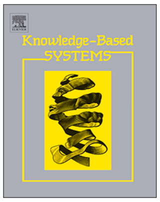
Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



A Hyper Learning Binary Dragonfly Algorithm for Feature Selection: A COVID-19 Case Study

Jingwei Too [a,*], Seyedali Mirjalili [b,c,d,1]

[a] Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia
[b] Center for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane, QLD 4006, Australia
[c] Yonsei Frontier Lab, Yonsei University, Seoul, Republic of Korea
[d] King Abdul Aziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 10 August 2020
Received in revised form 18 October 2020
Accepted 20 October 2020
Available online 31 October 2020

Keywords:
Binary Dragonfly Algorithm
Feature selection
Data mining
Optimization
Classification
Algorithm
Binary Optimization
Particle Swarm Optimization
Combinatorial Optimization
Artificial Intelligence

ABSTRACT

The rapid expansion of information science has caused the issue of “the curse of dimensionality”, which will negatively affect the performance of the machine learning model. Feature selection is typically considered as a pre-processing mechanism to find an optimal subset of features from a given set of all features in the data mining process. In this article, a novel Hyper Learning Binary Dragonfly Algorithm (HLBDA) is proposed as a wrapper-based method to find an optimal subset of features for a given classification problem. HLBDA is an enhanced version of the Binary Dragonfly Algorithm (BDA) in which a hyper learning strategy is used to assist the algorithm to escape local optima and improve searching behavior. The proposed HLBDA is compared with eight algorithms in the literature. Several assessment indicators are employed to evaluate and compare the effectiveness of these methods over twenty-one datasets from the University of California Irvine (UCI) repository and Arizona State University. Also, the proposed method is applied to a coronavirus disease (COVID-19) dataset. The results demonstrate the superiority of HLBDA in increasing classification accuracy and reducing the number of selected features.[2]

1. Introduction

With the rapid expansion of information science, the number of dimensions in datasets has been dramatically increased over the last decade [1,2]. According to [3], some features of the datasets are known to be irrelevant and redundant information, and these features can negatively affect the performance of the learning algorithm. Hence, potential dimension reduction algorithms are required to resolve this challenge.

Feature selection is typically considered as a pre-processing mechanism to find an optimal sub-set of features from a given set of all features in the data mining process. It aims to reduce the complexity, minify the unwanted features (irrelevant and redundant features), and enhance the prediction accuracy of the machine learning model [4]. Such pre-processing algorithms involve two main stages: search strategy and feature subset evaluation. The former uses some techniques to select a subset of features, while the latter employs the classifier to evaluate the quality of the selected sub-set of features [5]. All feature selection methods in the literature can be divided into two classes: filter-based and wrapper-based. The filter-based method utilizes the data-reliant specifications to assess the merits of the feature subset. On the one hand, the wrapper-based method employed a specific classifier to estimate the significant features [6]. As compared to the wrapper-based, filter-based method is less time-consuming. This is because no machine learning algorithm is involved in the feature selection process. However, the wrapper-based method can often find the best feature subset that expresses the original features [6,7].

In the past decade, metaheuristic algorithms have been regarded as the most efficient and reliable optimization techniques especially when solving challenging, high-dimensional problems. These algorithms have been widely used for performance improvement in real-world problems [5]. Some examples of the metaheuristic algorithms are particle swarm optimization [8], genetic algorithm [9], water cycle algorithm [10], animal migration optimization [11], forest optimization algorithm [12], human learning algorithm [13], heat transfer search [14], lightning search algorithm [15], monarch butterfly optimization [16], and dragonfly algorithm [17]. Recently, the metaheuristic algorithms have been successfully applied to various engineering and clinical

[*] Corresponding author.
E-mail addresses: jamesjames868@gmail.com (J. Too), ali.mirjalili@gmail.com (S. Mirjalili).

[1] https://seyedalimirjalili.com/.

[2] The source code of HLBDA is publicly available at https://seyedalimirjalili.com/da.

https://doi.org/10.1016/j.knosys.2020.106553
0950-7051/© 2020 Elsevier B.V. All rights reserved.

problems such as optimized connection weights in the neural network [18], numerical optimization [19], cloud computing [20], stock market index prediction [21], and solar photovoltaic (PV) parameter estimation [22].

Due to the powerful and excellent performance of metaheuristic algorithms, researchers employ them (wrapper-based methods) to solve the feature selection problems. Zhang et al. [23] proposed a Gaussian mutational chaotic fruit fly optimization algorithm to solve the feature selection problems in classification tasks. An optimized crow search algorithm was developed to select potential features for improving the diagnosis of Parkinson's disease [24]. The study proposed a novel fitness function with a random forest classifier to boost the performance of the chaotic crow search algorithm. However, the proposed algorithm may require more computational time when applied to high dimension dataset due to the high complexity of the random forest model.

Dragonfly algorithm (DA) is a recently established algorithm inspired by the swarming patterns of dragonflies in nature. It is originally designed to solve continuous optimization tasks. Later on, Mirjalili proposed a binary version of the dragonfly algorithm (BDA), which can be used to solve discrete problems (e.g. feature selection) [17]. Although BDA has shown its supreme performance in several benchmark tests, however, the strong exploitation of BDA may cause the local optima problem. Consequently, BDA may fail to find the global optimal solution [25]. Thus, different strategies can be implemented into BDA for performance enhancement.

In this research, we propose a Hyper Learning Binary Dragonfly Algorithm (HLBDA) to evaluate the most discriminating features for classification. The proposed HLBDA utilizes the hyper learning strategy, which enables the dragonflies to learn from the personal best and global best solutions during the search phase. The HLBDA algorithm is benchmarked using 21 benchmark datasets collected from the University of California, Irvine (UCI) repository and Arizona State University, and one coronavirus disease (COVID-19) dataset. Eight state-of-the-art feature selection algorithms are used to examine the efficacy of proposed HLBDA in this research. The experimental results show that the proposed HLBDA can often overwhelm other methods in finding the most informative features.

The main contributions of this research are summarized as follow:

- Proposal of HLBDA for feature selection problems. A hyper learning strategy is integrated into HLBDA to improve the exploratory behavior of BDA.
- Testing of HLBDA on twenty-one benchmark datasets in which the low, medium, and high dimension datasets are involved.
- Comparing and demonstrating the superiority of HLBDA against the other eight state-of-the-art feature selection algorithms.
- Solving a COVID-19 case study using the proposed HLBDA.

The organization of this article is as follows: Section 2 discusses the related works. Section 3 introduces the standard BDA method. Section 4 describes the proposed HLBDA and its application for feature selection. Section 5 presents the findings of the experiment. Section 6 demonstrates the adoption of the proposed HLBDA in a COVID-19 application. Section 7 discusses the experimental results. Finally, Section 8 concludes the outcome of the research.

2. Related work

As part of the popular technologies in information areas, data mining and data analysis have established the potential value of big data in the training process [26]. In recent years, many feature selection algorithms have been proposed and adopted for diagnosis, classification, categorization, pattern recognition, and detection [27–33]. For filter-based feature selection, Xue et al. [34] proposed a robust re-weighted multi-view feature selection algorithm. The proposed model allowed multiple relevant views as well as the specificity of each view in the process of selection. In [35], a feature selection model associated with a radial basis function network was designed to rank the importance of features. Kanimozhi and Manjula [36] proposed an incremental filtering feature selection approach for effective classification. They integrated a fuzzy rough set for assessing high-quality features. Another class discriminative degree-based feature selection approach was proposed in [37] for medical diagnosis.

Among the wrapper-based methods, particle swarm optimization (PSO) and its binary variant (BPSO) have been widely used in feature selection works. The self-adaptive parameter and strategy based PSO [38] and improved BPSO [39] were two recent approaches implemented in wrapper feature selection. Lately, authors in [40] introduced a multi-swarm mechanism to separate all the particles into several small size sub-swarms, and then a local version PSO was used to progress those sub-swarms. Besides, the proposed PSO approach utilized an elite learning strategy that allowed the exchange of information between sub-swarms, which can effectively improve the diversity of the population.

Another popular and well-known wrapper method was the genetic algorithm (GA). The GA algorithm solves optimization problems using principles of evolution in nature: selection, crossover, and mutation [41]. However, GA is a stochastic algorithm and requires tuning to ensure its maximum performance [42]. Therefore, Kromer et al. [43] designed a new crossover operator according to the Mergesort algorithm, which enabled the recombination of valid solutions and avoided invalid resolutions. Later on, Too and Abdullah [44] proposed a rival genetic algorithm and its fast version for feature selection problems. The rival genetic algorithm first divided the chromosomes into winner and loser groups, and then the parents were selected based on the tournament selection for crossover operation. Afterward, the mutation was performed using a dynamic mutation operator. Their study reported the proposed approaches can perform faster and overtake other conventional methods. However, the algorithm was complicated due to its complex structure.

In [4], Hancer et al. proposed a binary artificial bee colony based on the Jaccard coefficient dissimilarity. The authors borrowed a neighborhood mechanism from the differential evolution algorithm and integrated it into the similarity-based search to expand the global search capability. They reported that the proposed approach can usually overtake other traditional methods in feature subset selection. Nevertheless, the proposed approach was getting complicated because of its complex structure. Moreover, Wan et al. [45] modified the binary coded ant colony optimization with pheromone and visibility density models for feature selection problems. They performed the hybridization of binary ant colony optimization and genetic algorithm within two models to facilitate the searching behavior and improve the robustness of the algorithm. However, the proposed algorithm only validated on low dimensional datasets with the number of features less than 100.

An improved harris hawk optimizer (IHHO) based feature selection has been developed by [46]. The study performed a hybridization of harris hawk optimizer and salp swarm algorithm to enhance the searching behavior during optimization. In [47], the chaotic atom search optimization that used twelve different chaotic maps were proposed. The authors employed the chaotic map to alter the multiplier weight in order to boost the convergence behavior. Moreover, two variants of atom search

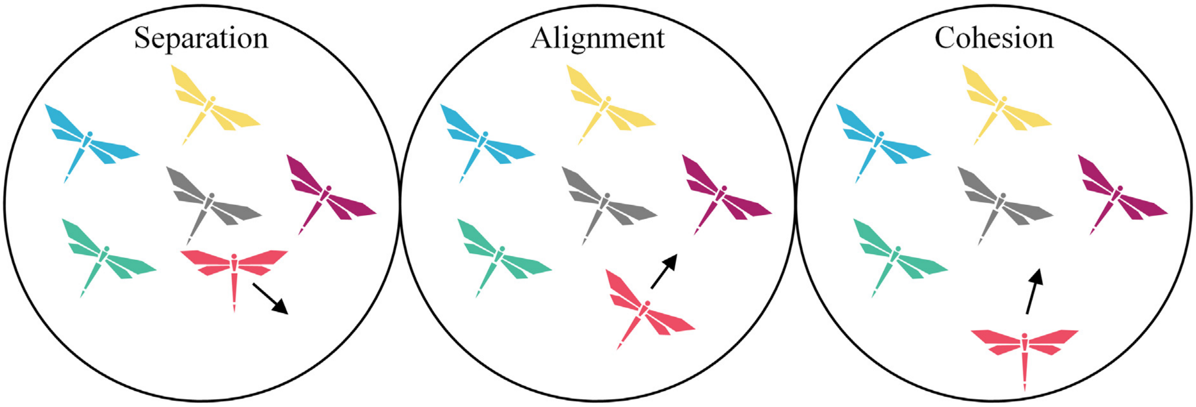


Fig. 1. Three principles of swarming in nature: separation, alignment, and cohesion.

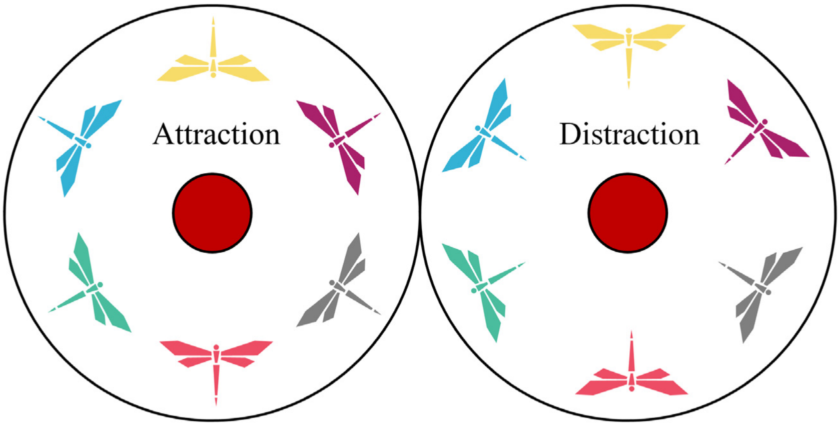


Fig. 2. Two more principles of swarming in nature considered in DA: attraction and distraction.

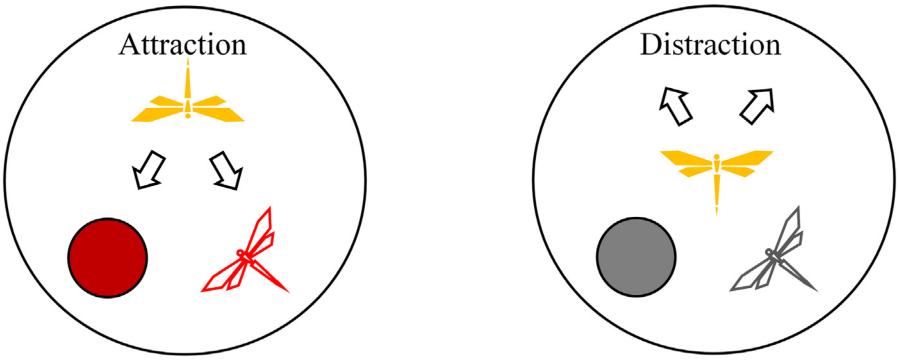


Fig. 3. The attraction and distraction behaviors of the proposed HLBDA.

optimization based feature selection approaches were proposed in [48,49]. Another feature selection study that employed the ant colony optimization, artificial bee colony and GA algorithm for keystroke dynamic was presented in [50].

Gray wolf optimizer (GWO) is a recent metaheuristic algorithm that mimics the leadership hierarchy and hunting behavior of gray wolves in nature [51]. Lately, the GWO and its binary variants (BGWO) have been used as the wrapper method for feature selection in [52,53]. Their studies showed that the proposed GWO variants outperformed other optimizers. Moreover, Chantar et al. [54] developed an elite crossover mechanism into BGWO for Arabic text classification. In addition, Too and Abdullah [55] implemented the opposition based learning into competitive gray wolf optimizer to improve the prediction accuracy of hand motions classification. Furthermore, authors in [56] proposed a fusion of gray wolf optimizer and bi-phase mutation operator to evolve the character of the leader during the search process.

Another inspiration based on the whale optimization algorithm (WOA) was developed for feature selection. As an example, Mafarja and Mirjalili [57] proposed WOA based feature selection approach in which the tournament selection was employed to replace the random mechanism in the selection process. In [58], Mafarja and Mirjalili hybridized the WOA with a simulated annealing algorithm, which strengthened the capability of the algorithm in exploring the feature space. Moreover, Tubishat et al. [59,60] proposed an improved WOA to solve the feature and attribute selection issues in sentiment analysis. In the former study, they first applied the information gain to filter some

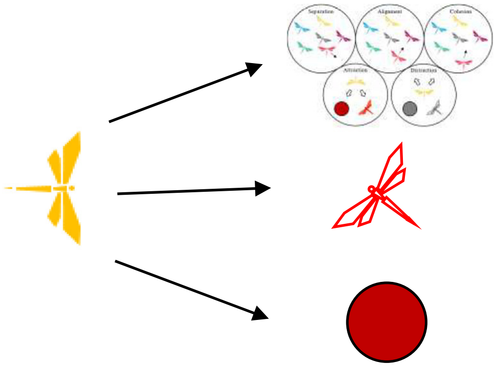


Fig. 4. General concept of the learning strategy.

unwanted features, and then a novel approach was introduced based on opposition-based learning principles to generate the new opposite solutions.

In COVID-19 patient detection, Shaban et al. [61] introduced a hybrid feature selection methodology to select the optimal features extracted from the chest computed tomography (CT) images. They first filtered some unwanted features and then accurately selected the informative feature subset using GA. Jain et al. [62] proposed a deep network model to identify the COVID-19 disease based on X-ray images. The proposed model was reliable and accurate, however, it heavy relies on the stage1 model [62]. To effectively remove the negative features from COVID-19 prediction, Tuncer et al. [63] designed an iterative ReliefF based feature selection algorithm. Moreover, the authors in [64] applied six popular machine learning algorithms for forecasting one, three, and six days-ahead the COVID-19 cumulative confirmed cases in ten Brazilian states. Another recent COVID-19 study was presented in [65].

The dragonfly algorithm (DA) was recently developed by Mirjalili for global optimization purpose. He then proposed a V-shaped transfer function and integrated it into DA to permit the dragonflies for searching on the binary search space [17]. In [66], the authors employed the Binary Dragonfly Algorithm (BDA) for feature selection. However, like other metaheuristic algorithms, the BDA also has the limitation of restricting the local optima. Therefore, Sayed et al. [25] integrated the chaos theorem into BDA to boost the convergence rate. Among ten different chaotic maps, they reported that the Gauss map was more capable of enhancing the performance of learning model. Moreover, an improved BDA approach was designed for infant cry classification [67]. The proposed model applied the elitism mechanism that stored the potential dragonfly and then inserted it to replace the worst dragonfly in the population, which enhanced the convergence behavior of the algorithm in finding the promising solutions.

3. Binary Dragonfly Algorithm

Binary Dragonfly Algorithm (BDA) is a discrete version of the Dragonfly Algorithm (DA) proposed by Mirjalili in 2016. As mentioned above, this algorithm mimics the swarming patterns of dragonfly in nature. The exploratory and exploitative mechanisms of DA are modeled by the interaction of dragonflies in avoiding the enemy (the worst solution) and finding the food source (the best solution) [17]. In DA, there are five main behaviors, namely separation, alignment, cohesion, attraction (see Fig. 1), and distraction, which are used in the position updating process.

Each of these behaviors is described as follows:

- *Separation* aims to prevent the static collision of the current individual with the individual in the neighborhood. Mathematically, separation is expressed as follow:

$$S_i = -\sum_{j=1}^{M} X - X_j \tag{1}$$

where X indicates a dragonfly's position in an D-dimensional space (D is the number of decision variables), X_j is the position of the neighbor individual, and M is the number of neighbor individuals.

- *Alignment* allows velocity matching of individuals in a swarm or sub-swarm. The alignment is calculated as below:

$$A_i = \frac{\sum_{j=1}^{M} V_j}{M} \tag{2}$$

where V_j is the velocity of neighbor individuals, and M is the number of neighbor individuals.

- *Cohesion* refers to the deviation of the current individual toward the center of the mass of neighbor individuals. The cohesion is defined as follow:

$$C_i = \frac{\sum_{j=1}^{M} X_j}{M} - X \tag{3}$$

where X_j indicates the position of the neighbor individual, and M is the total number of dragonflies in the neighborhood.

In addition to separation, alignment, and cohesion, individuals in natural swarms attract toward food sources and distract from predators. These two principals have been also mathematically modelled in the DA algorithm as follows (see Fig. 2):

- *Attraction* indicates the individual should be attracted toward sources of food. Mathematically, the attraction is expressed as:

$$F_i = Xf - X \tag{4}$$

where Xf indicates the position of a food source.

- *Distraction* implies the individual should be distracted outward from a predator. The distraction is calculated as below:

$$E_i = Xe + X \tag{5}$$

where Xe is the position of the enemy.

The movement of dragonflies in DA are controlled by these five behaviors. To update the position of each dragonfly, the following step vector is calculated:

$$\Delta X_i\,(t+1) = (sS_i + \alpha A_i + cC_i + fF_i + eE_i) + w\Delta X_i\,(t+1) \tag{6}$$

where s refers to the weight of separation, α shows the weight of alignment, c shows the weight of cohesion, f is the weight of food, e is the weight of a predator, w shows the inertia weight, and t is representing the current iteration.

In original DA, the positions of dragonflies are updated using the following equation:

$$X_i\,(t+1) = X_i\,(t) + \Delta X_i\,(t+1) \tag{7}$$

Such movement and navigations allow this algorithm to solve continuous problems. Unlike DA, the position vectors of BDA are updated using the following equations:

$$X_i^d\,(t+1) = \begin{cases} 1 - X_i^d\,(t) & rand < TF\left(\Delta X_i^d\,(t+1)\right) \\ X_i^d\,(t) & rand \geq TF\left(\Delta X_i^d\,(t+1)\right) \end{cases} \tag{8}$$

$$TF\,(\Delta X) = \left|\frac{\Delta X}{\sqrt{\Delta X^2 + 1}}\right| \tag{9}$$

where X_i^d is the d-th position of ith dragonfly, *rand* shows a randomly generated number in the interval of 0 and 1, ΔX is the step vector, t represents the current iteration, and $TF(.)$ is the transfer function as shown in Eq. (9). Algorithm 1 shows the pseudocode of BDA.

With separation, alignment, cohesion, the BDA can often provide different kinds of local and global searches during the optimizations. Besides, the attraction and distraction are the other factors that allow the dragonflies to exploit the best solutions and move away from the worst solutions. These five swarming behaviors have made the BDA algorithm superior.

BDA employs a V-shape transfer function to calculate altering probability for the position of dragonflies. With this transfer function BDA does not force the dragonfly to choose values of 1 and 0 [17] as opposed to other binary metaheuristics. Hence, BDA has high exploration that supported it to seek out the promising search space.

```
Algorithm 1. Binary Dragonfly Algorithm
Randomly initialize positions of $N$ dragonflies, $X$
Initialize the step vectors, $\Delta X$ to zeros
while (Maximum number of iterations is not met)
    Evaluate the fitness values of dragonflies
    Update the food source, $Xf$ and enemy, $Xe$
    Update $s, \alpha, c, f, e$, and $w$
    for $i = 1$ to number of dragonflies, $N$
        Calculate $S, A$, and $C$ using (1), (2), and (3)
        Compute $F$ and $E$ using (4) and (5)
        Update step vector using (6)
        Update the position of dragonfly ($i$-th) using (8)
    end for
end while
Output: Best food source, $Xf$
```

4. Proposed Hyper Learning Binary Dragonfly Algorithm for feature selection[2]

In this section, a novel Hyper Learning Binary Dragonfly Algorithm (HLBDA) is proposed and applied to feature selection. The HLBDA utilizes a hyper learning strategy that involves the concept of personal best and personal worst solutions in the position updating process (see Fig. 3). In the conventional BDA, the dragonflies concertize the global best solution (food source) and global worst solution (enemy) for attraction and distraction. However, by adding the personal best and personal worst dragonflies into these activities, it is believed that the potential of food finding and enemy fleeing behaviors can be improved.

Unlike BDA, the attraction and distraction of HLBDA are calculated using the following equations.

$$F_i = \frac{(Xpb_i - X_i) + (Xf - X_i)}{2} \tag{10}$$

$$E_i = \frac{(Xpw_i + X_i) + (Xe + X_i)}{2} \tag{11}$$

where Xpb is the position of the personal best dragonfly, Xpw shows the personal worst dragonfly, X is the dragonfly's position, Xf is the food source, and Xe is the enemy.

Furthermore, the hyper learning strategy enables the dragonflies to learn from both personal best and global best solutions during the search phase [13]. Fig. 4 shows the general concept of the learning strategy. Instead of updating the position based on the swarming behaviors, the dragonfly attempts to imitate from its personal best and the global best experiences.

In the proposed HLBDA, a dragonfly's position is updated as follows:

$$X_i^d(t+1) = \begin{cases} \overline{X}_i^d & 0 \le r_1 < pl \\ Xpb_i^d(t) & pl \le r_1 < gl \\ Xf^d(t) & gl \le r_1 \le 1 \end{cases} \tag{12}$$

$$\overline{X}_i^d = \begin{cases} 1 - X_i^d(t) & r_2 < TF\left(\Delta X_i^d(t+1)\right) \\ X_i^d(t) & r_2 \ge TF\left(\Delta X_i^d(t+1)\right) \end{cases} \tag{13}$$

where X is the position of the dragonfly, Xpb is the position of the personal best dragonfly, Xf is the food source, i is the order of dragonfly, d indicates the number of decision variables (dimension), t shows the current iteration, r_1 and r_2 are two independent random values in the interval of 0 and 1. The pl and gl are the personal learning rate and global learning rate with constant values in [0,1].

As given in Eq. (12), the pl and gl played important roles in the learning process. If the values of pl and gl are too low, the algorithm will mostly search around the personal best and global best solutions, which is easily getting trapped in the local optima. On the contrary, the position updating process will likely similar to BDA if the values of pl and gl are too high. Hence, the selections of pl and gl are critically important. The pseudocode of HLBDA is presented in Algorithm 2.

```
Algorithm 2. Hyper Learning Binary Dragonfly Algorithm
Randomly initialize positions of $N$ dragonflies, $X$
Initialize the step vectors, $\Delta X$ to zeros
while (Maximum number of iterations is not met)
    for $i = 1$ to number of dragonflies, $N$
        Calculate the fitness value dragonfly ($i$-th)
        Update the personal best dragonfly, $Xpb_i$
        Update the personal worst dragonfly, $Xpw_i$
    end for
    Update the food source, $Xf$ and enemy, $Xe$
    Update $s, \alpha, c, f, e$, and $w$
    for $i = 1$ to number of dragonflies, $N$
        Calculate $S, A$, and $C$ using (1), (2), and (3)
        Compute $F$ and $E$ using (10) and (11)
        Update step vector using (6)
        Update the position of the dragonfly ($i$-th) using (12)
    end for
end while
Output: Best food source, $Xf$
```

The principal steps of the proposed HLBDA are discussed in detail in the following sub-sections:

4.1. Initialization

The HLBDA starts with the setting of parameters, and then randomly initializes a population of dragonflies as below:

$$X = \begin{bmatrix} X_1^1 & X_1^2 & \cdots & X_1^D \\ X_2^1 & X_2^2 & \cdots & X_2^D \\ \vdots & \vdots & \ddots & \vdots \\ X_N^1 & X_N^2 & \cdots & X_N^D \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{14}$$

As given in Eq. (14), each position expressed a feature candidate. If the position has a value of 1, then the corresponding feature is selected; otherwise, the corresponding feature is unselected. The parameter setting of HLBDA is displayed in Table 1. According to [68,69], we set the number of solutions and maximum iterations at 10 and 100, respectively. In comparison with the conventional BDA, the HLBDA consists of two additional parameters (pl and gl) that need to be properly adjusted. The sensitivity analysis of these parameters is shown in Section 5.2.1.

4.2. Fitness evaluation

Iteratively, each dragonfly is evaluated using a pre-defined objective function. The objective function is used to assess the

Table 1
Parameter setting of HLBDA.

Parameter	Value
Number of dragonflies, N	10
Maximum iterations, T	100
Number of dimensions, D	Number of features in each dataset
Personal learning rate, pl	0.4
Global learning rate, gl	0.7

quality of the solutions [25]. In this article, the objective function is chosen in way to maximize classification accuracy and minimizes the number of selected features. The objective function is as follows [66,70]:

$$\downarrow Fit = \alpha ER + \beta \left(\frac{|S|}{|O|}\right) \tag{15}$$

where ER represents the error of classification, $|S|$ is the length of the sub-set of selected features, $|O|$ is the length of original features. The α ϵ [0,1] and β = (1 - α) are two weight infectors to indicate/prioritize the impact of classification error and feature size on the objective function [52,71].

In the present study, the dataset is partitioned randomly into the training set and validation set using the stratified 10-fold cross-validation method. The intension of the training set is to train the classification algorithm, while the set of remaining samples is used to assess the selected features [25]. As for wrapper-based feature selection, the K nearest neighbor (KNN, $k = 5$) is employed to compute the classification error. The reasons for choosing KNN as the learning algorithm are it is simple, easy to implement algorithm, and low computational cost [72,73].

4.3. Termination

In each iteration, dragonflies are updated (refer to Algorithm 2), and the global best solution is recorded. The optimization process is terminated when the end condition is met. It is assumed in this work that the algorithm will run for a pre-defined maximum number of iterations.

5. Results and analysis

5.1. Dataset

The performance of HLBDA is validated on 21 standard datasets gathered from the UCI repository [74] and Arizona State University [75]. Table 2 presents the detailed of datasets employed. From Table 2, the datasets are made up of various numbers of instances, features, and dimensions. That is, we test the performance of proposed HLBDA on different structures, which can demonstrate the efficacy of the algorithm in different dimensions [76].

5.2. Experimental results

To quantify the performance of HLBDA, five performance evaluation metrics are calculated. These metrics are the best fitness value, mean fitness value, standard deviation of fitness values, classification accuracy, and feature selection ratio. Due to the stochastic of optimization algorithm, each method is repeated 20 times. Finally, the averages of results are collected and reposed after 20 independent runs. The proposed HLBDA is programmed in the language of MATLAB 9.4 on a computer with 2.90 GHz Central Processing Unit (CPU) and 16.0G Random Access Memory (RAM).

Table 2
The utilized twenty-one datasets.

No.	Dataset	Number of training samples	Number of features	Dimension
1	Glass	214	10	Low
2	Hepatitis	155	19	Low
3	Lymphography	148	18	Low
4	Primary Tumor	339	17	Low
5	Soybean	307	35	Medium
6	Horse Colic	368	27	Low
7	Ionosphere	351	34	Medium
8	Zoo	101	16	Low
9	Musk 1	476	166	Medium
10	Arrhythmia	452	279	Medium
11	Dermatology	366	34	Medium
12	SPECT Heart	267	22	Low
13	Libras Movement	360	90	Medium
14	ILPD	583	10	Low
15	Seeds	210	7	Low
16	LSVT	126	310	Medium
17	SCADI	70	205	Medium
18	TOX_171	171	5748	High
19	Leukemia	72	7070	High
20	Lung discrete	73	325	Medium
21	Colon	62	2000	High

Table 3
Parameter settings of comparative feature selection algorithms (number of solutions and maximum iterations are set to 10 and 100 in all algorithms respectively).

Algorithm	Parameter	Value
BDA	All controlling parameters	Identical to the original paper
BABC	Maximum limits	5
BMVO	WEP	[0.02, 1]
	TDR	[0.6, 0]
BPSO	Inertia weight, w	[0.9, 0.4]
	Acceleration factors, c_1 and c_2	2
CCSA	AP	0.1
	fl	2
BCOA	Number of coyotes	5
	Number of packs	2
CMAES	Number of parents	$\lambda/4$
LSHADE	Minimum population size	4
	Memory size	5

5.2.1. Impact of pl and gl

As mentioned before, the proposed HLBDA algorithm has pl and gl parameters that need to be properly tuned. In this subsection, we study the effect of pl and gl on classification accuracy. Fig. 5 shows the boxplot analysis of pl and gl across twenty-one datasets. As can be seen, seven different pairs of pl and gl have been investigated. The higher the value of the median and mean, the better the performance is. Inspecting the result, the first pair (pl=0.3 and gl=0.65) and second pair (pl=0.4 and gl=0.7) yielded the median values greater than 87%. However, the optimal mean value (84.80%) was achieved when pl=0.4 and gl=0.7, and it overtook the first pair in at least thirteen datasets. Hence, only the pl and gl with 0.4 and 0.7 are used in the rest of this experiment.

5.2.2. Comparison with other feature selection methods

This subsection presents the performance comparison of the proposed HLBDA and the other eight well-known algorithms. These algorithms are BDA [17], binary artificial bee colony (BABC) [77], binary multiverse optimizer (BMVO) [72], binary particle swarm optimization (BPSO) [78], chaotic crow search algorithm (CCSA) [79], binary coyote optimization algorithm (BCOA) [80,81], evolution strategy with covariance matrix adaptation (CMAES) [82], and success-history based adaptive differential evolution with linear population size reduction (LSHADE) [83]. Table 3 exhibits the parameter settings of comparison algorithms. To ensure

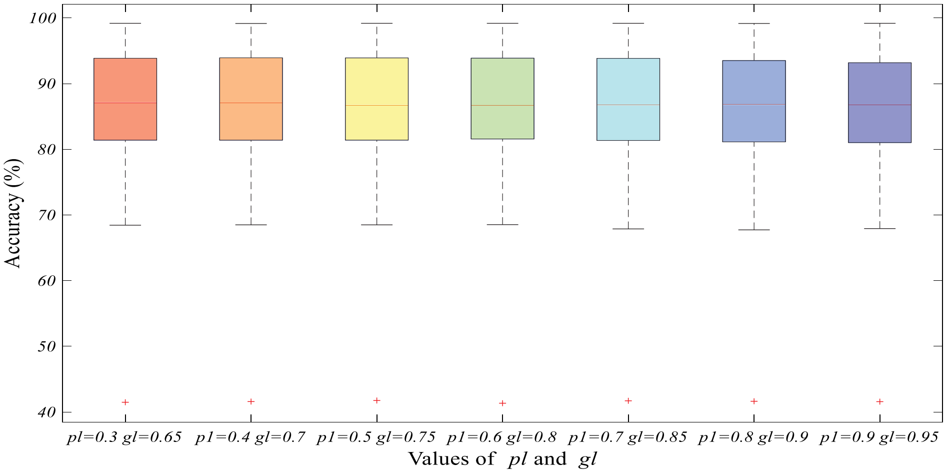


Fig. 5. Boxplot analysis of *pl* and *gl* across twenty-one datasets.

Table 4
The result of the best fitness value of algorithms.

No.	Dataset	Best fitness value								
		HLBDA	BDA	BABC	BMVO	BPSO	CCSA	BCOA	CMAES	LSHADE
1	Glass	**0.0067**	**0.0067**	**0.0067**	**0.0067**	**0.0067**	**0.0067**	**0.0067**	**0.0067**	**0.0067**
2	Hepatitis	**0.1154**	0.1245	0.1305	0.1226	0.1235	0.1309	0.1220	0.1229	0.1235
3	Lymphography	**0.1116**	0.1181	0.1122	0.1297	0.1177	0.1309	0.1257	0.1171	0.1226
4	Primary Tumor	0.5646	0.5731	0.5673	0.5887	**0.5623**	0.5756	0.5642	**0.5623**	0.5883
5	Soybean	**0.2009**	0.2073	0.2035	0.2421	0.2189	0.2294	0.2035	0.2010	0.2037
6	Horse Colic	**0.1298**	0.1330	0.1350	0.1440	0.1309	0.1418	0.1304	**0.1298**	0.1327
7	Ionosphere	**0.0694**	0.0731	0.0830	0.0980	0.0814	0.0906	0.0715	0.0746	0.0719
8	Zoo	0.0332	**0.0325**	0.0332	0.0332	**0.0325**	0.0338	**0.0325**	0.0334	**0.0325**
9	Musk 1	**0.0608**	0.0626	0.0879	0.0942	0.0783	0.0836	0.0662	0.0739	0.0633
10	Arrhythmia	**0.2926**	0.3179	0.3330	0.3352	0.3282	0.3399	0.3106	0.3269	0.2999
11	Dermatology	**0.0129**	0.0132	0.0161	0.0215	0.0156	0.0185	0.0161	0.0132	0.0159
12	SPECT Heart	0.1384	0.1384	0.1415	0.1455	0.1361	0.1411	**0.1337**	0.1388	0.1396
13	Libras Movement	**0.1666**	0.1808	0.1939	0.2022	0.1913	0.1916	0.1721	0.1751	0.1688
14	ILPD	**0.2672**	**0.2672**	0.2722	0.2698	**0.2672**	**0.2672**	**0.2672**	**0.2672**	**0.2672**
15	Seeds	**0.0453**	**0.0453**	**0.0453**	**0.0453**	**0.0453**	**0.0453**	**0.0453**	**0.0453**	**0.0453**
16	LSVT	**0.2389**	0.2696	0.3003	0.2796	0.2621	0.2978	0.3007	0.2866	0.3010
17	SCADI	**0.1157**	0.1167	0.1310	0.1312	0.1177	0.1320	0.1158	0.1166	0.1163
18	TOX_171	**0.1260**	0.1378	0.1897	0.1956	0.1722	0.1733	0.1485	0.1959	0.1383
19	Leukemia	0.0310	0.0313	0.0457	0.0445	0.0456	0.0579	**0.0307**	0.0457	0.0313
20	Lung discrete	**0.0553**	0.0597	0.0828	0.0846	0.0718	0.0828	0.0690	0.0737	0.0558
21	Colon	**0.0823**	0.0965	0.1154	0.1155	0.1131	0.1299	0.0985	0.1135	0.0992

a fair comparison, the same number of solutions and maximum iterations are adopted for each algorithm. From Table 3, the maximum limit of BABC was set at 5. In BMVO, the wormhole existence probability (*WEP*) was increasing from 0.02 to 1, while the traveling distance rate (*TDR*) was decreasing from 0.6 to 0. The 1-norm is adopted for normalization and roulette wheel mechansim. For BPSO, the acceleration factors are set at 2, and the inertia weight was linearly decreasing from 0.9 to 0.4. In CCSA, the awareness probability (*AP*) and flight length (*fl*) were set at 0.1 and 2, respectively. The number of coyotes and packs are set to 5 and 2 in BCOA. For CMAES, the number of parents is fixed at 25% of the solutions. Moreover, the memory size and minimum population size of LSHADE are set at 5 and 4.

Figs. 6 and 7 show the convergence curves of HLBDA on twenty-one datasets. It is seen that proposed HLBDA can usually provide excellent convergence behavior, which overtook other methods in optimal feature subset evaluation. Taking the high dimensional datasets (TOX_171, leukemia, and colon) as the examples, HLBDA can converge faster and deeper to find the global optimum. The result clearly reveals the superiority of proposed HLBDA in high dimensional feature selection problems. One of the reasons why HLBDA shows an accelerated convergence is due to the trait that the dragonflies retain improved during the hyper learning based explorative steps and finding preferable solutions.

Tables 4–6 outline the experiment results of the best fitness, mean fitness, and standard deviation of fitness for the proposed HLBDA. From Table 4, HLBDA perceived the optimal best fitness values on seventeen datasets. Successively, HLBDA has shown a very good effort when dealing with feature selection tasks. Based on the result obtained in Table 5, the proposed HLBDA contributed to the optimal mean fitness value in most cases (sixteen datasets), followed by BABC (three datasets). The result implies that the proposed HLBDA can often find the nearly optimal feature subset, thus leading to satisfactory performance. The outstanding search capability of HLBDA to solve the feature selection problem mainly benefits from the proposed hyper learning strategy. In Table 6, the lowest standard deviation was perceived by BABC (seven datasets), while HLBDA offered the optimal standard deviation in three datasets. Correspondingly, HLBDA can often provide highly consistent results compared to BDA.

Fig. 8 presents the result of the classification accuracy of proposed HLBDA. As can be seen, HLBDA yielded the highest accuracy in most datasets (sixteen datasets), which outperformed other methods in finding the optimal feature subset. On the other hand, the boxplot analysis of nine different methods is presented in Fig. 9. Judging from Fig. 9, the proposed HLBDA provided

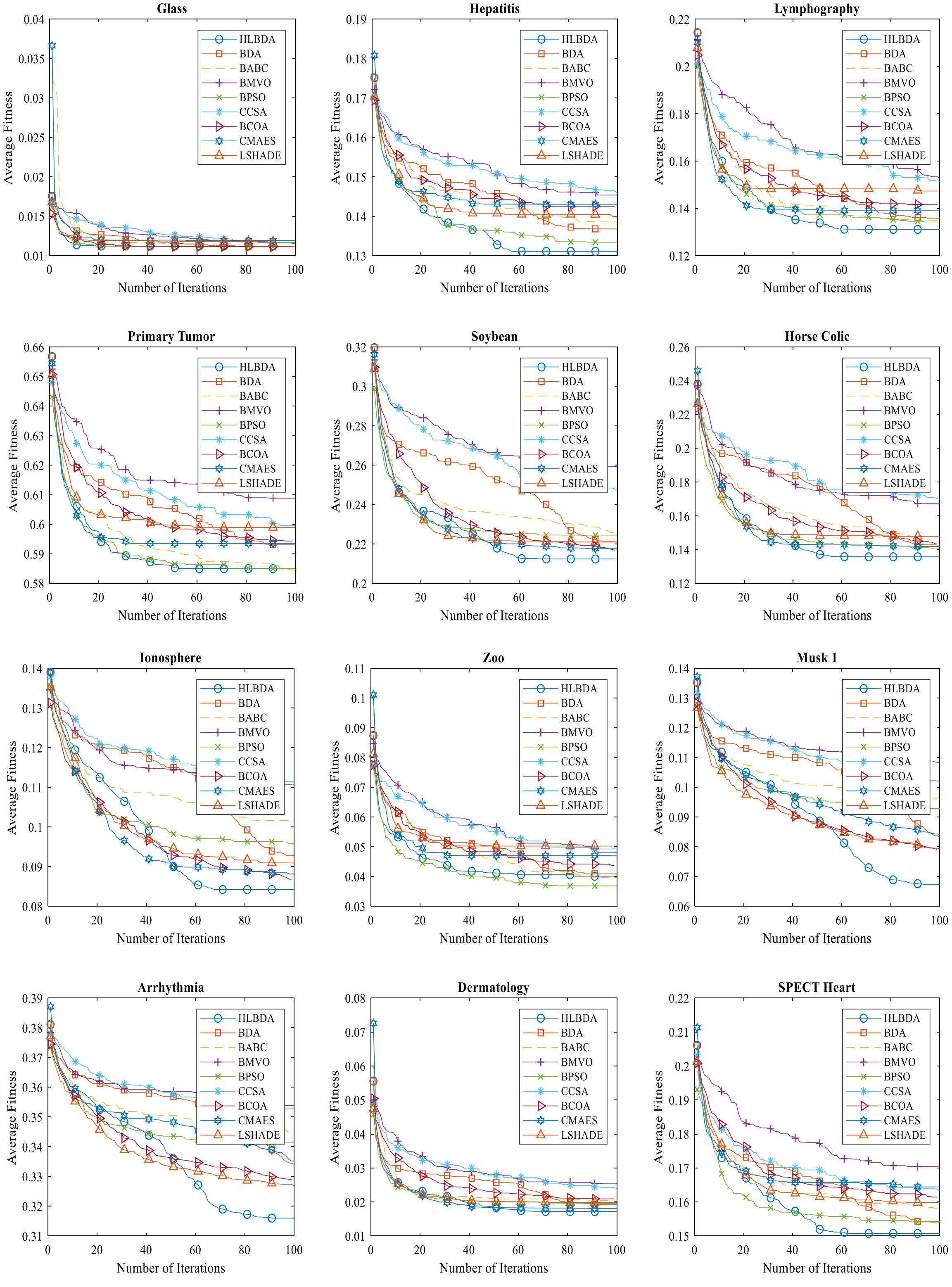


Fig. 6. Convergence curves of algorithms on dataset 1–12.

the best median and mean values in current work. The results obtained affirm the efficacy of the proposed method in retaining the highest classification accuracy.

Table 7 depicts the result of the feature selection ratio. In Table 7, the lower the feature selection ratio is, the smaller the length of the optimal sub-set of feature obtained by algorithms. Inspecting the results, BCOA achieved the smallest feature size in eight datasets, followed by HLBDA (seven datasets). In comparison with BDA, BABC, BMVO, BPSO, CCSA, CMAES, and LSHADE, the HLBDA can usually select a small subset of optimal features that can best describe the target concept. Owing to the hyper learning strategy, HLBDA can be capable to prevent the local optima and can effectively perceive the best solution in feature selection.

Several statistical tests are employed to examine the efficacy of the HLBDA. There are many statistical tests can be found in the

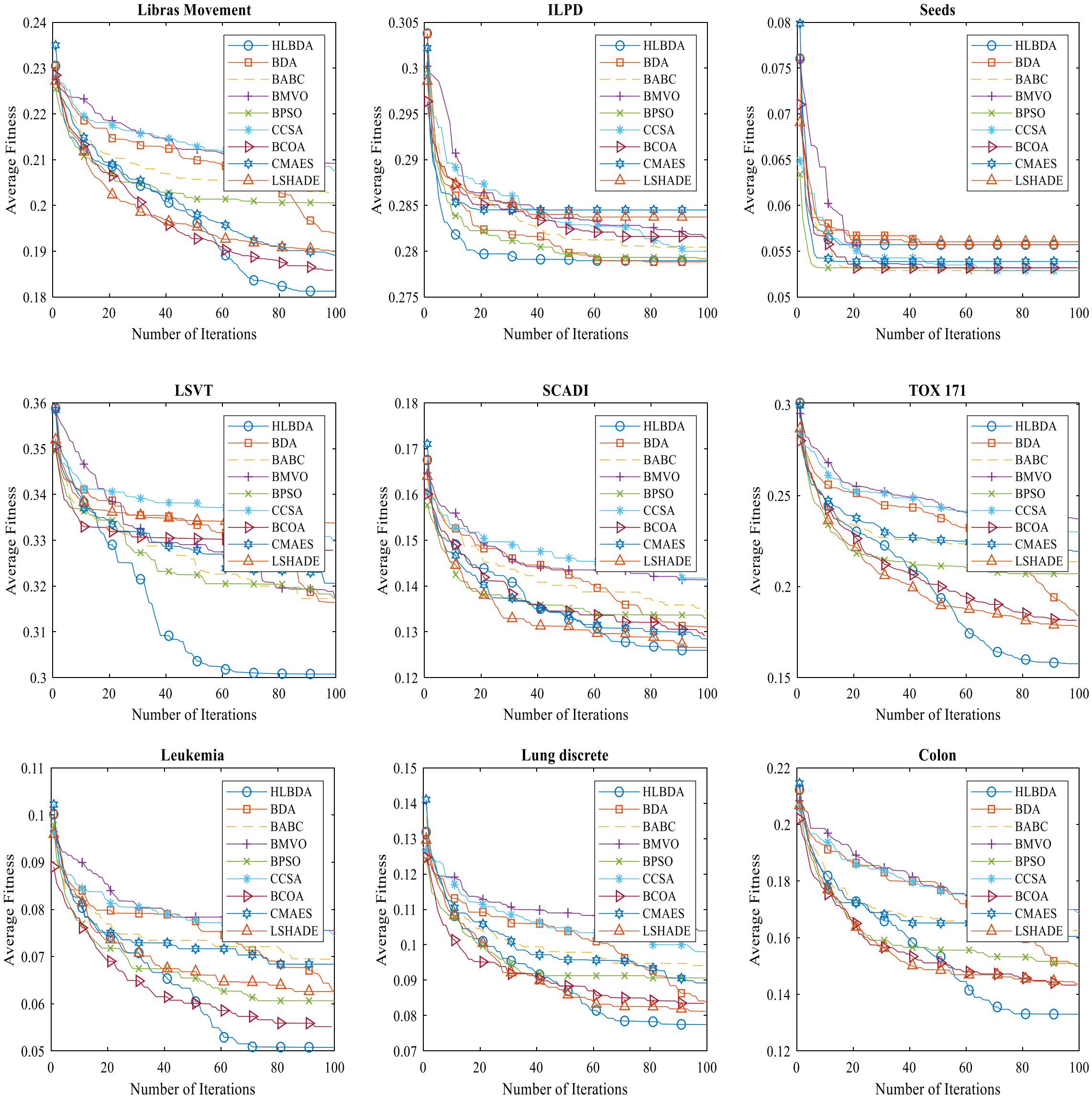


Fig. 7. Convergence curves of algorithms on dataset 13–21.

literature [84–86]. In this study, the statistical analysis is divided into three parts. Firstly, the Friedman test is used to test the accuracy perceived by all the algorithms. From this test, we found that there was a significant difference between all algorithms except Seed dataset (p-value > 0.05). Secondly, the Wilcoxon signed-rank test is applied for pairwise comparisons [85]. In this statistical test, the classification performances of the two methods are found to be similar if the p-value is greater than 0.05; otherwise, the two methods in the comparison are significantly different. Table 8 shows the result of the Wilcoxon test of HLBDA against other methods. Based on the results obtained, the classification performance of the proposed HLBDA was substantially better than other competitors in most cases. Thirdly, the two-way analysis of variance (Anova) and Fisher's least significant difference method is utilized for multiple comparison test. According to findings in Table 9, the HLBDA overtook other methods in many datasets. On the whole, the proposed HLBDA not only offers the highest classification accuracy but also good in dimensionality reduction.

6. Real-world dataset

In 2020, the World Health Organization (WHO) announced the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) started to attack China and rapidly spread over the world. The SARS-CoV-2 is also known as COVID-19, and it has caused the death of more than 600,000 people all around the world since August 2020 [87,88]. Recently, artificial intelligence has become the breakthrough of current technology, and it can be applied for diagnosis, prevention, and detection to fight against COVID-19 [89].

In this section, the proposed HLBDA is employed for COVID-19 patient health prediction. The dataset of COVID-19 patients was collected from the [90] at https://github.com/Atharva-Peshkar/Covid-19-Patient-Health-Analytics. This dataset comprises of fifteen features, as shown in Table 10. This study intends to predict the death and recovery conditions depending on the given factors. The patients' data that contain missing values for both "death" and "recov" status are removed from the main dataset. In this study, we partition the data equally into two sets of training

Table 5
The result of the mean fitness value of algorithms.

No.	Dataset	Mean fitness value								
		HLBDA	BDA	BABC	BMVO	BPSO	CCSA	BCOA	CMAES	LSHADE
1	Glass	0.0112	0.0112	**0.0111**	0.0116	**0.0111**	0.0116	**0.0111**	0.0119	0.0116
2	Hepatitis	**0.1311**	0.1368	0.1386	0.1454	0.1334	0.1457	0.1425	0.1430	0.1399
3	Lymphography	**0.1311**	0.1359	0.1350	0.1527	0.1342	0.1515	0.1416	0.1392	0.1474
4	Primary Tumor	0.5850	0.5932	**0.5845**	0.6088	0.5850	0.5996	0.5944	0.5935	0.5989
5	Soybean	**0.2124**	0.2212	0.2254	0.2594	0.2245	0.2479	0.2168	0.2177	0.2211
6	Horse Colic	**0.1358**	0.1427	0.1480	0.1674	0.1409	0.1699	0.1434	0.1419	0.1479
7	Ionosphere	**0.0842**	0.0928	0.1016	0.1106	0.0958	0.1114	0.0868	0.0882	0.0909
8	Zoo	0.0401	0.0409	0.0397	0.0482	**0.0369**	0.0493	0.0437	0.0470	0.0503
9	Musk 1	**0.0673**	0.0832	0.0961	0.1082	0.0929	0.1017	0.0792	0.0842	0.0793
10	Arrhythmia	**0.3160**	0.3341	0.3450	0.3538	0.3413	0.3529	0.3290	0.3352	0.3270
11	Dermatology	**0.0172**	0.0192	0.0202	0.0254	0.0195	0.0238	0.0209	0.0181	0.0198
12	SPECT Heart	**0.1507**	0.1542	0.1572	0.1704	0.1539	0.1632	0.1614	0.1644	0.1598
13	Libras Movement	**0.1813**	0.1937	0.2027	0.2092	0.2006	0.2073	0.1858	0.1890	0.1899
14	ILPD	0.2790	**0.2788**	0.2804	0.2814	0.2792	0.2800	0.2816	0.2845	0.2837
15	Seeds	0.0557	0.0557	**0.0529**	0.0532	**0.0529**	**0.0529**	0.0532	0.0539	0.0560
16	LSVT	**0.3008**	0.3164	0.3172	0.3167	0.3175	0.3294	0.3278	0.3206	0.3338
17	SCADI	**0.1259**	0.1310	0.1345	0.1413	0.1329	0.1410	0.1291	0.1284	0.1265
18	TOX_171	**0.1575**	0.1834	0.2137	0.2372	0.2070	0.2301	0.1814	0.2194	0.1780
19	Leukemia	**0.0507**	0.0623	0.0694	0.0755	0.0606	0.0743	0.0551	0.0684	0.0626
20	Lung discrete	**0.0774**	0.0835	0.0941	0.1039	0.0906	0.0979	0.0834	0.0892	0.0812
21	Colon	**0.1330**	0.1499	0.1626	0.1699	0.1508	0.1685	0.1432	0.1605	0.1436

Table 6
The result of the standard deviation of fitness value of algorithms.

No.	Dataset	Standard deviation of fitness value								
		HLBDA	BDA	BABC	BMVO	BPSO	CCSA	BCOA	CMAES	LSHADE
1	Glass	0.0033	0.0033	0.0033	0.0033	0.0033	**0.0032**	0.0033	0.0036	0.0033
2	Hepatitis	0.0093	0.0062	**0.0057**	0.0099	0.0080	0.0065	0.0139	0.0132	0.0091
3	Lymphography	0.0130	0.0121	0.0125	**0.0104**	0.0138	0.0117	0.0134	0.0151	0.0147
4	Primary Tumor	0.0104	0.0099	0.0085	**0.0080**	0.0116	0.0120	0.0135	0.0136	0.0134
5	Soybean	0.0087	0.0094	0.0089	0.0118	**0.0041**	0.0108	0.0103	0.0116	0.0109
6	Horse Colic	**0.0039**	0.0088	0.0073	0.0164	0.0069	0.0140	0.0110	0.0097	0.0158
7	Ionosphere	0.0086	0.0105	0.0101	**0.0053**	0.0056	0.0090	0.0104	0.0070	0.0099
8	Zoo	0.0079	0.0080	0.0073	0.0101	**0.0065**	0.0097	0.0092	0.0101	0.0116
9	Musk 1	**0.0062**	0.0099	0.0063	0.0076	0.0079	0.0079	0.0077	0.0075	0.0097
10	Arrhythmia	0.0094	0.0088	**0.0042**	0.0077	0.0073	0.0065	0.0100	0.0057	0.0130
11	Dermatology	0.0020	0.0034	0.0022	0.0025	**0.0019**	0.0038	0.0024	0.0022	0.0035
12	SPECT Heart	**0.0068**	0.0082	0.0081	0.0098	0.0078	0.0098	0.0205	0.0186	0.0170
13	Libras Movement	0.0084	0.0104	0.0086	**0.0050**	0.0074	0.0092	0.0102	0.0090	0.0122
14	ILPD	0.0051	0.0049	0.0049	0.0054	**0.0046**	0.0051	0.0064	0.0082	0.0065
15	Seeds	0.0152	0.0152	**0.0057**	0.0059	0.0057	0.0057	0.0062	0.0066	0.0152
16	LSVT	0.0312	0.0273	**0.0190**	0.0210	0.0243	0.0220	0.0226	0.0202	0.0217
17	SCADI	0.0080	0.0091	**0.0057**	0.0063	0.0062	0.0067	0.0118	0.0074	0.0079
18	TOX_171	0.0213	0.0273	0.0180	0.0156	0.0176	0.0199	0.0182	**0.0140**	0.0215
19	Leukemia	0.0133	0.0152	0.0139	0.0154	0.0115	0.0144	0.0133	**0.0111**	0.0197
20	Lung discrete	0.0096	0.0106	**0.0072**	0.0090	0.0088	0.0086	0.0109	0.0077	0.0104
21	Colon	0.0335	0.0349	**0.0255**	0.0307	0.0273	0.0266	0.0298	0.0335	0.0271

and testing for the validation process. All the features are converted into numeric form. For HLBDA, the parameter *pl* and *gl* are empirically set at 0.7 and 0.85, respectively.

Fig. 10 demonstrates the accuracy and feature size of the proposed HLBDA on the COVID-19 dataset. It is seen that HLBDA achieved the highest classification accuracy of 92.21%. On the other hand, the result shows that roughly three features were enough for HLBDA and BDA in patient health prediction. Based on the results obtained, the most selected features were id, age, and country. Besides, the analysis implies that the symptom4, symptom5, and symptom6 have never been selected by the HLBDA algorithm. The result reveals that those features cannot effectively differentiate the data patterns in the patient health prediction process. Furthermore, we analyse the performance of HLBDA by excluding the first feature (id) from the dataset, and it shows that an accuracy of roughly 90% can be perceived. In future, it is suggested that more clinical features can be obtained for accurate patient health prediction.

7. Discussion

According to the results, it can be inferred that the best feature selection model was HLBDA. In terms of convergence behavior, HLBDA accelerated to detect the global best solution over the complex feature space. Even though BPSO gave a very fast convergence speed in the early iterations, it started to decelerate after some iterations. In contrast, HLBDA was able to maintain a good acceleration over the course of the iterations. The results obtained in Tables 4–6 proved the supremacy of the HLBDA in convergence rate. Moreover, the HLBDA was able to perform more stably in comparison with the conventional BDA method, as smaller standard deviation values supported these findings.

Based on the empirical analysis, it can be observed that the proposed HLBDA can often give better classification results against BDA, BABC, BMVO, BPSO, CCSA, BCOA, CMAES, and LSHADE algorithms. Evidently, the features chosen by HLBDA can usually provide top-quality information, which was good at enhancing the prediction power. In a high dimensional dataset like TOX_171 and Colon, the HLBDA succeeded to select relevant attributes and

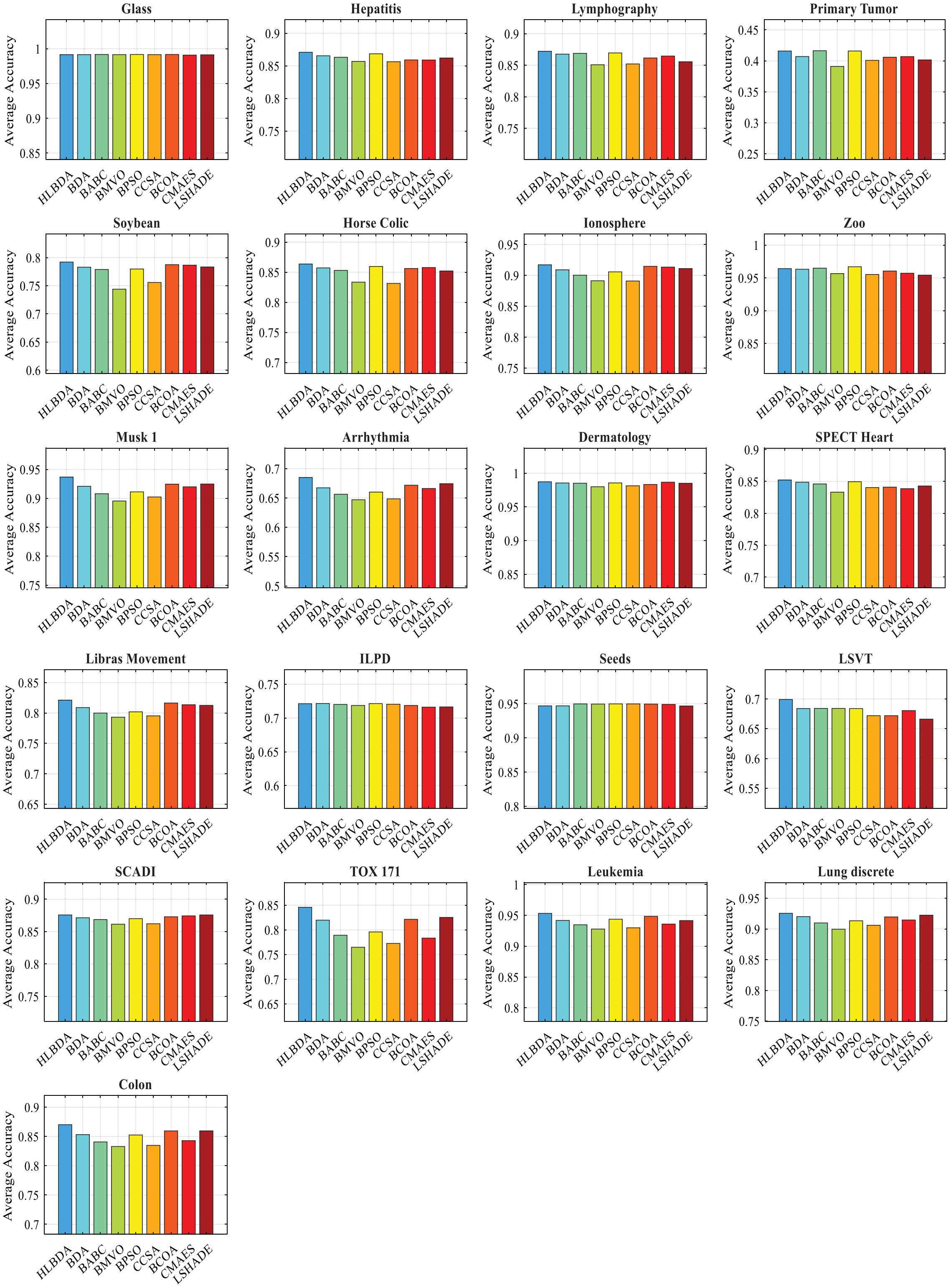


Fig. 8. The result of the accuracy of proposed HLBDA.

omit most of the redundant ones. The summarized simulation results concluded that the proposed HLBDA was more capable of selecting significant features compared to BDA, BABC, BMVO, BPSO, CCSA, BCOA, CMAES, and LSHADE methods. Furthermore, we applied the HLBDA on a COVID-19 application, and the result revealed that the proposed algorithm showing excellent performance in patient health prediction.

The improvement in the proposed HLBDA algorithm can be elaborated in two-sided. Firstly, HLBDA implicated the personal best and personal worst dragonflies in attraction and distraction behaviors. Instead of moving toward the food source, the dragonflies can also learn from their personal best experiences. This, in turn, may assist the HLBDA to escape from the local solution when the food source found by the best dragonfly has trapped in the local optima. Meanwhile, the inclusion of personal worst

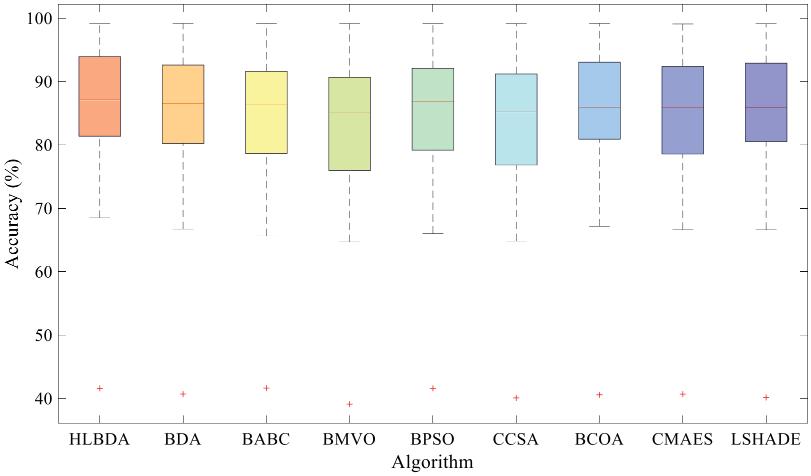


Fig. 9. Boxplots analysis of the HLBDA across twenty-one datasets.

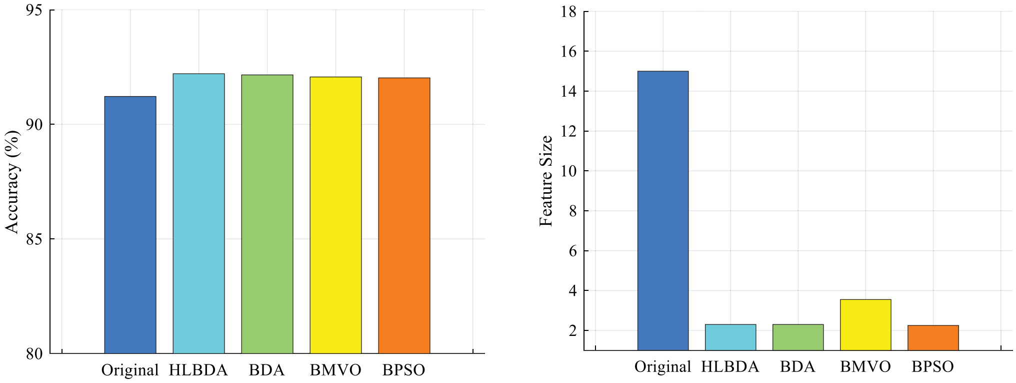


Fig. 10. Accuracy and feature size of the proposed HLBDA on the COVID-19 dataset.

Table 7
The result of the feature selection ratio of proposed HLBDA.

No.	Dataset	Feature selection ratio								
		HLBDA	BDA	BABC	BMVO	BPSO	CCSA	BCOA	CMAES	LSHADE
1	Glass	**0.2900**	**0.2900**	0.3050	0.3250	0.3050	0.3300	0.3050	**0.2900**	0.3050
2	Hepatitis	0.3184	0.3658	0.3158	0.3605	0.3263	0.3526	**0.3053**	0.3500	0.3368
3	Lymphography	0.4500	0.4806	0.5083	0.4889	0.5000	0.4972	0.4472	0.5083	**0.4333**
4	Primary Tumor	0.6676	0.6118	0.6706	**0.5941**	0.6676	0.6441	0.6059	0.6265	0.6412
5	Soybean	0.6529	0.6229	0.6429	**0.5743**	0.6414	0.5971	0.6371	0.6286	0.6486
6	Horse Colic	**0.0870**	0.1370	0.2426	0.2574	0.1963	0.2926	0.1056	0.1130	0.1500
7	Ionosphere	**0.2191**	0.2676	0.2897	0.2882	0.2441	0.3426	0.2265	0.2456	0.2824
8	Zoo	0.4563	0.4469	0.4969	0.5188	**0.4250**	0.4938	0.4531	0.4500	0.4938
9	Musk 1	0.4687	0.4783	0.4946	0.4602	0.4964	0.5033	**0.4458**	0.4946	0.4849
10	Arrhythmia	0.4050	0.4699	0.4803	0.4303	0.4706	0.4787	**0.4048**	0.4627	0.4495
11	Dermatology	0.4574	0.4824	0.5574	0.5426	0.5338	0.5426	**0.4368**	0.4926	0.5074
12	SPECT Heart	0.4477	0.4500	0.4864	0.5273	0.5045	0.5250	**0.4159**	0.4750	0.4273
13	Libras Movement	0.4161	0.4517	0.4600	0.4311	0.4489	0.4644	**0.4061**	0.4300	0.4322
14	ILPD	0.2950	0.3150	0.3350	**0.2800**	0.3250	0.3200	0.3050	0.3450	0.2950
15	Seeds	**0.3143**	**0.3143**	**0.3143**	0.3214	**0.3143**	**0.3143**	0.3214	0.3429	0.3214
16	LSVT	**0.2844**	0.3482	0.4503	0.4006	0.4427	0.4494	0.2984	0.4200	0.3165
17	SCADI	**0.2885**	0.3727	0.4373	0.4154	0.4249	0.4524	0.3180	0.3980	0.3488
18	TOX_171	0.4794	0.4829	0.5002	**0.4451**	0.4975	0.4982	0.4591	0.4981	0.4959
19	Leukemia	0.4517	0.4695	0.4908	0.4180	0.4943	0.4937	**0.4144**	0.4914	0.4773
20	Lung discrete	**0.3714**	0.4391	0.4860	0.4483	0.4808	0.4855	0.3806	0.4703	0.4378
21	Colon	0.4378	0.4629	0.4864	0.4444	0.4910	0.4884	**0.4178**	0.4884	0.4665

experience in distraction enabled the dragonflies to move away from both enemy (the worst solution) and personal worst solution. Secondly, the HLBDA integrated a new updating mechanism for improving the searching behavior. By applying this mechanism, the dragonflies have the opportunity to transport toward the personal best and global best positions. In this way, the HLBDA can usually converge faster to find the global optimum.

8. Conclusion

In this article, a novel Hyper Learning Binary Dragonfly Algorithm (HLBDA) was proposed as a wrapper-based method for

Table 8
The result of the Wilcoxon test of proposed HLBDA against other methods.

No.	Dataset	*p*-value							
		BDA	BABC	BMVO	BPSO	CCSA	BCOA	CMAES	LSHADE
1	Glass	1.00000	1.00000	1.00000	1.00000	0.75000	1.00000	0.15625	0.06250
2	Hepatitis	0.04410	0.00136	0.00096	0.49915	0.00029	0.00264	0.00307	0.00183
3	Lymphography	0.25511	0.29666	0.00015	0.87915	0.00045	0.03828	0.07969	0.00433
4	Primary Tumor	0.00485	0.70475	0.00010	0.64155	0.00063	0.01775	0.00270	0.00111
5	Soybean	0.01058	0.00019	9.00E−05	0.00051	9.00E−05	0.04408	0.09885	0.04737
6	Horse Colic	0.01135	0.00020	9.00E−05	0.01758	9.00E−05	0.00391	0.00110	0.00073
7	Ionosphere	0.01572	0.00022	9.00E−05	0.00193	9.00E−05	0.60109	0.11179	0.02412
8	Zoo	0.94434	0.98730	0.00439	0.20313	0.00049	0.11084	0.06042	0.01306
9	Musk 1	0.00045	9.00E−05	9.00E−05	9.00E−05	9.00E−05	0.00034	9.00E−05	0.00074
10	Arrhythmia	0.00012	9.00E−05	9.00E−05	9.00E−05	9.00E−05	0.00151	9.00E−05	0.01237
11	Dermatology	0.09298	0.02890	9.00E−05	0.04753	0.00046	0.00094	0.55493	0.06541
12	SPECT Heart	0.05508	0.00539	9.00E−05	0.10396	0.00025	0.05514	0.00212	0.02629
13	Libras Movement	0.00045	9.00E−05	9.00E−05	9.00E−05	9.00E−05	0.03746	0.00720	0.00444
14	ILPD	0.57813	0.13086	0.00269	1.00000	0.26172	0.01221	0.00061	0.00098
15	Seeds	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
16	LSVT	0.23511	0.17407	0.15496	0.16992	0.02287	0.07849	0.12256	0.00488
17	SCADI	0.14844	0.00195	0.00053	0.07227	0.00061	0.38770	0.78125	1.00000
18	TOX_171	0.00234	9.00E−05	9.00E−05	9.00E−05	9.00E−05	0.00068	9.00E−05	0.00381
19	Leukemia	0.00598	6.00E−05	0.00013	0.00195	0.00012	0.12280	0.00028	0.01233
20	Lung discrete	0.08325	0.00028	9.00E−05	0.00086	0.00040	0.01233	0.00342	0.33252
21	Colon	0.00220	0.00012	9.00E−05	0.00218	0.00011	0.03633	8.00E−05	0.08838

Table 9
The result of the multiple comparison test (HLBDA as control method).

No.	Dataset	*p*-value							
		BDA	BABC	BMVO	BPSO	CCSA	BCOA	CMAES	LSHADE
1	Glass	1.00000	0.48596	0.97219	0.46446	0.94441	0.46446	0.02253	0.38401
2	Hepatitis	0.06257	0.00798	0.00000	0.42630	0.00000	6.00E−05	5.00E−05	0.00246
3	Lymphography	0.27598	0.42102	0.00000	0.52110	0.00000	0.01053	0.06492	8.00E−05
4	Primary Tumor	0.01307	0.88485	0.00000	0.99598	4.00E−05	0.00504	0.01187	8.00E−05
5	Soybean	0.00422	4.00E−05	0.00000	0.00013	0.00000	0.14536	0.07959	0.00597
6	Horse Colic	0.05424	0.00163	0.00000	0.22536	0.00000	0.02665	0.08189	0.00071
7	Ionosphere	0.00202	0.00000	0.00000	2.00E−05	0.00000	0.32676	0.15352	0.01924
8	Zoo	0.75445	0.77832	0.01067	0.31011	0.00266	0.21199	0.01702	0.00086
9	Musk 1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
10	Arrhythmia	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00011
11	Dermatology	4.69E−02	0.02101	0.00000	0.06808	0.00000	1.00E−05	0.52899	0.01484
12	SPECT Heart	0.32751	0.08376	0.00000	0.46028	0.00104	0.00196	0.00018	0.00869
13	Libras Movement	0.00000	0.00000	0.00000	0.00000	0.00000	0.02520	0.00035	6.00E−05
14	ILPD	0.76800	0.34944	0.02306	0.94246	0.50005	0.02723	2.00E−05	5.00E−05
15	Seeds	1.00000	0.15742	0.19475	0.15742	0.15742	0.19475	0.28829	0.90591
16	LSVT	0.03216	0.03438	0.03400	0.03006	0.00015	0.00016	0.00850	1.00E−05
17	SCADI	0.07858	0.00366	0.00000	0.01949	0.00000	0.23961	0.55587	1.00000
18	TOX_171	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	5.00E−05
19	Leukemia	9.00E−05	0.00000	0.00000	0.00110	0.00000	0.09518	0.00000	7.00E−05
20	Lung discrete	0.03437	0.00000	0.00000	0.00000	0.00000	0.02032	3.00E−05	0.22248
21	Colon	6.00E−05	0.00000	0.00000	4.00E−05	0.00000	0.01087	0.00000	0.01178

feature selection purpose as an essential pre-processing phase for classification algorithms. A hyper learning strategy was employed to substantially improve the performance of the algorithm in feature selection. Twenty-one datasets collected from the UCI repository and Arizona State University were used to validate the performance of the proposed method in this work. Besides, the proposed model was applied to a COVID-19 application. Our findings show that the proposed HLBDA outperformed the other algorithms in most of the datasets. Among rivals, the HLBDA was able to identify a subset of high discriminative features that were good in describing the target concepts. Therefore, the HLBDA can often achieve the highest accuracy when applied to feature selection works.

For future studies, HLBDA can be applied to various engineering and clinical applications such as COVID-19 diagnosis, electromyography pattern recognition, optimized deep neural network, and power quality diagnosis. The readers are also encouraged to investigate the best combination of pl and gl for

Table 10
The description of COVID-19 dataset.

No.	Features	Description
1	id	The ID of patients
2	location	The location where patients belong to
3	country	The country where patients belong to
4	gender	The gender of patients
5	age	The ages of patients
6	sym_on	The date patients have symptoms
7	hosp_vis	The date patients visit hospital
8	vis_wuhan	Whether the patients visited Wuhan, China
9	from_wuhan	Whether the patients from Wuhan, China
10	symptom1	Symptoms of patients
11	symptom2	Symptoms of patients
12	symptom3	Symptoms of patients
13	symptom4	Symptoms of patients
14	symptom5	Symptoms of patients
15	symptom6	Symptoms of patients

attaining the optimal performance. Furthermore, different enhanced initialization strategies can be implemented into HLBDA for performance enhancement.

CRediT authorship contribution statement

Jingwei Too: Conceptualization, Methodology, Software, Formal analysis, Validation, Visualization, Writing - original draft, Writing - review & editing. **Seyedali Mirjalili:** Visualization, Writing - original draft, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Han, G. Zhou, Y. Zhou, Binary symbiotic organism search algorithm for feature selection and analysis, IEEE Access 7 (2019) 166833–166859, http://dx.doi.org/10.1109/ACCESS.2019.2953800.

[2] J.C. Quiroz, A. Banerjee, S.M. Dascalu, S.L. Lau, Feature selection for activity recognition from smartphone accelerometer data, Intell. Autom. Soft Comput. (2017) 1–9, http://dx.doi.org/10.1080/10798587.2017.1342400.

[3] I. Aljarah, M. Habib, H. Faris, et al., A dynamic locality multi-objective salp swarm algorithm for feature selection, Comput. Ind. Eng. 147 (2020) 106628, http://dx.doi.org/10.1016/j.cie.2020.106628.

[4] E. Hancer, B. Xue, D. Karaboga, M. Zhang, A binary ABC algorithm based on advanced similarity scheme for feature selection, Appl. Soft Comput. 36 (2015) 334–348, http://dx.doi.org/10.1016/j.asoc.2015.07.023.

[5] S. Arora, P. Anand, Binary butterfly optimization approaches for feature selection, Expert Syst. Appl. 116 (2019) 147–160, http://dx.doi.org/10.1016/j.eswa.2018.08.051.

[6] C. Zhong, Y. Chen, J. Peng, Feature selection based on a novel improved tree growth algorithm, Int. J. Comput. Intell. Syst. 13 (2020) 247–258, http://dx.doi.org/10.2991/ijcis.d.200219.001.

[7] Q. Al-Tashi, S.J.A. Kadir, H.M. Rais, et al., Binary optimization using hybrid grey wolf optimization for feature selection, IEEE Access 7 (2019) 39496–39508, http://dx.doi.org/10.1109/ACCESS.2019.2906757.

[8] J. Kennedy, Particle swarm optimization, in: Encyclopedia of Machine Learning, Springer, Boston, MA, 2011, pp. 760–766.

[9] J.H. Holl, Genetic algorithms, Sci. Am. 267 (1992) 66–73.

[10] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166, http://dx.doi.org/10.1016/j.compstruc.2012.07.010.

[11] X. Li, J. Zhang, M. Yin, Animal migration optimization: an optimization algorithm inspired by animal migration behavior, Neural Comput. Appl. 24 (2014) 1867–1877, http://dx.doi.org/10.1007/s00521-013-1433-8.

[12] M. Ghaemi, M.-R. Feizi-Derakhshi, Forest optimization algorithm, Expert Syst. Appl. 41 (2014) 6676–6687, http://dx.doi.org/10.1016/j.eswa.2014.05.009.

[13] L. Wang, R. Yang, H. Ni, et al., A human learning optimization algorithm and its application to multi-dimensional knapsack problems, Appl. Soft Comput. 34 (2015) 736–743, http://dx.doi.org/10.1016/j.asoc.2015.06.004.

[14] V.K. Patel, V.J. Savsani, Heat transfer search (HTS): a novel optimization algorithm, Inf. Sci. 324 (2015) 217–246, http://dx.doi.org/10.1016/j.ins.2015.06.044.

[15] H. Shareef, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. 36 (2015) 315–333, http://dx.doi.org/10.1016/j.asoc.2015.07.028.

[16] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput. Appl. (2015) 1–20, http://dx.doi.org/10.1007/s00521-015-1923.

[17] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073, http://dx.doi.org/10.1007/s00521-015-1920-1.

[18] I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, Soft Comput. 22 (2018) 1–15, http://dx.doi.org/10.1007/s00500-016-2442-1.

[19] S.R. KS, S. Murugan, Memory based hybrid dragonfly algorithm for numerical optimization problems, Expert Syst. Appl. 83 (2017) 63–78, http://dx.doi.org/10.1016/j.eswa.2017.04.033.

[20] M. Abdel-Basset, L. Abdle-Fatah, A.K. Sangaiah, An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment, Clust. Comput. (2018) 1–16, http://dx.doi.org/10.1007/s10586-018-1769-z.

[21] H. Hu, Y. Ao, Y. Bai, et al., An improved Harris's hawks optimization for SAR target recognition and stock market index prediction, IEEE Access 8 (2020) 65891–65910, http://dx.doi.org/10.1109/ACCESS.2020.2985596.

[22] Dalia Yousri, Hegazy Rezk, Ahmed Fathy, Identifying the parameters of different configurations of photovoltaic models based on recent artificial ecosystem-based optimization approach, Int. J. Energy Res. (2020) 11302–11322.

[23] X. Zhang, Y. Xu, C. Yu, et al., Gaussian mutational chaotic fruit fly-built optimization and feature selection, Expert Syst. Appl. 141 (2020) 112976, http://dx.doi.org/10.1016/j.eswa.2019.112976.

[24] D. Gupta, S. Sundaram, A. Khanna, et al., Improved diagnosis of Parkinson's disease using optimized crow search algorithm, Comput. Electr. Eng. 68 (2018) 412–424, http://dx.doi.org/10.1016/j.compeleceng.2018.04.014.

[25] G.I. Sayed, A. Tharwat, A.E. Hassanien, Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection, Appl. Intell. 49 (2019) 188–205, http://dx.doi.org/10.1007/s10489-018-1261-8.

[26] J. Wang, Y. Yang, T. Wang, et al., Big data service architecture: A survey, J. Internet Technol. 21 (2020) 393–405.

[27] S. AbdEl-Fattah Sayed, E. Nabil, A. Badr, A binary clonal flower pollination algorithm for feature selection, Pattern Recognit. Lett. 77 (2016) 21–27, http://dx.doi.org/10.1016/j.patrec.2016.03.014.

[28] S. Ahmed, K.K. Ghosh, P.K. Singh, et al., Hybrid of harmony search algorithm and ring theory-based evolutionary algorithm for feature selection, IEEE Access 8 (2020) 102629–102645, http://dx.doi.org/10.1109/ACCESS.2020.2999093.

[29] H. Wang, X. Jing, B. Niu, A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data, Knowl.-Based Syst. 126 (2017) 8–19, http://dx.doi.org/10.1016/j.knosys.2017.04.004.

[30] H. Alazzam, A. Sharieh, K.E. Sabri, A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer, Expert Syst. Appl. 148 (2020) 113249, http://dx.doi.org/10.1016/j.eswa.2020.113249.

[31] C. De Stefano, F. Fontanella, C. Marrocco, A. Scotto di Freca, A GA-based feature selection approach with an application to handwritten character recognition, Pattern Recognit. Lett. 35 (2014) 130–141, http://dx.doi.org/10.1016/j.patrec.2013.01.026.

[32] B. Ma, Y. Xia, A tribe competition-based genetic algorithm for feature selection in pattern classification, Appl. Soft Comput. 58 (2017) 328–338, http://dx.doi.org/10.1016/j.asoc.2017.04.042.

[33] X. Lian, L. Zhang, J. Jiang, W. Goss, An approach for optimized feature selection in large-scale software product lines, J. Syst. Softw. 137 (2018) 636–651, http://dx.doi.org/10.1016/j.jss.2017.02.044.

[34] Y. Xue, NanWang, Y. Niu, et al., Robust re-weighted multi-view feature selection, Comput. Mater. Contin. 60 (2019) 741–756, http://dx.doi.org/10.32604/cmc.2019.05611.

[35] A. Erdal Tümer, A. Akkuş, Application of radial basis function networks with feature selection for GDP per capita estimation based on academic parameters, Comput. Syst. Sci. Eng. 34 (2019) https://www.techscience.com/csse/v34n3/40035.

[36] U. Kanimozhi, D. Manjula, An intelligent incremental filtering feature selection and clustering algorithm for effective classification, Intell. Autom. Soft Comput. (2017) 1–9, http://dx.doi.org/10.1080/10798587.2017.1307626.

[37] S. Fang, Z. Cai, W. Sun, et al., Feature selection method based on class discriminative degree for intelligent medical diagnosis, CMC-Comput. Mater. Contin. 55 (2018) http://dx.doi.org/10.3970/cmc.2018.02289.

[38] Y. Xue, T. Tang, W. Pang, A.X. Liu, Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers, Appl. Soft Comput. 88 (2020) 106031, http://dx.doi.org/10.1016/j.asoc.2019.106031.

[39] B. Ji, X. Lu, G. Sun, et al., Bio-inspired feature selection: An improved binary particle swarm optimization approach, IEEE Access 8 (2020) 85989–86002, http://dx.doi.org/10.1109/ACCESS.2020.2992752.

[40] C. Qiu, A novel multi-swarm particle swarm optimization for feature selection, Genet. Program. Evolvable Mach. (2019) http://dx.doi.org/10.1007/s10710-019-09358-0.

[41] C.-L. Huang, C.-J. Wang, A GA-based feature selection and parameters optimizationfor support vector machines, Expert Syst. Appl. 31 (2006) 231–240, http://dx.doi.org/10.1016/j.eswa.2005.09.024.

[42] D. Jude Hemanth, J. Anitha, Modified genetic algorithm approaches for classification of abnormal magnetic resonance brain tumour images, Appl. Soft Comput. 75 (2019) 21–28, http://dx.doi.org/10.1016/j.asoc.2018.10.054.

[43] P. Krömer, J. Platoš, J. Nowaková, V. Snášel, Optimal column subset selection for image classification by genetic algorithms, Ann. Oper. Res. 265 (2018) 205–222, http://dx.doi.org/10.1007/s10479-016-2331-0.

[44] J. Too, A.R. Abdullah, A new and fast rival genetic algorithm for feature selection, J. Supercomput. (2020) 1–31, http://dx.doi.org/10.1007/s11227-020-03378-9.

[45] Y. Wan, M. Wang, Z. Ye, X. Lai, A feature selection method based on modified binary coded ant colony optimization algorithm, Appl. Soft Comput. 49 (2016) 248–258, http://dx.doi.org/10.1016/j.asoc.2016.08.011.

[46] Y. Zhang, R. Liu, X. Wang, et al., Boosted binary Harris hawks optimizer and feature selection, Eng. Comput. (2020) http://dx.doi.org/10.1007/s00366-020-01028-5.

[47] J. Too, A.R. Abdullah, Chaotic atom search optimization for feature selection, Arab. J. Sci. Eng. (2020) http://dx.doi.org/10.1007/s13369-020-04486-7.

[48] K. Kanti Ghosh, R. Guha, S. Ghosh, et al., Atom search optimization with simulated annealing – a hybrid metaheuristic approach for feature selection, 2020, ArXiv E-Prints 2005:arXiv:2005.08642.

[49] J. Too, A.R. Abdullah, Binary atom search optimisation approaches for feature selection, Connect. Sci. 0 (2020) 1–25, http://dx.doi.org/10.1080/09540091.2020.1741515.

[50] P. Baynath, S. Soyjaudah, M. Khan, Feature selection and representation of evolutionary algorithm on keystroke dynamics, Intell. Autom. Soft Comput. (2018) 1, http://dx.doi.org/10.31209/2018.100000060.

[51] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

[52] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381, http://dx.doi.org/10.1016/j.neucom.2015.06.083.

[53] H.M. Zawbaa, E. Emary, C. Grosan, V. Snasel, Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach, Swarm Evol. Comput. 42 (2018) 29–42, http://dx.doi.org/10.1016/j.swevo.2018.02.021.

[54] H. Chantar, M. Mafarja, H. Alsawalqah, et al., Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification, Neural Comput Appl. (2019) http://dx.doi.org/10.1007/s00521-019-04368-6.

[55] J. Too, A.R. Abdullah, Opposition based competitive grey wolf optimizer for EMG feature selection, Evol. Intell. (2020) http://dx.doi.org/10.1007/s12065-020-00441-5.

[56] M. Abdel-Basset, D. El-Shahat, I. El-henawy, et al., A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection, Expert Syst. Appl. 139 (2020) 112824, http://dx.doi.org/10.1016/j.eswa.2019.112824.

[57] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453, http://dx.doi.org/10.1016/j.asoc.2017.11.006.

[58] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312, http://dx.doi.org/10.1016/j.neucom.2017.04.053.

[59] M. Tubishat, M.A.M. Abushariah, N. Idris, I. Aljarah, Improved whale optimization algorithm for feature selection in Arabic sentiment analysis, Appl. Intell. 49 (2019) 1688–1707, http://dx.doi.org/10.1007/s10489-018-1334-8.

[60] Mohammad Tubishat, Norisma Idris, Explicit and implicit aspect extraction using whale optimization algorithm and hybrid approach, Int. Conf. Ind. Enterp. Syst. Eng. (2019).

[61] W.M. Shaban, A.H. Rabie, A.I. Saleh, M.A. Abo-Elsoud, A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier, Knowl.-Based Syst. 205 (2020) 106270, http://dx.doi.org/10.1016/j.knosys.2020.106270.

[62] G. Jain, D. Mittal, D. Thakur, Mittal M.K., A deep learning approach to detect Covid-19 coronavirus with X-ray images, Biocybern. Biomed. Eng. 40 (2020) 1391–1405, http://dx.doi.org/10.1016/j.bbe.2020.08.008.

[63] T. Tuncer, S. Dogan, F. Ozyurt, An automated residual exemplar local binary pattern and iterative relieff based COVID-19 detection method using chest X-ray image, Chemom. Intell. Lab. Syst. 203 (2020) 104054, http://dx.doi.org/10.1016/j.chemolab.2020.104054.

[64] M.H.D.M. Ribeiro, R.G. da Silva, V.C. Mariani, L. dos S. Coelho, Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil, Chaos Solitons Fractals 135 (2020) 109853, http://dx.doi.org/10.1016/j.chaos.2020.109853.

[65] R.G. da Silva, M.H.D.M. Ribeiro, V.C. Mariani, L. dos S. Coelho, Forecasting Brazilian and American COVID-19 cases based on artificial intelligence coupled with climatic exogenous variables, Chaos Solitons Fractals 139 (2020) 110027, http://dx.doi.org/10.1016/j.chaos.2020.110027.

[66] M.M. Mafarja, D. Eleyan, I. Jaber, et al., Binary dragonfly algorithm for feature selectionn, in: 2017 International Conference on New Trends in Computing Sciences, ICTCS, 2017, pp. 12–17.

[67] M. Hariharan, R. Sindhu, V. Vijean, et al., Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification, Comput, Methods Programs Biomed, 155 (2018) 39–51, http://dx.doi.org/10.1016/j.cmpb.2017.11.021.

[68] N. Neggaz, A.A. Ewees, M.A. Elaziz, M. Mafarja, Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection, Expert Syst. Appl. 145 (2020) 113103, http://dx.doi.org/10.1016/j.eswa.2019.113103.

[69] A.I. Hammouri, M. Mafarja, M.A. Al-Betar, et al., An improved dragonfly algorithm for feature selection, Knowl.-Based Syst. 203 (2020) 106131, http://dx.doi.org/10.1016/j.knosys.2020.106131.

[70] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing 213 (2016) 54–65, http://dx.doi.org/10.1016/j.neucom.2016.03.101.

[71] M. Mafarja, I. Aljarah, H. Faris, et al., Binary grasshopper optimisation algorithm approaches for feature selection problems, Expert Syst. Appl. 117 (2019) 267–286, http://dx.doi.org/10.1016/j.eswa.2018.09.015.

[72] N. Al-Madi, H. Faris, S. Mirjalili, Binary multi-verse optimization algorithm for global optimization and discrete problems, Int. J. Mach. Learn. Cybern. (2019) http://dx.doi.org/10.1007/s13042-019-00931-8.

[73] E. Emary, H.M. Zawbaa, Feature selection via Lèvy Antlion optimization, PAA Pattern Anal. Appl. (2018) 1–20, http://dx.doi.org/10.1007/s10044-018-0695-2.

[74] UCI Machine Learning Repository, 2019, https://archive.ics.uci.edu/ml/index.php, (Accessed 24 March 2019).

[75] Datasets | Feature Selection @ ASU, 2019, http://featureselection.asu.edu/datasets.php, (Accessed 3 October 2019).

[76] N. Neggaz, E.H. Houssein, K. Hussain, An efficient henry gas solubility optimization for feature selection, Expert Syst. Appl. 152 (2020) 113364, http://dx.doi.org/10.1016/j.eswa.2020.113364.

[77] Y. He, H. Xie, T.-L. Wong, X. Wang, A novel binary artificial bee colony algorithm for the set-union knapsack problem, Future Gener. Comput. Syst. 78 (2018) 77–86, http://dx.doi.org/10.1016/j.future.2017.05.044.

[78] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Computational Cybernetics and Simulation 1997 IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, 1997, pp. 4104–4108.

[79] G.I. Sayed, A.E. Hassanien, A.T. Azar, Feature selection via a novel chaotic crow search algorithm, Neural Comput. App.l (2017) 1–18, http://dx.doi.org/10.1007/s00521-017-2988-6.

[80] J. Pierezan, L. Dos Santos Coelho, Coyote optimization algorithm: A new metaheuristic for global optimization problems, in: 2018 IEEE Congress on Evolutionary Computation, CEC, 2018, pp. 1–8.

[81] R.C. Thom de Souza, C.A. de Macedo, L. dos Santos Coelho, et al., Binary coyote optimization algorithm for feature selection, Pattern Recognit, 107 (2020) 107470, http://dx.doi.org/10.1016/j.patcog.2020.107470.

[82] N. Hansen, S. Kern, Evaluating the CMA evolution strategy on multimodal test functions, in: Parallel Problem Solving from Nature - PPSN VIII, Springer, Berlin, Heidelberg, 2004, pp. 282–291.

[83] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation, CEC, 2014, pp. 1658–1665.

[84] J. Derrac, S. García, S. Hui, et al., Analyzing convergence performance of evolutionary algorithms: A statistical approach, Inf. Sci. 289 (2014) 41–58, http://dx.doi.org/10.1016/j.ins.2014.06.009.

[85] J. Carrasco, S. García, M.M. Rueda, et al., Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm Evol. Comput. 54 (2020) 100665, http://dx.doi.org/10.1016/j.swevo.2020.100665.

[86] J. Derrac, S. García, D. Molina, Herrera F., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.

[87] X. Chen, Y. Tang, Y. Mo, et al., A diagnostic model for coronavirus disease 2019 (COVID-19) based on radiological semantic and clinical features: a multi-center study, Eur. Radiol. (2020) http://dx.doi.org/10.1007/s00330-020-06829-2.

[88] Coronavirus Update (Live): 18 447 723 Cases and 697, 244 Deaths from COVID-19 Virus Pandemic - Worldometer, 2020, https://www.worldometers.info/coronavirus/, (Accessed 4 August 2020).

[89] Ahmed T. Sahlol, Dalia Yousri, Ahmed A. Ewees, Mohammed A.A. Al-Qaness, Robertas Damasevicius, Mohamed Abd Elaziz, COVID-19 image classification using deep features and fractional-order marine predators algorithm, Sci. Rep. (2020).

[90] C. Iwendi, A.K. Bashir, A. Peshkar, et al., COVID-19 patient health prediction using boosted random forest algorithm, Front. Publ. Health 8 (2020) http://dx.doi.org/10.3389/fpubh.2020.00357.