Improved feature selection using a hybrid side-blotched lizard algorithm and genetic algorithm approach

Amr Abdel-aal, Ibrahim El-Henawy
Department of Computer Science, Faculty of Computer and Informatics, Zagazig University, Alsharqia, Egypt

Article Info

Article history:

Received Nov 11, 2022
Revised Apr 16, 2023
Accepted Apr 24, 2023

Keywords:

Classification
Feature selection
Optimization
Side-blotched lizard algorithm
Transfer function

ABSTRACT

Feature selection entails choosing the significant features among a wide collection of original features that are essential for predicting test data using a classifier. Feature selection is commonly used in various applications, such as bioinformatics, data mining, and the analysis of written texts, where the dataset contains tens or hundreds of thousands of features, making it difficult to analyze such a large feature set. Removing irrelevant features improves the predictor performance, making it more accurate and cost-effective. In this research, a novel hybrid technique is presented for feature selection that aims to enhance classification accuracy. A hybrid binary version of side-blotched lizard algorithm (SBLA) with genetic algorithm (GA), namely SBLAGA, which combines the strengths of both algorithms is proposed. We use a sigmoid function to adapt the continuous variables values into a binary one, and evaluate our proposed algorithm on twenty-three standard benchmark datasets. Average classification accuracy, average number of selected features and average fitness value were the evaluation criteria. According to the experimental results, SBLAGA demonstrated superior performance compared to SBLA and GA with regards to these criteria. We further compare SBLAGA with four wrapper feature selection methods that are widely used in the literature, and find it to be more efficient.

Corresponding Author:

Amr Abdel-aal
Department of Computer Science, Faculty of Computer and Informatics, Zagazig University
Alsharqia, Egypt
Email: AMEbrahem@fci.zu.edu.eg

1. INTRODUCTION

Over the past few years, an enormous amount of data is available, resulting in an increased need to process this data to extract information and knowledge. This has made data mining a hot research topic [1]. One among the most popular data mining functions is classification where refers to assigning items in a collection into classes. Problems like dimensionality may reduce the classification accuracy [2]. High dimensional data sets with hundred or thousand features make it difficult for a model to interpret and understand [3]. Features selection preprocessing technique plays an important role in enhancing the dataset quality. Feature selection process gets rid of irrelevant features and keeps only the significant ones which results in a decrease in the total quantity of features in the dataset [4]. Feature selection is an important technique which leads to model interpretability, smaller training set, less training time, and minimizing overfitting. The two primary categories of feature selection techniques are as follows. First, filter methods that do not use any learning algorithm and depend on data properties [5]. Second, wrapper methods which employ learning-based algorithm [6] including but not limited to k-nearest neighbors (KNN) [7], neural networks [8], decision tree (DT) [9], and support vector machine (SVM) [10].

Optimization problems can be categorized based on the solution produced [11]: classical algorithms [12] and non-classical algorithms [13]. Classical search techniques divided into two categories: gradient based algorithms [14], which are used when the objective function has continuous derivatives, and direct search algorithms [15], which are used with partially continuous or non-differentiable objective function [16]. One of the main issues facing classical search methods is the vastness of the search space. Assuming that a dataset comprises k features, there will be 2^k potential solutions which requires high computational cost [17]. Metaheuristic approaches are considered useful for optimization problems since they can find good solutions with less computational power and time.

Usually, meta-heuristic optimization algorithms used alone or enhanced or hybrid with other algorithms to solve feature selection problem. For example, particle swarm optimization (PSO) [18], [19], which emulates the motion of bird flocks and schooling fish, GA [20], [21], which is inspired by the natural selection process, ant colony optimization (ACO) [22], [23], which emulates ants foraging behavior, cuckoo search (CS) [24], [25], which is inspired of the cuckoo search behavior and their reproduction strategy, bat algorithm (BA) [26], [27], which is inspired by the behavior of bats, firefly algorithm (FFA) [28], [29], which simulates the flashing behavior of fireflies during mating, grey wolf optimizer (GWO) [30], [31], which mimics the grey wolves hunting mechanism in nature, dragonfly algorithm (DA) [32], [33], which emulates the dragonflies behavior, flower pollination algorithm (FPA) [34], [35], which draws inspiration from the pollination behavior of flowers, ant lion optimizer (ALO) [36], [37], which mimics the antlions hunting mechanism in nature, whale optimization algorithm (WOA) [38], [39], which is modeled after the hunting behavior of humpback whales, salp swarm algorithm (SSA) [40], [41], which is based on the salps swarming mechanism, and henry gas solubility optimization (HGSO) [42], [43], which leverages Henry’s law of gas solubility in liquids to solve optimization problems.

A new meta-heuristic algorithm, called side-blotched lizard algorithm (SBLA) [44], has been proposed which emulate polymorphic population of the lizard. The experiments results showed the superiority of SBLA over some recent meta-heuristic algorithms in some engineering problems. Some issues such as stucking into local minima and achieving a proper trade-off between the exploration and exploitation faces SBLA as many metaheuristic algorithms. More modification and hybridization strategies are required to get better results. The main contributions of this work: i) we developed a binary form of SBLA by using sigmoid transfer function and ii) a hybrid method was introduced by combining the binary SBLA with GA then the experiments were performed in two phases: First, the hybrid method compared with SBLA and GA and the outcomes revealed the superiority of the hybrid method. Second, the hybrid method evaluated against a variety of well-known algorithms used in studies in the literature including HGSO, binary dragonfly algorithm (BDA), binary grey wolf optimizer (BGWO), and binary whale optimization algorithm (BWOA) and the outcomes demonstrated the advantages of the hybrid approach. The method applied in this study is presented in section 5738 while sections 3 and 4 give the results and conclusion, respectively.

2. METHOD

The SBLA with genetic algorithm (SBLAGA) approach for feature selection in machine learning is specifically tailored for classification tasks and involves the following steps as described in Figure 1. Firstly, the input data comprises a data set with an equal number of features (greater than one), a label with a non-negative value, and features that are characterized by real-valued numerical descriptions. Secondly, the input data is partitioned randomly into training and testing sets, with 80% of the data assigned to the training set and 20% assigned to the testing set during the holdout cross-validation phase. The SBLAGA algorithm is then employed for feature selection, with the KNN classifier used for each iteration of the algorithm. The aim of the optimization problem is to achieve optimal predictive performance while utilizing the fewest possible number of features, and the best individual is determined based on the value of the objective function, with the minimal value indicating the best individual. Lastly, the classifier’s performance is assessed.

Figure 2 and algorithm 1 are used to describe the proposed approach SBLAGA, respectively. The SBLAGA consists of several key stages, including the transformation function, initialization, KNN, and evaluation. These phases will be extensively covered in the upcoming subsections.

2.1. Transfer function

SBLA is typically employed for solving optimization problems that involve continuous variables, whereas feature selection is a type of optimization problem that deals with binary variables. Each lizard position should be transformed to its corresponding binary solution. To transform a continuous search space into a binary one, transfer functions are utilized for mapping purposes. S-shaped and V-shaped are the categories of transfer functions. The proposed approach uses the sigmoid function described in (1) which is an example of S-shaped transfer function.

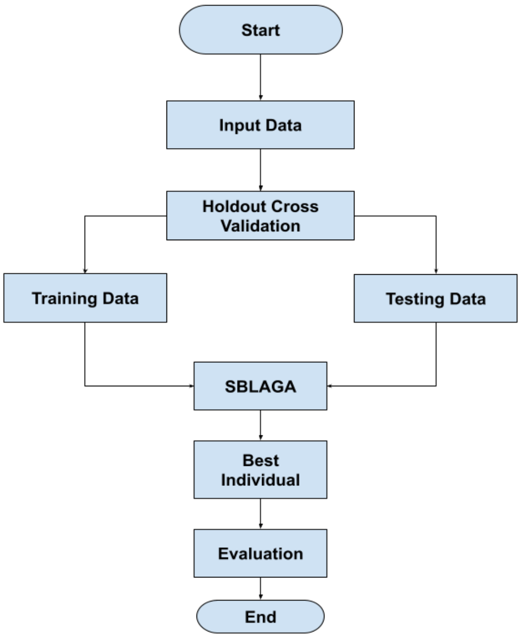

Figure 1. General framework of SBLAGA applied to the task of feature selection

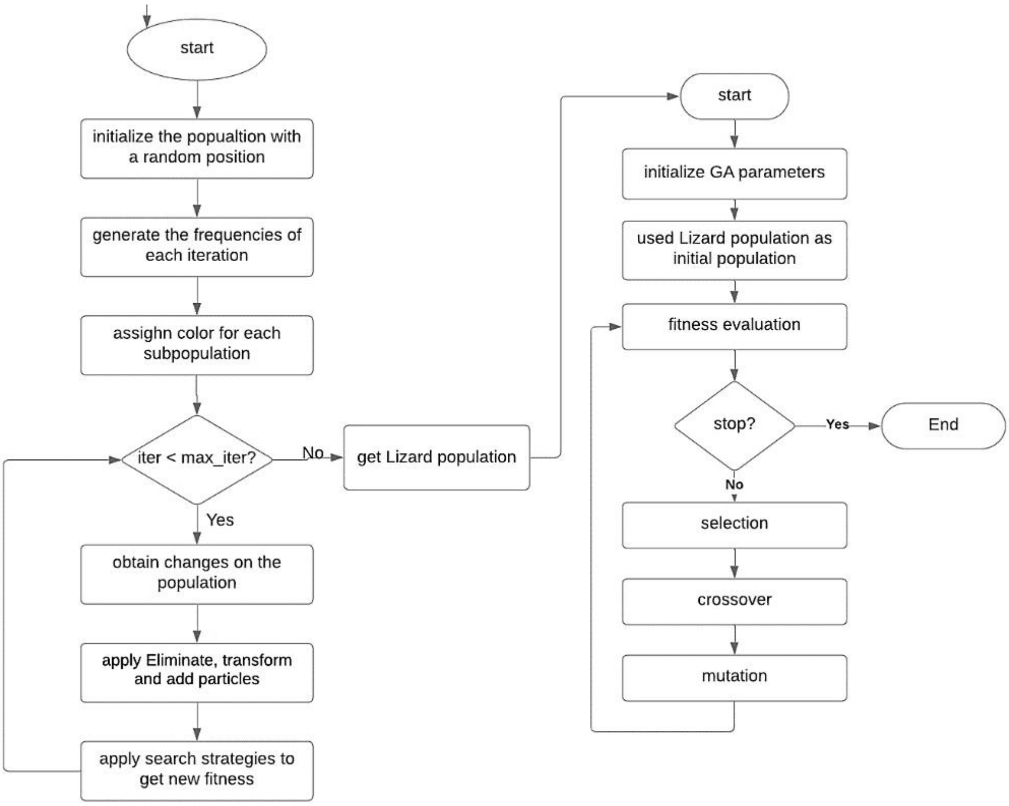
start
initialize the popualtion with a random position
generate the frequencies of each iteration
assighn color for each subpopulation
iter < max_iter?
No
get Lizard population
Yes
obtain changes on the population
apply Eliminate, transform and add particles
apply search strategies to get new fitness
start
initialize GA parameters
used Lizard population as initial population
fitness evaluation
stop?
Yes
End
No
selection
crossover
mutation

Figure 2. Proposed method abstraction

$$S\left(x_i^d(t)\right) = \frac{1}{1+e^{-10(x_i^d(t)-0.5)}} \tag{1}$$

where x_i^d represents the i^{th} lizard position in the d^{th} dimension at iteration number t, (1) is applied to determine x_i. The output of sigmoid function still continuous number ∈ [0, 1] so, the (2) is used to convert it to binary one.

$$x_i^d(t+1) = \begin{cases} 1, & r \geq S\left(x_i^d(t)\right) \\ 0, & r < S\left(x_i^d(t)\right) \end{cases} \tag{2}$$

where r is a value chosen at random from [0, 1].

Algorithm 1. Overview of SBLAGA in pseudo code

```
Set the parameters for the SBLA algorithm, including the maximum number of iterations
($max\_iter$) and the population size.
Initialize each lizard position in the population.
Transform each lizard position into binary.
Evaluate each lizard in the population using KNN classifier.
Generate every subpopulation size.
Assign color for each subpopulation.
Define $i \leftarrow 0$
While ($i < max\_iter$) do
    Get the current season.
    Calculate population changes.
    Apply eliminate, transform, and add particles functions depends on the current season
    and population changes.
    Apply defensive search strategy on blue lizards.
    Apply expansion search strategy on orange lizards.
    Apply sneaky search strategy on yellow lizards.
End
Use the returned lizards population as an input to GA.
Initialize the GA parameters: mutation rate, crossover rate and iterations number.
Evaluate each lizard in the population using the fitness function.
While (stopping criteria have not been met) do
    Choose two pairs of lizards using roulette wheel selection operator.
    Employ crossover operator with probability specified in crossover rate parameter.
    Employ mutation operator with probability specified in mutation rate parameter.
    Evaluate Offsprings.
    Update the population with the new offsprings.
    Apply fitness function to the new population.
End
Return the best solution in the population.
```

2.2. Initialization

The lizard population is initially created at random. Each lizard is represented by a vector of size d, where d denotes the size of the dataset's features. The vector's values are all either 1 or 0. 1 signifies that the feature has been selected, and 0 indicates that it has not been selected. As illustrated in Figure 3, five features are chosen while the rest are excluded.

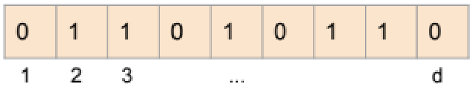

Figure 3. Binary possible solution

2.3. K-nearest neighbor (KNN)

KNN is among the most frequently utilized supervised machine learning techniques for classification tasks. KNN classifies a new data point based on the classification on k-neighbors, where k represents the maximum number of nearest neighbors to be considered. KNN is very simple, and extremely powerful. Figure 4 show an example of KNN.

Several techniques are available for computing the distance between a new data point and each of the training points. Among the most widely recognized methods are Euclidean, Manhattan, and Hamming distance. The method used in this paper is the Euclidean distance. The Euclidean distance can be computed by taking the square root of the sum of the squared differences between the new point (x) and the existing point. Euclidean distance is shown as:

$$\sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \tag{3}$$

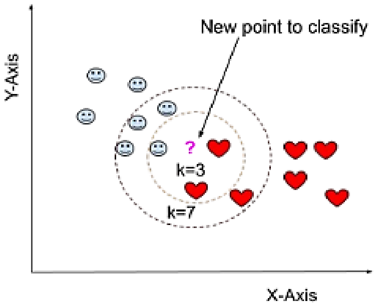

Figure 4. KNN example

2.4. Evaluation

Optimizing the classification accuracy and reducing the features number are the two objectives in solving feature selection problem. The evaluation function formulated in [37] simultaneously addresses the conflicting objectives as (4).

$$f = \alpha * Y_R(D) + \beta * \frac{|S|}{|N|} \tag{4}$$

where $\alpha \in [0,1]$, $\beta = (1-\alpha)$, $Y_R(D)$ represents the classification error rate, $|S|$ represents the number of selected features, and $|N|$ represents the total number of features in the dataset.

3. RESULTS AND DISCUSSION

3.1. Datasets

In this work, the experiments were conducted on a set of 23 benchmark datasets sourced from the UCI repository. Details regarding the number of features and instances in each dataset can be found in Table 1. It is worth noting that the datasets selected represent a diverse range of real-world problems from various domains such as healthcare and finance. Furthermore, the datasets have been extensively used in the literature for evaluating the effectiveness of various metaheuristic algorithms used to solve feature selection problem, which allows for a fair comparison of our approach against state-of-the-art techniques.

Table 1. Summary of the datasets used in the experiments

Dataset	Number of features	Number of instances
IonospereEW	34	351
BreastEW	30	569
carevaluation	6	1728
heartEW	13	270
lymphography	18	148
Parliment1984	16	435
wineEW	13	178
HeartFCR	12	299
WaveEW	40	5000
Glass-identification	10	214
m-of-n	13	1000
Sonar	60	208
Spect	44	267
Vehicle	18	846
Exactly	13	1000
Breastcancer	9	699
Exactly2	13	1000
Vote	16	300
Fri_c0_500_10	10	500
Fri_c0_1000_10	10	1000
Fri_c1_1000_10	10	1000
Fri_c1_1000_25	25	1000
Fri_c2_1000_25	25	1000

3.2. Parameter settings

Each dataset is divided into two sets; the first set is used as training set and represents 80% of the dataset and the second set used as test set and represents 20% of the dataset. This partitioning has been used in previous works by many researchers. KNN classifier is evaluated by using K-fold-cross-validation where the parameter K of KNN classifier is equals five as in [45]. For all experiments, the parameters were set as: a maximum of 200 iterations, a population size of 10, and a dimension corresponding to the number of features. The common parameters for all the algorithms are presented in Table 2. Each algorithm was executed 10 times with a random seed on a computer equipped with an Intel® Core™ i5-6500 processor with a clock speed of 3.20 GHz and 16 GB of RAM.

Table 2. Common parameters used in the experiments

Parameter	Value
K parameter for KNN classifier	5
∝ parameter for fitness function	0.99
ho parameter for partitioning	0.2
Dmax of BDA	6
a of BGWO	From 2 to 0
a of BWOA	From 2 to 0
b of BGWO	1
w1 of HGSO	0.99
w2 of HGSO	0.01
GA Crossover ratio	0.7
GA Mutation ratio	0.3

3.3. Comparison of SBLA, GA, and SBLAGA

In this section the performance of SBLA, GA and SBLAGA is outlined due to the average classification accuracy and average number of features selected. In the proposed method SBLAGA, GA begins to execute after SBLA terminates and the final solution from SBLA is used as initial solution for GA. Table 3 demonstrates the comparison among the three algorithms on 23 data sets. We notice that the proposed algorithm SBLAGA is better than both SBLA and GA in 19 datasets due to the average classification accuracy. It is important to note that there is a small discrepancy in the number of selected features between SBLA and SBLAGA, but the difference in average classification accuracy between the two is significant. Therefore, SBLAGA is still considered the better algorithm.

Table 3. Comparison between SBLA, GA and SBLAGA due to the average classification accuracy and the average number of the selected features

Dataset	Average Accuracy			Average number of features		
	SBLA	GA	SBLAGA	SBLA	GA	SBLAGA
IonospereEW	.957142	.918573	**.969997**	**1.8**	12.6	2.5
BreastEW	.964602	.962856	**.969912**	**2.5**	15.9	3.2
carevaluation	.9021745	**.94286**	.915219	**2**	3.8	2.5
heartEW	.877778	.848147	**.9**	**2.9**	5.5	3.4
lymphography	.92069	.920689	**.934483**	**3.1**	8.7	3.4
Parliment1984	.972416	.972416	**.982761**	**2.5**	7.4	3.3
wineEW	.980002	.960002	**.982858**	**1.8**	5.2	2.6
HeartFCR	.86441	.817142	**.877968**	**1.1**	5.1	1.3
WaveEW	.8214	**.8517**	.844	**18**	19.8	18.9
Glass-identification	.990327	.988095	**.992857**	**1.1**	4.9	1.2
m-of-n	.9345	.949999	**.9985**	**5**	6.6	5.2
Sonar	.929269	.95122	**.960976**	**3.6**	27.1	11.3
Spect	.907546	.920641	**.920756**	**4.4**	20.2	6.1
Vehicle	.978107	.967923	**.988165**	**2.8**	8	5.9
Exactly	.872	.799998	**.9895**	**4.5**	5.7	5.6
Breastcancer	.983454	.979243	**.985613**	**2.3**	4.4	2.6
Exactly2	.779	**.864148**	.784	**2.9**	6.3	4.2
Vote	.97833	.976666	**.986665**	**2**	6.9	2.3
Fri_c0_500_10	.882	.860001	**.888**	**1.6**	5.4	3.4
Fri_c0_1000_10	.8695	**.874997**	.8705	**2.1**	5.4	5.3
Fri_c1_1000_10	.9025	.849999	**.9235**	**1.8**	3.9	2.4
Fri_c1_1000_25	.8935	.824999	**.903**	**2.5**	9.2	3.1
Fri_c2_1000_25	.901	.855	**.904**	**1.8**	8	3

3.4. Comparison with other meta-heuristic-based approaches

This section objective is to compare the hybrid algorithm SBLAGA with other optimization algorithms. The algorithms used in the comparison are popular population-based algorithms commonly utilized for feature selection problem: BGWO, BDA, HGSO, and BWOA. These algorithms were compared and tested on the data sets mentioned previously in Table 1 and the performance indicators are average classification accuracy, average fitness value and average number of features selected. The average classification accuracy and average fitness values of all approaches are presented in Tables 4 and 5 respectively. We notice that SBLAGA obtained the highest average of fitness values and classification accuracy in 18 datasets while HGSO is superior in 3 datasets and BDA is superior in 2 datasets. Also, standard deviation in Tables 4 and 5 refers that SBLAGA behaves more robust than the other algorithms on almost the data sets. Table 6 presents the average number of selected features. We notice that SBLAGA algorithm obtained the highest average of selected features in all the data sets.

Table 1. Comparison between BGWO, BDA, HGSO, BWOA, and SBLAGA due to the average classification accuracy (AvgAc)

Dataset	BGWO		BDA		HGSO		BWOA		SBLAGA	
	AvgAc	StndDev	AvgAc	StndDev	AvgAc	StndDev	AvgAc	StndDev	AvgAc	StndDev
IonospereEW	.916214	.0225	.94467	.0191	.96	.0154	.922858	.0301	**.969997**	**.0135**
BreastEW	.939823	.01716	.969489	.0119	.961062	.0126	.947788	.01396	**.969912**	**.0090**
carevaluation	.898696	**.0178**	.908038	.0182	**.92087**	.0234	.899130	.0232	.915219	.0187
heartEW	.848148	.0437	.882795	.0226	.877778	**.0222**	.846296	0.0454	**.9**	.0277
lymphography	.875862	.0493	.931638	.0461	.92069	.0379	.889655	.0402	**.934483**	**.0154**
Parliment1984	.965517	.0170	.974429	.0140	.973563	**.008977**	.957471	.0136	**.982761**	.0106
wineEW	.96	.0229	.985686	**.0138**	.977143	.0214	.962857	.0257	**.985715**	.0189
HeartFCR	.845763	.0544	.863763	.0337	.857627	.0493	.849152	.0464	**.877968**	**.0249**
WaveEW	.8392	.0248	.838277	.0082	.8306	.0059	.836	**.0079**	**.844**	.0083
Glass-identification	.983393	.0214	**.995843**	**.0074**	.992857	.0109	.992857	.0109	.992857	.0095
m-of-n	.956	.0312	.995385	**.0000**	.9875	.0118	.947	.0442	**.9985**	.0045
Sonar	.9248	.0226	.9463	.0229	.94878	.023	.902439	.03778	**.960976**	**.0223**
Spect	.909433	.0313	.924837	.0249	.90943	.0184	.875471	.02947	**.926417**	**.0132**
Vehicle	.976331	.0095	.981552	**.0055**	.985799	0.0065	.96804	.0103	**.988165**	.0083
Exactly	.766	.0427	**.995385**	**.0000**	.906	.0749	.8515	.1161	.9895	.0282
Breastcancer	.98489	.0087	.978194	.0098	.978417	.012	.98273	.0092	**.985613**	**.0071**
Exactly2	.7775	.0131	.780093	**.0057**	.781	.0073	.7695	.0106	**.784**	.008
Vote	.96666	.0223	.985012	.0117	.98	.0163	.961666	.0198	**.986665**	**.01**
Fri-c0-500-10	.86	.0309	.88006	.0194	.876	**.0143**	.867	.0261	**.888**	.0198
Fri-c0-1000-10	.85	.0224	.872335	.0135	**.88**	**.0097**	.862	.0148	.8705	.0166
Fri-c1-1000-10	.8855	.0211	.917395	.015	.9095	.0134	.8865	.0268	**.9235**	**.0131**
Fri-c1-1000-25	.776	.0211	.87666	.0412	.875	.0241	.8465	.0429	**.903**	**.0122**
Fri-c2-1000-25	.804	.0263	.90711	.0146	**.9095**	.0113	.8605	.0381	.904	**.0076**

Table 5. Comparison between BGWO, BDA, HGSO, BWOA, and SBLAGA due to the average fitness value (AvgFit)

Dataset	BGWO		BDA		HGSO		BWOA		SBLAGA	
	Avgfit	StndDev	Avgfit	StndDev	Avgfit	StndDev	Avgfit	StndDev	Avgfit	StndDev
IonospereEW	.084285	.0225	.055337	.0191	.039429	.0154	.0668	.0301	**.03**	**.0135**
BreastEW	.0602	.01716	.02986	.0119	.03893	.0126	.0522	.01396	**.03**	**.0090**
carevaluation	.1013	**.0178**	.0919	.0182	**.0791**	.0234	.1008	.0232	.0930	.0187
heartEW	.1518	.0437	.1172	.0226	.1222	**.0222**	.1537	.0454	**.0999**	.0277
lymphography	.1241	.0493	.0683	.0461	.0793	.0379	.1103	.0402	**.0655**	**.0154**
Parliment1984	.0345	.0170	.0256	.0140	.0264	**.008977**	.0425	.0136	**.0172**	.0106
wineEW	.039998	.0229	.0143	**.0138**	.0228	.0214	.0371	.0257	**.0142**	.0189
HeartFCR	.1542	.0544	.1362	.0337	.1424	.0493	.1508	.0464	**.1220**	**.0249**
WaveEW	.1608	.0248	.1617	.0082	.1694	.0059	.164	**.0079**	**.156**	.0083
Glass-identification	.0166	.0214	**.0041**	**.0074**	.0071	.0109	.0071	.0109	.0071	.0095
m-of-n	.044	.0312	.004615	**.0000**	.0125	.0118	.053	.0442	**.0015**	.0045
Sonar	.0752	.0226	.0537	.0229	.05122	.023	.09756	.03778	**.03902**	**.0223**
Spect	.090567	.0313	.07516	.0249	.09057	.0184	.12453	.02947	**.07358**	**.0132**
Vehicle	.023669	.0095	.01844	**.0055**	.0142	.0065	.03196	.0103	**.011835**	.0083
Exactly	.234	.0427	**.004615**	**.0000**	.094	.0749	.1484	.1161	.0105	.0282
Breastcancer	.01511	.0087	.021806	.0098	.02158	.012	.01727	.0092	**.01438**	**.0071**
Exactly2	.2225	.0131	.219907	**.0057**	.219	.0073	.2305	.0106	**.216**	.008
Vote	.03334	.0223	.014988	.0117	.02	.0163	.0383	.0198	**.013335**	**.01**
Fri-c0-500-10	.14	.0309	.11994	.0194	.124	**.0143**	.133	.0261	**.112**	.0198
Fri-c0-1000-10	.15	.0224	.127665	.0135	**.12**	**.0097**	.138	.0148	.1295	.0166
Fri-c1-1000-10	.1145	.0211	.082605	.015	.0905	.0134	.1135	.0268	**.0765**	**.0131**
Fri-c1-1000-25	.224	.0211	.12334	.0412	.125	.0241	.1535	.0429	**.097**	**.0122**
Fri-c2-1000-25	.196	.0263	.09289	.0146	**.0905**	.0113	.1395	.0381	.096	**.0076**

Table 6. Comparison between BGWO, BDA, HGSO, BWOA, and SBLAGA due to the average number of selected features (AvgNf)

Dataset	BGWO		BDA		HGSO		BWOA		SBLAGA	
	AvgNf	StndDev	AvgNf	StndDev	AvgNf	StndDev	AvgNf	StndDev	AvgNf	StndDev
IonospereEW	20.5	2.8017	9.5	3.0083	5.1	1.57797	7.6	5.1807	**2.5**	**1.5**
BreastEW	15.6	2.4166	4.6	1.6852	8.5	4.2953	14.4	6.4218	**3.2**	**1.1661**
carevaluation	4.7	**.4582**	4.3	**.4582**	4.3	**.4582**	4.7	.6403	**2.5**	2.1095
heartEW	7	1.2649	4.5	1.02469	5	**1.3416**	5	1.8973	**3.4**	1.7435
lymphography	11.3	2.0024	5.7	**1.3453**	6.5	3.0741	10.8	3.5721	**3.4**	2.6153
Parliment1984	8.8	1.077	6	1.4142	6.1	2.2113	6	2.236	**3.3**	**.781**
wineEW	6.9	1.044	4.7	.9	5.2	2.0396	6	1.7888	**2.6**	**.8**
HeartFCR	6.7	1.6763	1.7	**.4582**	4.6	2.0099	4.3	1.6155	1.3	**.5482**
WaveEW	31.5	**2.0124**	23.3	2.2825	29.4	2.2	33.7	3.0016	**18.9**	5.0685
Glass-identification	6	**1.2649**	1.7	.4582	5.7	1.6155	4.5	1.3601	1.2	**.4**
m-of-n	9	1.8439	6	**.0000**	7.2	.74833	9.4	2.2	**5.2**	2.4413
Sonar	34.8	**4.5782**	17.9	4.5923	16.1	6.09015	25.3	11.9084	**11.3**	7.7980
Spect	26.4	3.826	18.4	4.0049	8.1	3.3	17.2	7.97245	**6.1**	**3.14**
Vehicle	11.7	1.9	7.9	**1.4456**	8	2.2	9.5	3.20156	**5.9**	2.8
Exactly	9.9	1.6401	6	**.0000**	7.7	.9	8.9	1.86815	**5.6**	1.9595
Breastcancer	5.6	1.2	3.6	1.0198	4.7	1.7916	5.1	1.44568	**2.6**	**.9165**
Exactly2	7.4	1.8	6.6	**1.3564**	7.6	2.1541	7	3.0983	**4.2**	2.856
Vote	9	2.4899	5.5	**2.3345**	4.8	2.5219	7.1	3.0479	**2.3**	1.4177
Fri-c0-500-10	7.2	1.2489	5.1	**.8306**	6.1	1.5779	5.6	1.9596	**3.4**	1.562
Fri-c0-1000-10	6.6	.9165	5.4	.9165	5.9	1.3	6.8	1.4	**5.3**	**.78102**
Fri-c1-1000-10	5.7	.781	3.9	.8306	3.4	**.4899**	4	1	**2.4**	1.2806
Fri-c1-1000-25	14.3	2.0025	6.4	2.1071	3.7	**1.1874**	4.1	1.2206	**3.1**	1.5779
Fri-c2-1000-25	13	3.2249	4.4	.4899	3.5	.8062	4.4	4.0299	**3.1**	**.8**

4. CONCLUSION

In this work, SBLAGA was introduced as a hybrid feature selection approach. Twenty-three bench-mark data sets from the UCI repository were collected to investigate the performance of the proposed approach with GA and the original SBLA. The experimental results indicate that the SBLAGA approach outperformed both GA and SBLA in terms of average classification accuracy. SBLAGA then compared with recent well-known meta-heuristic algorithms used to solve feature selection problem including BGWO, BDA, HGSO, and BWOA. The experiments were conducted on the same datasets, measuring average classification accuracy, fitness value, and number of selected features. SBLAGA outperformed the four recent well-known algorithms in terms of these metrics. In future studies, a potential direction for improvement would be to parallelize the algorithm, particularly for handling high-dimensional datasets, in order to reduce the computation time. Other classification algorithms such as neural network and SVM can be used to investigate the proposed algorithm. Real world problems like spam email detection and medical diagnosis can be investigated using the proposed approach.

REFERENCES

[1] J. Han, M. Kamber, and P. Jian, *Data mining: concepts and techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 2011.

[2] G. I. Sayed, A. Tharwat, and A. E. Hassanien, "Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection," *Applied Intelligence*, vol. 49, no. 1, pp. 188–205, Jan. 2019, doi: 10.1007/s10489-018-1261-8.

[3] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 593–637, Jan. 2021, doi: 10.1007/s10462-020-09860-3.

[4] M. Abdel-Basset, R. Mohamed, R. K. Chakrabortty, M. J. Ryan, and S. Mirjalili, "An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection," *Computers & Industrial Engineering*, vol. 153, Mar. 2021, doi: 10.1016/j.cie.2020.107078.

[5] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 70–76, Feb. 2007, doi: 10.1109/TSMCB.2006.883267.

[6] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997, doi: 10.1016/S0004-3702(97)00043-X.

[7] S. R. Amendolia, G. Cossu, M. L. Ganadu, B. Golosio, G. L. Masala, and G. M. Mura, "A comparative study of k-nearest neighbour, support vector machine and multi-layer perceptron for thalassemia screening," *Chemometrics and Intelligent Laboratory Systems*, vol. 69, no. 1–2, pp. 13–20, Nov. 2003, doi: 10.1016/S0169-7439(03)00094-7.

[8] Z.-B. Xu, H. Qiao, J. Peng, and B. Zhang, "A comparative study of two modeling approaches in neural networks," *Neural Networks*, vol. 17, no. 1, pp. 73–85, Jan. 2004, doi: 10.1016/S0893-6080(03)00192-8.

[9] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree ID3 and C4.5," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, 2014, doi: 10.14569/SpecialIssue.2014.040203.

[10] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, "A comparative study of support vector machine and naive bayes classifier for sentiment analysis on amazon product reviews," in *2020 International Conference on Contemporary Computing and Applications (IC3A)*, Feb. 2020, pp. 217–220, doi: 10.1109/IC3A48958.2020.233300.

[11] X.-S. Yang, *Engineering optimization*. Wiley, 2010.

[12] N. Pham, A. Malinowski, and T. Bartczak, “Comparative study of derivative free optimization algorithms,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 592–600, Nov. 2011, doi: 10.1109/TII.2011.2166799.

[13] M. Francisco, S. Revollar, P. Vega, and R. Lamanna, “A comparative study of deterministic and stochastic optimization methods for integrated design of processes,” *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 335–340, 2005, doi: 10.3182/20050703-6-CZ-1902.00917.

[14] J. Creswell, “Qualitative, quantitative and mixed methods approaches,” *Sage*, 2014.

[15] T. G. Kolda, R. M. Lewis, and V. Torczon, “Optimization by direct search: new perspectives on some classical and modern methods,” *SIAM Review*, vol. 45, no. 3, pp. 385–482, Jan. 2003, doi: 10.1137/S003614450242889.

[16] M. O. Okwu and L. K. Tartibu, “Future of nature inspired algorithm, swarm and computational intelligence,” in *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*, Springer International Publishing, 2021, pp. 147–151.

[17] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3, pp. 1157–1182, 2003.

[18] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[19] R. Bello, Y. Gomez, A. Nowe, and 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)*, Oct. 2007, pp. 691–696, doi: 10.1109/ISDA.2007.101.

[20] J. H. Holland, “Genetic algorithms,” *Scientific American*, vol. 267, no. 1, pp. 66–72, Jul. 1992, doi: 10.1038/scientificamerican0792-66.

[21] M. M. Kabir, M. Shahjahan, and K. Murase, “A new local search based hybrid genetic algorithm for feature selection,” *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, Oct. 2011, doi: 10.1016/j.neucom.2011.03.034.

[22] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/CI-M.2006.248054.

[23] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, “MLACO: A multi-label feature selection algorithm based on ant colony optimization,” *Knowledge-Based Systems*, vol. 192, Mar. 2020, doi: 10.1016/j.knosys.2019.105285.

[24] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, Jan. 2013, doi: 10.1007/s00366-011-0241-y.

[25] D. Rodrigues *et al.*, “BCS: A binary cuckoo search algorithm for feature selection,” in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, May 2013, pp. 465–468, doi: 10.1109/ISCAS.2013.6571881.

[26] X.-S. Yang and A. Hossein Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” *Engineering Computations*, vol. 29, no. 5, pp. 464–483, Jul. 2012, doi: 10.1108/02644401211235834.

[27] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, “BBA: A binary bat algorithm for feature selection,” in *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug. 2012, pp. 291–297, doi: 10.1109/SIBGRAPI.2012.47.

[28] X. S. Yang and X. He, “Firefly algorithm: recent advances and applications,” *International Journal of Swarm Intelligence*, vol. 1, no. 1, 2013, doi: 10.1504/IJSI.2013.055801.

[29] Y. Zhang, X. Song, and D. Gong, “A return-cost-based binary firefly algorithm for feature selection,” *Information Sciences*, vol. 418–419, pp. 561–574, Dec. 2017, doi: 10.1016/j.ins.2017.08.047.

[30] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[31] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque, and S. Mirjalili, “A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection,” *Expert Systems with Applications*, vol. 139, Jan. 2020, doi: 10.1016/j.eswa.2019.112824.

[32] S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, May 2016, doi: 10.1007/s00521-015-1920-1.

[33] A. I. Hammouri, M. Mafarja, M. A. Al-Betar, M. A. Awadallah, and I. Abu-Doush, “An improved dragonfly algorithm for feature selection,” *Knowledge-Based Systems*, vol. 203, Sep. 2020, doi: 10.1016/j.knosys.2020.106131.

[34] X.-S. Yang, “Flower pollination algorithm for global optimization,” in *Unconventional Computation and Natural Computation*, Springer Berlin Heidelberg, 2012, pp. 240–249.

[35] D. Rodrigues, X.-S. Yang, A. N. de Souza, and J. P. Papa, “Binary flower pollination algorithm and its application to feature selection,” in *Studies in Computational Intelligence*, Springer International Publishing, 2015, pp. 85–100.

[36] S. Mirjalili, “The ant lion optimizer,” *Advances in Engineering Software*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[37] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary ant lion approaches for feature selection,” *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016, doi: 10.1016/j.neucom.2016.03.101.

[38] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[39] M. M. Mafarja and S. Mirjalili, “Hybrid whale optimization algorithm with simulated annealing for feature selection,” *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017, doi: 10.1016/j.neucom.2017.04.053.

[40] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, “Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,” *Advances in Engineering Software*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[41] M. Tubishat *et al.*, “Dynamic salp swarm algorithm for feature selection,” *Expert Systems with Applications*, vol. 164, Feb. 2021, doi: 10.1016/j.eswa.2020.113873.

[42] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, “Henry gas solubility optimization: A novel physics-based algorithm,” *Future Generation Computer Systems*, vol. 101, pp. 646–667, Dec. 2019, doi: 10.1016/j.future.2019.07.015.

[43] N. Neggaz, E. H. Houssein, and K. Hussain, “An efficient henry gas solubility optimization for feature selection,” *Expert Systems with Applications*, vol. 152, Aug. 2020, doi: 10.1016/j.eswa.2020.113364.

[44] O. Maciel C., E. Cuevas, M. A. Navarro, D. Zaldívar, and S. Hinojosa, “Side-blotched lizard algorithm: a polymorphic population approach,” *Applied Soft Computing*, vol. 88, Mar. 2020, doi: 10.1016/j.asoc.2019.106039.

[45] H. Chantar, M. Tubishat, M. Essgaer, and S. Mirjalili, “Hybrid binary dragonfly algorithm with simulated annealing for feature selection,” *SN Computer Science*, vol. 2, no. 4, Jul. 2021, doi: 10.1007/s42979-021-00687-5.

BIOGRAPHIES OF AUTHORS

Amr Abdel-aal received the bachelor's degree from Zagazig University, in 2017. He is currently pursuing the master's degree with the Faculty of Computer and Informatics, Zagazig University, Egypt. His research interests include multi-objective optimization, evolutionary algorithms, computational intelligence, and natural language processing. He can be contacted at email: AMEbrahem@fci.zu.edu.eg.

Ibrahim El-Henawy received the M.S. and Ph.D. degrees in computer science from State University of New York, USA in 1980 and 1983, respectively. Currently, he is a professor in computer science department, Zagazig University. His current research interests are mathematics, networks, artificial intelligence, optimization, digital image processing, and pattern recognition. He can be contacted at email: ielhenawy@zu.edu.eg.