Knowledge-Based Systems 248 (2022) 108789

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

BCOVIDOA: A Novel Binary Coronavirus Disease Optimization Algorithm for Feature Selection

Asmaa M. Khalid [a], Hanaa M. Hamza [a], Seyedali Mirjalili [b,c], Khalid M. Hosny [a,*]

[a] Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt
[b] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane 4006, QLD, Australia
[c] Yonsei Frontier Lab, Yonsei University, Seoul, South Korea

ARTICLE INFO

Article history:
Received 21 July 2021
Received in revised form 8 April 2022
Accepted 8 April 2022
Available online 18 April 2022

Keywords:
Coronavirus
Optimization
Frameshifting
Best cost
Convergence
Evolutionary algorithm
Feature selection
Meta-heuristic
Big data

ABSTRACT

The increased use of digital tools such as smart phones, Internet of Things devices, cameras, and microphones, has led to the produuction of big data. Large data dimensionality, redundancy, and irrelevance are inherent challenging problems when it comes to big data. Feature selection is a necessary process to select the optimal subset of features when addressing such problems. In this paper, the authors propose a novel Binary Coronavirus Disease Optimization Algorithm (BCOVIDOA) for feature selection, where the Coronavirus Disease Optimization Algorithm (COVIDOA) is a new optimization technique that mimics the replication mechanism used by Coronavirus when hijacking human cells. The performance of the proposed algorithm is evaluated using twenty-six standard benchmark datasets from UCI Repository. The results are compared with nine recent wrapper feature selection algorithms. The experimental results demonstrate that the proposed BCOVIDOA significantly outperforms the existing algorithms in terms of accuracy, best cost, the average cost (AVG), standard deviation (STD), and size of selected features. Additionally, the Wilcoxon rank-sum test is calculated to prove the statistical significance of the results.

1. Introduction

With the rapid use of computer and internet technologies, immense quantities of data with hundreds of features are produced. In data mining, useful information must be extracted from such big data to decide. Selecting only the relevant and useful features would have a significant effect in many applications such as text mining [1], image processing [2], Bioinformatics [3], and industrial applications [4]. Internet of things (IoT) is a modern and powerful technology in which physical objects embedded with sensors are connected through a network to exchange data [5]. Challenges to IoT applications include storing and processing such a vast amount of data gathered by IoT sensors [6]. Another challenge is the existence of redundant, irrelevant, and noisy features. A solution to these challenges is to use feature selection to select the optimum subset of features.

Feature selection is a preprocessing, mining, and machine learning problem since it removes redundant and irrelevant variables in a dataset [7]. Feature selection aims to reduce data dimensionality, reduce training time, and increase generalization. A feature selection model consists of three main factors; classification (e.g., Support Vector Machine (SVM), K-Nearest Neighbors (KNN), etc.), evaluation criteria (such as classification accuracy), and search algorithm [8]. Classification assigns each subset to a specific class, evaluation criteria should be selected to evaluate each subset, and the searching algorithm is used to select the optimum subset of features.

The main categories of feature selection methods are wrappers and filters. The wrapper methods involve classifiers and detect the interactions between variables. Filter methods evaluate feature subsets based on the data regardless of the model [9]. Filter methods are much faster than wrapper methods as they do not involve classifiers but may fail to find the best subset of features. However, wrapper methods usually provide the best performing feature subset for a predetermined classifier [10]. For filtering algorithms, Xu et al. [11] proposed an SVMs Classification based two-side cross-domain collaborative filtering algorithm by inferring intrinsic user and item. The major innovation of the proposed model is that domain-independent intrinsic features of users and items can be inferred from domain-dependent rating matrices. The results have shown that the proposed model significantly outperforms all the state-of-the-art algorithms at various sparsity levels.

Additionally, a Cross-Domain Collaborative Filtering (CDCF) Algorithm is proposed in [12]. In the proposed algorithm, knowledge can be transferred from user- and item-side auxiliary domains to the target domain by expanding the original feature vector. The experimental results have shown that the proposed algorithm outperforms the state-of-the-art baseline algorithms.

[*] Corresponding author.
E-mail addresses: asmaa.elhenawy@gmail.com (A.M. Khalid), hanaa_hamza2000@yahoo.com (H.M. Hamza), ali.mirjalili@gmail.com (S. Mirjalili), k_hosny@zu.edu.eg (K.M. Hosny).

https://doi.org/10.1016/j.knosys.2022.108789
0950-7051/© 2022 Elsevier B.V. All rights reserved.

According to a specific evaluation metric, a feature selection method searches for the best feature subset from all possible subsets. Searching algorithms can be classified into exact search methods and metaheuristics [13]. The exact methods search the entire search space. For example, for the feature set with k features, the size of the search space is proportional to $2^{k,}$ which requires too much computational time [14]. On the other hand, metaheuristic search strategies can be used to find a (near) optimum subset from the original set by using the local search or imitating a natural process.

In the feature selection problem, a dataset can be represented by a two-dimensional matrix X with N rows representing instances where $x = 1, 2, \ldots, N$, and M columns representing features in each instance where $y = 1, 2, \ldots, M$. X_{ij} is the value of the yth feature in and xth instance. Fig. 1 shows a matrix representing a dataset. Each class has similar values, whereas two different classes have elements with different values.

Several metaheuristic algorithms have been proposed in the literature to solve feature selection problems. These algorithms include binary Cuckoo Search (BCS) [15], Binary Flower Pollination Algorithm (BFPA) [16], Binary Dragonfly algorithm (BDA) [17], Simulated Annealing (SA) [18], Particle Swarm Optimization (PSO) [19], Genetic Algorithm (GA) [20], Differential Evolution (DE) [21,22], Artificial Bee Colony (ABC) [23], Ant Colony Optimization (ACO) [24], Grey Wolf Optimization (GWO) [14], Whale Optimization Algorithm (WOA) [25], and Bat Algorithm (BA) [26]. In addition to these algorithms, where two or more algorithms are combined to solve feature selection problems. In [27], three types of hybridization of GA and SA are proposed to solve some non-linear optimization cases. The algorithm is tested using five benchmark functions, and the obtained results showed that hybrid GA-SA could enhance the performance of GA and SA to provide better results. Mafarja and Mirjalili [28] proposed two hybridization modes based on WOA. The first model embeds the SA algorithm to WOA, while SA improves the best solution found after each iteration in the second model. The performance is evaluated on 18 UCI datasets. The results confirm the efficiency of the proposed approaches in enhancing classification accuracy compared to other feature selection algorithms. In addition, a binary version of hybrid GWO and PSO algorithms to solve feature selection problems is proposed in [29]. 18 UCI benchmark datasets are employed. The results showed that the proposed algorithm outperformed the state-of-the-art binary algorithms using performance measures such as accuracy and computational time.

Additionally, recent optimization algorithms are introduced and applied to feature selection, such as the meta-heuristic quantum-inspired immune clone optimization algorithm (QICO) [30] used for optimal feature selection from gene expression data to develop the cancer data classification. Also, a clustering-based hybrid approach [31] is introduced for gene subset selection of microarray gene expression data. The experimental outcomes denote that the proposed model achieves efficient results in selecting the best gene subsets.

The Coronavirus Optimization Algorithm (COVIDOA) [32] is a recent technique that mimics the Coronavirus mechanism inside the human body. COVIDOA has been tested on many benchmarks and real-world problems compared to other metaheuristic techniques. COVIDOA has been proven to have a better performance when compared to other well-known metaheuristics in addition to its high exploration and exploitation capabilities. Additionally, the no-free-lunch theorem has logically proved that no one metaheuristic algorithm, in particular, is best suited for solving all optimization problems. These motivated the authors to propose a binary version of COVIDOA as a wrapper feature selection method to improve the performance of feature selection and classification tasks.

The main contributions of this paper can be summarized as follows:

1. A binary version of the recent COVIDOA algorithm is proposed to solve the feature selection problem.
2. The performance of the proposed algorithm is tested using 26 standard benchmark datasets.
3. A number of evaluation measures are utilized, including classification accuracy, best fitness, average fitness, standard deviation, and selection size.
4. The Wilcoxon rank-sum test was conducted, and the results proved the significance of the proposed algorithm.

This paper is organized as follows: Section 2 provides a brief overview of COVIDOA. In Section 3, the proposed binary COVIDOA is introduced. The datasets and experimental results are discussed in Section 4. Finally, conclusions and future work are given in Section 5.

2. Coronavirus optimization algorithm

COVIDOA is a new evolutionary optimization algorithm proposed by Khalid et al. [32]. COVIDOA is inspired by the replication mechanism of Coronavirus particles when attacking the human body. Four stages are considered to simulate the replication lifecycle of Coronavirus, as follows:

1. **Virus entry and uncoating**
   The virus uses the spike protein on its surface as a key to enter a human cell. Once inside the cell, the virus contents (RNA) are released inside the cell cytoplasm [33].
2. **Virus replication**
   The virus uses the Ribosomal frameshifting technique for replication [32]. Frameshifting is a process when a specific reading frame of RNA molecule shifts to another reading frame to provide a new protein sequence [33–35]. Frameshifting results in the creation of several viral proteins. In the proposed algorithm, a solution (virus particle) is selected for replication. The frameshifting technique produces several viral proteins that are then combined to form a new virus particle (solution). The most popular type of frameshifting is +1 frameshifting [36] which can be modeled as follows:

   - **+1 frameshifting technique**
     The parent solution's values are shifted in the right direction by 1, and the value in the first position is set as a random value in the range [minVal maxVal] as follows.

$$S_k(1) = \text{rand}(\text{minVal}, \text{maxVal}), \tag{1}$$

$$S_k(2:D) = P(1:D-1), \tag{2}$$

     Where minVal and maxVal are the minima and maximum values for the variables in each solution.

3. **Virus mutation**
   Coronavirus accumulates random mutations to escape from the human immune system [37]. In the proposed algorithm, a mutation operator is applied to the solution created in the previous step to generate a new mutated solution as follows:

$$Z_i = \begin{cases} r & \text{if rand}\,(0,1) < MR \\ X_i & \text{otherwise} \end{cases} \tag{3}$$

   X is the solution before mutation, Z is the mutated solution, X_i and Z_i are the ith element in the old and new solutions, respectively, $i = 1, \ldots, D$, and r are random values in the range [minVal, maxVal]. MR is the mutation rate.

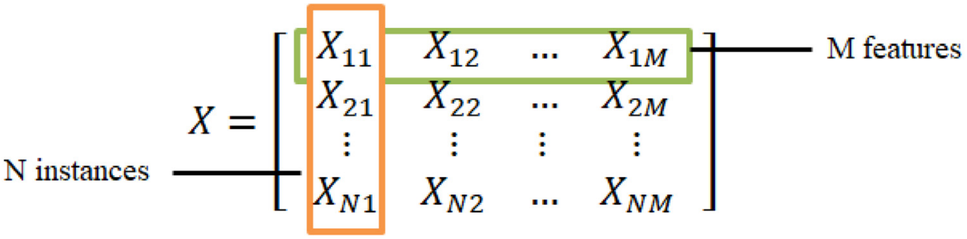

M features
N instances
$X =$
X_{11} X_{12} ... X_{1M}
X_{21} X_{22} ... X_{2M}
X_{N1} X_{N2} ... X_{NM}

Fig. 1. Dataset representation.

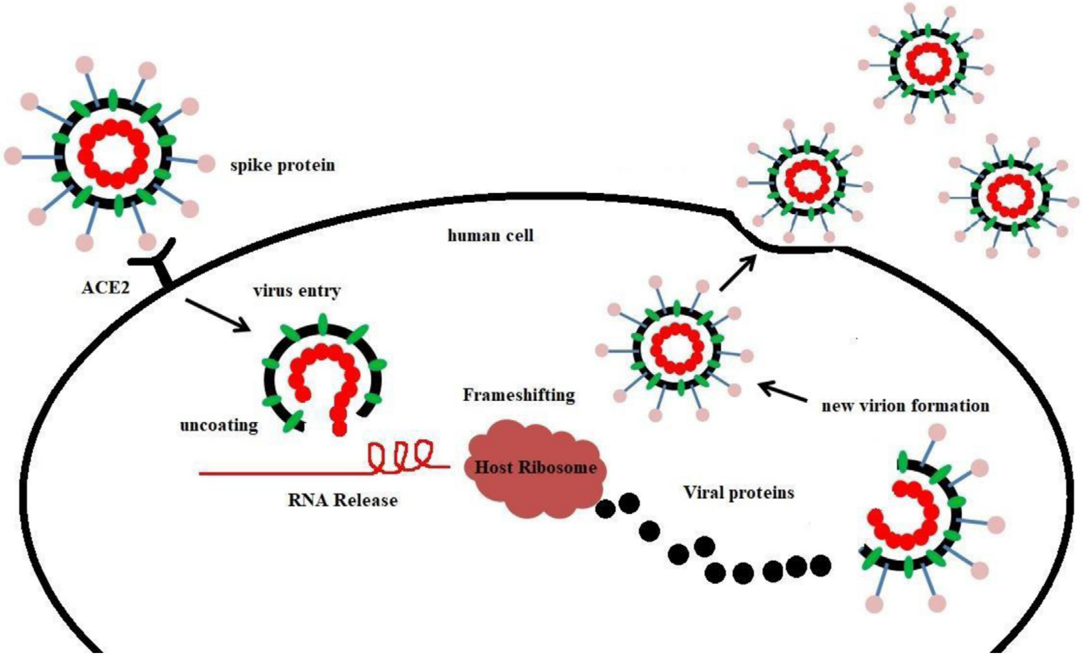


Fig. 2. Coronavirus replication lifecycle.

4. **New virion release**
   Finally, the new virion is released, trying to hijack new healthy cells. The replication lifecycle of Coronavirus is shown in Fig. 2. The flowchart of COVIDOA is shown in Fig. 3.
   The parameters of COVIDOA are described as follows:

- *PopNo* : number of solutions in the initial population.
- *Max_Iter* : maximum number of iterations.
- *MinVal, MaxVal* : lower and upper bounds of each variable in COVID solution. The values of MinVal, and MaxVal depend on the problem; however, in the case of feature selection, they are set to 0 and 1, respectively.
- *D*: refers to the problem dimension. As in the case of MinVal and maxVal, the problem dimension depends on the problem; however, it is equal to the size of features in the dataset in the feature selection problem.
- *MR*: represents the mutation rate. As mentioned in [35], the mutation rate of Coronavirus is 1 × $10^{-6,}$ which is very low; however, the mutation rate in the proposed algorithm is set at a larger value in the range [0.1 0,001], which helps in exploring new promising regions and avoid getting stuck in a local minimum.
- *Shifting Number*: represents the number by which the variables of each solution are shifted in the frameshifting technique. The most common type of frameshifting is +1 frameshifting which uses a shifting number of 1.
- *numOfProtiens*: represents the number of viral proteins generated from one virus particle in the replication process. Each virus particle can generate millions of viral proteins; however, we set it to 2 proteins to avoid computational complexity. As mentioned in the coming sections, parameter tuning is applied to test the impact of changing parameter values on the performance of the proposed algorithm.

Pseudocode of the native COVIDOA is given in Box I.

3. The proposed binary approach

In binary problems, such as feature selection, each solution is represented by a one-dimensional vector that contains only zeros and ones, where one indicates that the feature is selected and 0 indicates it is ignored. The number of elements in the vector equals the size of features in the dataset. The binary representation of a COVID solution for a dataset with the number of features *D* is shown in Fig. 4.

```
Set the initial values of the population size PopNo, maximum number of iterations
Max_Iter, minimum and maximum values MinVal, MaxVal, problem dimension D, cost
function, mutation rate MR=0.001, frameshifting technique shifting = 1, number of
generated proteins numOfProtiens = 2.

For $(i = 1: i \leq n)$ do
        Generate initial random population $X_i(t)$.
        Evaluate the fitness function of each solution in the population.
End for
Order the solutions ascendingly according to the fitness function.
Set the first solution as the optimum solution $X_i^*(t)$.
Set t=1;
Repeat
        For $(i = 1: i \leq n)$ do
                Select a parent solution P,
                For $(k = 1: k \leq numOfProtiens)$ do
                        Generate protein $S_k$ from parent P using equations (1) and (2).
                End for
                Apply uniform crossover between the set of generated proteins to
                produce new virion (new solution).
                If (rand(0,1)< MR) then
                        Mutate the new solution using equation (3)
                End if
        End for

Until t > MaxIt
```

Box I.

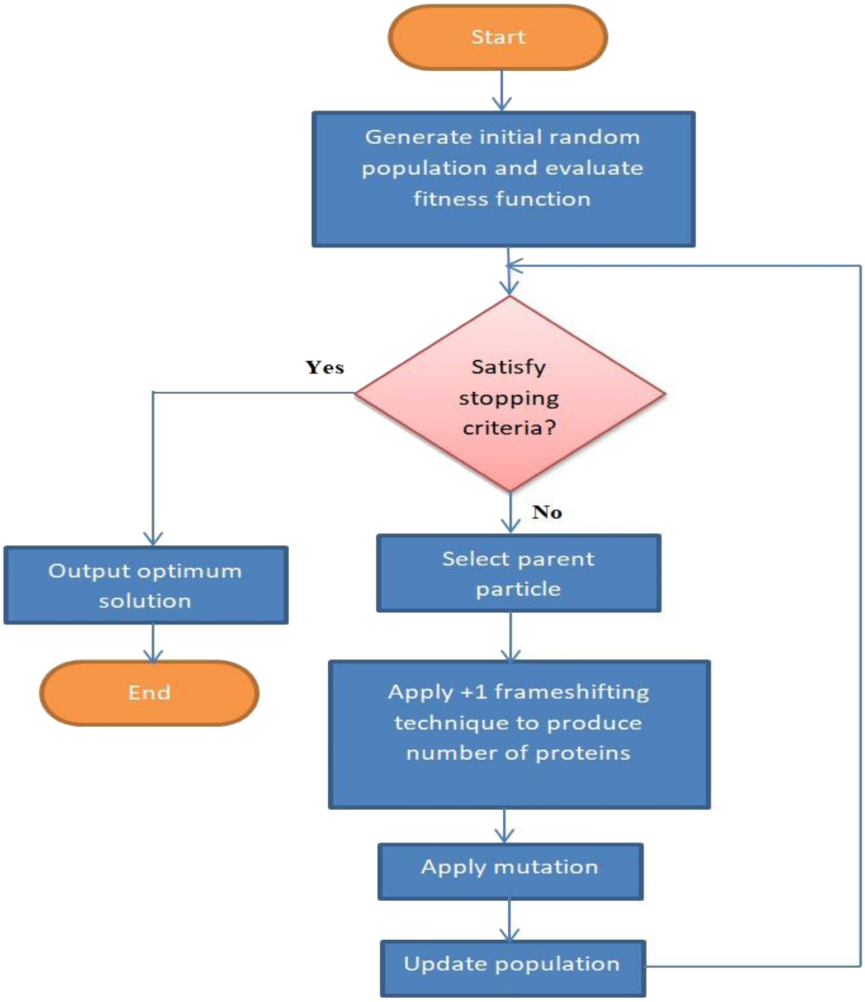

Start
Generate initial random population and evaluate fitness function
Satisfy stopping criteria?
Yes
No
Output optimum solution
End
Select parent particle
Apply +1 frameshifting technique to produce number of proteins
Apply mutation
Update population

Fig. 3. The flowchart of COVIDOA.

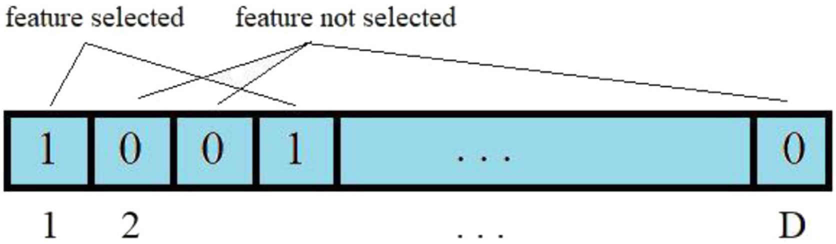


Fig. 4. The binary representation of the COVID solution.

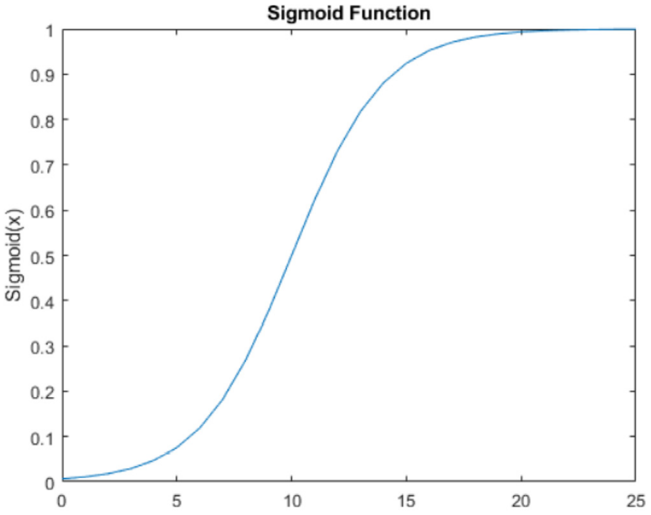


Fig. 5. The sigmoid function.

3.1. Initial stage

In the initial stage of the COVID algorithm, a population *pop* of randomly generated agents is created, where each agent represents a solution to the problem. The following equation generates the initial population:

$$x_i = minVal_i + \alpha_i \times (maxVal_i - minVal_i)\ , i = 1, 2, \ldots, D \tag{4}$$

Where x_i is the solution at the ith location in the population *pop;* α_i is a random value between 0 and 1; $maxVal_i$ and $minVal_i$ are the upper and lower boundaries of the problem. However, in the binary version, $maxVal_i = 1$ and $minVal_i = 0$. In the proposed binary algorithm, each solution is converted into its binary representation using a binarization technique. The sigmoid function is one of the most used transformation functions belonging to the S-shaped family [28,38]. It maps each value in the real-valued solution to a value of 0 or 1 as follows:

$$S\,(x_i) = \frac{1}{1 + e^{-x_i}}, x_{binary=} \begin{cases} 1 & if\ rand \geq S\,(x_i) \\ 0 & otherwise \end{cases} \tag{5}$$

The curve of the sigmoid function is shown in Fig. 5.

3.2. The KNN classifier

The KNN classifier (where $K = 5$) is used to get the classification accuracy of each solution for the following advantages [22,39]:

- It is straightforward to implement. Only two parameters are required to implement KNN, i.e., the parameter *K*, which represents the number of neighbors, and the distance function (e.g., Euclidean or Manhattan, etc.).
- It is a memory-based approach that allows the algorithm to respond quickly to changes in the input [40].
- Efficiency in finding the best subset of features and high robustness to noisy data.
- K-NN algorithm gives the user the flexibility to choose distance while building the K-NN model (e.g., Euclidian distance, Hammig distance, Manhattan distance, etc.).

The role of the KNN classifier is to assign each data point to a class to which most of the closest K neighbors belong. Each dataset is divided into training, validation, and testing sets using cross-validation in the same way as in [41]. The $K - 1$ folds are used for training and validation in cross-validation, and the remaining is utilized for testing.

The K-NN classifier works as follows:

For the test dataset, the kNN algorithm must determine the K closest neighbors for each sample from the training dataset by computing the Euclidean as follows:

$$D(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{6}$$

Where d is the number of features in a given dataset.

As shown in Fig. 6, the classifier assigns the unknown sample to class B (when $k = 3$) because 2 of its closest points are from class B. The classification accuracy for the classifier determines how accurate the class prediction is for the classifier and can be obtained by dividing the correct instances by the total number of instances found in the dataset. On the other hand, the classification error rate can be obtained by dividing the incorrect instances by the total number of instances in the dataset.

The classification accuracy rate must be calculated using the KNN classifier to evaluate each solution, where the best solution is the one with the highest accuracy rate.

3.3. The fitness function

Not only the classification accuracy rate is the only measure used to compare solutions, but an additional objective is needed, which is the number of selected features. When two solutions have the same classification accuracy, the one with the minimum number of selected features is the best. Therefore, the fitness function is to maximize the classification accuracy rate (minimize classification error) and minimize the number of selected features as follows:

$$\text{fitness} = \alpha\gamma_c + (1 - \alpha)\frac{S}{N} \tag{7}$$

Where α is a value between 0 and 1, γ_c is the error rate of the classifier, S represents the number of selected features, and N is the total number of features.

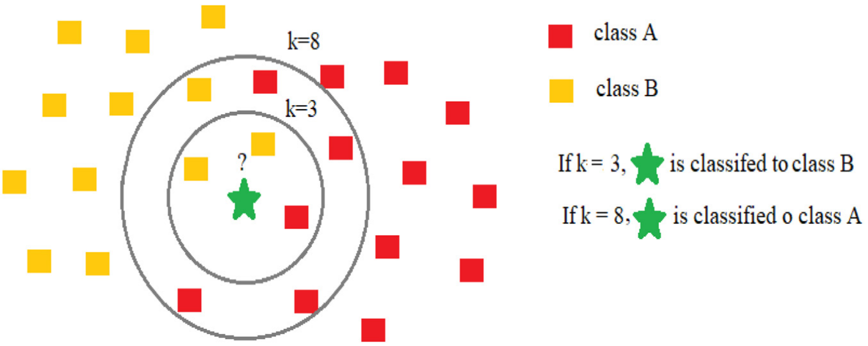

k=8
k=3
?
class A
class B
If k = 3, is classifed to class B
If k = 8, is classified o class A

Fig. 6. KNN classifier.

Table 1
Dataset description.

Dataset		No. of attributes	No. of instances	No. of classes	Area
1	Heart	13	270	5	Life
2	Zoo	16	101	7	Life
3	Breast_cancer	9	699	2	Life
4	Glass_identification	11	214	6	Physical
5	Australian	15	690	2	N/A
6	Spambase	57	4601	2	computer
7	EEG Eye State	15	14980	2	Life
8	Segment	19	2310	7	N/A
9	Waveform	21	5000	3	Physical
10	Auto MPG	8	398	2	N/A
11	House Voting	16	435	2	Social
12	Wine	13	178	3	Physical
13	Vowel	13	990	11	N/A
14	Dermatology	33	366	6	Life
15	Cryotherapy	7	90	N/A	Life
16	M-of-n	44	267	N/A	N/A
17	kr-vs-kp	36	3196	2	Game
18	Optical recognition	64	5620	10	Computer
19	Page blocks	10	5473	2	Computer
20	Semion	256	1593	10	Computer
21	Pendigits	16	10992	2	Handwriting
22	Movement_libras	91	360	15	N/A
23	arrhythmia	279	452	13	Life
24	isolet5	617	7797	26	Computer
25	Mturk	500	180	N/A	Computer
26	pixraw10P	10000	100	10	Face image data

3.4. Position update

As mentioned in [32], virus particles (solutions) use the frameshifting technique for producing multiple protein sequences, which are then combined to form a new particle (solution). In the original COVIDOA version, the +1 frameshifting technique is applied to a solution by shifting its variables to the right by 1, and the value in the first position is assigned with a random value in the range [minVal maxVal] as mentioned in Eqs. (1) and (2). In the binary version, the frameshifting technique is applied as follows:

$$S_k(1) = \begin{cases} 1 & if\ rand(0,1) < 0.5 \\ 0 & otherwise \end{cases} \tag{8}$$

$$S_k(2:D) = P(1:D-1), \tag{9}$$

S_k refers to the kth generated protein, P is the parent solution, D is the problem dimension, $rand()$ is a random value between 0 and 1. After replication, a mutation in the binary version can be applied as follows:

$$Z_i = \begin{cases} \begin{cases} 1 & \text{if rand}(0,1) < 0.5 \\ 0 & \text{otherwise} \end{cases} & if\ rand\,(0,1) < MR \\ X_i & \text{otherwise} \end{cases} \tag{10}$$

X is the solution before mutation, Z is the mutated solution, and X_i and Z_i are the ith elements in the old and new solutions.

4. Experimental results

This section presents the proposed algorithm results and the comparisons with the state-of-the-art algorithms. The proposed and state-of-the-art algorithms were run on a laptop with the following specifications: Intel(R) Core(TM) i7-1065G7 CPU, 8 GB RAM, Windows 10 operating system, and MATLAB R2016a development environment.

4.1. Datasets

We applied the proposed method to 26 different datasets from the UC Irvine Machine Learning Repository [42] to prove the efficiency of the proposed algorithm. Each data set is described in terms of the number of features, number of instances, number of classes, and the area to which they belong. We utilized many datasets to ensure the efficiency of the proposed algorithm in feature selection. These 26 datasets are selected based on the variety in dimension size (number of features) and the area they belong to. We utilized datasets with small (9, 11, 13), medium (64, 91, 256), and large (500, 617, 10000) dimension size. Furthermore, the datasets belong to different areas such as life, computer, physical, game, and social. A detailed description of the datasets is presented in Table 1. N/A means that this information is not known.

Pseudocode of the proposed binary COVIDOA is given in Box II.

```
Set the initial values of the population size PopNo, maximum number of iterations
Max_Iter, minimum and maximum values MinVal=0, MaxVal=1, problem dimension D, cost
function, mutation rate MR=0.01 , frameshifting technique shifting = 1, number of
generated proteins numOfProtiens = 2,

For $(i = 1: i \leq n)$ do
          Generate initial random population $X_i(t) \in [0,1]$.
          Evaluate the fitness function of each solution in the population.
End for
Order the solutions ascendingly according to the fitness function.
Set the first solution as the optimum solution $X_i^*(t)$.
Set t=1;
Repeat
          For $(i = 1: i \leq n)$ do
                    Select a parent solution P,
                    For $(k = 1: k \leq numOfProtiens)$ do
                              Generate protein $S_k$ from parent P using equations (8) and (9).
                    End for
                    Apply uniform crossover between the set of generated proteins to
                    produce new virion (new solution).
                    If (rand(0,1)< MR) then
                              Mutate the new solution using equation (10)
                    End if
          End for
```

Box II.

Table 2
Parameter setting.

Algorithm	Parameter	Value

GA	Selection
	Crossover
	Mutation
		Roulette wheel
		Probability = 0.9
		Probability = 0.05

DE	Scaling factor Crossover probability	0.5 0.5
PSO	topology Cognitive and social constant	Fully connected (C1, C2) 2, 2
WOA	a r	2 to 0 [0,1]
GWO	a r	2 to 0 [0,1]
HH	a	2 to 0
AOA	α μ	5 0.5
COVID	Shifting No. No. of proteins Mutation	1 2 0.1

Table 3
Scenarios of the tuning parameters.

Scenario	Parameters	
	MR	numOfProtiens
1	0.1	2
2	0.01	2
3	0.001	2
4	0.1	4
5	0.01	4
6	0.001	4
7	0.1	6
8	0.01	6
9	0.001	6

4.2. Parameter setting

In all algorithms, the number of solutions in the population is 50, and the maximum number of iterations is 100. The proposed and state-of-the-art algorithms were run 20 times and the best results gained from these runs are reported. The problem dimension equals the number of features in the dataset, and the search domain is set to [0,1]. The remaining parameters of the proposed and state-of-the-art algorithms are set as shown in Table 2.

4.3. Parameter tuning

To test the impact of changing parameter values on the performance of the COVID algorithm, we used nine different scenarios by changing the values of the parameters MR (Mutation Rate) and numOfProtiens. We utilized the values of 0.1, 0.01, ad 0.001 for MR, 2, 4, and 6 for numOfProtiens which produces nine scenarios, as shown in Table 3. The feature selection results (for the zoo dataset) of each scenario are shown in Table 4. It is observed that scenario 1 yields the best results, followed by scenario 3, which means that a higher mutation rate and a lower number of proteins improve the performance of the proposed algorithm.

4.4. Evaluation measures

The results of the proposed algorithm are compared to the state-of-the-art feature selection algorithms such as GA [43], DE [44], PSO [45], WOA [25], WOASA [28], GWOPSO [46], HH [47], GWO [48], and AOA [49]. The parameters of these algorithms are selected as suggested by their authors. The comparison is made according to the following measures:

- **Classification Accuracy**
  It measures how accurate the classifier is in selecting the optimum subset of features. The maximum classification accuracy can be calculated as follows:

  $$BestAcc = \max(Acc_n) \tag{11}$$

  Where Acc_n refers to the accuracy at run n, where $n = 1, \ldots, M$ and M is the number of runs.
- **Best cost**
  The best cost at run n can be calculated as follows:

  $$BestCost_n = \min_i Cost_i \tag{12}$$

  Where $Cost_i$ is the cost obtained at iteration i where i =1, …,$MaxIter$. The best cost obtained over the M runs can be calculated as follows:

  $$BestCost = Min(BestCost_n) \tag{13}$$

  Where $n = 1, \ldots,$ M and M is the number of runs.

Table 4
The results of parameter tuning on Zoo dataset.

Metric	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7	Scenario 8	Scenario 9
Accuracy	**1**	**1**	0.96078	**1**	0.98039	0.98039	0.98039	0.96078	0.98039
Best fitness	**0.00375**	0.004375	0.042574	0.005625	0.024412	0.025037	0.025037	0.043824	0.024412
Average fitness	**0.0037**	0.0044	0.0429	0.0056	0.0247	0.0250	0.0328	0.0474	0.0279
STD	**5.1021e−18**	6.2304e−18	4.9024e−04	5.2304e−18	3.0894e−04	1.7435e−17	0.0095	0.0086	0.0067
Selection size	**6**	7	**6**	9	8	9	9	8	8

Table 5
Results obtained from the Proposed Binary COVIDOA.

Dataset		Accuracy	Best Cost	Average Cost	Standard Deviation (STD)	Selection Size
1	Zoo	1	0.00187	0.00332	1.0548e−05	3
2	Heart	0.87407	0.12774	0.1394	0.0083	4
3	Breast_cancer	0.98	0.026467	0.0272	1.8282e−04	5
4	Glass_identification	1	0.002	0.0021	1.0000e−04	2
5	Australian	0.87826	0.12124	0.12342	0.0032	1
6	spambase	0.92047	0.083823	0.0867	0.0031	29
7	EEG Eye State	0.96662	0.04233	0.0439	0.0024	13
8	Segment	0.97576	0.028737	0.0306	9.1212e−04	9
9	Waveform	0.8008	0.20435	0.2051	0.0028	15
10	Auto MPG	0.88889	0.11286	0.1129	1.9739e−17	2
11	House voting	0.8945	0.10645	0.1064	1.1158e−16	3
12	Wine	0.98876	0.01529	0.0153	2.6152e−17	4
13	Vowel	0.97571	0.031549	0.0318	7.2473e−04	8
14	Dermatology	0.99454	0.011586	0.0182	0.0024	13
15	Cryotherapy	0.97778	0.025333	0.0253	1.3948e−17	2
16	M-of-n	1	0.004615	0.0051	0.0023	20
17	kr-vs-kp	0.83917	0.16293	0.1629	2.7895e−16	13
18	Optical recognition	0.99444	0.013006	0.0136	9.7768e−04	31
19	Page blocks	0.96346	0.040171	0.0403	3.2698e−04	4
20	Semion	0.98369	0.020903	0.0220	0.0024	125
21	Pendigits	0.99314	0.014292	0.0144	2.3527e−04	11
22	Movement_libras	0.87222	0.13094	0.1314	0.0015	36
23	arrhythmia	0.66814	0.33151	0.3401	0.0016	73
24	isolet5	0.84743	0.156873	0.1651	0.0047	250
25	pixraw10P	0.8482	0.16326	0.1633	3.6043e−06	2861
26	mturk	0.65773	0.35454	0.3623	0.0142	289
Average		**0.9250**	**0.0898**	**0.0920**	**0.0019**	**147.15**

- **Average cost:**
  The average cost at each run n can be calculated as follows:

$$AvgCost_n = \frac{1}{MaxIter}\sum_{i=1}^{MaxIter} cost_i \tag{14}$$

  Where $cost_i$ refers to the cost at iteration i.
  The minimum average cost over M runs can be calculated as follows:

$$AvgCost = \min_n AvgCost_n \tag{15}$$

  Where $n = 1, \ldots$, M and M is the number of runs.

- **Standard Deviation (STD)**
  It shows how the cost values are far from the average cost. STD at run n can be calculated as follows:

$$STD_n = \sqrt{\frac{\sum_{i=1}^{MaxItr} |cost_i - AvgCost|^2}{MxIter}} \tag{16}$$

  The minimum STD over M runs can be calculated as follows:

$$MinSTD = \min_n STD_n \tag{17}$$

- **Selection size**
  The selection size at run n can be calculated as follows:

$$SelSize_n = \frac{SF_n}{D} \tag{18}$$

  SF_n refers to the number of selected features in the optimum solution obtained at run n. The minimum number of features obtained over the M runs can be calculated as follows:

$$MinSelSize = \min_n SelSize_n \tag{19}$$

- **Wilcoxon rank-sum test**
  Null hypothesis [50] is a type of hypothesis widely used in Statistics. It is used to prove the results' statistical significance. The test results of the 15 datasets are compared using Wilcoxon rank-sum test at the %5 significance level [51]. A small p-value (typically ≤ 0.05) indicates strong evidence against the null hypothesis. In this hypothesis, researchers assume no significant difference between the two methods' average values.

4.5. Results

This section presents the numerical results of the proposed binary COVIDOA and the comparisons with the state-of-the-art algorithms. The results of the proposed binary COVIDOA for feature selection are presented in Table 5. The proposed binary COVIDOA achieved the highest accuracy rate (100%) for the datasets named Zoo, Glass identification, and M-of-n. The minimum accuracy achieved is 65% for the mturk dataset. Moreover, the binary COVIDOA achieved the minimum best fitness (0.001) and average fitness (0.0032) for the Zoo dataset and minimum STD value (1.3948e−17) for the Cryotherapy dataset. Besides, the proposed algorithm achieved the minimum selection size of 1 from 15 features for the Australian dataset. The average accuracy, best fitness, average fitness, standard deviation, and selection size for all 26 datasets using the binary COVIDOA are achieved: 92.5%,

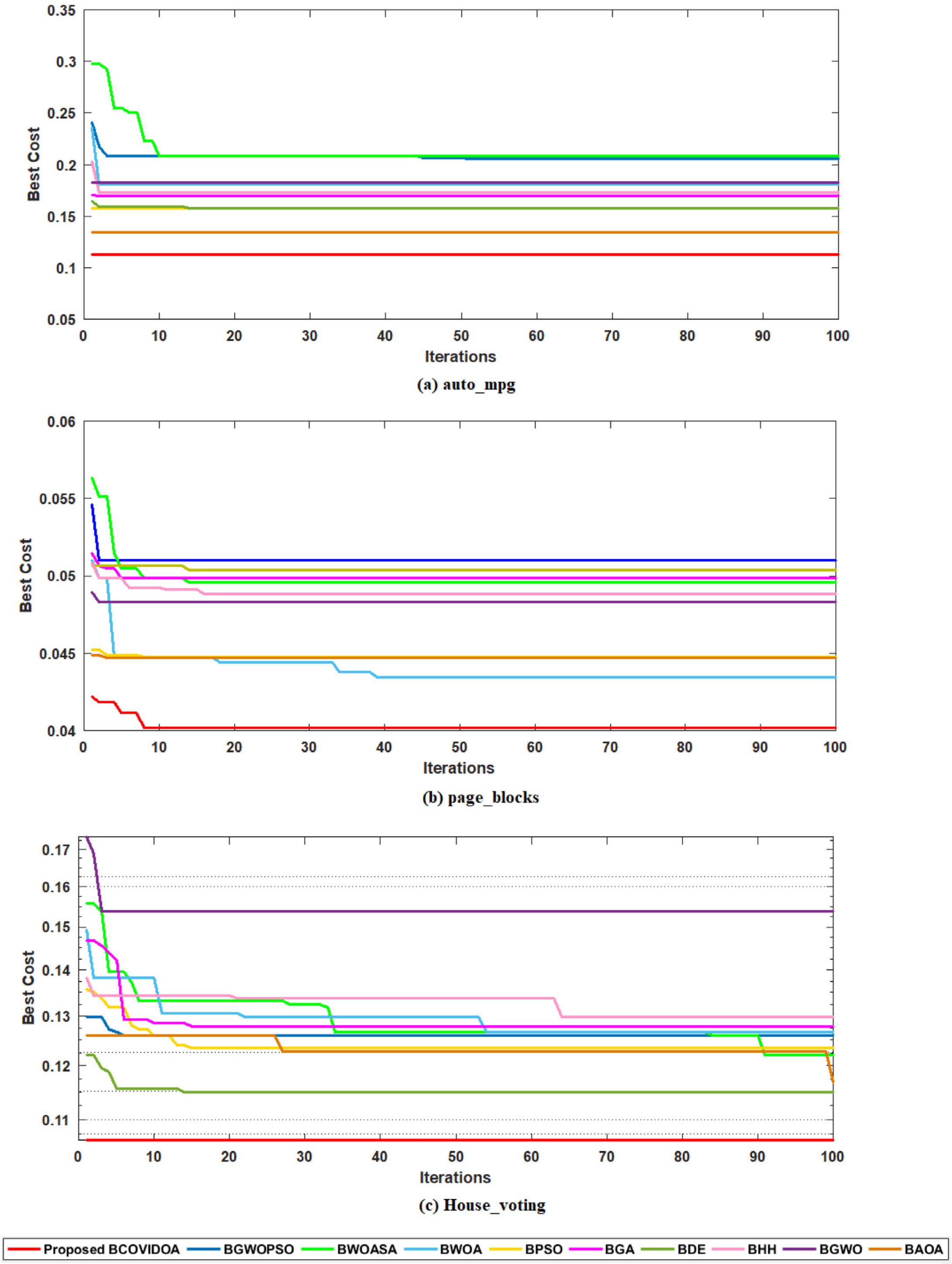
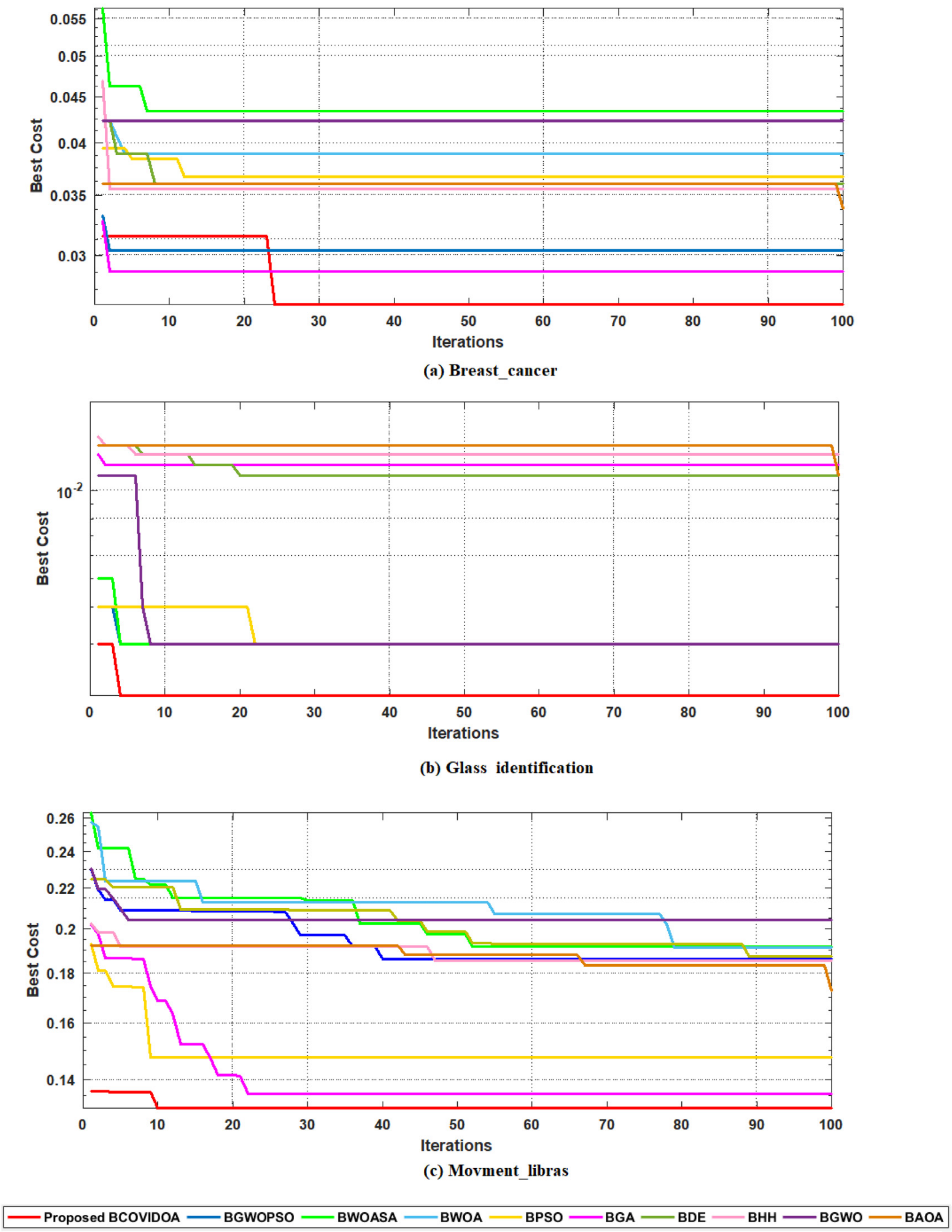


Fig. 7. Comparison of convergence curves of the binary COVIDOA and the state-of-the-art algorithms for (a)Auto_mpg, (b) and Page_blocks, and (c) House_voting datasets.

0.0898, 0.0920, 0.0019, and 147.15, respectively. The obtained results reveal the efficiency of the proposed algorithm, which indicates its strong exploration and exploitation capabilities.

Figs. 7–10 show the convergence curves of applying the proposed algorithm in the feature selection of various datasets. The curves represent the relationship between the iterations from 1 to 100 and the corresponding fitness values for the proposed binary COVIDOA and the state-of-the-art algorithms (GA, DE, PSO, WOA, WOASA, GWOPSO, HH, GWO, and AOA). It is evident from the figure that the proposed binary COVIDOA outperforms the state-of-the-art algorithms' overall datasets in terms of fitness values. In addition, the figure indicates the rapid convergence of the proposed algorithm as it reaches the global optimum during the first few iterations.

The numerical results of the proposed algorithm against the state-of-the-art algorithms are shown in Tables 6–10. Table 6 shows the comparison in terms of classification accuracy. This table shows that the proposed algorithm reaches the maximum accuracy in 22 out of 26 datasets; however, GA reaches the maximum accuracy in only 5 out of 26 datasets.



(a) Breast_cancer

(b) Glass identification

(c) Movment_libras

Fig. 8. Comparison of convergence curves of the binary COVIDOA and the state-of-the-art algorithms for (a) Breast_cancer, (b) glass_identification, and (c) movement_libras datasets.

The proposed algorithm achieves the highest total average accuracy over all datasets. The bar chart in Fig. 11 presents a comparison in terms of the total average classification accuracy. The figure shows that the proposed algorithm comes in the first position with total average accuracy (92.5%), followed by the GWOPSO algorithm with total average accuracy (90%).

The statistical results of the best, average, and standard deviation of the fitness values of each algorithm are presented in Tables 7–9. The best results are in bold. As can be seen from the results, the binary COVIDOA outperforms the other algorithms in 19 out of 26 datasets in terms of best fitness. Each algorithm's average best fitness is recorded in Table 7 and shown in the bar chart in Fig. 12.

The proposed binary COVIDOA algorithm archives the minimum average best fitness value (0.092) for overall datasets followed by (0.0106) for the GWOPSO algorithm. The proposed algorithm achieves the minimum average fitness in 17 out of 26 datasets, followed by GA, which achieves the minimum average fitness in 7 out of 26 datasets, as seen in Table 8. A comparison between the algorithms in terms of the total average for the average fitness values is shown in Fig. 13. The figure shows that the proposed algorithm comes in the first position with a value

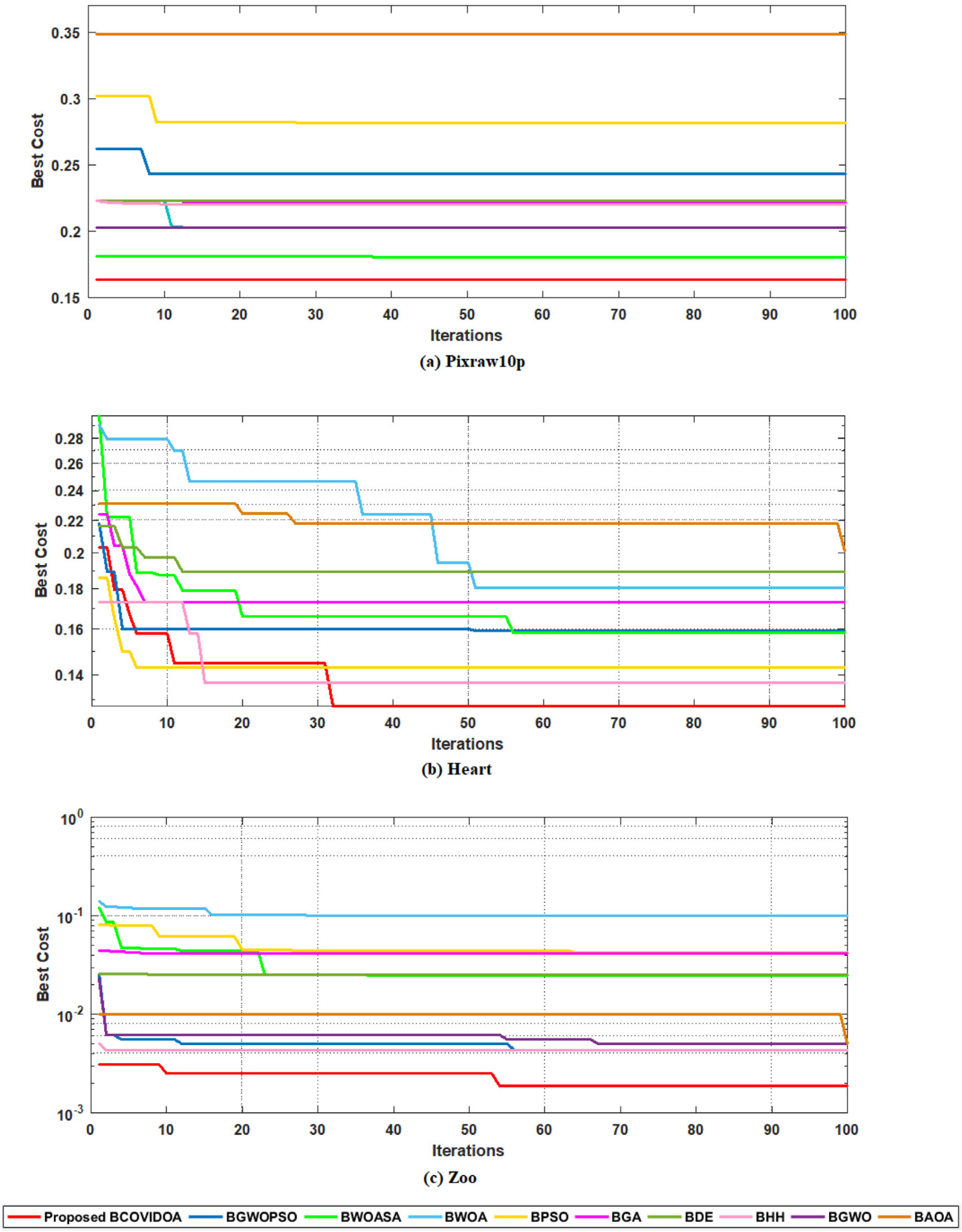


Fig. 9. Comparison of convergence curves of the binary COVIDOA and the state-of-the-art algorithms for (a) Pixraw10p, (b) Heart, and (c) Zoo datasets.

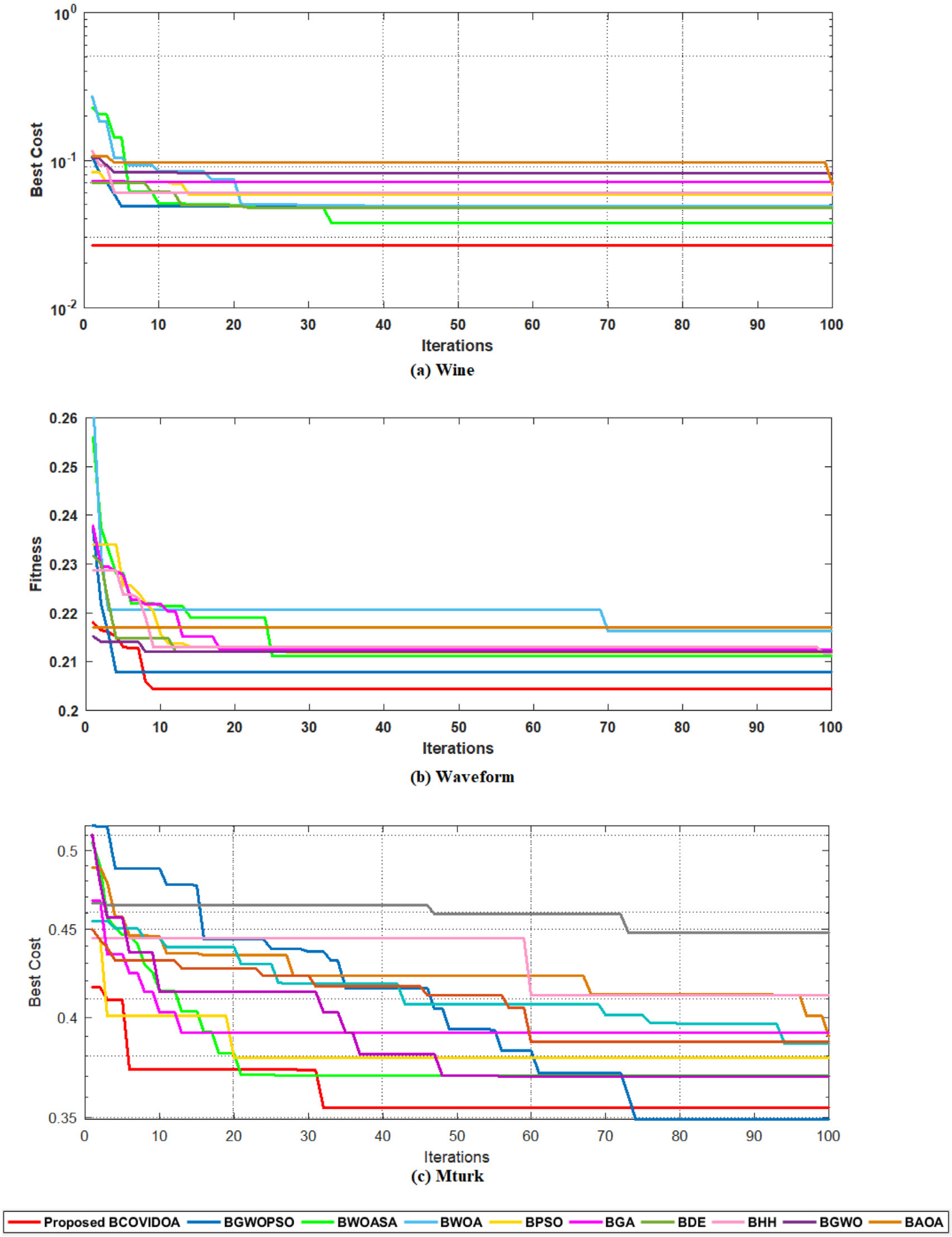
of (0.0898), followed by the GWOPSO algorithm with a value of (0.01024).

The standard deviation is one of the essential metrics to evaluate the proposed algorithm. Lower standard deviation values indicate that the fitness values are clustered closely around the mean value, proving the proposed algorithm's stability. As shown in Table 9, the proposed algorithm comes in the first rank as it achieves the minimum STD values in 17 out of 26 datasets and has the minimum average STD value (0.0019) compared to its peers, see Fig. 14.

While maintaining the highest classification accuracy, another objective is to achieve the minimum number of selected features. Table 10 reports the number of selected features for all datasets. The proposed algorithm achieves the minimum selection size for 17 out of 26 datasets. It achieves the minimum average selection size (147.15) for all datasets, which means that the proposed algorithm has high size reduction capabilities. As shown in Fig. 15, the proposed algorithm is superior to its peers in the selected size.

The Wilcoxon rank-sum test is a nonparametric statistical test that compares two paired groups. This test calculates the difference between sets of pairs and analyzes them to establish if they are statistically significantly different. The test results of the 26 benchmark datasets are compared using Wilcoxon rank-sum test at the %5 significance level.

Table 11 introduces the p values computed by Wilcoxon rank-sum test that compares the binary COIDOA with nine well-known



(a) Wine

(b) Waveform

(c) Mturk

Fig. 10. Comparison of convergence curves of the binary COVIDOA and the state-of-the-art algorithms for (a) Wine, (b) Waveform, and (c) Mturk datasets.

metaheuristic algorithms for the 26 benchmark datasets. We observed from the table that all p values are less than a 5% significance level for all comparative algorithms; this is strong evidence against the null hypothesis. Therefore, we conclude that the binary COVIDOA is better than all comparative algorithms.

The frameshifting technique applied to the population in the replication stage of COVIDOA helps enhance the population diversity of the search space and converge to global optima. The binary COVIDOA reaches the minimum selection size while maintaining the highest classification accuracy, achieving the two objectives of the feature selection problem. The convergence curves and standard deviation values also prove its high convergence as it rapidly reaches the global optimum.

5. Conclusions and future work

Feature selection is a way to eliminate redundant and irrelevant data. This process leads to improved learning accuracy, reduced computational time, and enhanced understanding of learning models. This paper proposed a binary approach to the Coronavirus Optimization algorithm based on a wrapper method to solve feature selection problems. The proposed algorithm used the KNN classifier because its simplicity has two

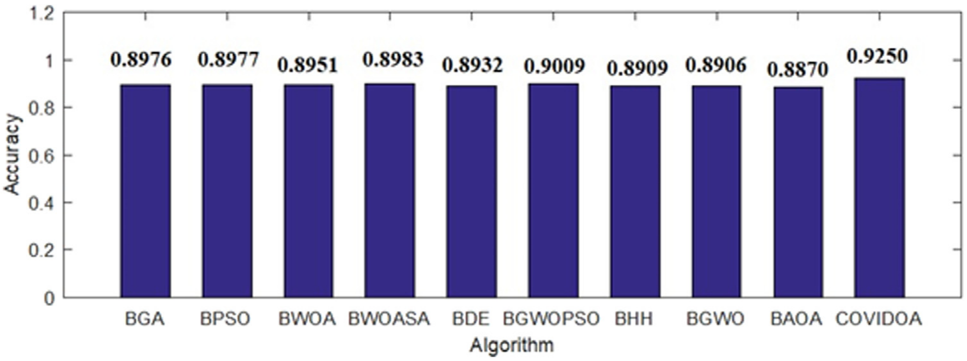


Fig. 11. : The total average accuracy for all datasets.

Table 6
The results of classification accuracy of the proposed and state-of-the-art algorithms.

Dataset		Algorithm									
		BGA [43]	BPSO [45]	BWOA [23]	BWOASA	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
1	Heart	0.8296	0.859259	0.822222	0.844444	0.814815	0.844444	0.866667	0.807407	0.80000	**0.87407**
2	Zoo	0.9608	0.960784	0.960784	0.980392	0.960784	**1**	**1**	**1**	**1**	**1**
3	Breast_cancer	0.9743	0.968571	0.968571	0.977143	0.968571	0.977143	0.968571	0.971429	0.971429	**0.98**
4	Glass_identification	0.9907	**1**	0.990654	**1**	0.990654	**1**	**1**	**1**	0.990654	**1**
5	Australian	0.8348	0.872464	0.857971	0.849275	0.811594	0.843478	0.843478	0.834783	0.8724q64	**0.87826**
6	spambase	**0.9387**	0.922642	0.917427	0.928292	0.924815	0.932203	0.916993	0.923077	0.912647	0.92047
7	EEG Eye State	**0.96889**	**0.966889**	0.960080	0.966088	0.966622	0.962350	0.965154	0.965688	0.966355	0.966622
8	Segment	0.9680	0.967965	0.959307	0.965368	0.961905	0.969697	0.961039	0.968831	0.961905	**0.97576**
9	Waveform	0.7916	0.793600	0.791200	0.794400	0.794000	0.799600	0.792800	0.794400	0.794000	**0.8008**
10	Auto MPG	0.8333	0.848485	0.823232	0.792929	0.843434	0.828283	0.828283	0.818182	0.868687	**0.88889**
11	House Voting	0.8716	0.880734	0.876147	0.857798	0.889908	0.876147	0.871560	0.844037	0.885321	**0.8945**
12	Wine	0.9326	0.943820	0.955056	0.966292	0.955056	0.955056	0.943820	0.955056	0.932584	**0.98876**
13	Vowel	0.9656	0.957490	0.943320	0.969636	0.967611	0.961538	0.951417	0.969636	0.973684	**0.97571**
14	Dermatology	0.9836	0.989071	0.983607	0.989071	0.989071	0.983607	0.983607	0.983607	0.989071	**0.99454**
15	Cryotherapy	0.8889	0.977778	0.955556	0.955556	0.977778	0.977778	0.933333	0.911111	0.955556	**0.97778**
16	M-of-n	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
17	kr-vs-kp	0.8242	0.831665	0.823529	0.814768	0.825407	0.831039	0.811640	0.808511	0.804130	**0.83917**
18	Optical recognition	0.9933	0.992214	0.991101	0.994438	0.992214	0.989989	0.983315	0.993326	0.982202	**0.99444**
19	Page blocks	0.9547	0.959810	0.960175	0.957252	0.952137	0.952503	0.957252	0.954695	0.960906	**0.96346**
20	Semion	**0.9925**	0.984944	0.984944	0.982434	0.986198	0.992472	0.976161	0.984944	0.985539	0.98369
21	Pendigits	**0.9943**	0.994282	0.990280	0.992567	0.994282	0.993139	0.99313	0.991424	0.983689	0.99314
22	Movement_libras	0.8667	0.855556	0.811111	0.811111	0.816667	0.850	0.816667	0.800000	0.833333	**0.87222**
23	arrhythmia	0.67234	0.663717	**0.703540**	0.690265	0.616071	0.672566	0.615044	0.6239	0.609091	**0.66814**
24	isolet5	0.8769	0.807692	0.833333	0.84359	0.826923	0.837692	0.816667	0.819231	0.805128	**0.84743**
25	pixraw10P	0.7800	0.720000	0.800000	0.822222	0.780000	0.760000	0.780000	0.80000	0.65306	**0.8482**
26	mturk	0.6517	0.622222	0.611111	0.611111	0.617978	0.633333	0.588889	0.633333	**0.556818**	**0.65773**
Average		0.8976	0.8977	0.8951	0.8983	0.8932	0.9009	0.8909	0.8906	0.8870	**0.9250**

Table 7
The results of best fitness of the proposed and state-of-the-art algorithms.

Dataset		Algorithm									
		BGA [43]	BPSO [45]	BWOA [23]	BWOASA [28]	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
1	Heart	0.17328	0.143179	0.180615	0.158615	0.189487	0.159385	0.136615	0.195282	0.201077	**0.12774**
2	Zoo	0.041324	0.042574	0.042574	0.024412	0.042574	0.003125	0.005000	0.004375	0.005000	**0.00187**
3	Breast_cancer	0.0287	0.036670	0.040003	0.028184	0.038892	0.030406	0.035559	0.033841	0.033841	**0.026467**
4	Glass_identification	0.012252	0.003	0.011252	0.003000	0.011252	0.003	0.004	0.012252	0.011252	**0.002**
5	Australian	0.16499	0.128404	0.141323	0.151360	0.192950	0.158528	0.158528	0.168565	0.126975	**0.12124**
6	spambase	**0.066805**	0.082023	0.088414	0.078184	0.081626	0.072206	0.088669	0.083873	0.091918	0.083823
7	EEG Eye State	**0.039951**	0.042065	0.048806	0.042858	0.042330	0.046559	0.043784	0.043255	0.042594	0.0439
8	Segment	0.034872	0.034346	0.042917	0.037444	0.041925	0.033158	0.044361	0.034541	0.039820	**0.028737**
9	Waveform	0.21251	0.211955	0.216236	0.211163	0.212035	0.207915	0.211795	0.212115	0.212035	**0.20435**
10	Auto MPG	0.16929	0.151429	0.180714	0.207857	0.157857	0.172857	0.172857	0.182857	0.134286	**0.11286**
11	House Voting	0.12782	0.123407	0.126615	0.142780	0.114991	0.125948	0.129823	0.159737	0.116865	**0.10645**
12	Wine	0.070908	0.058951	0.048661	0.037537	0.047828	0.048661	0.060618	0.050328	0.068408	**0.01529**
13	Vowel	0.040735	0.048752	0.063613	0.036727	0.040398	0.045577	0.054764	0.036727	.033553	**0.031549**
14	Dermatology	0.020347	0.015526	0.022700	0.015526	0.016996	0.020053	0.022406	0.021524	0.014643	**0.011586**
15	Cryotherapy	0.11333	0.027000	0.049000	0.049000	0.027000	0.027000	0.071000	0.091333	0.049000	**0.025333**
16	M-of-n	0.004615	0.004615	0.004615	0.004615	0.004615	0.004615	0.004615	0.004615	0.004615	**0.004615**
17	kr-vs-kp	0.17751	0.172938	0.179563	0.189094	0.177704	0.171557	0.193048	0.196146	0.198197	**0.16293**

(*continued on next page*)

Table 7 (*continued*).

Dataset		Algorithm									
		BGA [43]	BPSO [45]	BWOA [23]	BWOASA [28]	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
18	Optical recognition	**0.010826**	0.013177	0.015372	0.011444	0.014584	0.014755	0.023550	0.015201	0.018974	0.013006
19	Page blocks	0.049852	0.044788	0.043426	0.046320	0.050384	0.051022	0.048320	0.048852	0.044703	**0.040171**
20	Semeion	**0.011642**	0.019963	0.022975	0.020113	0.021135	0.013604	0.029526	0.022340	0.023378	0.020903
21	Pendigits	**0.013785**	**0.013785**	0.014858	0.016498	**0.013785**	0.014292	0.014292	0.015991	0.025771	0.014292
22	Movement_libras	0.13533	0.147444	0.191444	0.191667	0.187278	0.151722	0.185611	0.204111	0.170000	**0.13094**
23	arrhythmia	**0.28637**	0.337938	0.298263	0.311225	0.388476	0.327528	0.386698	0.379299	0.395315	0.33151
24	isolet5	**0.12621**	0.195133	0.170640	0.16005	0.180260	0.147530	0.1899	0.186919	0.197429	0.156873
25	pixraw10P	0.22169	0.281957	0.202528	0.180455	0.222666	0.243217	0.220459	0.202906	0.34823	**0.16326**
26	mturk	**0.3492**	0.378970	0.390251	0.390351	0.386258	0.370114	0.412251	0.369754	0.448129	**0.35454**
	Average	0.1040	0.1061	0.1091	0.1056	0.1117	0.1024	0.1133	0.1144	0.1175	**0.0898**

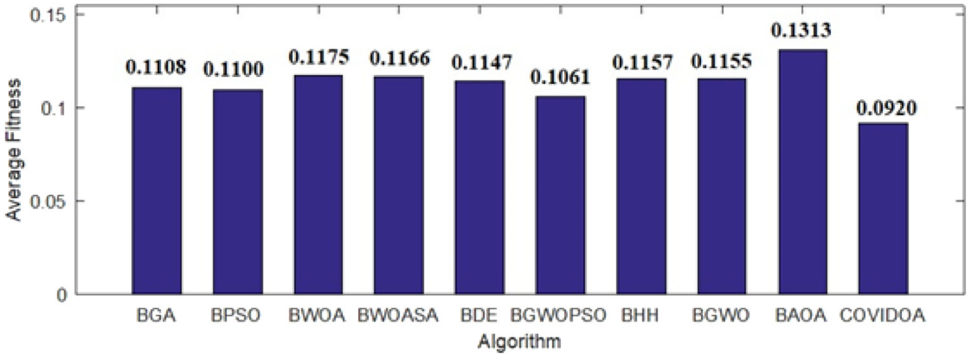


Fig. 12. The total average best fitness overall datasets.

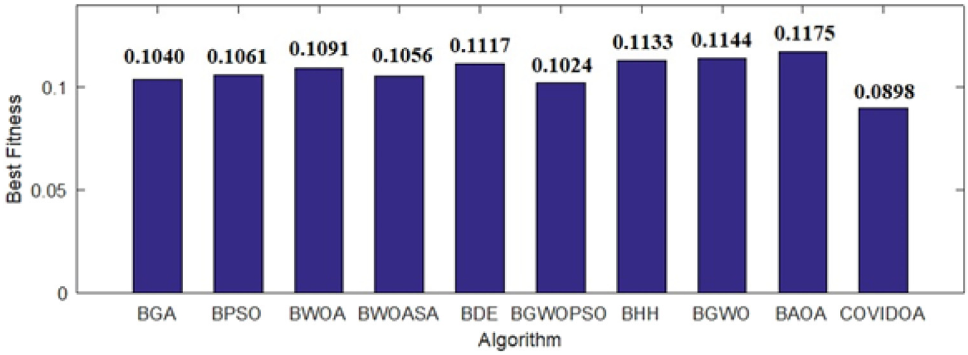


Fig. 13. The total average mean fitness overall datasets.

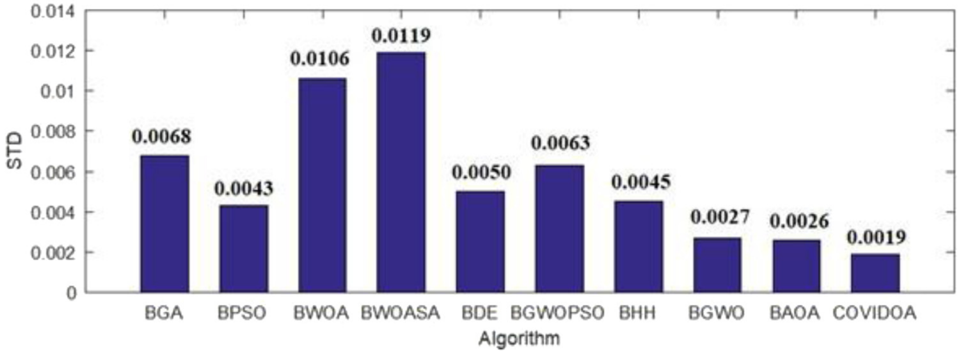


Fig. 14. The total average STD overall datasets.

parameters: K and distance function. Many evaluation metrics are utilized to evaluate the performance, such as classification accuracy, best fitness, average fitness, standard deviation, and selection size. The proposed algorithm is tested on 26 benchmark datasets, and the results are compared with nine well-known metaheuristics.

Additionally, the Wilcoxon rank-sum test is evaluated to prove the significance of the proposed algorithm. The statistical results reveal that the proposed algorithm performs better than the state-of-the-art algorithms. The convergence curves proved that it has a high convergence speed as it reaches the global optimum rapidly.

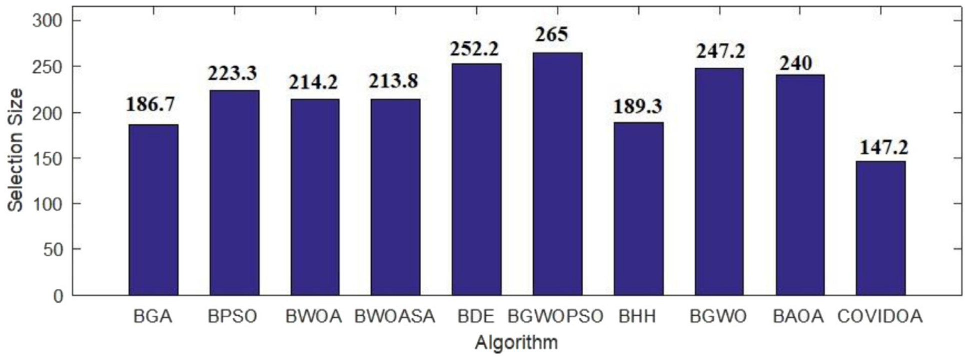


Fig. 15. The total average selection size of overall datasets.

Table 8
The results of the average fitness of the proposed and state-of-the-art algorithms.

Dataset	Algorithm									
	BGA [43]	BPSO [45]	BWOA [23]	BWOASA [28]	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
1 Heart	0.2015	0.1702	0.2043	0.1851	0.1522	0.1611	0.1432	0.1955	0.2208	**0.1394**
2 Zoo	0.0527	0.0826	0.0452	0.0818	0.0645	0.0047	0.0059	0.0044	0.0100	**0.00332**
3 Breast_cancer	0.0311	0.0328	0.0317	0.0385	0.0321	0.0350	0.0357	0.0338	0.0360	**0.0272**
4 Glass_identification	0.0123	0.0032	0.0115	0.0031	0.0116	0.0245	0.0041	0.0123	0.0142	**0.0021**
5 Australian	0.1655	0.1305	0.1615	0.1621	0.1931	0.1599	0.1601	0.1687	0.2172	**0.12342**
6 spambase	**0.0713**	0.0836	0.0989	0.0838	0.0850	0.0750	0.0901	0.0842	0.1053	0.0867
7 EEG Eye State	**0.0401**	0.0424	0.0498	0.0454	0.0424	0.0467	0.0442	0.0433	0.0426	0.0439
8 Segment	0.0351	0.0345	0.0475	0.0413	0.0429	0.0334	0.0448	0.0346	0.0444	**0.0306**
9 Waveform	0.2142	0.2137	0.2198	0.2141	0.2127	0.2084	0.2140	0.2123	0.2284	**0.2051**
10 Auto MPG	0.1693	0.1514	0.1813	0.2126	0.1581	0.1729	0.1732	0.182857	0.1343	**0.1129**
11 House Voting	0.1288	0.1242	0.1293	0.1479	0.114991	0.1261	0.1324	0.1600	0.1235	**0.1064**
12 Wine	0.0710	0.0607	0.0602	0.0418	0.0503	0.0499	0.0618	0.0504	0.0965	**0.0153**
13 Vowel	0.0411	0.0494	0.0657	0.0385	0.0407	0.0477	0.0558	0.0370	0.0336	**0.0318**
14 Dermatology	0.0228	**0.0179**	0.0294	0.0254	0.0222	0.0205	0.0245	0.0215	0.0215	0.0182
15 Cryotherapy	0.1133	0.0270	0.0497	0.0499	0.0270	0.0276	0.0710	0.0913	0.0490	**0.0253**
16 M-of-n	0.0055	0.0046	0.0279	0.0204	0.0094	0.0069	0.0129	**0.0049**	0.1233	0.0051
17 kr-vs-kp	0.1801	0.1800	0.1809	0.1983	0.1861	0.1740	0.1950	0.1962	0.2137	**0.1629**
18 Optical recognition	**0.0122**	0.0139	0.0191	0.0147	0.0158	0.0183	0.0251	0.0161	0.0190	0.0136
19 Page blocks	0.0499	0.0448	0.0440	0.0468	0.0504	0.0511	0.0483	0.0489	0.0447	**0.0403**
20 Semeion	**0.0129**	0.0213	0.0260	0.014858	0.0239	0.0181	0.0302	0.0226	0.0234	0.0220
21 Pendigits	**0.0138**	0.0139	0.0154	0.0169	0.0140	0.0144	0.0149	0.0160	0.0259	0.0144
22 Movement_libras	0.1421	0.1499	0.2091	0.2036	0.2012	0.1539	0.1887	0.2049	0.1796	**0.1314**
23 arrhythmia	**0.3106**	0.3420	0.3233	0.3428	0.4058	0.3437	0.3925	0.3799	0.3959	0.3401
24 isolet5	**0.1450**	0.1991	0.1952	0.1779	0.1899	0.1605	0.1882	0.1906	0.2049	0.1651
25 pixraw10P	0.2345	0.2836	0.2048	0.1807	0.2227	0.2446	0.2206	0.2029	0.3482	**0.1633**
26 mturk	0.4044	0.3841	0.4248	0.4439	0.4142	0.3810	0.4313	0.3901	0.4587	**0.3623**
Average	0.1108	0.1100	0.1175	0.1166	0.1147	0.1061	0.1157	0.1155	0.1313	**0.0920**

Table 9
The results of the proposed and state-of-the-art algorithms' standard deviation (STD).

Dataset	Algorithm									
	BGA [43]	BPSO [45]	BWOA [23]	BWOASA [28]	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
1 Heart	0.0085	0.0101	0.0133	0.0117	0.0094	0.0158	0.0168	**0.0023**	0.0056	0.0083
2 Zoo	5.4847e−04	2.4328e−04	0.0035	0.0106	1.3690e−04	0.0039	0.0018	3.4007e−04	5.0000e−05	**1.0548e−05**
3 Breast_cancer	6.3965e−04	4.0826e−04	0.0025	0.0014	0.0015	1.9050e−04	0.0011	2.4166e−04	2.2222e−04	**1.8282e−04**
4 Glass_identification	1.7145e−04	4.0936e−04	4.3519e−04	3.4289e−04	8.6199e−04	2.0000e−04	1.9695e−04	0.0020	3.0000e−04	**1.0000e−04**
5 Australian	0.0046	0.0063	0.0323	0.0285	**5.5268e−04**	0.0108	0.0069	0.0014	0.0091	0.0045
6 spambase	0.0086	0.0028	0.0151	0.0071	0.0071	0.0049	0.0025	0.0015	**0.0014**	0.0031
7 EEG Eye State	0.0011	0.0015	0.0060	0.0118	9.9593e−04	**9.7810e−04**	0.0021	0.0025	0.0022	0.0018
8 Segment	0.0011	9.0169e−04	0.0063	0.0083	0.0019	0.0011	7.7223e−04	**2.3733e−04**	4.6306e−05	9.1212e−04
9 Waveform	0.0045	0.0049	0.0050	0.0067	0.0029	0.0033	0.0036	0.0045	0.0033	**0.0028**
10 Auto MPG	1.4286e−04	1.1158e−16	0.0055	0.0173	7.8019e−04	1.4286e−04	0.0030	2.7895e−17	2.5106e−16	**1.9739e−17**
11 House Voting	0.0038	0.0025	0.0039	0.0020	0.0012	6.7722e−04	0.0021	0.0014	0.0016	**1.1158e−16**
12 Wine	3.6507e−04	0.0049	0.0315	0.0221	0.0064	0.0070	0.0071	0.0041	0.0033	**2.6152e−17**
13 Vowel	0.0015	0.0030	0.0034	0.0067	0.0011	0.0041	0.0012	0.0012	2.5000e−04	**7.2473e−04**
14 Dermatology	0.0064	0.0048	0.0194	0.0151	0.0086	0.0012	0.0037	0.0011	**6.8804e−04**	0.0024
15 Cryotherapy	6.9739e−17	6.2765e−17	0.0043	0.0055	6.2765e−17	0.0063	6.9739e−17	1.3948e−16	6.9739e−17	**1.3948e−17**
16 M-of-n	0.0044	0.0139	0.0448	0.0499	0.0184	0.0179	0.0266	0.0044	0.0120	**0.0023**
17 kr-vs-kp	0.0059	0.0100	0.0041	0.0096	0.0114	0.0057	0.0026	5.5751e−04	0.0016	**2.7895e−16**
18 Optical recognition	0.0027	0.0018	0.0037	0.0043	0.0018	0.0033	0.0019	0.0020	0.0062	**9.7768e−04**

(continued on next page)

Table 9 (*continued*).

Dataset		Algorithm									
		BGA [43]	BPSO [45]	BWOA [23]	BWOASA [28]	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
19	Page blocks	2.0136e−04	6.4991e−05	0.0012	0.0013	9.3483e−05	3.5320e−04	6.3829e−05	2.8766e−04	**2.3956e−04**	3.2698e−04
20	Semeion	0.0028	0.0015	0.0025	0.0024	0.0023	0.0034	0.0012	0.0011	0.0060	**8.7931e−05**
21	Pendigits	2.6000e−04	5.2196e−04	0.0020	6.1602e−04	5.5680e−04	6.3477e−04	5.9333e−04	**5.6604e−05**	2.4575e−04	2.3527e−04
22	Movement_libras	0.0160	0.0086	0.0122	0.0153	0.0109	0.0052	0.0036	0.0036	0.0024	**0.0015**
23	arrhythmia	0.0339	0.0082	0.0218	0.0237	0.0159	0.0210	0.0076	0.0034	0.0045	**0.0016**
24	isolet5	0.0214	0.0082	0.0100	0.0120	0.0086	0.0145	0.0037	0.0053	**0.0027**	**0.0047**
25	pixraw10P	5.2430e−05	0.0054	0.0060	1.1680e−04	4.1843e−16	0.0049	3.5025e−04	6.6087e−06	2.7035e−04	**3.6043e−06**
26	mturk	0.0493	**0.0121**	0.0158	0.0376	0.0186	0.0264	0.0160	0.0285	0.0270	0.0142
	Average	0.0068	0.0043	0.0106	0.0119	0.0050	0.0063	0.0045	0.0027	**0.0026**	**0.0019**

Table 10
The results of selection size of the proposed and state-of-the-art algorithms.

Dataset		Algorithm									
		BGA [43]	BPSO [45]	BWOA [23]	BWOASA [28]	BDE [44]	BGWOPSO [46]	BHH [47]	BGWO [48]	BAOA [49]	BCOVIDOA
1	Heart	5	4	5	5	6	6	5	5	4	**3**
2	Zoo	5	6	7	10	7	6	8	7	6	**4**
3	Breast_cancer	**3**	5	8	5	7	7	4	5	5	5
4	Glass_identification	2	2	2	3	2	2	2	3	2	**2**
5	Australian	3	3	3	3	9	5	5	7	1	**1**
6	spambase	35	31	38	40	40	30	30	44	31	**29**
7	EEG Eye State	13	13	13	13	13	13	13	13	13	**13**
8	Segment	13	**5**	5	5	8	13	11	7	**4**	9
9	Waveform	**13**	15	17	16	16	16	14	18	16	15
10	Auto MPG	3	2	4	2	2	2	2	2	3	**2**
11	House Voting	3	8	6	3	9	5	4	8	5	**3**
12	Wine	5	4	5	5	4	5	6	7	**2**	4
13	Vowel	8	8	8	8	10	9	8	8	8	**8**
14	Dermatology	14	16	22	16	22	14	21	18	13	**13**
15	Cryotherapy	2	3	3	3	3	3	2	2	3	**2**
16	M-of-n	20	20	20	20	20	20	20	20	20	**20**
17	kr-vs-kp	13	22	17	17	17	15	23	23	15	**13**
18	Optical recognition	**27**	35	42	38	44	31	33	54	35	31
19	Page blocks	5	5	4	4	**3**	5	6	4	6	4
20	Semeion	125	133	136	146	197	125	156	195	258	**125**
21	Pendigits	13	13	11	12	13	12	12	12	12	**11**
22	Movement_libras	**29**	39	39	42	52	38	38	38	42	36
23	arrhythmia	98	140	73	73	234	128	156	194	232	**73**
24	isolet5	**218**	269	293	348	550	410	322	491	278	250
25	pixraw10P	**3890**	4757	4528	4455	4866	5616	3659	4906	4758	**2861**
26	Mturk	290	248	262	**267**	402	355	362	337	468	289
Average		186.7	223.3	214.2	213.8	252.2	265	189.30	247.23	240	**147.15**

Table 11
The results of Wilcoxon rank sum test.

Dataset		COVIDOA vs GA	COVIDOA vs DE	COVIDOA vs PSO	COVIDOA vs WOA	COVIDOA vs WOASA	COVIDOA vs GWOPSO	COVIDOA vs GWO	COVIDOA vs HH	COVIDOA Vs AOA
1	Heart	5.6032e−39	1.7658e−37	5.6003e−39	2.1981e−36	2.7477e−36	1.8002e−37	3.8011e−4	1.3865e−35	4.342e−33
2	Zoo	1.7534e−36	1.6259e−14	6.9608e−20	1.7181e−33	9.1823e−29	9.9344e−19	1.4233e−39	2.0497e−40	2.4533e−35
3	Breast_cancer	6.3499e−43	1.2756e−28	1.0760e−39	4.0981e−43	6.8100e−42	9.7131e−43	4.6912e−43	7.4605e−43	6.5322e−40
4	Glass_identification	2.3159e−44	2.6903e−42	3.5850e−44	5.4813e−44	3.3051e−26	1.5851e−40	2.5793e−34	3.5850e−44	1.6437e−30
5	Australian	2.1302e−43	8.5032e−35	4.1592e−23	3.9384e−17	1.1815e−34	4.2307e−30	1.9596e−18	9.7510e−15	7.3434e−22
6	Spambase	4.3465e−31	1.9975e−24	1.1097e−35	1.9636e−06	2.4298e−32	1.0498e−33	9.0754e−38	4.4347e−25	1.4765e−32
7	EEG Eye State	1.3910e−37	1.3954e−05	2.6110e−34	2.7204e−34	8.8026e−08	1.6882e−30	1.9302e−05	3.3901e−06	4.6433e−18
8	Segment	7.6710e−43	1.3937e−40	7.6484e−43	6.1152e−38	3.0861e−39	9.6344e−36	9.4052e−38	8.6619e−41	9.6023e−40
9	Waveform	7.6385e−31	4.7017e−35	5.5021e−31	3.8192e−15	4.5960e−31	2.1784e−32	2.9884e−29	1.9970e−40	2.4340e−25
10	Auto MPG	3.5216e−45	9.2924e−45	5.7625e−45	2.6851e−37	1.8127e−43	3.5216e−45	5.7625e−45	3.5216e−45	2.7850e−45
11	House Voting	3.4467e−40	8.6598e−41	3.9014e−40	7.8992e−40	2.4713e−39	5.9000e−44	4.8986e−37	3.5147e−41	6.3344e−39
12	Wine	4.8033e−42	2.6353e−43	9.2924e−45	9.8266e−41	2.0061e−42	1.4777e−44	3.6008e−40	1.6868e−40	5.3830e−40
13	Vowel	2.9574e−42	2.9557e−32	7.6711e−39	5.6250e−25	1.3419e−41	2.5666e−37	5.0621e−38	1.0285e−41	1.2673e−45
14	Dermatology	2.4166e−09	4.4232e−18	1.8017e−07	5.4397e−11	1.2786e−14	1.2684e−30	1.1505e−35	1.0661e−42	2.5742e−14
15	Cryotherapy	3.5216e−45	3.5216e−45	3.5216e−45	2.9642e−43	5.7625e−45	3.5216e−45	3.5216e−45	3.5216e−45	4.6472e−45
16	M-of-n	3.1186e−11	1.4396e−06	1.9623e−10	3.5325e−05	9.0593e−07	2.7821e−10	1.0333e−04	1.9571e−15	1.6467e−12
17	kr-vs-kp	1.1945e−40	1.2793e−39	8.3757e−40	4.6916e−39	4.6202e−40	4.1952e−39	4.5437e−39	9.9695e−40	1.3575e−40
18	Optical recognition	7.9569e−19	1.1762e−07	7.0097e−33	3.4097e−08	6.4914e−29	2.4700e−28	5.9524e−29	1.3792e−13	2.7345e−20
19	Page blocks	4.8485e−43	1.6000e−42	3.0580e−39	5.5883e−41	1.2629e−41	5.7625e−45	1.3127e−43	2.4083e−41	3.8567e−35
20	Semeion	1.4273e−37	6.0392e−37	8.5955e−27	1.2883e−36	1.7441e−36	5.2803e−42	1.9126e−39	6.9880e−33	6.7486e−32
21	Pendigits	1.5393e−40	1.1753e−38	4.5141e−38	7.2579e−38	1.2680e−25	6.8678e−40	6.6390e−24	7.9184e−31	9.3431e−33

(*continued on next page*)

Table 11 (*continued*).

Dataset		COVIDOA vs GA	COVIDOA vs DE	COVIDOA vs PSO	COVIDOA vs WOA	COVIDOA vs WOASA	COVIDOA vs GWOPSO	COVIDOA vs GWO	COVIDOA vs HH	COVIDOA Vs AOA
22	Movement_libras	6.2654e−27	4.5994e−35	6.9745e−39	1.7639e−35	1.1697e−35	2.2858e−34	1.1526e−34	4.3247e−39	4.6727e−35
23	Arrhythmia	4.5345e−32	5.3278e−31	8.3454e−28	4.6544e−25	1.3249e−38	5.3453e−40	6.3437e−35	5.3453e−31	1.6346e−32
24	isolet5	1.2343e−10	1.6489e−08	5.3475e−14	4.6479e−11	5.6324e−15	6.2140e−18	3.5359e−10	1.2536e−20	2.5362e−22
25	pixraw10P	1.3534e−28	4.6273e−24	5.3455e−12	6.3081e−32	1.3455e−35	5.2343e−34	2.1536e−33	3.5389e−25	2.3125e−28
26	Mturk	3.4537e−05	4.6471e−08	6.4340e−13	3.5467e−12	6.2138e−17	4.7543e−10	5.3572e−25	1.5366e−18	4.5920e−12

Future work may apply the binary COVIDOA with different classifiers such as support vector machines (SVM). Also, applying the proposed algorithm to solving real-world problems such as medical diagnoses, image processing, and industrial applications would be interesting. Another possible future work is hybridizing COVIDOA with another metaheuristic algorithm such as SA or PSO.

CRediT authorship contribution statement

Asmaa M. Khalid: Conceptualization, Methodology, Validation, Software, Writing – original draft. **Hanaa M. Hamza:** Validation, Software, Supervision. **Seyedali Mirjalili:** Conceptualization, Methodology, Writing – review & editing. **Khalid M. Hosny:** Conceptualization, Methodology, 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] L.P. Jing, H.K. Huang, H.B. Shi, Improved feature selection approach TFIDF in text mining, in: Proceedings. International Conference on Machine Learning and Cybernetics, Vol. 2, IEEE, 2002, pp. 944–946.

[2] K. Huang, S. Aviyente, Wavelet feature selection for image classification, IEEE Trans. Image Process. 17 (9) (2008) 1709–1720.

[3] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.

[4] S. Egea, A.R. Mañez, B. Carro, A. Sánchez-Esguevillas, J. Lloret, Intelligent IoT traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments, IEEE Internet Things J. 5 (3) (2017) 1616–1624.

[5] L. Zhao, X. Dong, An industrial Internet of Things feature selection method based on potential entropy evaluation criteria, IEEE Access 6 (2018) 4608–4617.

[6] P. Wongthongtham, J. Kaur, V. Potdar, A. Das, Big data challenges for the Internet of Things (IoT) paradigm, in: Z. Mahmood (Ed.), Connected Environments for the Internet of Things, in: Computer Communications and Networks, Springer, Cham, 2017, http://dx.doi.org/10.1007/978-3-319-70102-8_3.

[7] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (6) (2017) 1–45.

[8] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognit. Lett. 28 (13) (2007) 1825–1844.

[9] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.

[10] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (1–4) (1997) 131–156.

[11] X. Yu, Y. Chu, F. Jiang, Y. Guo, D. Gong, SVMs classification based two-side cross-domain collaborative filtering by inferring intrinsic user and item features, Knowl.-Based Syst. 141 (2018) 80–91.

[12] X. Yu, F. Jiang, J. Du, D. Gong, A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains, Pattern Recognit. 94 (2019) 96–109.

[13] L. Jourdan, M. Basseur, E.G. Talbi, Hybridizing exact methods and metaheuristics: A taxonomy, European J. Oper. Res. 199 (3) (2009) 620–629.

[14] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.

[15] D. Rodrigues, L.A. Pereira, T.N.S. Almeida, J.P. Papa, A.N. Souza, C.C. Ramos, X.S. Yang, BCS: A binary cuckoo search algorithm for feature selection, in: 2013 IEEE International Symposium on Circuits and Systems, ISCAS, IEEE, 2013, pp. 465–468.

[16] D. Rodrigues, X.S. Yang, A.N. De Souza, J.P. Papa, Binary flower pollination algorithm and its application to feature selection, in: Recent Advances in Swarm Intelligence and Evolutionary Computation, Springer, Cham, 2015, pp. 85–100.

[17] M.M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, S. Mirjalili, Binary dragonfly algorithm for feature selection, in: 2017 International Conference on New Trends in Computing Sciences, ICTCS, IEEE, 2017, pp. 12–17.

[18] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, European J. Oper. Res. 171 (3) (2006) 842–858.

[19] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, European J. Oper. Res. 171 (3) (2006) 842–858.

[20] M.M. Kabir, M. Shahjahan, K. Murase, A new local search based hybrid genetic algorithm for feature selection, Neurocomputing 74 (17) (2011) 2914–2928.

[21] X, Q. Zhang, N. Sun, Y. Dong, Feature selection with discrete binary differential evolution, in: 2009 International Conference on Artificial Intelligence and Computational Intelligence, Vol. 4, IEEE, 2009, pp. 327–330.

[22] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Amer. Statist. 46 (3) (1992) 175–185.

[23] M. Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP J. Image Video Process. 2013 (1) (2013) 1–8.

[24] S. Kashef, H. Nezamabadi-pour, An advanced ACO algorithm for feature subset selection, Neurocomputing 147 (2015) 271–279.

[25] A.G. Hussien, A.E. Hassanien, E.H. Houssein, S. Bhattacharyya, M. Amin, S-shaped binary whale optimization algorithm for feature selection, in: Recent Trends in Signal and Image Processing, Springer, Singapore, 2019, pp. 79–87.

[26] R.Y. Nakamura, L.A. Pereira, K.A. Costa, D. Rodrigues, J.P. Papa, X.S. Yang, BBA: A binary bat algorithm for feature selection, in: 2012 25th SIBGRAPI Conference on Graphics, Patterns, and Images, IEEE, 2012, pp. 291–297.

[27] J. Wu, Z. Lu, L. Jin, A novel hybrid genetic algorithm and simulated annealing for feature selection and kernel optimization in support vector regression, in: 2012 IEEE 13th International Conference on Information Reuse & Integration, IRI, IEEE, 2012, pp. 401–406.

[28] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.

[29] Q. Al-Tashi, S.J.A. Kadir, H.M. Rais, S. Mirjalili, H. Alhussian, Binary optimization using hybrid grey wolf optimization for feature selection, IEEE Access 7 (2019) 39496–39508.

[30] N.R. Eluri, G.R. Kancharla, S. Dara, V. Dondeti, Cancer data classification by quantum-inspired immune clone optimization-based optimal feature selection using gene expression data: deep learning approach, Data Technol. Appl. (2021).

[31] C.S.R. Annavarapu, S. Dara, Clustering-based hybrid feature selection approach for high dimensional microarray data, Chemometr. Intell. Lab. Syst. 213 (2021) 104305.

[32] Asmaa M. Khalid, Khalid M. Hosny, Seyedali Mirjalili, COVIDOA: A novel evolutionary optimization algorithm based on coronavirus replication lifecycle, Research Square 1 (2022) http://dx.doi.org/10.21203/rs.3.rs-1592094/v1.

[33] Y. Yamauchi, U.F. Greber, Principles of virus uncoating, cues and the snooker ball, Traffic 17 (6) (2016) 569–592.

[34] Jyoti Sharma, M. Keeling, M. Rowe, Pharmacological approaches for targeting cystic fibrosis nonsense mutations, Eur. J. Med. Chem. (2020).

[35] Y.M. Bar-On, A. Flamholz, R. Phillips, R. Milo, SARS-CoV-2 (COVID-19) By the Numbers, Science Forum, Elife, 2020.

[36] M.R. Pascual, Coronavirus SARS-CoV-2: Analysis of subgenomic mRNA transcription, 3CLpro, and PL2pro protease cleavage sites and protein synthesis, 2020, arXiv preprint arXiv, 2020.

[37] M.I. Khan, Z.A. Khan, M.H. Baig, I. Ahmad, A.E. Farouk, Y.G. Song, J.J. Dong, Comparative genome analysis of novel coronavirus (SARS-CoV-2) from different geographical locations and the effect of mutations on major target proteins, an in-silico insight, PLoS One 15 (9) (2020).

[38] M. Seyedali, L. Andrew, S-shaped versus V-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14.

[39] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.

[40] V. Losing, B. Hammer, H. Wersing, KNN classifier with self-adjusting memory for heterogeneous concept drift, in: 2016 IEEE 16th International Conference on Data Mining, ICDM, IEEE, 2016, pp. 291–300.

[41] H. Pei, J.-S. Pan, S.-C. Chu, Improved binary grey wolf optimizer and its application for feature selection, Knowledge-Based Systems (2020).

[42] C.L. Blake, CJ Merz UCI Repository of Machine Learning Databases Ph.D. dissertations, Dept. Inform. Comput. Sci. Univ. California, Irvine, CA, USA, 1998.

[43] M.M. Kabir, M. Shahjahan, K. Murase, A new local search based hybrid genetic algorithm for feature selection, Neurocomputing 74 (17) (2011) 2914–2928.

[44] X. He, Q. Zhang, N. Sun, Y. Dong, Feature selection with discrete binary differential evolution, in: 2009 International Conference on Artificial Intelligence and Computational Intelligence, 2009, pp. 327–330, http://dx.doi.org/10.1109/AICI.2009.438.

[45] R. Bello, Y. Gomez, A. Nowe, M.M. Garcia, Two-step particle swarm optimization to solve the feature selection problem, in: Seventh International Conference on Intelligent Systems Design and Applications, ISDA 2007, 2007, pp. 691–696, http://dx.doi.org/10.1109/ISDA.2007.101.

[46] Q. Al-Tashi, S.J. Abdul Kadir, H.M. Rais, S. Mirjalili, H. Alhussian, 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.

[47] Too, et al., A new quadratic binary harris hawk optimization for feature selection, Electronics 8 (10) (2019) 1130, http://dx.doi.org/10.3390/electronics8101130, MDPI AG.

[48] H. Hamouda, M. Mafarja, M. Alsawalqah, et al., Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification, Neural Computing and Applications 32 (2020) 12201–12220.

[49] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609.

[50] D. Szucs, J. Ioannidis, When null hypothesis significance testing is unsuitable for research: a reassessment, Front. Human Neurosci. 11 (390) (2017).

[51] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.