase begins, the search agent intends to choose its location based on two strategies. If we assume there is equal chance r for each strategy, the first strategy is to be near other family members positions, for the condition of $\boldsymbol{r < 0.5}$ in eq (23), the other strategy is to choose a random position within the group range as modeled in eq(23) for the condition of $\boldsymbol{r \geq 0.5}$.

$$\vec{X}(t+1) = \begin{cases} X_r(t) - C\,|X_r(t) - 2r_1X(t)|, & r \geq 0.5 \\ (X_P(t) - X_m(t)) \\ \quad - C(LB + r_2(UB - LB)), & r < 0.5 \end{cases} \tag{23}$$

$$\vec{C} = 2.\vec{r}_3. \tag{24}$$

Where $\vec{X}\,\boldsymbol{(t+1)}$ is the next position of the search agent, $\boldsymbol{X_r(t)}$ is a random search agent from the current population, $\boldsymbol{X_P(t)}$ indicates the position of the prey, $\boldsymbol{X_m(t)}$ represents the average position of the search agents, C is a random coefficient calculated from eq (24), r, r_1, r_2 are random coefficients $\in [0-1]$. The parameter C allows the algorithm to exhibit a high level of randomness to avoid local optima. The parameters r, r_1 and r_2 are scaling coefficients to increase further the random way the algorithm generates random locations within the range between LB to UB.

When $|A| < 1$, the search agent will be obliged to update its position according to the position of the three best agents found so far using Eq (21). The parameters X_1, X_2, and X_3 are the three best solutions and mathematically expressed using Eq (6) through Eq (8).

Based on the energy of the prey, the parameter $\vec{A}$ is used to control the exploration phase and exploitation phase since the value of $\vec{A}$ is a random constant in the range from $-2a$ to 2a, where $\vec{a}$ is a random value linearly decreased from 2 to 0 through the number of iterations. When the $|\boldsymbol{A}| < \mathbf{1}$ that means the prey lost its energy and stopped moving then the search agent start attacking (exploitation), while if $|\boldsymbol{A}| \geq \mathbf{1}$ that means the prey is not applicable to be attacked and the search agent will attend to explore new search space (exploration). The $|A|$ is determined by eq (25.)

$$|\vec{A}| = |2\vec{a}.\vec{r}_1 - \vec{a}| \tag{25}$$

The pseudo-code of the proposed HBGWOHHO is presented in Algorithm 1.

Feature selection generally aims to minimize the number of features and classification error rates [66]. In other words, the classification accuracy is maximized by removing irrelevant and redundant features and retaining only relevant features [50]. In the present study, the KNN classifier was utilized due to its simplicity in evaluating the fitness function [71], [72]. The fitness function used is expressed in Eq (26), which was adopted.

$$\text{fitness} = \alpha\,(ER) + (1-\alpha)\frac{|S_f|}{|T_f|} \tag{26}$$

where $\alpha = [0, 1]$ is constant to control accuracy [37], ER is the error rate, $\boldsymbol{S_f}$ is the number of selected feature and $\boldsymbol{T_f}$ is total number of features.

V. EXPERIMENTAL RESULTS

A. DATASETS

Table 1 outlines the benchmark datasets that were used to check the validity of the proposed HBGWOHHO. These datasets were obtained from the UCI machine learning repository [73].

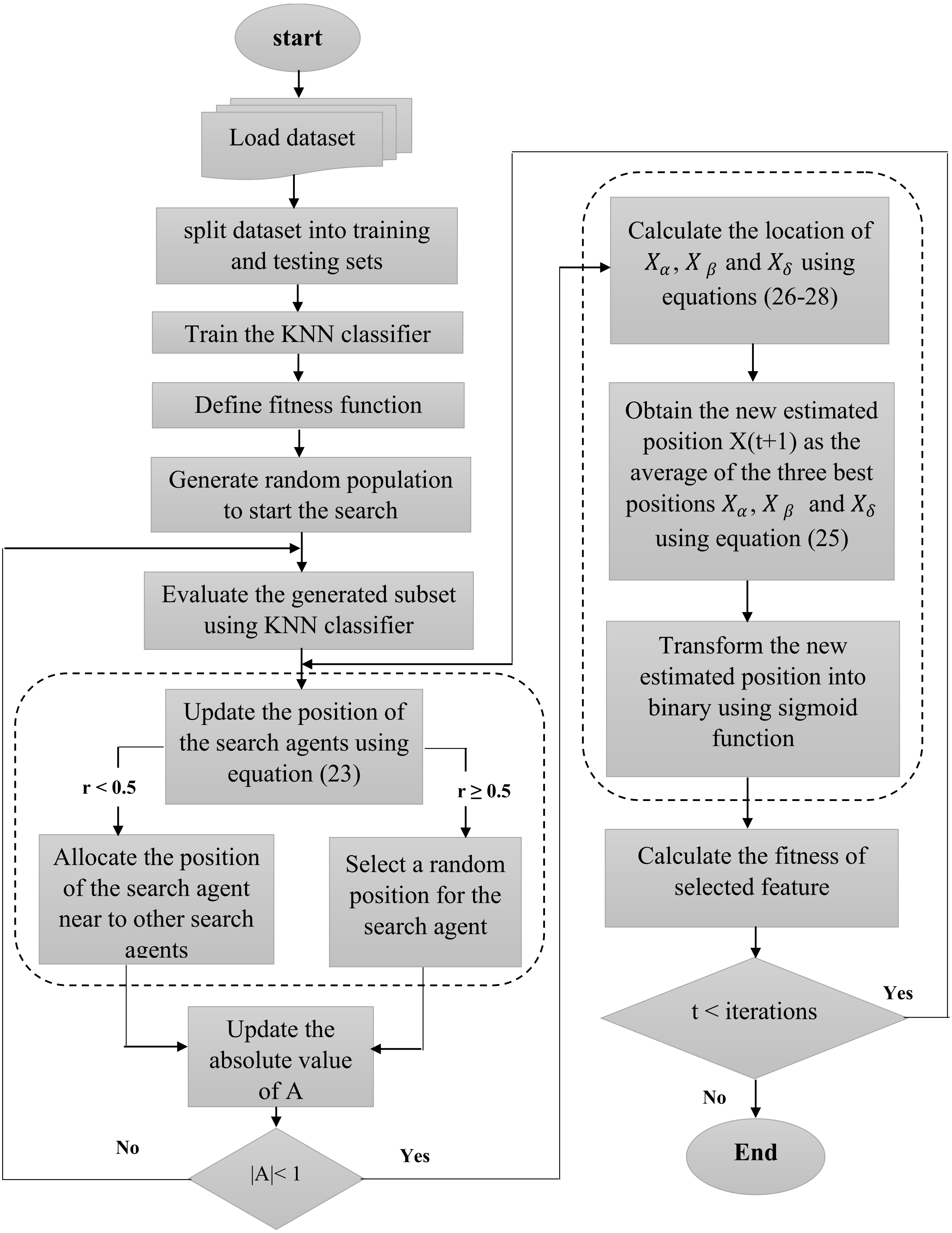


FIGURE 1. Schematic of the proposed hybridization method.

Algorithm 1 Pseudo-Code of HBGWOHHO Algorithm

Input: number of search agent (N) and Number of iterations (T)
Output: the location of the fittest solution and its value
Initialize search agents population
initialize a, A, C, and r
Calculate the fitness value of the search agents X_α, X_β and X_δ
While (t < maximum no. of iterations)
  Set X_P as the location of the prey (Best position)
  For (each iteration)
Evaluate the fitness value of the search agents
End for
For (each agent)
    Update the absolute value of A using Eq (3)
    If (|A| ≥ 1) then (Exploration phase)
      Initialize a random value of q
        If $r \geq 0.5$
update the position of the search agents to be perched on a random position
using Eq (23)
else if $r < 0.5$
update the position of the search agents to be perched near to other search agents positions
using Eq (9)
end if
   else if (|A| < 1) (Exploitation phase)
         update the location of α, β, and δ
calculate the new estimated position X(t + 1) using eq(33)
transform the new estimated position X(t + 1) into binary by applying sigmoid function Eq(22)
end if
calculate the fitness of the search agents
  t = t + 1
End for
End while
Return X_P

B. PARAMETERS SETTINGS

The K-nearest neighbor (KNN) algorithm with k = 5 and Euclidean distance was used to evaluate the best solutions. The value of K is utilized based on a trial and error basis. In the experiments, the value of k = 5 was selected as it has the best performance in all datasets. This procedure was adopted from [50], [66], [71]. Partitioning the datasets into K segments was achieved by utilizing the K-fold cross-validation as in [74]. One segment is used as a testing set, while the other segments (K-1) are used for training sets [75].

The training and testing datasets segments are shuffled and repeated randomly until the best solution is obtained [74].

It is noteworthy to mention that the iteration number (T) and population size (N) are two key factors that influence the practicality of a particular algorithm, as indicated in [12]. The parameters including the value of k, N, UB, LB, and T are set following standard parameter setting outlined in [50] and [66]. Table 2 outlines the parameters' setting for the proposed HBGWOHHO algorithm. Intel(R) CoreTMi7-6700 machine with 3.4GHz CPU, and 8GB of RAM was used to run the algorithm. The same machine was utilized to run the original code BGWO obtained from [70], and BHHO from [49] and results were compared. Further comparisons were made between the HBGWOHHO algorithm and BHHO, BPSO, BGA [66], [50], and BGWOPSO algorithms. The findings were analyzed and compared based on classification accuracy, mean fitness, best fitness, worst fitness, number of selected features, and computational time.

TABLE 1. Benchmark datasets used.

#	Dataset	Number of Instances	Number of features
1	Breastcancer	699	9
2	BreastEW	569	30
3	CongressEW	434	16
4	Exactly	1000	13
5	Exactly2	1000	13
6	HeartEW	270	13
7	IonosphereEW	351	34
8	KrvskpEW	3196	36
9	Lymphography	148	18
10	M-of-N	1000	13
11	PenglungEW	73	325
12	SonarEW	208	60
13	SpectEW	267	22
14	Tic-tac-toe	958	9
15	Vote	300	16
16	WaveformEW	5000	40
17	WineEW	178	13
18	Zoo	101	16

TABLE 2. Parameter setting for HBGWOHHO.

Parameter	Value
Population size (N)	10
Upper search bound (UB)	1
Lower search bound (LB)	0
Number of iterations (T)	100
Number of repeated runs (M)	20
Fitness function constant (α)	0.99

C. ASSESSMENT METRICS

To evaluate the quality of generated solutions, we used statistical metrics including average accuracy, mean fitness, worst fitness, average selected feature size, and average computational time.

1) THE AVERAGE ACCURACY

This metric refers to the precision of the classifier in selecting the most relevant features. When the algorithm is run M times, its average accuracy is calculated as in Eq (27):

$$Average\ Accuracy = \frac{1}{M}\sum\nolimits_{k=1}^{M} Accuracy^{K} \tag{27}$$

where $Accuracy^{K}$ is the accuracy achieved at run k.

2) MEAN FITNESS

This metric refers to the average value of the achieved fitness function. When the algorithm is run M times, its mean fitness is calculated as in Eq (28):

$$Mean\ fitness = \frac{1}{M}\sum\nolimits_{k=1}^{M} g_{*}^{k} \tag{28}$$

where g_{*}^{k} is the value of fitness function achieved at run k

$$Mean\ fitness = \frac{1}{M}\sum\nolimits_{k=1}^{M} g_{*}^{k} \tag{29}$$

where g_{*}^{k} is the value of fitness function achieved at run k

3) BEST FITNESS

This metric refers to the minimum value of the fitness function. When the algorithm is run M times, its best fitness is calculated as in Eq (30):

$$Best\ fitness = Min_{K=1}^{M} g_{*}^{k} \tag{30}$$

where $\boldsymbol{g_{*}^{k}}$ is the optimal fitness value achieved at run k.

4) BEST FITNESS

This metric refers to the minimum value of the fitness function. When the algorithm is run M times, its best fitness is calculated as in Eq (31):

$$Best\ fitness = Min_{K=1}^{M} g_{*}^{k} \tag{31}$$

Where $\boldsymbol{g_{*}^{k}}$ is the optimal fitness value achieved at run k.

5) WORST FITNESS

This metric refers to the maximum value of the fitness function. When the algorithm is run M times, its worst fitness is calculated as in Eq (32):

$$Worst\ fitness = Max_{K=1}^{M} g_{*}^{k} \tag{32}$$

Where $\boldsymbol{g_{*}^{k}}$ is the optimal fitness value achieved at run k.

6) AVERAGE SELECTED FEATURES SIZE

This metric refers to the average size of the selected features to the total number of features. When the algorithm is run M times, its average selection is calculated as in Eq (33):

$$Average\ selection = \frac{1}{M}\sum\nolimits_{k=1}^{M} \frac{Avg.size^{k}}{T_f} \tag{33}$$

where T_f is the total number of features, $Avg.size^{k}$ is the average size of selected features achieved at run k.

7) AVERAGE COMPUTATIONAL TIME

This metric refers to the average computational time in seconds. When the algorithm is run M times, its average computational time is calculated as in Eq (34):

$$Average\ time = \sum\nolimits_{k=1}^{M} Avg.time^{k} \tag{34}$$

Where $\boldsymbol{Avg.time^{k}}$ average computational time spent at run k.

VI. RESULTS AND DISCUSSION

This section discusses the results given by the proposed hybrid algorithm.

A. HBGWOHHO ALGORITHM

Table 3 presents the results obtained by applying the proposed HBGWOHHO algorithm across 18 datasets. The results presented include average accuracy values, mean fitness, feature selected size, and computational time. For each dataset, the algorithm was run 20 times. The highest reported average accuracy was 100%, which was achieved in PenglungEW and WineEW datasets. This was followed by an average accuracy of 99% in CongressEW, BreastEW, Breastcancer datasets, and 98% in Vote dataset. The two datasets, Zoo and KrvskpEW, showed an average accuracy of 97%. In terms of feature selected, the smallest size (4.3 features) was reported in Exactly2 dataset. It was followed by Breastcancer and Tic-tac-toe datasets with a selected feature size of 4.9 and 5.6 features, respectively. The last time was spent on PenglungEW, CongressEW, SonarEW, and Zoo with a computational time of 2.49, 2.54, 2.59, and 2.59 seconds.

B. COMPARISON OF PROPOSED HBGWOHHO ALGORITHM WITH RELATED ALGORITHMS

HBGWOHHO algorithm is proposed here to resolve premature convergence associated with the exploration phase of the BGWO algorithm. Table 4 through Table 9 provide a statistical comparison between the proposed HBGWOHHO and some other wrapper-based algorithms in term of classification accuracy, mean fitness, best fitness, worst fitness, selected feature size, and computational time.

Table 4 compares the proposed HBGWOHHO and other algorithms in terms of classification accuracy. By comparing with native BGWO, it is evident that the proposed HBGWOHHO yielded a better classification accuracy in 16 datasets out of the 18 tested datasets. Besides, the proposed HBGWOHHO algorithm outperformed BPSO and BGA algorithms across all the 18 tested datasets.

The proposed HBGWOHHO has also performed better than BGWO, BPSP, and BGA in terms of mean fitness (Table 5), and best fitness (Table 6) in all the 18 Exactly and M-of-n datasets.

Table 7 also shows that the proposed HBGWOHHO algorithm is not worse than any of the other algorithms across all the 18 tested datasets. In terms of the selected features' size as shown in Table 8, the proposed HBGWOHHO algorithm provided better results than other algorithms in most of the

TABLE 3. Results obtained by the proposed HBGWOHHO represented by Average (AVG) values and the corresponding standard deviation (SD) algorithm.

#	Dataset	Accuracy		Fitness		Feature selected size		Computational time	
		AVG	SD	AVG	SD	AVG	SD	AVG	SD
1	Breastcancer	0.99	0.00	0.00	0.00	4.9	0.96	2.7	0.83
2	BreastEW	0.99	0.01	0.01	0.01	16.9	2.30	2.8	0.71
3	CongressEW	0.99	0.01	0.01	0.01	7.9	2.03	2.5	0.81
4	Exactly	0.74	0.03	0.26	0.03	7.1	2.10	2.7	0.80
5	Exactly2	0.77	0.01	0.23	0.01	4.3	2.37	2.8	0.80
6	HeartEW	0.90	0.02	0.10	0.02	7.8	1.25	3.3	0.63
7	IonosphereEW	0.93	0.01	0.07	0.01	15.3	2.62	2.7	0.84
8	KrvskpEW	0.97	0.01	0.03	0.01	20.3	2.80	3.2	0.92
9	Lymphography	0.90	0.02	0.10	0.02	9.9	2.20	3.5	1.02
10	M-of-n	0.93	0.04	0.07	0.04	8.2	1.40	2.9	0.83
11	PenglungEW	**1.00**	0.01	0.00	0.01	161.4	8.50	2.4	0.68
12	SonarEW	0.92	0.02	0.08	0.02	30.4	3.50	2.5	0.70
13	SpectEW	0.91	0.02	0.09	0.02	11.8	2.13	2.6	0.72
14	Tic-tac-toe	0.79	0.02	0.21	0.02	5.6	1.24	3.1	0.86
15	Vote	0.98	0.01	0.02	0.01	7.9	1.35	2.8	0.80
16	WaveformEW	0.84	0.01	0.17	0.01	22.5	2.85	4.0	1.40
17	WineEW	**1.00**	0.00	0.00	0.00	7.8	1.10	3.2	0.60
18	Zoo	0.97	0.02	0.03	0.02	8.9	1.45	2.6	0.83

datasets. In 13 datasets, HBGWOHHO provided a smaller number of selected features than BGWO, BPSO, and BGA algorithms. From the reported statistical results, it is evident that the proposed method has improved the BGWO algorithm's performance. The proposed HBGWOHHO showed an outstanding ability to select a smaller number of features while maintaining high accuracy.

Furthermore, the computational time (in seconds) of the proposed HBGWOHHO. (Table 9) is far less than that of native BGWO across all the 18 tested datasets. It is noteworthy to mention the average accuracy and fitness values of BHHO, and hybrid GWOPSO were slightly better than that of the proposed HBGWOHHO in some of the datasets.

C. COMPARISON OF ALGORITHMS' PERFORMANCE AT DIFFERENT DATA SIZES

Further assessment for the proposed algorithm versus other algorithms at different sizes was conducted. The 18 tested datasets were categized according to [54] into small datasets (with 0-15 features), medium datasets (with 16-25 features), and large datasets (with 30-350 features).

The classification accuracy and selected feature size achieved by the proposed HBGWOHHO and other algorithms were analyzed across the three categories.

In small datasets, as shown in Figure 2, the classification accuracy (refer to Figure 2a) achieved by the proposed HBGWOHHO is better than that attained by BGWO, BPSO, and BGA in most of the datasets. Figure 2b demonstrates that HBGWOHHO produced the least number of selected features compared to BGWO, BPSO, and BGA in all the small datasets For medium-size datasets, as shown in Figure 3, the classification accuracy (refer to Figure 3a) achieved by the proposed HBGWOHHO is better than that accomplished by BGWO, BPSO, and BGA in all the datasets. Figure 3b also reveals that HBGWOHHO selected the lowest number of features as compared to BGWO, BPSO, and BGA in all the medium datasets. In large datasets, accuracy (refer to Figure 4a) achieved by the proposed HBGWOHHO is better than that accomplished by BGWO, BPSO, and BGA in all six datasets. Figure 4b also reveals that HBGWOHHO selected the lowest number of features as compared to BGWO, BPSO, and BGA in the majority of the large size datasets.

Such findings demonstrate the proposed algorithm's excellent performance in small, medium, and large size compared to the native BGWO. The developed algorithm shows superior results compared to other state-of-art algorithms, including BPSP and BGA. It is essential to mention that in small, medium, and large datasets, the accuracy and feature size of BHHO and BGWOPSO were slightly better than that of HBGOHHO.

D. SIGNIFICANCE ANALYSIS

Wilcoxon test was used to assess the statistical significance of the differences between the resulting mean fitness obtained by proposed HBGWOHHO versus other optimizers. The

TABLE 4. Comparison between classification accuracy and corresponding standard deviation (SD) of proposed HBGWOPSO and related work methods.

Dataset	HBGWOHHO		BGWO		BHHO		BPSO		BGA		BGWOPSO	
	AVG	SD	AVG	SD	AVG	SD	AVG	SD	AVG	SD	AVG	SD
Breastcancer	**0.99**	0.00	0.97	0.00	0.97	0.01	0.95	0.00	0.96	0.00	0.98	0.00
BreastEW	**0.99**	0.01	0.97	0.00	0.99	0.01	0.94	0.00	0.94	0.01	0.97	0.00
CongressEW	**0.99**	0.01	0.96	0.00	0.98	0.00	0.94	0.00	0.94	0.00	0.98	0.00
Exactly	0.74	0.03	**0.96**	0.02	**0.95**	0.13	0.68	0.03	0.67	0.00	**1.00**	0.00
Exactly2	**0.77**	0.01	0.76	0.00	**0.78**	0.01	0.75	0.01	0.76	0.01	0.76	0.01
HeartEW	**0.90**	0.02	0.86	0.00	**0.93**	0.02	0.78	0.01	0.82	0.01	0.85	0.02
IonoshpereEW	**0.93**	0.01	0.90	0.01	**0.93**	0.00	0.84	0.01	0.83	0.02	**0.95**	0.01
KrvskpEW	**0.97**	0.01	**0.97**	0.00	**0.98**	0.01	0.94	0.01	0.92	0.02	**0.98**	0.00
Lymphography	**0.90**	0.02	0.85	0.01	0.86	0.02	0.69	0.02	0.71	0.02	**0.92**	0.01
M-of-n	0.93	0.04	**0.97**	0.01	**0.99**	0.01	0.86	0.02	0.93	0.03	**1.00**	0.00
PenglungEW	**1.00**	0.01	0.58	0.01	**1.00**	0.00	0.72	0.04	0.70	0.04	0.96	0.02
SonarEW	**0.92**	0.02	0.86	0.01	**0.97**	0.02	0.74	0.03	0.73	0.03	**0.96**	0.02
SpectEW	**0.91**	0.02	0.85	0.00	0.85	0.01	0.77	0.01	0.78	0.01	0.88	0.00
Tic-tac-toe	**0.79**	0.02	0.77	0.00	**0.87**	0.01	0.73	0.01	0.71	0.01	**0.81**	0.01
Vote	**0.98**	0.01	0.96	0.00	**0.99**	0.01	0.89	0.01	0.89	0.01	0.97	0.01
WaveformEW	**0.86**	0.01	0.85	0.00	0.85	0.01	0.76	0.00	0.77	0.00	0.80	0.00
WineEW	**1.00**	0.00	0.98	0.01	0.99	0.00	0.95	0.01	0.93	0.00	**1.00**	0.01
Zoo	**0.97**	0.02	0.95	0.01	0.96	0.00	0.83	0.03	0.88	0.03	**1.00**	0.04
Average	0.92		0.88		0.93		0.82		0.83		0.93	

TABLE 5. Comparison between mean fitness function of the proposed HBGWOHHO and related work method.

Dataset	HBGWOHHO	BGWO	BHHO	BPSO	BGA	BGWOPSO
Breastcancer	**0.00**	0.03	0.02	0.03	0.03	0.03
BreastEW	0.01	0.03	**0.00**	0.03	0.04	0.04
CongressEW	**0.01**	0.04	0.02	0.04	0.04	0.03
Exactly	0.26	**0.04**	0.05	0.28	0.28	**0.00**
Exactly2	0.23	0.24	**0.22**	0.25	0.25	0.24
HeartEW	0.10	0.14	**0.07**	0.15	0.14	0.15
IonosphereEW	**0.07**	0.10	0.07	0.14	0.13	**0.05**
KrvskpEW	0.03	0.03	0.03	0.05	0.07	**0.02**
Lymphography	0.10	0.15	0.15	0.19	0.17	**0.08**
M-of-n	0.07	0.03	0.02	0.11	0.08	**0.00**
PenglungEW	**0.00**	0.14	0.00	0.22	0.22	0.05
SonarEW	0.08	0.15	**0.04**	0.13	0.13	0.05
SpectEW	**0.09**	0.15	0.15	0.13	0.14	0.12
Tic-tac-toe	0.21	0.23	**0.13**	0.24	0.24	0.19
Vote	0.02	0.05	**0.00**	0.05	0.05	0.03
WaveformEW	**0.16**	0.17	0.16	0.22	0.20	0.21
WineEW	**0.00**	0.03	0.02	0.02	0.01	0.00
Zoo	0.03	0.05	0.04	0.10	0.08	**0.00**
Average	0.08	0.10	0.06	0.13	0.13	0.07

TABLE 6. Best fitness function comparison between the proposed HBGWOHHO and related work methods.

Dataset	HBGWOHHO	BGWO	BHHO	BPSO	BGA	BGWOPSO
Breastcancer	**0.00**	0.02	0.01	0.03	0.02	0.02
BreastEW	**0.00**	0.03	0.00	0.02	0.02	0.03
CongressEW	**0.00**	0.05	**0.02**	0.03	0.03	0.02
Exactly	0.17	**0.00**	**0.00**	0.21	0.27	**0.00**
Exactly2	**0.22**	0.24	**0.21**	0.22	0.22	0.23
HeartEW	**0.03**	0.13	0.05	0.13	0.12	0.12
IonosphereEW	0.05	0.08	0.07	0.12	0.09	**0.04**
KrvskpEW	**0.01**	0.02	0.02	0.03	0.03	0.02
Lymphography	**0.06**	0.12	0.11	0.14	0.12	**0.06**
M-of-n	**0.00**	**0.00**	**0.00**	0.06	0.02	**0.00**
PenglungEW	**0.00**	0.12	**0.00**	0.13	0.13	0.03
SonarEW	0.05	0.12	**0.00**	0.07	0.07	0.02
SpectEW	**0.05**	0.14	0.13	0.10	0.12	0.11
Tic-tac-toe	0.16	0.23	**0.11**	0.21	0.21	0.17
Vote	0.01	0.04	**0.00**	0.03	0.03	0.03
WaveformEW	0.16	0.16	**0.15**	0.21	0.19	0.20
WineEW	**0.00**	0.01	0.01	**0.00**	**0.00**	**0.00**
Zoo	**0.00**	0.03	0.02	0.03	**0.00**	**0.00**
Average	**0.05**	0.08	0.05	0.10	**0.09**	**0.06**

TABLE 7. Worst fitness function comparison between the proposed HBGWOHHO and related work methods.

Dataset	HBGWOHHO	BGWO	BHHO	BPSO	BGA	BGWOPSO
Breastcancer	0.00	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**
BreastEW	0.01	0.03	0.02	0.03	**0.04**	**0.04**
CongressEW	0.01	**0.04**	0.03	**0.04**	**0.04**	0.03
Exactly	0.26	0.04	**0.29**	0.28	0.28	0.00
Exactly2	0.23	0.24	0.24	**0.25**	**0.25**	**0.25**
HeartEW	0.10	0.14	0.09	0.15	0.14	**0.17**
IonosphereEW	0.07	0.10	0.08	**0.14**	0.13	0.07
KrvskpEW	0.03	0.03	0.05	0.05	**0.07**	0.03
Lymphography	0.10	0.15	0.13	**0.19**	0.17	0.10
M-of-n	0.07	0.03	0.024	**0.11**	0.08	0.00
PenglungEW	0.00	0.14	0.004	**0.22**	**0.22**	0.08
SonarEW	0.08	**0.15**	0.06	0.13	0.13	0.07
SpectEW	0.09	**0.15**	0.14	0.13	0.14	0.13
Tic-tac-toe	0.21	0.23	0.14	**0.24**	**0.24**	0.20
Vote	0.02	**0.05**	0.02	**0.05**	**0.05**	**0.05**
WaveformEW	0.16	0.17	0.17	**0.22**	0.20	0.22
WineEW	0.00	**0.03**	0.02	0.02	0.01	0.03
Zoo	0.03	0.05	0.03	**0.10**	0.03	0.10
Average	0.08	0.10	0.09	0.13	0.12	0.08

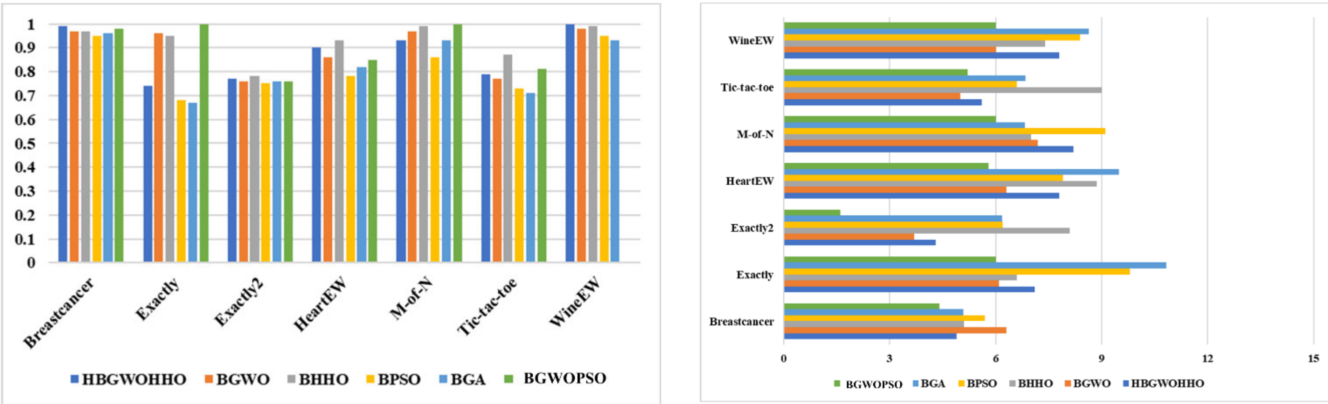
TABLE 8. Comparison between average selected feature size of proposed HBGWOHHO and related work methods.

Dataset	HBGWOHHO	BGWO	BHHO	BPSO	BGA	BGWOPSO
Breastcancer	**4.9**	6.3	5.1	5.7	5.1	4.4
BreastEW	**14.6**	**14.6**	20.1	16.6	16.3	13.6
CongressEW	6.4	**4.0**	9.3	6.8	6.6	4.4
Exactly	**5.5**	6.1	6.6	9.8	10.8	6.0
Exactly2	**3.4**	3.7	8.1	6.2	6.2	1.6
HeartEW	**5.9**	6.3	8.9	7.9	9.5	5.8
IonosphereEW	**14.5**	15.2	12.8	19.2	17.3	13.0
KrvskpEW	20.0	**11.9**	24.8	20.8	22.4	15.8
Lymphography	**7.5**	7.8	10.8	9.0	11.0	9.2
M-of-n	**6.9**	7.2	7.0	9.1	6.8	6.0
PenglungEW	**161.4**	168.9	167.4	178.8	177.1	130.8
SonarEW	**30.4**	32	36.7	31.2	33.3	31.2
SpectEW	11.8	**9.4**	12.9	12.5	11.7	8.4
Tic-tac-toe	**4.8**	5.0	9.0	6.6	6.8	5.2
Vote	7.9	**4.8**	8.7	8.8	6.6	3.4
WaveformEW	22.5	**16.3**	27.8	22.7	25.30	14.2
WineEW	**6.0**	**6.0**	7.4	8.4	8.6	6.0
Zoo	**8.2**	8.5	9.9	21.66	21.8	6.8

TABLE 9. Comparison between the computational time (in seconds) of proposed HBGWOHHO and BGWO, BHHO and PGWOPSO.

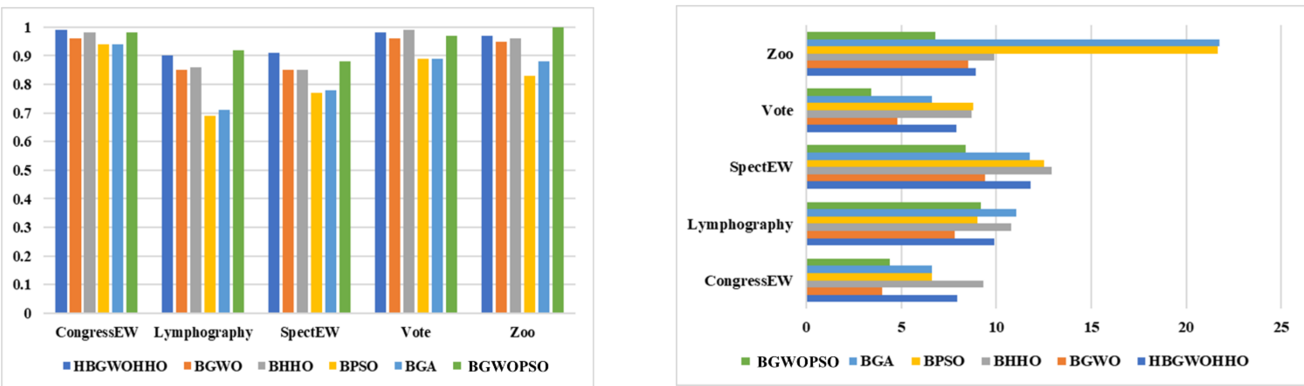
Dataset	HBGWOHHO	BGWO	BHHO	BGWOPSO
Breastcancer	**2.70**	23.4	32.68	5.69
BreastEW	**2.89**	23.1	31.02	5.38
CongressEW	**2.54**	22.89	32.69	5.35
Exactly	**2.77**	23.93	35.21	5.61
Exactly2	**2.89**	24.65	34.67	5.41
HeartEW	**3.35**	23.00	32.57	5.25
IonosphereEW	**2.76**	22.87	30.56	5.38
KrvskpEW	**3.22**	33.14	40.61	19.61
Lymphography	**3.57**	22.03	35.91	5.33
M-of-n	**2.92**	24.84	34.27	5.60
PenglungEW	**2.49**	20.8	31.14	4.58
SonarEW	**2.59**	21.6	30.35	5.44
SpectEW	**2.68**	22.55	29.94	5.71
Tic-tac-toe	**3.06**	23.3	34.39	5.30
Vote	**2.89**	24.2	30.96	5.73
WaveformEW	**4.02**	45.79	56.68	33.53
WineEW	**3.17**	22.6	30.91	5.62
Zoo	**2.59**	22.31	30.76	5.22

results of this analysis reveal if the results of the two tests are independent. The null hypothesis assumes no significant difference between the proposed HBGWOHHO optimizer and other optimizers' mean fitness. Significance levels greater than 5% support the null hypothesis assumption and vice versa. The procedures of this analysis are adopted from [40] and [76]. As indicated by the p-values presented in Table 10, it can be seen that the proposed HBGWOHHO achieves significant improvement over the BGWO, achieving a p-value of 0.007. Besides, the enhancement achieved by HBGWOHHO was substantial as compared to BHHO, BPSO, BGA, and BGWOPSO. Therefore, the null hypothesis



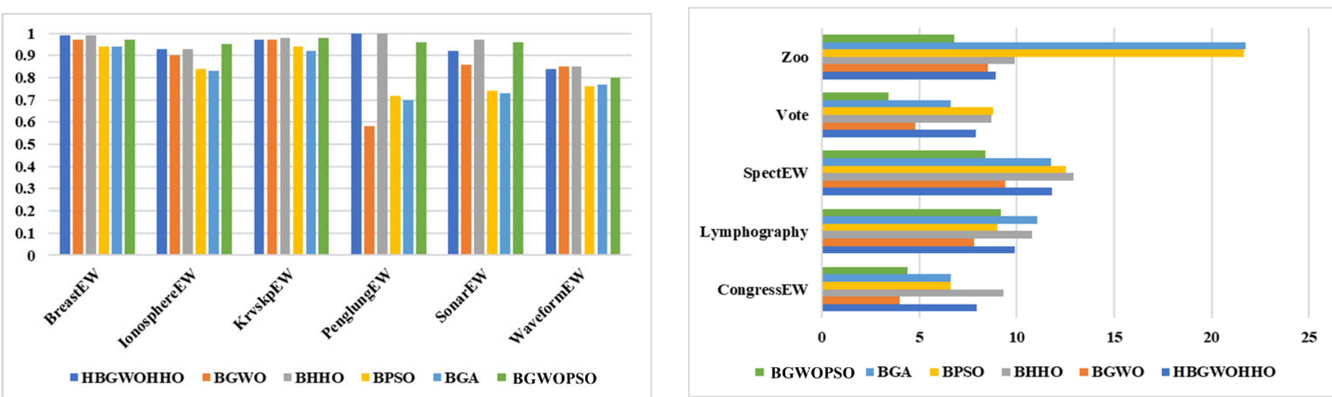
(a) (b)

FIGURE 2. a) Classification accuracy on five different benchmarking datasets and (b) the number of selected features achieved by HBGWOHHO and other algorithms in benchmarking datasets.



(a) (b)

FIGURE 3. a) Classification accuracy on five different benchmarking datasets and (b) the number of selected features achieved by HBGWOHHO and other algorithms in benchmarking datasets.



(a) (b)

FIGURE 4. a) Classification accuracy on five different benchmarking datasets and (b) the number of selected features achieved by HBGWOHHO and other algorithms in benchmarking datasets.

of similar mean fitness at the default 5% significance level is rejected.

E. COMPUTATIONAL COMPLEXITY

The computational complexity of the BGWOHHO relies on three main steps: initialization, fitness evaluation, and updating of the search agents. Since the number of search agents is N agents, the computational complexity of the initialization step is O(N). For updating process, it consists of searching for the best solution and updating the position of all search agents. Therefore, the computational complexity of updating process is expressed as $O(T \times N) + O(T \times$

TABLE 10. p-value by Wilcoxon test for the average fitness obtained by the different optimizers.

Comparison	p-value
HBGWOHHO vs. BGWO	0.007
HBGWOHHO vs. BHHO	0.001
HBGWOHHO vs. BPSO	9.6E-06
HBGOWHHO vs. BGA	8.4E-06
HBGWOHHO vs. BGWOPSO	0.006

$N \times B$) where T is the number of iteration and B is the search space boundary. The overall computational complexity of the proposed algorithm (BGWOHHO) is $O(N \times (T + TD + 1))$.

VII. CONCLUSION

A hybrid feature selection algorithm was proposed to enhance the performance of the existing wrapped-based BGWO algorithm. The proposed algorithm called HBGWOHHO replaced the complex exploration phase of native BGWO by exploring the HHO algorithm. The performance of the proposed technique was tested and compared across 18 standard UCI benchmark datasets. The efficacy new hybrid method was also compared with that of native BGWO and other wrapper-based feature selection algorithms including BPSO and BGA. The findings revealed that the proposed method was effective in improving the performance of the BGWO algorithm. The new hybrid algorithm outperformed the native BGWO algorithm in terms of accuracy, selected feature size, and computational time. When comparing with BPSO and BGA feature selection algorithms, the proposed HBGWOHHO surpassed them yield better accuracy, the smaller size of selected features in much lower computational time. These results indicate that the new hybridization approach effectively improved the performance of the native BGWO algorithm. Besides, the enhancement achieved by HBGWOHHO was significant compared to BHHO, BPSO, as indicated by the p-value, which was less than 5%.

REFERENCES

[1] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, Jan. 2014, doi: 10.1007/s00521-013-1368-0.

[2] M. T. Sadiq, X. Yu, Z. Yuan, M. Z. Aziz, S. Siuly, and W. Ding, "A matrix determinant feature extraction approach for decoding motor and mental imagery EEG in subject specific tasks," *IEEE Trans. Cognit. Develop. Syst.*, early access, Nov. 25, 2020, doi: 10.1109/TCDS.2020.3040438.

[3] M. T. Sadiq, X. Yu, Z. Yuan, and M. Z. Aziz, "Motor imagery BCI classification based on novel two–dimensional modelling in empirical wavelet transform," *Electron. Lett.*, vol. 56, no. 25, pp. 1367–1369, Dec. 2020, doi: 10.1049/el.2020.2509.

[4] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016, doi: 10.1109/TEVC.2015.2504420.

[5] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 1, pp. 131–156, Jan. 1997, doi: 10.1016/S1088-467X(97)00008-5.

[6] K. Ren, W. Fang, J. Qu, X. Zhang, and X. Shi, "Comparison of eight filter-based feature selection methods for monthly streamflow forecasting—Three case studies on CAMELS data sets," *J. Hydrol.*, vol. 586, Jul. 2020, Art. no. 124897.

[7] M. T. Sadiq, X. Yu, Z. Yuan, F. Zeming, A. U. Rehman, I. Ullah, G. Li, and G. Xiao, "Motor imagery EEG signals decoding by multivariate empirical wavelet transform-based framework for robust brain–computer interfaces," *IEEE Access*, vol. 7, pp. 171431–171451, 2019, doi: 10.1109/ACCESS.2019.2956018.

[8] M. T. Sadiq, X. Yu, and Z. Yuan, "Exploiting dimensionality reduction and neural network techniques for the development of expert brain–computer interfaces," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 114031, doi: 10.1016/j.eswa.2020.114031.

[9] J. Zhang, Y. Xiong, and S. Min, "A new hybrid filter/wrapper algorithm for feature selection in classification," *Analytica Chim. Acta*, vol. 1080, pp. 43–54, Nov. 2019, doi: 10.1016/j.aca.2019.06.054.

[10] A. G. Karegowda, A. S. Manjunath, and M. A. Jayaram, "Feature subset selection problem using wrapper approach in supervised learning," *Int. J. Comput. Appl.*, vol. 1, no. 7, pp. 13–17, Feb. 2010.

[11] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, Jun. 2011, doi: 10.1016/S1672-6529(11)60020-6.

[12] R. Al-wajih, S. J. Abdulakaddir, N. B. A. Aziz, and Q. Al-tashi, "Binary grey wolf optimizer with K-nearest neighbor classifier for feature selection," in *Proc. Int. Conf. Comput. Intell. (ICCI)*, Oct. 2020, pp. 130–136, doi: 10.1109/ICCI51257.2020.9247792.

[13] S. Ledesma, G. Cerda, G. Avina, D. Hernandez, and M. Torres, "Feature selection using artificial neural networks," in *Proc. Mex. Int. Conf. Artif. Intell.* Berlin, Germany: Springer, Oct. 2008, pp. 351–359.

[14] F. Song, D. Mei, and H. Li, "Feature selection based on linear discriminant analysis," in *Proc. Int. Conf. Intell. Syst. Design Eng. Appl.*, vol. 1, Oct. 2010, pp. 746–749.

[15] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, "Local geometric structure feature for dimensionality reduction of hyperspectral imagery," *Remote Sens.*, vol. 9, no. 8, p. 790, Aug. 2017. [Online]. Available: https://www.mdpi.com/2072-4292/9/8/790

[16] G. Shi, H. Huang, and L. Wang, "Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1425–1429, Aug. 2020, doi: 10.1109/LGRS.2019.2944970.

[17] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1100–1103, Sep. 1971, doi: 10.1109/T-C.1971.223410.

[18] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inf. Theory*, vol. 9, no. 1, pp. 11–17, Jan. 1963, doi: 10.1109/TIT.1963.1057810.

[19] Q. Al-Tashi, S. J. Abdulkadir, H. Md Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," *IEEE Access*, vol. 8, pp. 125076–125096, 2020, doi: 10.1109/ACCESS.2020.3007291.

[20] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994, doi: 10.1016/0167-8655(94)90127-9.

[21] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, H. Alhussian, M. G. Ragab, and A. Alqushaibi, "Binary multi-objective grey wolf optimizer for feature selection in classification," *IEEE Access*, vol. 8, pp. 106247–106263, 2020, doi: 10.1109/ACCESS.2020.3000040.

[22] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013, doi: 10.1109/TSMCB.2012.2227469.

[23] E. Hancer, B. Xue, D. Karaboga, and M. Zhang, "A binary ABC algorithm based on advanced similarity scheme for feature selection," *Appl. Soft Comput.*, vol. 36, pp. 334–348, Nov. 2015, doi: 10.1016/j.asoc.2015.07.023.

[24] H. R. Kanan and K. Faez, "An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 716–725, Nov. 2008, doi: 10.1016/j.amc.2008.05.115.

[25] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognit. Lett.*, vol. 10, no. 5, pp. 335–347, Nov. 1989, doi: 10.1016/0167-8655(89)90037-8.

[26] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Oct. 2007, doi: 10.1007/s11721-007-0002-0.

[27] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[28] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Dept. Comput. Eng., Fac. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep. tr06, 2005, vol. 200, pp. 1–10.
[29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Feb. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[30] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.
[31] T. K. Sharma, A. K. Sahoo, and P. Goyal, "Bidirectional butterfly optimization algorithm and engineering applications," in *Proc. Mater. Today*, 2020.
[32] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.
[33] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[34] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.
[35] V. Ayumi, L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Optimization of convolutional neural network using microcanonical annealing algorithm," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2016, pp. 506–511.
[36] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, p. 671, 1983, doi: 10.1126/science.220.4598.671.
[37] M. Yesilbudak, S. Sagiroglu, and I. Colak, "A novel implementation of kNN classifier based on multi-tupled meteorological input data for wind power prediction," *Energy Convers. Manage.*, vol. 135, pp. 434–444, Mar. 2017, doi: 10.1016/j.enconman.2016.12.094.
[38] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: Wiley, 2009.
[39] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018, doi: 10.1016/j.knosys.2018.05.009.
[40] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
[41] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Appl. Soft Comput.*, vol. 76, pp. 16–30, Mar. 2019, doi: 10.1016/j.asoc.2018.11.047.
[42] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque, and S. Mirjalili, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112824, doi: 10.1016/j.eswa.2019.112824.
[43] Q. Al-Tashi, H. M. Rais, S. J. Abdulkadir, and S. Mirjalili, "Feature selection based on grey wolf optimizer for oil & gas reservoir classification," in *Proc. Int. Conf. Comput. Intell. (ICCI)*, Oct. 2020, pp. 211–216, doi: 10.1109/ICCI51257.2020.9247827.
[44] A. O. Balogun, S. Basri, S. Mahamad, S. J. Abdulkadir, M. A. Almomani, V. E. Adeyemo, Q. Al-Tashi, H. A. Mojeed, A. A. Imam, and A. O. Bajeh, "Impact of feature selection methods on the predictive performance of software defect prediction models: An extensive empirical study," *Symmetry*, vol. 12, no. 7, p. 1147, Jul. 2020.
[45] Q. Al-Tashi, H. Rais, and S. Jadid, "Feature selection method based on grey wolf optimization for coronary artery disease classification," in *Recent Trends in Data Science and Soft Computing*, F. Saeed, N. Gazem, F. Mohammed, and A. Busalim, Eds. Cham, Switzerland: Springer, 2019, pp. 257–266.
[46] Q. Al-Tashi, H. M. Rais, S. J. Abdulkadir, S. Mirjalili, and H. Alhussian, "A review of Grey Wolf optimizer-based feature selection methods for classification," in *Evolutionary Machine Learning Techniques: Algorithms and Applications*, S. Mirjalili, H. Faris, and I. Aljara, Eds. Singapore: Springer, 2020, pp. 273–286.
[47] E. H. Houssein, M. E. Hosney, D. Oliva, W. M. Mohamed, and M. Hassaballah, "A novel hybrid Harris hawks optimization and support vector machines for drug design and discovery," *Comput. Chem. Eng.*, vol. 133, Feb. 2020, Art. no. 106656, doi: 10.1016/j.compchemeng.2019.106656.
[48] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," *Eng. Comput.*, pp. 1–30, 2020.
[49] J. Too, A. R. Abdullah, and N. M. Saad, "A new quadratic binary Harris hawk optimization for feature selection," *Electronics*, vol. 8, no. 10, p. 1130, Oct. 2019.
[50] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017, doi: 10.1016/j.neucom.2017.04.053.
[51] E. G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, Sep. 2002, doi: 10.1023/A:1016540724870.
[52] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004, doi: 10.1109/TPAMI.2004.105.
[53] R. R. Rajalaxmi, "A hybrid binary cuckoo search and genetic algorithm for feature selection in type-2 diabetes," *Current Bioinf.*, vol. 11, no. 4, pp. 490–499, Aug. 2016, doi: 10.2174/1574893611666151228190309.
[54] S. Nemati, M. E. Basiri, N. Ghasem-Aghaee, and M. H. Aghdam, "A novel ACO–GA hybrid algorithm for feature selection in protein function prediction," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12086–12094, Dec. 2009, doi: 10.1016/j.eswa.2009.04.023.
[55] P. Vasant, "Hybrid simulated annealing and genetic algorithms for industrial production management problems," *Int. J. Comput. Methods*, vol. 07, no. 02, pp. 279–297, Jun. 2010.
[56] Y. Li, H. Guo, L. Wang, and J. Fu, "A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under E-supply chain environment," *Sci. World J.*, vol. 2013, Dec. 2013, Art. no. 125893, doi: 10.1155/2013/125893.
[57] C.-L. Huang and J.-F. Dun, "A distributed PSO–SVM hybrid system with feature selection and parameter optimization," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1381–1391, Sep. 2008, doi: 10.1016/j.asoc.2007.10.007.
[58] H. H. Inbarani, A. T. Azar, and G. Jothi, "Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis," *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 175–185, Jan. 2014, doi: 10.1016/j.cmpb.2013.10.007.
[59] G. Eappen and T. Shankar, "Hybrid PSO-GSA for energy efficient spectrum sensing in cognitive radio network," *Phys. Commun.*, vol. 40, Jun. 2020, Art. no. 101091, doi: 10.1016/j.phycom.2020.101091.
[60] F. Javidrad, M. Nazari, and H. R. Javidrad, "Optimum stacking sequence design of laminates using a hybrid PSO-SA method," *Composite Struct.*, vol. 185, pp. 607–618, Feb. 2018, doi: 10.1016/j.compstruct.2017.11.074.
[61] R. Purushothaman, S. P. Rajagopalan, and G. Dhandapani, "Hybridizing gray wolf optimization (GWO) with grasshopper optimization algorithm (GOA) for text feature selection and clustering," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106651, doi: 10.1016/j.asoc.2020.106651.
[62] H. Bouzary, F. F. Chen, and M. Shahin, "Optimal composition of tasks in cloud manufacturing platform: A novel hybrid GWO-GA approach," *Procedia Manuf.*, vol. 34, pp. 961–968, 2019, doi: 10.1016/j.promfg.2019.06.103.
[63] M. Barman and N. B. D. Choudhury, "A similarity based hybrid GWO-SVM method of power system load forecasting for regional special event days in anomalous load situations in assam, india," *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102311, doi: 10.1016/j.scs.2020.102311.
[64] N. Singh and S. B. Singh, "A novel hybrid GWO-SCA approach for optimization problems," *Eng. Sci. Technol., Int. J.*, vol. 20, no. 6, pp. 1586–1601, Dec. 2017, doi: 10.1016/j.jestch.2017.11.001.
[65] P. J. Gaidhane and M. J. Nigam, "A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems," *J. Comput. Sci.*, vol. 27, pp. 284–302, Jul. 2018, doi: 10.1016/j.jocs.2018.06.008.
[66] 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, doi: 10.1109/ACCESS.2019.2906757.
[67] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao, "A hybrid feature selection algorithm for gene expression data classification," *Neurocomputing*, vol. 256, pp. 56–62, Sep. 2017, doi: 10.1016/j.neucom.2016.07.080.
[68] Q. Li, H. Chen, H. Huang, X. Zhao, Z. Cai, C. Tong, W. Liu, and X. Tian, "An enhanced Grey Wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," *Comput. Math. Methods Med.*, vol. 2017, Jan. 2017, Art. no. 9512741, doi: 10.1155/2017/9512741.
[69] C. Yan, J. Liang, M. Zhao, X. Zhang, T. Zhang, and H. Li, "A novel hybrid feature selection strategy in quantitative analysis of laser-induced breakdown spectroscopy," *Analytica Chim. Acta*, vol. 1080, pp. 35–42, Nov. 2019, doi: 10.1016/j.aca.2019.07.012.

[70] J. Too, A. Abdullah, N. M. Saad, N. M. Ali, and W. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," *Computers*, vol. 7, no. 4, p. 58, Nov. 2018.
[71] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016, doi: 10.1016/j.neucom.2016.03.101.
[72] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–38, Feb. 2008, doi: 10.1016/j.compbiolchem.2007.09.005.
[73] C. Blake. (1998). *UCI Repository of Machine Learning Databases*. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[74] S. B. Imandoust and M. Bolandraftar, "Application of K-nearest neighbor (KNN) approach for predicting economic events: Theoretical background," *Int. J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 605–610, 2013.
[75] L. Wang, L. Khan, and B. Thuraisingham, "An effective evidence theory based K-nearest neighbor (KNN) classification," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, vol. 1, Dec. 2008, pp. 797–801.
[76] G. A. Sharifai and Z. Zainol, "Feature selection for high-dimensional and imbalanced biomedical data based on robust correlation based redundancy and binary grasshopper optimization algorithm," *Genes*, vol. 11, no. 7, p. 717, Jun. 2020. [Online]. Available: https://www.mdpi.com/2073-4425/11/7/717

RANYA AL-WAJIH received the B.Sc. degree in communication and computer engineering from Taiz University, Yemen, in 2015. She is currently pursuing the M.Sc. degree in information technology with Universiti Teknologi PETRONAS. Her current research interests include feature selection and classification.

SAID JADID ABDULKADIR (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science and the Ph.D. degree in information technology, in 2016. He is currently a Senior Lecturer with the Department of Computer and Information Sciences and a member of the Executive Committee for the IEEE Computational Intelligence Society, Malaysia, and the Data Science Association Malaysia. His current research interests include machine learning, data clustering, and predictive analytics. He is currently serving as the Editor-in-Chief for the *Journal of Applied Artificial Intelligence*, a Section Editor for *Platform Journal*, and a Reviewer for *Artificial Intelligence Review*, IEEE Access, and *Knowledge-Based Systems*.

NORSHAKIRAH AZIZ received the Diploma degree in management technology, the bachelor's and M.Sc. degrees in information technology (IT), and the Ph.D. degree in e-business(e-SCM). She is currently a Senior Lecturer with Universiti Teknologi PETRONAS, Malaysia. She is also a Researcher with the UTP Centre of Research in Data Sciences (CeRDaS) and the Data Governance Leader with the High-Performance Cloud Computing Data Centre (HPCCC). She is a Qualified Technologist. She has a total of 19 years of experience gained from academic institutions and industry under her belt. Her industry working experience is related to business intelligence, e-business, and IT project management. Her research interests include business intelligence, data analytics, data governance, and digital addiction.

QASEM AL-TASHI received the B.Sc. degree in software engineering from Universiti Teknologi Malaysia, in 2012, and the M.Sc. degree in software engineering from Universiti Kebangsaan Malaysia, in 2017. He is currently a Research Scientist with Universiti Teknologi PETRONAS and an Academic Staff with Albaydha University, Yemen. His research interests include machine learning, multi-objective optimization, and feature selection, classification, and data analytics. He is currently serving as a Section Editor for the *Journal of Applied Artificial Intelligence* and *Journal of Information Technology and Computing*. He also serves as a Journal Reviewer for *Applied Soft Computing*, *Knowledge-Based Systems*, *Journal of Ambient Intelligence and Humanized Computing*, *Artificial Intelligence Review*, IEEE Access, and PLOS ONE.

NOUREEN TALPUR received the master's degree in information technology (IT) from Universiti Tun Hussein Onn Malaysia (UTHM). She is currently pursuing the Ph.D. degree with Universiti Teknologi PETRONAS (UTP), Malaysia. She has coauthored nine articles in areas of adaptive neuro-fuzzy inference system (ANFIS), membership functions, neuro-fuzzy systems, metaheuristic algorithms, and optimization. Her research publications include research papers and articles in well-known journals and conference proceedings. Her research interests include machine learning, deep learning, classification, metaheuristics, optimization, neuro-fuzzy systems, such as ANFIS and deep neuro-fuzzy systems.