

Received April 19, 2022, accepted June 3, 2022, date of publication June 13, 2022, date of current version June 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3182685

Variable Length Black Hole for Optimization and Feature Selection

TARA OTHMAN QADIR¹, NORFAIZA FUAD², AND NIK SHAHIDAH AFIFI MD TAUJUDDIN²

¹Department of Software and Informatics Engineering, College of Engineering, Salahaddin University, Erbil 44002, Iraq

²Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Johor 86400, Malaysia

Corresponding author: Tara Othman Qadir (alsaraftara@gmail.com)

ABSTRACT In the high dimensional space, the problem of feature selection (FS) can be regarded as combinatorial optimization problem with high complexity due to the huge number of candidate features. In this article, a novel type of meta-heuristic searching based on variable length of solution space is proposed in order to solve the high dimensionality issue of the FS and to obtain more optimal results. The proposed algorithm uses the original black hole optimization as baseline for development. Blackhole optimization assumes in fixed solution space which decreases the efficiency when the number of features is high. Furthermore, the algorithm is subject to stagnation due to the single exemplar or black hole selection. Hence, a novel variable length black hole modifies the original black hole algorithm with various aspects, namely, it enables decomposing the solution space into subset of dimensions and searching within each dimension separately with selecting an exemplar for each dimension which represents the black hole of the corresponding dimension. In addition, it enables length changing of the solutions based on stagnation criterion. Furthermore, it proposes new concept of energy to the black hole which indicates to the decrease in the effectiveness of black hole with time in an exponential way and use it to replace the black hole when it is not effective anymore. The proposed algorithm which is designated as variable length black hole optimization VLBHO is compared with the variable length particle swarm optimization. The approach has increased the accuracy from 50% to 67% for forest cover dataset and from 38% to 80% for wine dataset.

INDEX TERMS Features selection, black hole optimization, high dimension solution space, variable length.

I. INTRODUCTION

In the era of artificial intelligence and machine learning, data plays an important role in generating knowledge among learners. Typically, data is not used directly but pre-processed before being presented for training or testing [1]. One of the essential stages conducted with the data before using it for learning is feature selection [2]. This means choosing a sub-set of the data for reducing the length of the record used for training, as well as providing more discriminative and less redundant information [3]. Feature selection can be carried out in different ways: offline or online, fixed or variable length, low or high dimension, and supervised and unsupervised [4]. The majority of the approaches used for feature selection adopt certain criteria to indicate the importance of

the attributes in considering whether the class information exists or not. Some of these are entropy, information gain, symmetric uncertainty and error Bayesian rate [5]. In addition, many approaches use various searching algorithms to identify the most important features to be selected and the less important or redundant features that are to be de-selected. The searching is called combinatorial searching due to the large candidate number of choices and their combinatorial nature.

Most feature selection approaches can be classified into two main categories: filter and wrapper [6]. Filter methods evaluate features based on the rankings of their importance to categorise classes using statistical methods (e.g., Student's t-test and the Chi-square test); information theory-based methods (e.g., entropy, Kullback–Leibler divergence and the information gain measure), or other search techniques (e.g., the correlation-based feature selection algorithm [7]). However, filter methods may remove features that are particularly

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu¹.

relevant to some classes with fewer instances [8]. Unlike filter methods, wrapper methods use the classifier as an evaluation function for feature subsets, which require classification accuracy as a form of feedback to evaluate the classifier. Thus, wrapper methods are usually modelled as optimisation problems from which to select the optimal feature subset. Because of this, wrapper methods require more computational time than filter methods, as the latter only use a separate metric to evaluate and select features. However, wrapper methods usually have better accuracy than filter methods. For example, there is a large margin hybrid algorithm for the feature selection [9] and hybrid meta-heuristic approaches.

Meta-heuristic searching is a family of random search algorithms that use various heuristics while searching for the way to optimise certain types of objective function [10]. The elements of meta-heuristic searching are the solution space, which carries the solution representation; the objective function, which provides the way of evaluating the candidate solution; and the approach of changing the set of solutions while searching. Two main classes of meta-heuristic searching algorithm exist, namely swarm, where solutions are represented by a set of moving solutions to generate new positions from an existing one; and evolutionary, where solutions are represented by a pool of mating for solutions to generate off-spring from existing ones. An example of the swarm searching algorithm is the particle swarm optimisation, while an example of the evolutionary searching meta-heuristic algorithm is the genetic algorithm.

Swarm algorithms, as a subset of meta-heuristic algorithms, indicate optimisation algorithms, in which solutions are represented by swarms or sets that interact with each other and evolve based on the interaction until they reach maturity and convergence. The length of the solution space can be determined in advance and is called the fixed solution space length. Alternatively, it can change according to the solution, when it is called variable length (VL) size. An example of the variable length meta-heuristic algorithm with the swarm type is given in the work of [11]. In this study, particle swarm optimisation enables searching based on different lengths of solutions in order to solve the problem of feature selection in high dimensional space.

One recently developed swarm-based meta-heuristic searching approach is Black Hole Optimisation (BHO), [12] which imitates in its representation the state of the stars, their interaction with themselves or with a black hole and their mobility in the universe. This algorithm has recently been used for the purpose of clustering; however, no development was made for it to be used for feature selection. The aim of this article is to propose a novel development of the BHO searching algorithm for feature selection with supporting supervised data and using the Variable Length (VL) type of selected features while searching. Contrary to the work of [11], the developed VLBHO in this article solves the issues of stagnation that are caused when solutions do not improve based on certain exemplars.

The use of feature selection for improving classification accuracy is a recent topic in the literature. Researchers have indicated the importance of features selection in the improvement of the model performance [13]. In [11], a novel variable length optimisation algorithm based on particle swarm optimisation was proposed. The algorithm is distinguished by its capability to provide different lengths of particles. It starts by sorting the features in a descending manner based on their importance and evaluating them based on an objective function that combines both inter- and intra-class distances. Additionally, the algorithm contains a solution changing mechanism to enable efficient and effective solution finding. In [14], an integrated particle swarm optimisation with k-means was proposed for feature selection. In [15], a wrapper-based feature selection based on a heuristic was proposed. The heuristic uses Ant Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO) to integrate both face and fingerprint features. In [16], two prevailing sets of features were combined for use in iris recognition: first-order and second-order statistical measures as textural feature descriptors. A hybrid statistical dependency-based feature selection algorithm is also applied to the extracted feature descriptors to remove noisy and redundant features, thus reducing the size of the feature vector. A back propagation neural network using the Levenberg-Marquardt training algorithm is used for the recognition task. In [17], a whale optimisation algorithm was used to build two hybridisation models for feature selection. Simulated Annealing was embedded in the whale optimisation algorithm to enhance the exploration. In [18], an improved meta-heuristic selection was proposed based on the Chaotic dragonfly algorithm. Their framework consisted of multiple stages starting from pre-processing, which aimed to use an over-sampling technique to solve an imbalanced dataset. Next, a feature selection phase was used for selecting discriminant features using the Chaotic dragonfly algorithm. For classification, they used a support vector machine. In [19], a new feature selection algorithm was presented for two stages of a forecasting engine, consisting of both a recursive neural network and the Elman neural network. In [20], a feature selection method was proposed for iris spoofing detection using top-k feature selection based on the Friedman test and fused through score-level fusion. In addition, the work includes score level fusion of handcrafted and data-driven features. Hence, the feature selection approach worked on both of them. In [21], a feature selection method was proposed based on hybrid Particle Swarm Optimisation (PSO) and a genetic algorithm. It is based on the Gabor filter and the Local Tetra Pattern. The feature selection approach is based on a multi-model biometric framework that includes the face, iris and fingerprint. In [22], the feature selection was proposed based on a genetic algorithm with a cross-generational elitist selection, heterogeneous recombination and cataclysmic mutation for feature selection. In [11], the first variable-length PSO representation for feature selection was proposed, enabling particles to have different and shorter lengths, which defines

a smaller search space and, therefore, improves the performance of PSO. By rearranging features in descending order of their relevance, they facilitate the capacity of particles with shorter lengths to achieve better classification performance. Furthermore, using the proposed length changing mechanism, PSO can jump out of local optima, further narrowing the search space and focusing its search on smaller and more fruitful areas. However, this approach suffers from a lack of stagnation handling, which happens when the exemplar does not improve its associated solutions for a long period of time. In other words, the algorithm might fall in premature convergence.

There are two issues with existing BHO [12], namely, fixed length searching which causes inefficiency when the length of solution space increase and relying on single blackhole which might cause stagnation or falling in local minima. In this article, we propose a novel swarm-based meta-heuristic searching algorithm developed under the framework of blackhole optimisation and incorporating two novel aspects, namely the variable length of VLPSO and energy-based stagnation handling. The role of the first aspect is to enable efficiency in searching in high dimensional data for feature selection and the role of the latter is to enable avoidance of premature convergence.

The remainder of the article is organised as follows. In II, the proposed method is presented. The results and discussion are provided in III. Lastly, the conclusion is given in IV.

II. THE PROPOSED METHODS

This section presents the developed methodology for VLBHO optimisation. In sub-section A, an overview is presented of traditional black hole optimisation. In sub-section B, the general algorithm of variable length black hole optimisation is presented. Afterwards, the evaluation metrics are given in C.

A. PROBLEM FORMULATION

$$\{i_k\}_{k=1..K} = \operatorname{argmin}(f_1(D, i_k), f_2(D, i_k) \dots f_o(D, i_k)) \quad (1)$$

where

f_j denotes an objective function considered in the feature selection

$j = 1, 2, \dots, o$ is an index for the optimization function

o denotes the number of the objective function

Many researchers consider only one objective function such as the inverse of accuracy (because there is a minimisation) [23]. However, some researchers have formulated the issue of feature selection as a bi-objective problem with two objectives: the number of neurons and the classification error; one such study is the work of [24]. In this article, we are only concerned with the classification error rate or accuracy as an objective for optimisation.

B. AN OVERVIEW BLACK HOLE OPTIMIZATION

This sub-section presents the original BHO before adding our subsequent modification. The algorithm uses two equations:

the first one is Equation (2), which is used to move stars or solutions toward the black hole. The second one is Equation (3), which is used to calculate the event horizon of the black hole [12].

$$x_i(t+1) = x_i(t) + \operatorname{rand} \times (x_{BH} - x_i(t)) \quad i = 1, 2, \dots, N \quad (2)$$

$$R = \frac{f_{BH}}{\sum_{i=1}^N f_i} \quad (3)$$

where

x_i denotes the position of the solution

x_{BH} denotes the black hole position

f_{BH} denotes the fitness value of the black hole

f_i denotes the fitness value of the solution i

t denotes the iteration

The difference between the mobility equation provided in 1 and 2 and the one of particle swarm optimisation is its use of the concept of the event horizon, in which solutions close enough to the black hole are removed. The radius is computed based on the fitness value of the black hole and the fitness value of the solution.

C. GENERAL ALGORITHM

Variable length black hole optimisation VLBHO is presented in Algorithm 1. The inputs of the algorithm are: *Max_iteration*: the number of iterations that will be executed on the algorithm; *numOfStars*: the number of stars that will be initialised in the population;

and *rangeOfDimension*: the range of dimension related to the searching in the solution space. The output of the algorithm is *bho*, which represents a black hole object that includes the global best *gBest* and other information.

The algorithm starts by initialising the black hole object BHO using *Max_iteration*, *numOfStars* and *rangeOfDimension*. It returns an original black hole object *bho*. In this initialisation process, the initial population is created. Next, the algorithm iterates until *Max_iteration* and it loops over the stars one by one to do the following: 1- it updates the position of the star using the function *updatePosition()* 2- it updates the fitness of the star using *updateFitness()* 3- it updates the global best using *best()* 4- it updates the energy using *UpdateEnergy* Next, the algorithm performs an inner loop over each dimension and over each exemplar in the corresponding dimension in order to check the energy of the exemplar, thus disabling its role as exemplar if the energy is lower than *bho.Emin*. In order to disable its role as an exemplar, the algorithm finds its follower stars and assigns each of them a new exemplar according to the dimension of the star. On the other hand, the algorithm is responsible for replacing the black hole when stagnation happens or when no improvement is made over a certain period of time.

Observing the procedure of the algorithm, it was noted that it introduces two concepts: the first is a black hole, which represents the global best, and the second is an exemplar, which represents a solution with the same dimension as its

follower. The exemplar loses its role when its energy is below a certain level while the black hole loses its role when stagnation happens.

Algorithm 1 The General Algorithm of Variable Length Black Hole Optimization.

Input

Max_iteration
numOfStars
rangeOfDimension
Emin

Output

bho object that includes: gBest and other information

Start

```

1- bho=initBHO(Max_iteration,numOfStars,
   rangeOfDimnesion);
2- for each iteration of Max_iteration
3-   for each star of bho.Stars
4-     bho =updatePosition(bho,star)
5-     bho=updateFitnessANDpersonalBest(bho,star,itr)
6-     bho.gBest = best(bho.stars)
7-     bho = UpdateEnergy(bho, star, itr)
8-     for each dimension in rangeOfDimension
9-       for each exemplar
10-        if bho.Stars(star).Energy(dimension)
           <bho.Emin
11-          set_stars = get the stars who use this star as
            their exemplar in this dimension.
12-          for each star in set_stars
13-            bho.Stars (star).Exemplar(dimension) =
              ExemplarAssignment (bho,star,dimension)
14-          end
15-        end
16-      end
17-    end
18-    if gBest not improved for a time period
19-      bho = lengthChanging(bho)
20-    end
21-  end
22- end

```

End

The algorithm of the exemplar assignment is presented in Algorithm 2 of the ExemplarAssignment() function. It takes an existing bho object, a current star, an iteration and a logical variable of perspective as the inputs and it provides an updated version of the bho object with its updated exemplar as the output.

The algorithm works as follows: it scans all the stars and decides whether they are part of the radius of the position circle or the fitness circle, according to the perspective. Next, it adds all the stars within the radius as a candidate exemplar. The exemplar is selected probabilistically based on a roulette wheel model.

The algorithm for updating the energy of the star is presented in Algorithm 3. As it is provided, the algorithm takes

Algorithm 2 The Algorithm of Updating the Exemplar According Two Perspectives.

Input:

bho object
curr_star: the star we are getting its exemplar
itr: iteration number
perspective: the type of version (fitness, position).
Ra_Position: function to get the radius from the perspective of the positions.
Ra_fitness: function to get the radius from the perspective of the fitness.

Output:

Exemplar

Start

```

1- If perspective == POSITION
2-   Ra ← Ra_Position (bho) % if the version is the
   position version.
3- else
4-   Ra ← Ra_fitness(bho) % if the version is the
   position version.
5- end
6- For star = 1 ← bho. Stars-number
7-   If (bho. Stars (star). Position (dimension) <=Ra)
   AND (perspective == POSITION)
8-     add star to candidate exemplars
9-   end
10-  If (bho. Stars (star). Position (dimension) <= Ra)
   AND (perspective == FITNESS)
11-    add star to candidate exemplars
12-  end
13- end
14- Exemplar for curr_star in this dimension ←
   Roulette_Wheel_Selection (bho, candidate exemplars)
15-End

```

the star and the iteration as the input along with the bho object. Next, it iterates on the stars in the same dimension and adds the ones that use a subject star as an exemplar. Next, the energy is updated according to the improvement percentage of the exemplar over its associated stars, as is presented in line numbers 4 and 6, multiplying by the inversely exponential factor based on the iteration. The energy update is given according to Equation (4).

$$E = IR \times E_{max} e^{-(t-t_0)} \quad (4)$$

It is important to track the improvement of any star in order to change its energy and thus to decide whether to keep it as an exemplar or not and whether to kill it or not. The algorithm of updating the improvement of the star is given in Algorithm 4. It accepts the black hole object BHO, the star, and the iteration number and it gives the output, which is an updated BHO object. The algorithm uses the star current cost to compare it with the personal best cost and triggers the improvement flag if the former is lower. Otherwise, the improvement star is set to false. The improvement flag is used to count how many

times the star has not been improved. When the star has not been improved a certain number of times, it is killed and a new star is generated.

Algorithm 3 Algorithm of Updating the Energy of the Star.

Input:

- bho object
- star
- itr: iteration number

Output:

updated bho object

Start

- 1-FOR stars in star dimensions
- 2- set_stars = get the stars who use this star as their exemplar in this dimension.
- 3- FOR starSpace in set_stars
- 4- IR = IR + bho.Stars (starSpace). Improvement
- 5- end
- 6- bho.Stars(star).Energy (dimension)= IR*bho.Emax*exp(-(itr-t0));
- 7- end

End

Algorithm 4 The Algorithm of Updating the Fitness Value and the Personal Best of a Star and Kill it in Case in it Does not Improve.

Input:

- 1) BHO object
- 2) star
- 3) iteration number

Output:

updated bho object

Start

- 1-If bho. Stars (star).Cost >= bho.Stars(star).pBest.Cost
- 2- pbest ← star
- 3- bho. Stars (star).Improvement= true
- 4-else
- 5- bho. Stars(star).Improvement= false

End

The flowchart of the developed variable length blackhole optimization is presented in Figure 1.

D. TRANSFER FUNCTIONS

When dealing with binary space, such as with the problem of feature selection, the Equation (5) is modified to Equation (6).

$$x_i(t+1) = \begin{cases} 0 & \text{if } rand < T(v_i(t+1)) \\ 1 & \text{if } rand \geq T(v_i(t+1)) \end{cases} \quad i = 1, 2, \dots \quad (5)$$

$$v_i(t+1) = rand \times (x_{BH} - x_i(t)) \quad (6)$$

where

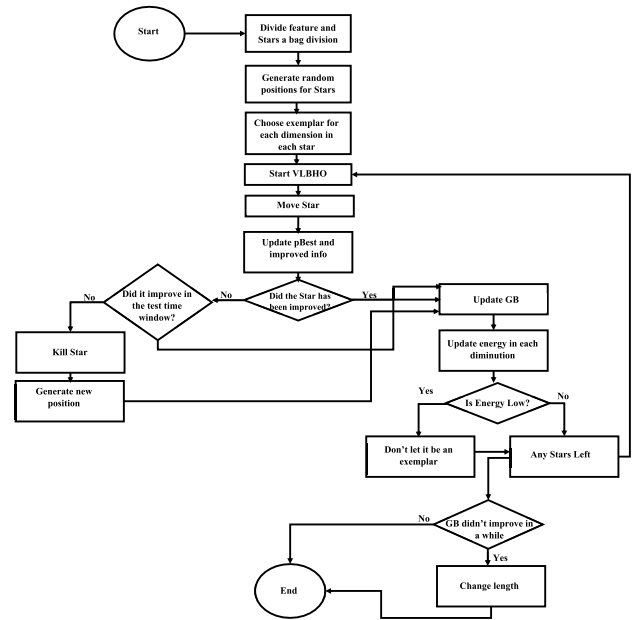


FIGURE 1. Flowchart of the developed variable length blackhole optimization.

T denotes the transfer function that might be an S-shaped or V-shaped transfer function. Two types of transfer functions, S and V are given in Equations (7,8) and presented in Figure 2.

$$y = S(x) = \frac{1}{1 + e^{-ax}} \quad (7)$$

$$z = V(x) = |\tanh(bx)| \quad (8)$$

where

a and b are constants

E. EVALUATION METRICS

After the evaluation, there were four types of records, namely true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Afterwards, the metrics were calculated as shown in Equations (9) to (14).

- 1- Precision or positive predictive values and it is given as in Equation (9). It aims at finding the proportion of predicted Positives that is truly Positive

$$PPV = \frac{TP}{TP + FP} = 1 - FDR \quad (9)$$

where FDR indicates to false discovery rate

- 2- Sensitivity or recall (true positive rate) it aims at finding the proportion of actual Positives that is correctly classified. It is calculated based on the equation

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR \quad (10)$$

- 3- F-measure it indicates to the harmonic mean of precision and recall

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN} \quad (11)$$

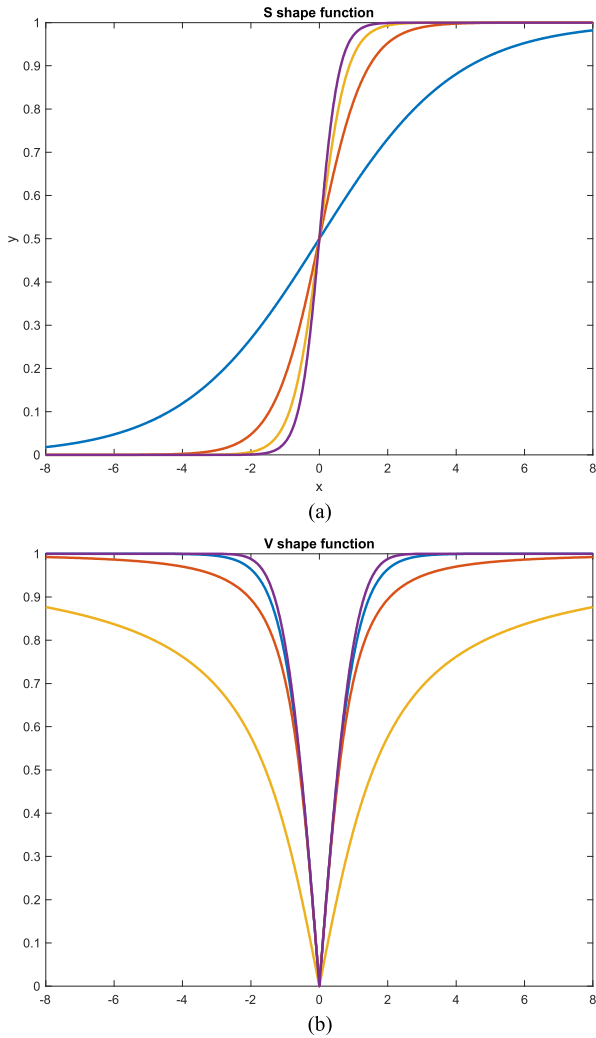


FIGURE 2. Curve of two types of transfer functions -a- S shape -b- V shap.

4- G-mean it indicates to the geometric mean between precision and recall

$$G - mean = \sqrt{recall \ precision} \quad (12)$$

5- Accuracy: it is defined as the proportion of true results among the total number of cases examined.

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

Another way of calculating the accuracy is using this Equation (12)

$$acc = \frac{number \ of \ true \ predictions}{total \ number \ prediction} \quad (14)$$

F. DATASETS

For the datasets, two datasets were used, as presented below:

1- Forest Cartographic shows various cartographic variables to classify forest categories that range from 1 to 7. It was downloaded from Kaggle 1 and is comprised of 15,120 samples with 54 features [25].

TABLE 1. Parameters for operating the variable length black hole optimization algorithm VLBHO.

Parameter name	Meaning	Value
Time window length	period to check the stars improvement for the killing process	5
Stars number	Number of stars in the population	40
E _{max}	Max energy for black hole	10
E _{min}	Min energy to switch between the Black hole and the exemplar	1e-3
T _{max}	Max number of iterations	20
T	Threshold for the stagnation	5

TABLE 2. Benchmarking mathematical functions used for evaluating VLBHO.

Function Name	Formula
Rastrigin	$f(x) = An + \sum_{i=1}^n [x_i^2 - A\cos(2\pi x_i)]$ where: $A = 10$
Ackley	$f(x, y) = -20\exp[-0.2\sqrt{0.5(x^2 + y^2)}] - \exp[0.5(\cos 2\pi x + \cos 2\pi y)] + e + 20$
Sphere	$f(x) = \sum_{i=1}^n x_i^2$
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$
Griewangk	$1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$
Michaelwicz	$f(x) = - \sum_{i=1}^d \sin(x_i) \sin^{2m}(\frac{ix_i^2}{\pi})$
Shubert	$f(x) = \left(\sum_{i=1}^5 \text{icos}((i+1)x_1 + i) \right) \left(\sum_{i=1}^5 \text{icos}((i+1)x_2 + i) \right)$
Schwefel	$f(x) = 418,9829d - \sum_{i=1}^d x_i \sin(\sqrt{ x_i })$

2- Wine Dataset: these data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines [26].

III. RESULTS AND DISCUSSION

For evaluation, we ran our developed VLBHO with is four variants, namely S-shaped position, V-shaped position, S-shaped fitness and V-shaped fitness. Firstly, we tested their performance on the benchmarking mathematical function to investigate their optimisation performance. Secondly, we tested their performance on the different datasets.

A. A EVALUATION ON MATHEMATICAL FUNCTIONS

The first stage of evaluation compared the performance of the two variants of VLBHO with particle swarm optimisation (PSO). The parameters used to operate VLBHO are given in Table 1. In addition, the benchmarking mathematical

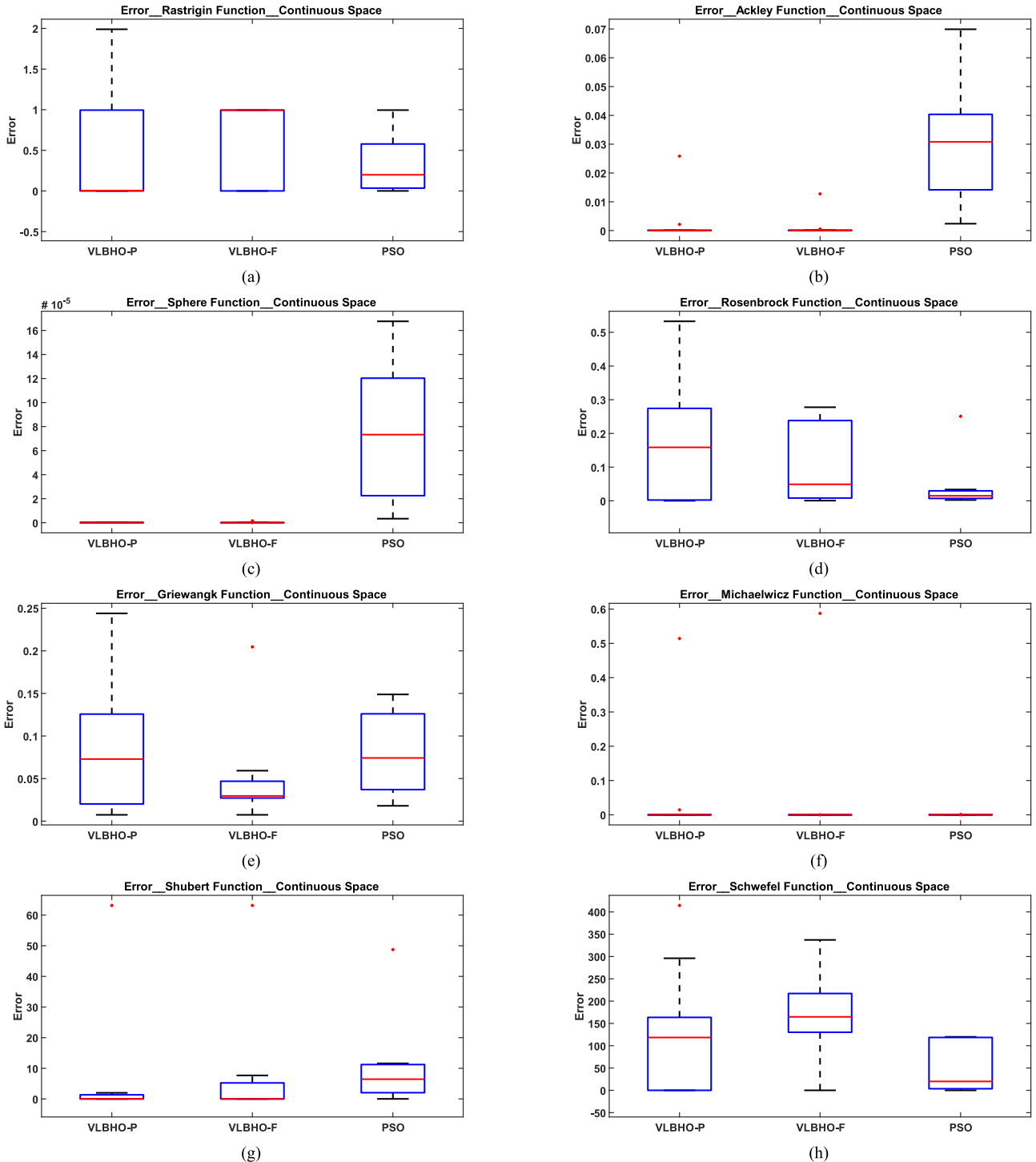


FIGURE 3. Boxplot visualisation of the obtained fitness value of VLBHO (position variant or fitness variant) and comparison with PSO for mathematical functions -a- Rastrigin, -b- Ackley, -c- Sphere, -d- Rosenbrock -e- Griewangk -f- Michalewicz -g- Shubert -h- Schwefel.

functions are provided in Table 2. These functions were selected because they are considered challenging benchmarking optimisation functions with non-convexity, non-linearity and multi-modal optimisation surfaces.

Observing Figure 3, we found that VLBHO provided the least fitness value for Ackley, Rastrigin, Sphere, and Shubert. The least fitness value indicates better performance, as the

functions need to be minimised. The fitness value is similar to PSO for Michalewicz

B. EVALUATION ON BENCHMARKING DATASET

The evaluation is composed of two parts: the first is the evaluation based on the optimisation error, presented in A.

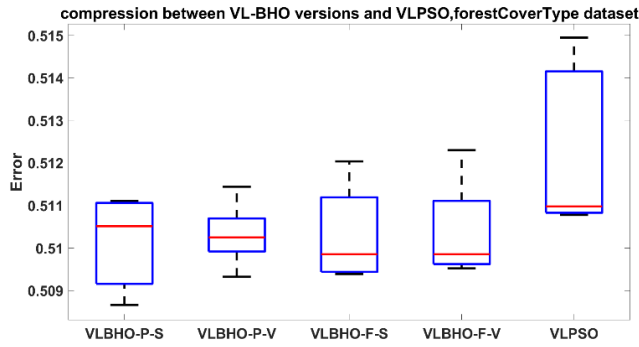


FIGURE 4. Cost value minimisation results based on the feature selection of VL-BHO (four variants) and the comparison with the benchmark VL-PSO for the forest cover dataset [27].

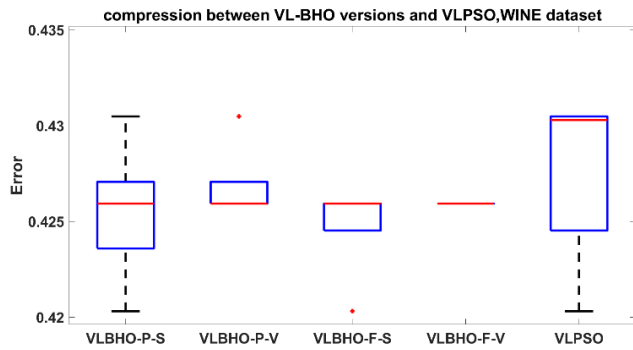


FIGURE 5. Cost value minimisation results based on the feature selection of VL-BHO (four variants) and the comparison with the benchmark VLPSO for the wine dataset [28].

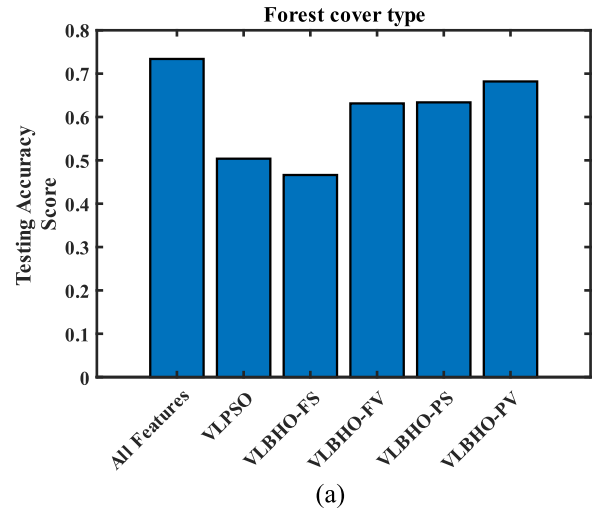
TABLE 3. Summary of reached accuracy for wine and forest cover.

	VLPSO	VLBHO
Forest Cover	50%	67%
Wine Dataset	38%	80%

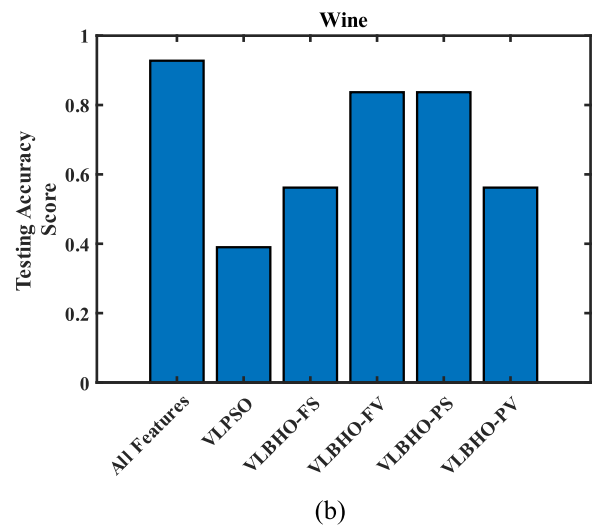
The second is the evaluation based on the benchmarking dataset, which is presented in B.

1) OPTIMIZATION ERROR

Each of the algorithms was run ten times and the performance is reported in Figures 4 and 5 for the forest cover dataset, in which the minimised cost of the optimisation of feature selection follows different statistical behaviour according to the algorithm. However, each of the VL-BHO variants accomplished lower cost values and more narrow distribution compared with the benchmark VL-PSO. On the other hand, the S-shaped position variant of VL-BHO was the best at providing more minimum cost compared with the other variants. In addition, the algorithms were run on the wine dataset for feature selection and generated the fitness values, as depicted in Figure 4, for our developed VL-BHO with its four variants and for the benchmark of VL-PSO. Obviously, the algorithm accomplished lower values of cost compared with the benchmark. The best performance was observed for the S-Shaped function-based VL-BHO. This can be interpreted from the dimension decomposition perspective of the algorithm, which is done through dealing with solutions



(a)



(b)

FIGURE 6. Accuracy comparison between our developed VLBHO variants, the benchmark VLPSO and using all the features for -a- the forest cover dataset -b- the wine dataset.

or stars according to their dimension, where each dimension has its own exemplar. In addition, the superiority can be interpreted because of the effectiveness of the concepts of energy and the number of improvements, which assists in deciding when to kill a certain solution or replace it with another.

2) TESTING ACCURACY

The evaluation of our developed VLBHO with its four variants, FV, FS, PV and PS was evaluated and compared with the benchmark VLPSO [11] and with the inclusion of all features. It was observed from Figure 6 that the highest testing accuracy was accomplished with a value of 67% with VLBHO-PV, which was close to the case of the classification with all the features. The least performance was observed with the benchmark and VLBHO-FS, with an accuracy rate of around 50%.

The testing accuracy for the wine dataset is presented in Figure 7, where the best accomplished accuracy of nearly

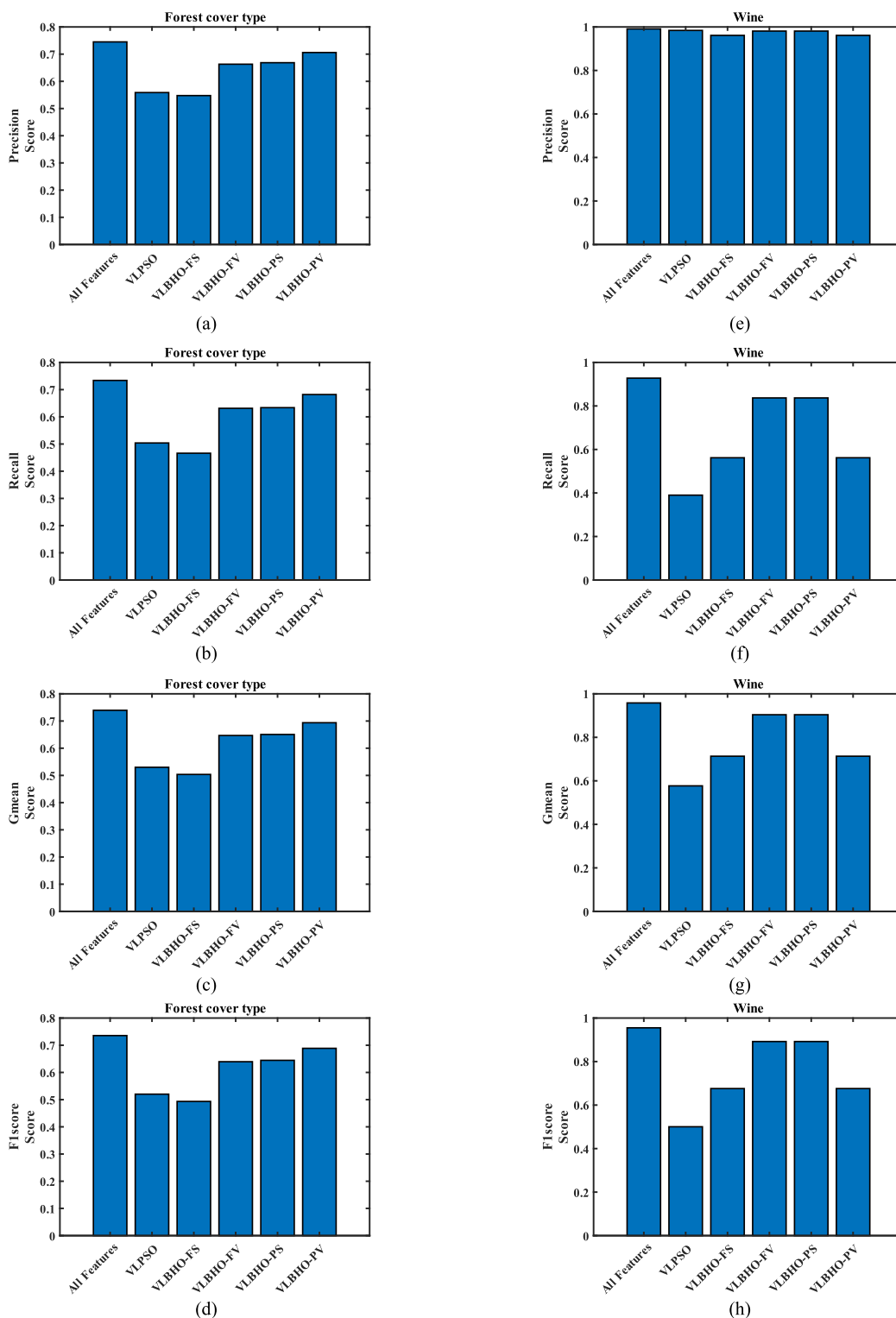


FIGURE 7. Classification metrics for our developed VLBHO for Wine dataset (a-d) forest dataset (e-h) Wine dataset.

80% resulted from VLBHO-FV and VLBHO-PS. The least performance resulted from VLPSO, with an accuracy of lower than 40%. For further elaboration of the performance, the other classification metrics for the two datasets are shown in Figure 7. We present the numerical results of the

algorithm in Tables 4 and 5 for Forest and Wine datasets respectively.

For further elaboration, the confusion matrix matrices are presented in Figure 8 for the forest cover dataset and for the wine dataset in Figure 9. The numbers show that each

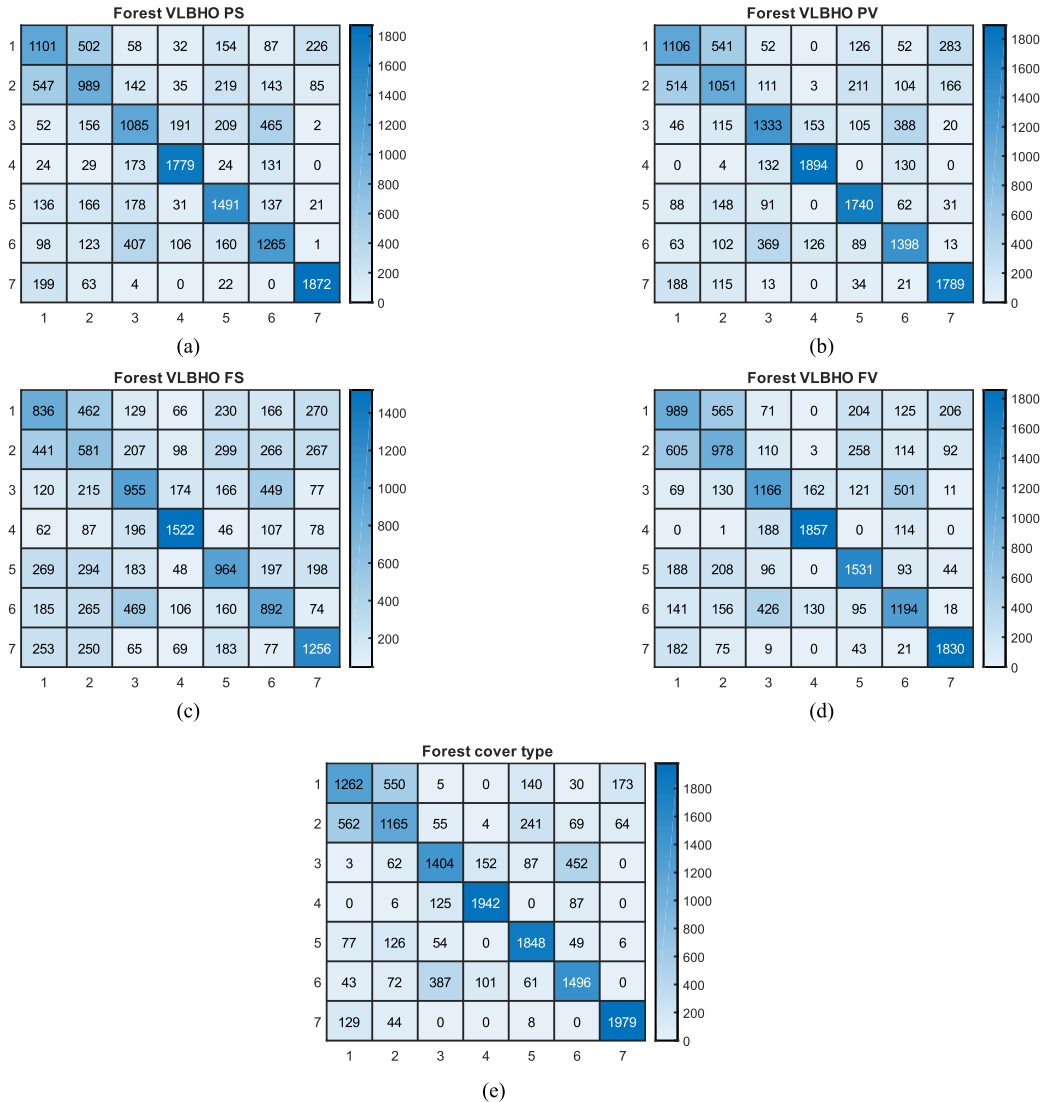


FIGURE 8. Confusion matrix comparison between our developed VLBHO variants (a) VLBHO PS (b) VLBHO PV (c) VLBHO FS (d) VLBHO FV, the benchmark (e) VLP SO and using all the features, for the forest cover dataset.

TABLE 4. Classification metrics for our developed VLBHO for forest dataset.

Forest	All Feature	VLBSO	VLBHO-FS	VLBHO-FV	VLBHO-PS	VLBHO-PV
Testing Accuracy	0.734	0.504	0.466	0.631	0.634	0.682
F1 Score	0.735	0.520	0.493	0.635	0.644	0.689
Precision Score	0.745	0.559	0.548	0.663	0.668	0.706
Recall Score	0.734	0.504	0.466	0.631	0.634	0.682
G-mean Score	0.739	0.530	0.504	0.647	0.651	0.694

TABLE 5. Classification metrics for our developed VLBHO for Wine dataset.

Wine	All Feature	VLBSO	VLBHO-FS	VLBHO-FV	VLBHO-PS	VLBHO-PV
Testing Accuracy	0.928	0.383	0.562	0.837	0.837	0.562
F1 Score	0.955	0.496	0.679	0.892	0.892	0.679
Precision Score	0.991	0.982	0.975	0.981	0.981	0.975
Recall Score	0.928	0.383	0.562	0.837	0.837	0.562
G-mean Score	0.958	0.572	0.719	0.904	0.904	0.719

algorithm generated a different number of misclassifications. However, VLBHO-PV accomplished the best performance for the forest cover dataset as its diagonal values are higher compared with the other algorithms.

The results of applying our developed VLBHO on the wine dataset are presented in Figure 9. The confusion matrices indicate that the best accomplished results of predictions were for VLBHO-PS and VLBHO-FV, with a diagonal value of

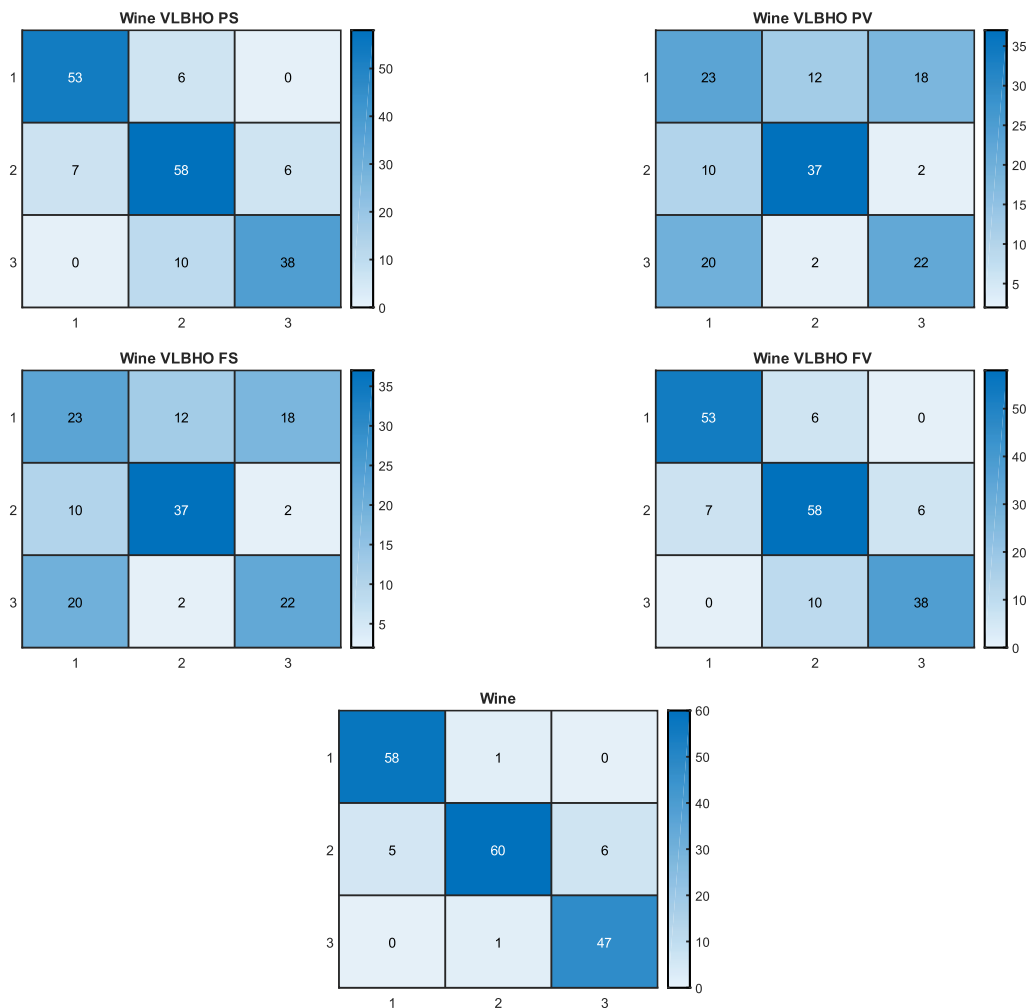


FIGURE 9. Confusion matrix comparison between our developed VLBHO variants, the benchmark VLPSO and using all the features, for the wine dataset.

35, 58 and 38, which is consistent with the accuracy results presented in Figure 6.

In summarising the performance, the best accuracy was achieved for VLBHO, and its comparison with the benchmark VLPSO are presented in Table 3. The results show that VLBHO reached an accuracy of 67%, compared with an accuracy of only 50% for VLPSO for the forest cover dataset. In addition, it was observed that the accuracy reached for the wine dataset was 80% for VLBHO, compared with only 38% for the VLPSO dataset.

IV. CONCLUSION

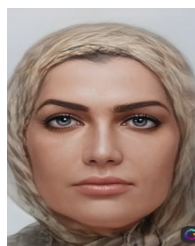
This article has presented a variable length meta-heuristic algorithm for feature selection. The algorithm uses the metaphor of original black hole optimisation as the baseline for development. It was modified with various aspects: namely, it enables the decomposing of the solution space into a subset of dimensions and can search within each dimension separately by selecting an exemplar for each dimension, which represents the black hole of the corresponding dimension. In addition, it enables length changing of the solutions

based on the stagnation criterion. Furthermore, it proposes a new concept of energy to the black hole to replace it when it is no longer effective. The proposed algorithm, designated as variable length black hole optimisation VLBHO, is compared with the benchmark of variable length particle swarm optimisation. The evaluation showed its superiority in terms of testing accuracy and other classification metrics. The accuracy of VLBHO reached 67% for the forest cover dataset and 80% for the wine dataset, compared with VLPSO, the accuracy of which was only 50% and 38%, respectively. Future work should apply VLBHO in optimisation problems for object identification and the recognition of biometric authentication. Here, feature selection and optimisation are carried out for high dimensional data, which requires variable length searching.

REFERENCES

- [1] I. Cordón, J. Luengo, S. García, F. Herrera, and F. Charte, "Smartdata: Data preprocessing to achieve smart data in R," *Neurocomputing*, vol. 360, pp. 1–13, Sep. 2019.
- [2] J. Huang, G. Li, Q. Huang, and X. Wu, "Joint feature selection and classification for multilabel learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 876–889, Mar. 2018.

- [3] A. Chaudhuri and T. P. Sahu, "Feature selection using binary crow search algorithm with time varying flight length," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114288.
- [4] A. K. Das, S. Goswami, A. Chakrabarti, and B. Chakraborty, "A new hybrid feature selection approach using feature association map for supervised and unsupervised classification," *Expert Syst. Appl.*, vol. 88, pp. 81–94, Dec. 2017.
- [5] H. Peng and Y. Fan, "Feature selection by optimizing a lower bound of conditional mutual information," *Inf. Sci.*, vols. 418–419, pp. 652–667, Dec. 2017.
- [6] W. Wei, S. Chen, Q. Lin, J. Ji, and J. Chen, "A multi-objective immune algorithm for intrusion feature selection," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106522.
- [7] N. Gopika and A. M. Kowshalya, "Correlation based feature selection algorithm for machine learning," in *Proc. 3rd Int. Conf. Commun. Electron. Syst. (ICCES)*, Oct. 2018, pp. 692–695.
- [8] J. Lee, J.-Y. Jeong, and C.-H. Jun, "Markov blanket-based universal feature selection for classification and regression of mixed-type data," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113398.
- [9] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC hybrid)," *Swarm Evol. Comput.*, vol. 36, pp. 27–36, Oct. 2017.
- [10] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106040.
- [11] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, Jun. 2019.
- [12] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2012.
- [13] A. Ya, "Feature extraction techniques for iris recognition system: A survey," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 8, no. 2, pp. 2347–5552, Mar. 2020.
- [14] L. Kumar and K. K. Bharti, "An improved BPSO algorithm for feature selection," in *Recent Trends in Communication, Computing, and Electronics*. Singapore: Springer, 2019, pp. 505–513.
- [15] P. Kumari, "A fast feature selection technique in multi modal biometrics using cloud framework," *Microprocessors Microsyst.*, vol. 79, Nov. 2020, Art. no. 103277.
- [16] A. A. Nithya and C. Lakshmi, "Enhancing iris recognition framework using feature selection and BPNN," *Cluster Comput.*, vol. 22, no. S5, pp. 12363–12372, Sep. 2019.
- [17] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neuro Comput.*, vol. 260, pp. 302–312, Oct. 2017.
- [18] G. I. Sayed, A. Tharwat, and A. E. Hassanien, "Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection," *Int. J. Speech Technol.*, vol. 49, no. 1, pp. 188–205, Jan. 2019.
- [19] N. Ghadimi, A. Akbarimajd, H. Shayeghi, and O. Abedinia, "Two stage forecast engine with feature selection technique and improved metaheuristic algorithm for electricity load forecasting," *Energy*, vol. 161, pp. 130–142, Oct. 2018.
- [20] M. Choudhary, V. Tiwari, and U. Venkanna, "Iris anti-spoofing through score-level fusion of handcrafted and data-driven features," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106206.
- [21] K. Vairavel and S. Valarmathy, "Implementing feature selection for multimodal biometrics," *Int. J. Biomed. Eng. Technol.*, vol. 23, nos. 2–4, pp. 242–260, 2017.
- [22] S. Rathee and S. Ratnoo, "Feature selection using multi-objective CHC genetic algorithm," *Proc. Comput. Sci.*, vol. 167, pp. 1656–1664, Jan. 2020.
- [23] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," *Eng. Appl. Artif. Intell.*, vol. 101, May 2021, Art. no. 104216.
- [24] X.-H. Wang, Y. Zhang, X.-Y. Sun, Y.-L. Wang, and C.-H. Du, "Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106041.
- [25] *Forest Cover Type Prediction*. Accessed: 2014. [Online]. Available: <https://www.kaggle.com/c/forest-cover-type-prediction/data>
- [26] *Wine Dataset*. Accessed: 2005. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/wine>
- [27] *Forest Cover Dataset*. Accessed: 2015. [Online]. Available: <https://www.kaggle.com/c/forest-cover-type-prediction/data>
- [28] *Datasets*. Accessed: 2018. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets.php>



TARA OTHMAN QADIR received the B.Sc. degree in computer science and the M.Sc. degree in biometric in Baghdad. After that, she was employed in Iraqi Commission for Computers Informatics, Scientific Technology Information Center in Baghdad. She is currently with the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Johor, Malaysia; a Lecturer with the Department of Software and Informatics, College of Engineering, Salahaddin University, Erbil, Iraq; and a Lecturer in communication engineering in Sulaymaniyah.



NORFAIZA FUAD received the B.Eng. degree (Hons.) in computer engineering from the Universiti Teknologi Malaysia, in 2003, the M.Sc. degree in computer system engineering from the Universiti Putra Malaysia, in 2006, and the Ph.D. degree in advance signal processing from the Faculty Electrical Engineering, University Teknologi MARA, Shah Alam, Malaysia. She is currently a Senior Lecturer with the Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Batu Pahat Johor, Malaysia.



NIK SHAHIDAH AFIFI MD TAUJUDDIN received the B.Sc. (Hons.) and M.Sc. degrees in electrical and electronic engineering from the Universiti Teknologi Malaysia and the Ph.D. degree in image processing from the Universiti Tun Hussein Onn Malaysia. She is currently with the Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia. She also used to be a Visiting Researcher with the Nagaoka University of Technology, Japan. Her research areas include image processing, computer security, and computer networks.

• • •