



A Novel Binary Drawer Algorithm for Feature Selection in AI Application

Hasan Azeez^{1*}

¹*Department of Computer Engineering, Azad University, South Tehran Branch, College of Engineering, Iraq*

* Corresponding author's Email: mschasanazeez@gmail.com

Abstract: In artificial intelligence (AI) applications, to make informed decisions, relevant data must be gathered from vast databases. Choosing only the relevant and desirable characteristics would have a significant impact on the accuracy of the model. The primary goal of feature selection is to remove unneeded features, reducing complexity. This paper presents Binary Drawer Algorithm (BinDA) for feature selection. The Drawer Algorithm (DA) is a novel metaheuristic algorithm inspired by the process of selecting objects from several drawers to build an optimal combination. The standard DA has been enhanced with major features to increase its overall performance. The local search algorithm is a novel addition to the DA algorithm that improves its exploitation capacity. In order to determine how well the algorithm works, it is compared to others and tested on a collection of 20 datasets. The proposed BinDA is assessed in contrast to 7 modern wrapper feature selection techniques. The results show that the proposed BinDA algorithm regularly performs better than existing algorithms.

Keywords: Artificial intelligence, Drawer algorithm (DA), Metaheuristic, Local search algorithm, Feature selection.

1. Introduction

The explosive development of computer technologies has led to the generation of a huge amount of data with several features. The standard DA has been enhanced with major features to increase its overall performance. The local search algorithm is a novel addition to the DA algorithm that improves its exploitation capacity [1]. In the AI model, the existence of redundant, irrelevant, and chaotic records in high-dimensional datasets affects classification accuracy and increases complexity [2,3]. Feature selection is an essential component of data preparation, playing a substantial part in constructing resilient models [4, 5]. The use of metaheuristics in solving feature selection problems is dependent on their capacity to produce a solution [6, 7]. Metaheuristics display considerable adaptability when utilized in certain problem domains due to their intrinsic simplicity and straightforward implementation. Metaheuristic are categorized into four types: swarm intelligence [8], human-based methods [9], physics-based methods [10], and evolutionary algorithms [11]. Animals in swarms

demonstrate collective behavior, which influences swarm intelligence techniques. The application of this specific metaheuristic method has made major contributions to efficiently addressing feature selection difficulties.

The algorithms considered are the Binary Flower Pollination Algorithm (BFPA) [12], Binary Horse Herd Optimization (BinHOA) [13], Binary Dragonfly algorithm (BDA) [14], Binary Cuckoo Search (BCS) [15], and Particle Swarm Optimization (PSO) [16]. Human-based methods are based on how people act and connect with each other in society. In his work on knowledge-based gains sharing (GSK), Prachi [17] presented a new binary version. Imperial competition algorithms are one kind of algorithm that relies on human-like methods (ICA) [18], the cultural evolution algorithm [19, 20] and teaching learning-based optimization (TLBO) [21]. From these algorithms, lightning search methods [22], multi-verse methods [23], Henry gas solubility method [24], and gravitational search methods [25, 26]. An improved equilibrium optimizer was presented by Shameem et al. [27]. A binary optimizer called BinEO was introduced by Mohamed Mostafa Saleh

et al. [28]. The genetic algorithm is a type of evolutionary method [29]. Alternative evolutionary approaches include differential evolution algorithms [30, 31] and stochastic fractal search [32]. In addition to, extended stochastic coati optimizer introduced by P. D. Kusuma [33]. The result presents the superiority of ESCO among five shortcoming metaheuristics in solving optimization problem. Rami S. present Model of Whale Optimization Algorithm and K-nearest Neighbors [34]. This novel approach is more adept at steering clear of local optima. The experimental results affirm the superiority of the model over other optimization such as classification accuracy, fitness value, and average feature. Sajjad A. D. proposes Ring Toss Game-Based Optimization Algorithm for Solving Various Optimization Problems [35]. The main idea of Ring Toss Game-Based Optimization (RTGBO) is to simulate the behaviour of players and rules of the ring toss game in the design of the proposed algorithm the results of optimizing the multi-model type objective functions indicate the acceptable exploration ability of RTGBO. The results also confirm the superiority of the proposed RTGBO algorithm over mentioned optimization techniques. P. D. Kusuma present a new metaphor-free metaheuristic search called the swarm bipolar algorithm (SBA) [36]. SBA is developed mainly based on the non-free-lunch (NFL) doctrine, which mentions the non-existence of any general optimizer appropriate to answer all varieties of problems. The construction of SBA is based on splitting the swarm into two equal-sized swarms to diversify the searching process while performing intensification within the sub swarms. The result shows that the search toward the middle between the two finest sub-swarm members is the best among the four searches in SBA.

The Drawer Algorithm (DA) [37] is a recently developed metaheuristic algorithm that draws inspiration and simulates the selection of objects from several drawers in order to achieve an optimal combination. The Data Analysis (DA) assumes the presence of multiple drawers, each of which holds a specific quantity of objects. To form a correct combination of goods inside the drawers, it is necessary to select one item from each drawer. Assembling the suitable items from the drawers and combining them is an optimization procedure that can serve as a source of inspiration for algorithm creation. Metaheuristics have had a beneficial impact on feature selection problems in recent times, as previously stated. Further enhanced outcomes can be attained by implementing further optimization procedures. Despite thorough investigation, a considerable amount of metaheuristics still face

numerous obstacles that require resolution. Despite thorough investigation, a substantial amount of metaheuristics still face numerous obstacles that require solutions. This motivated us to establish an improved binary version of DA as a binary optimization method for feature selection problems. Our contributions consist of the following:

- BinDA: A binary improved version of the DA optimizer is introduced to overcome the feature selection problem.

- The local search algorithm is implemented to improve the DA's exploitation capacity. It combined with DA to prevent the occurrence of local optima.

The rest of the research is organised like this: The Drawer algorithm (DA) is introduced in Section 2, the proposed BinDA algorithm in Section 3, the experiments and discussion in Section 4, and finally, the findings in Section 5.

2. Drawer algorithm

The DA is an iterative solution to optimization issues that is based on metaheuristics. In order to find the best solution, the DA's population members traverse the problem's search space throughout each iteration. The population can be represented mathematically as:

$$D = [\vec{D}_1, \vec{D}_2, \dots, \vec{D}_N] = \begin{bmatrix} d_{1,1} & \dots & d_{1,j} & \dots & d_{1,dim} \\ \vdots & \dots & \vdots & \dots & \vdots \\ d_{i,1} & \dots & d_{i,j} & \dots & d_{i,dim} \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ d_{N,1} & \dots & d_{N,j} & \dots & d_{N,dim} \end{bmatrix} \quad (1)$$

where D indicates the population matrix of the DA, N indicates the number of population members, dim is the number of variables, $i = 1, \dots, N$ being the i-th solution, and $d_{i,j}$ is a j-th component.

In the initial stage, the entire population must be randomly initialized through a process of random assignment.

$$D_{i,j} = lb_j + rand \times (ub_j - lb_j), \quad i = 1, \dots, N \text{ and } j = 1, \dots, dim. \quad (2)$$

The function rand creates a random integer uniformly in the interval [0, 1]. The lower and upper limits of the j-th variable are represented by lb_j and ub_j , respectively. The function rand creates a random integer uniformly in the interval [0, 1]. The lower and upper limits of the j-th variable are represented by lb_j and ub_j , respectively.

The proposed solution algorithm's population determines the evaluation of the objective function Fun, which has dim variables. The objective function values are displayed by a provided statement.

$$\vec{Fun} = \begin{bmatrix} \vec{Fun}_1 \\ \vdots \\ \vec{Fun}_i \\ \vdots \\ \vec{Fun}_N \end{bmatrix} = \begin{bmatrix} Fun(\vec{D}_1) \\ \vdots \\ Fun(\vec{D}_i) \\ \vdots \\ Fun(\vec{D}_N) \end{bmatrix} \quad (3)$$

The vector Fun represents the objective function values acquired.

$Fun_i = Fun(D_{i,1}, \dots, D_{i,j}, \dots, D_{i,dim})$, with $i = 1 \dots N$.

More precisely, the DA presupposes that there is a commode with a number of drawers equal to the number of variables in the optimization problem. Every compartment within the commode houses is distinct and recommends the values of the corresponding variables. The commode and drawers can be quantitatively represented as:

$$p = \begin{bmatrix} p_{1,1} & \dots & p_{1,k} & \dots & p_{1,N_D} \\ \vdots & \dots & \vdots & \dots & \vdots \\ p_{j,1} & \dots & p_{i,k} & \dots & p_{i,N_D} \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ p_{m,1} & \dots & p_{m,k} & \dots & p_{m,N_D} \end{bmatrix} \quad (4)$$

$$N_D(t) = \left\lceil \left(1 - \frac{t}{T}\right) \times N \right\rceil, \quad t=1, \dots, T, \quad (5)$$

$$\vec{P}_i = \left(\vec{D}_{rand(N)}, |K = 1, 2, \dots, N_D(t) \right), \quad j = 1, \dots, m, \quad (6)$$

where P is the matrix of the drawer, \vec{P}_i is the vector of j-th drawer, for $j = 1, \dots, m$, $\lceil \cdot \rceil$ is the usual mathematical ceiling function, T represents the overall iterations, $N_D(t)$ is the drawer number in the t-th iteration, and $\vec{D}_{rand(N),j}$ is the corresponding element of the i-th column.

Metaheuristic methods that utilize random search inside the relevant space have the capability to discover appropriate answers for optimization challenges. In order to achieve efficient search, metaheuristic algorithms must possess the capability to thoroughly explore the search space at both macro and micro stage exploration and exploitation. Utilizing a random combination in the update process of the DA design results in significant movements of the population within the search space, hence enhancing the exploration capability. Furthermore, in

DA, the quantity of suggested in drawer values diminishes in accordance with Eq. (5). This results in lower displacements, which enhances the algorithm's exploitation potential. Eq. (5) is chosen to ensure that, during the first iterations, the maximum values suggested for all variables are used in order to enhance exploration. Thus, in the construction of the DA algorithm, a harmonious equilibrium has been achieved between the processes of exploration and exploitation throughout its iterations.

The Differential Evolution algorithm uses a stochastic combination generated by data from drawers to update every individual in the population. This arbitrary combination guides the individuals in the population toward the exploration. The procedure of generating a random collection of drawers ensures that one value is chosen from the drawer. Subsequently, the amalgamation of these specifically chosen values from the drawers generates a fortuitous arrangement to direct the individual among the populace. The procedure for generating this arbitrary amalgamation is depicted as follows:

$$\vec{C}_i = \{p_{j,rand(N_D(t))} | j = 1, \dots, m\}, \quad i=1, \dots, N \quad (7)$$

Once the random composition is determined, each member of the population is updated in the search space by the provided equations

$$d_{i,j}^{new} = \begin{cases} d_{i,j} + rand(0,1) \cdot (C_{i,j} - rand(2) \cdot d_{i,j}), & Fun_i^C < Fun_i \\ d_{i,j} + rand(0,1) \cdot (d_{i,j} - C_{i,j}), & else \end{cases} \quad (8)$$

$$D_i = \begin{cases} D_i^{new} + rand(0,1) \cdot (C_{i,j} - rand(2) \cdot D_{i,j}), & Fun_i^{new} < Fun_i \\ D_i, & else \end{cases} \quad (9)$$

where D_i^{new} indicates the updated state of the i-th suggested solution, $d_{i,j}^{new}$ indicates its j-th dimension, Fun_i^{new} indicates its value of the objective function and Fun_i^C indicates the objective function of random combining to direct the i-th population component. Upon completing the population update, one iteration of the algorithm is performed. The algorithm population is continuously updated until the completion of its iteration, in accordance with Eqs. (5)-(9).

3. Proposed BinDA

3.1 Initialization step

In the initial stage, the entire population must be randomly initialized through a process of random assignment. The initial population is generated by the equation below:

$$D_{i,j} = lb_j + rand \times (ub_j - lb_j), \quad i = 1, \dots, N \text{ and } j = 1, \dots, dim. \quad (10)$$

The j-th variable value is generated by the i-th member. The function rand creates a random integer uniformly in the interval [0, 1].

$$N_D(t) = \left\lceil \left(1 - \frac{t}{T}\right) \times N. \right\rceil, \quad t=1, \dots, T, \quad (11)$$

$$\vec{P}_i = \left(\vec{D}_{rand(N)}, |K = 1, 2, \dots, N_D(t) \right), \quad j = 1, \quad (12)$$

3.2 Transformation function

Conventional wisdom holds that Feature Selection is best understood as a black-and-white issue. The standard Drawer Algorithm (DA) uses continuous values to describe the positions of the particles. Therefore, a transformation function must be included in order to change the continuous space of DA into a binary search space. When it comes to feature subset selection problems, particle concentrations can only take on two possible values: 0 and 1. In Fig. 1, we can see the binary version of a DA solution that was developed for a dataset of D characteristics. The related characteristic is either selected or unselected, denoted by the values 1 and 0, respectively. Among the transfer functions that fall into the S-shaped category, the sigmoidal function [34] is defined as follows:

$$T(P_i(t)) = \frac{1}{1 + e^{-P_i(t)}} \quad (13)$$

$$P_i(t+1) = \begin{cases} 1 & rand \geq T(P_i(t)) \\ 0 & rand < T(P_i(t)) \end{cases} \quad (14)$$

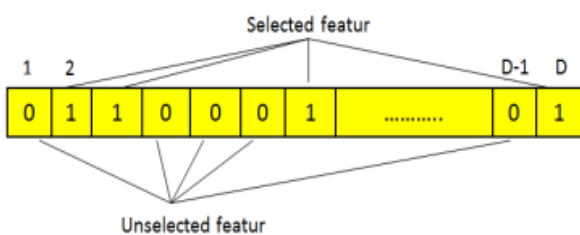


Figure. 1 Binary representation for BinDA solution

3.3 Applying the local search algorithm (LSA) to BinDA

In each iteration, the current optimal solution S is forwarded to a local search algorithm in an effort to identify an improved solution [13]. Each iteration of the Local search algorithm utilizes a random process to select three features from the current optimal solution. The fitness value for the novel solution is subsequently computed. Algorithm 1 states that the Local search algorithm will update the value B only if the fitness value of the new solution is greater than that of the present solution.

Algorithm 1: Local search algorithm

1. t_value = S where S is the optimum solution.
 2. **while** t < maximum_iterations
 3. Three features are selected at random from t_value.
 4. **if** selected_feature == 1 where 1 indicates that the feature is selected and 0 indicates that it is not selected) **then**
 5. selected_feature = 0
 6. **else**
 7. selected_feature = 1
 8. **end if**
 9. Assess t_value based on kNN or SVM classifiers
 10. Compute the fitness value of t_value
 11. **if** f(t_value) < f(S) **then**
 12. S = t_value
 13. **end if**
 14. t = t + 1
 15. **end while**
 16. **return** S
-

3.4 The evaluation function

Choosing more features from the data isn't always easy because classifier performance usually drops when faced with superfluous or unnecessary features. Hence, it is imperative to tackle this problem by decreasing the data's dimensionality. When assessing solutions, both the accuracy of classification and the quantity of selected features are important factors to consider. If two solutions achieve the same level of accuracy in classification, priority is given to the option that uses the least number of specified characteristics. By minimizing the number of characteristics chosen and the classification error, the fitness function seeks to maximize the classification accuracy. To find a happy medium between these two

main goals, we use the fitness function that is given below to measure BinDA solutions.

$$fitness = \omega\delta + \alpha \frac{Q}{N} \quad (20)$$

Where $\omega \in [0,1]$, δ is the rate of classification error calculated by the kNN or SVM, $\alpha = 1 - \omega$, Q represents the chosen features, whereas N is the overall number of features. The kNN or SVM serves as a classifier. [43, 44] in our proposed algorithm. In Algorithm 2, we can see the BinDA procedure in action.

Table 1. Description of the datasets

NO	Dataset	No. of features	No. of instances	No. of classes
1	Breast Cancer	9	699	2
2	Fri_c0_500_10	10	500	2
3	Fri_c0_1000_10	10	1000	2
4	Vowel	12	990	11
5	Australian	14	690	5
6	Zoo	17	101	2
7	Hepatitis	19	155	2
8	Parkinsons	22	195	2
9	IonosphereEW	34	351	2
10	Satellite	36	6435	6
11	WaveForm	40	5000	3
12	Lung Cancer	56	32	3
13	Spambase	57	4601	2
14	SonarEW	60	208	2
15	Splice	60	3190	3
16	Movementlibras	90	360	15
17	Robot1	90	88	4
18	Hillvalley	100	1212	2
19	Clean1	168	476	2
20	Leukemia	7129	72	2

Table 2. BinDA Parameter

parameter	value
Number. of runs	30
Number. of iterations	50
Number. of search agents	15
α	0.01
ω	0.99
K-neighbors	5
K-folder cross-validation	10

4. Experiments and discussion

4.1 Datasets

We employed a group of 20 datasets to evaluate BinDA in comparison to modern techniques. The datasets were chosen based on their wide range of instances and attributes to comprehensively assess BinDA across many situations. Table 1 presents a brief description of the analyzed datasets [28,45].

4.2 Setting BinDA parameter

Table 2 lists all of the BinDA parameters.

4.3 Experimental result

This research investigates the influence of integrating the performance of the proposed BinDA. Table 3 displays comparison analyses between the proposed BinDA and the standard DA in terms of average fitness, classification accuracy, and number of selected characteristics. The table demonstrates that BinDA consistently achieves higher average fitness than the standard DA in all 20 datasets. The BinDA consistently exceed the standard DA in Accuracy categorization across all 20 datasets as shown in Table 3. The quantity of chosen features for each method is additionally documented in Table 3. Fig. 2, 3, and 4 display a comparative analysis between DA and BinDA.

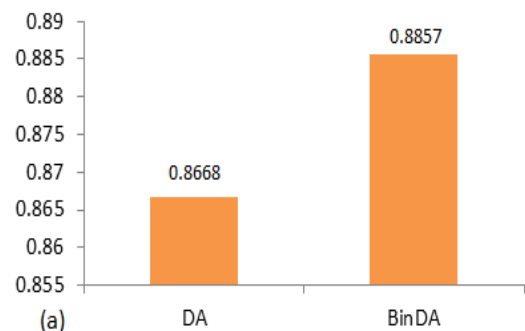


Figure. 2 The average Fitness value of BinDA compared to DA

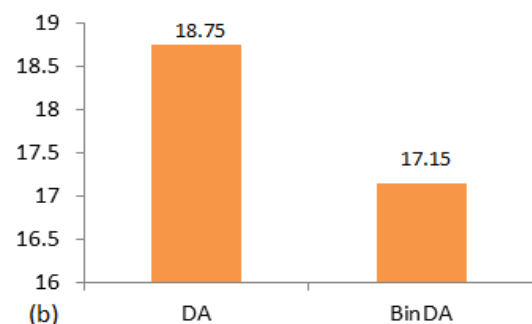


Figure. 3 Average classification accuracy value of BinDA compared to DA

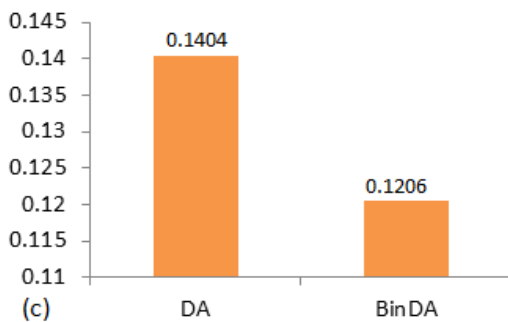


Figure. 4 Average No. of selected feature value of BinDA compared to DA

Algorithm 2: proposed algorithm BinDA

1. Put the iterations number T and the members number of the population N.
2. Create the initial population D at random by Eqs. (1) and (2).
3. Evaluate the initial population D by Eq. (3).
4. **For** t = 1: T
5. Transform the particle's positions to binary space depending on a transfer function by Eqs. (13) and (14).
6. Assess each particle in the population by kNN or SVM classifiers.
7. Measure the entire population fitness for the particle using Eq. (20).
8. Update the optimum proposed solution.
9. Compute the drawer matrix depending on Eqs. (4) and (6).
10. **For** i = 1: N
11. Construct a random combination depending on Eq. (7).
12. Compute a new status of population member depending on Eq. (8).

13. Update the i-th population member by Eq. (9).
 14. Apply a Local search algorithm (LSA) on the best solution to find if there is a better solution.
 15. **End**
 16. Save the best proposed solution so far.
 17. **End**
- Return** the best obtained proposed solution.

Results were compared with the most recent feature selection algorithms to examine the proposed BinDA algorithm's performance. Our feature selection algorithms include six popular ones: BinGWO[41], BinEO[28], BinMFO[342], BinPSO[43], BinSSA[44], and BinHOA [13]. Statistical results comparing the suggested BinDA method to state-of-the-art feature selection algorithms are shown in Tables 4-6. According to Table 4, after 50 iterations on each of the 20 datasets, the suggested BinDA algorithm and the participants' approaches were determined to be accurate. BinDA has achieved the highest accuracy of the instances, namely in all datasets. So, it has the best average accuracy in all datasets. A bar chart comparing the average overall accuracy is shown in Fig. 5. Table 5 lists the Fitness values of all the algorithms, including BinDA, for 20 datasets. Results from 19 of the 20 datasets show that the proposed BinDA method outperforms other algorithms. Fig. 6 is a bar chart that compares the average fitness values of the other algorithms with BinDA. To achieve the lowest average fitness value, the BinDA algorithm outperforms the HOA method, which ranks second with a value depicted in fig. 6.

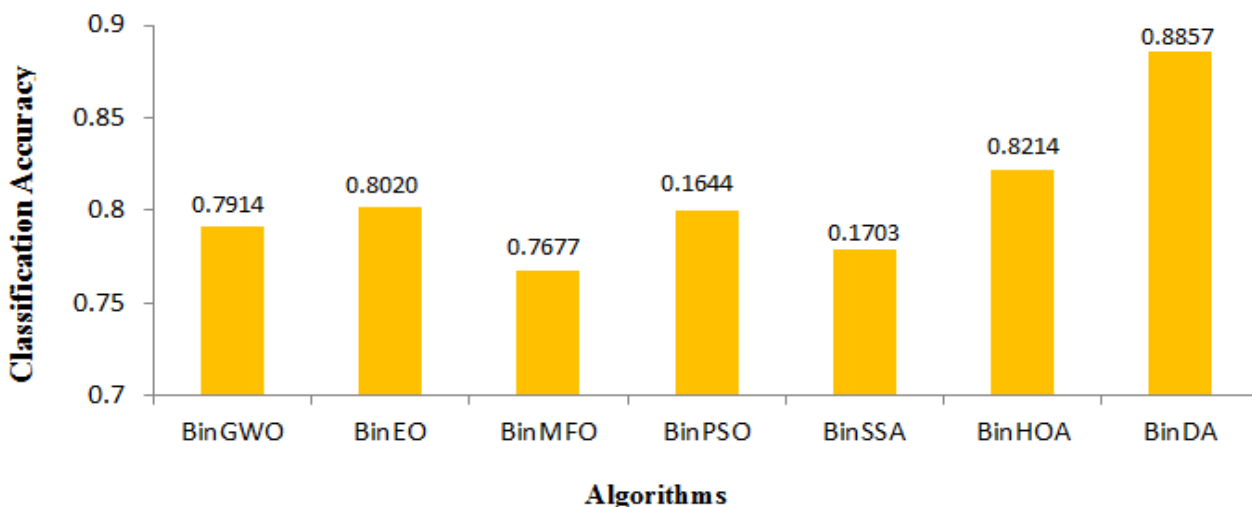


Figure. 5 Comparing BinDA with newer algorithms in terms of classification accuracy

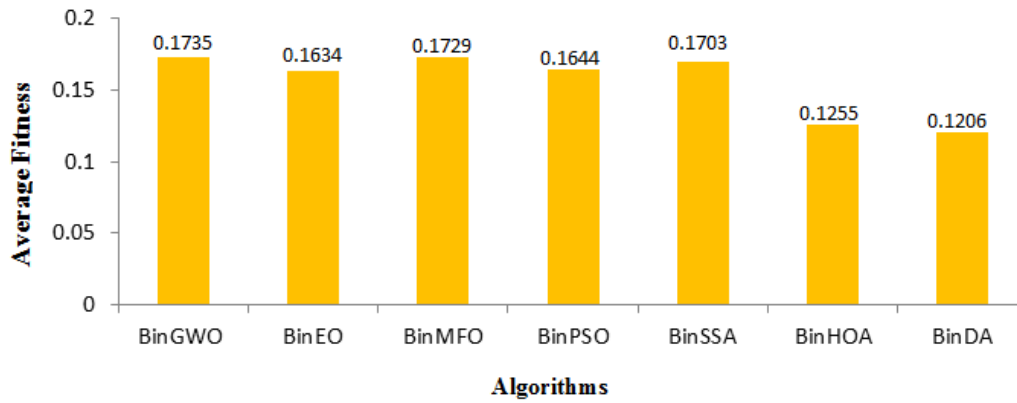


Figure. 6 Comparing BinDA with newer algorithms in terms of Average Fitness

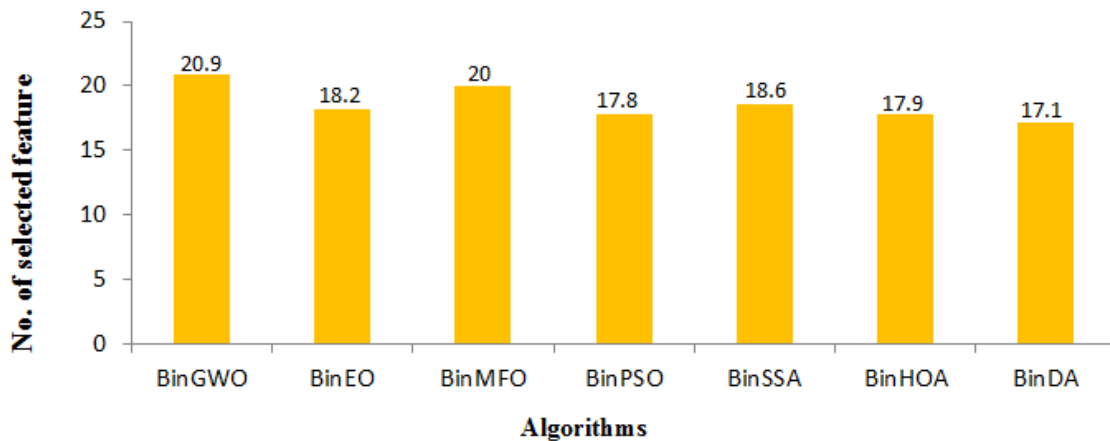


Figure. 7 Comparing BinDA with newer algorithms in terms of No. of selected features

Table. 3 Comparison between BinDA and DA overall datasets.

NO	Dataset	Average Fitness		Accuracy classification		No. of selected feature	
		DA	BinDA	DA	BinDA	DA	BinDA
1	Breast Cancer	0.0343	0.033	0.9348	0.9917	6	5
2	Fri_c0_500_10	0.1219	0.1036	0.8674	0.9087	6	5
3	Fri_c0_1000_10	0.1051	0.1007	0.9454	0.9407	5	3
4	Vowel	0.0671	0.0629	0.9567	0.9535	9	7
5	Australian	0.3538	0.311	0.6028	0.6942	6	5
6	Zoo	0.0367	0.0345	0.9664	0.9903	5	6
7	Hepatitis	0.1309	0.1287	0.8865	0.9000	9	8
8	Parkinsons	0.0639	0.0671	0.9409	0.9579	7	6
9	IonosphereEW	0.0508	0.0477	0.9598	0.9703	11	10
10	Satellite	0.0297	0.0241	0.9766	0.9897	16	14
11	WaveForm	0.1502	0.1342	0.8653	0.8859	20	19
12	Lung Cancer	0.2927	0.2687	0.6168	0.6674	15	16
13	Spambase	0.0762	0.0535	0.9780	0.9508	28	27
14	SonarEW	0.1306	0.1003	0.8874	0.8921	23	21
15	Splice	0.1906	0.1332	0.8716	0.8462	22	18
16	Movementlibras	0.2751	0.2124	0.7706	0.8013	31	30
17	Robot1	0.1223	0.1247	0.8716	0.9086	28	28
18	Hillvalley	0.3007	0.2308	0.7077	0.7297	32	29
19	Clean1	0.2659	0.2318	0.7498	0.7647	44	37
20	Leukemia	0.0095	0.0085	0.9797	0.9709	53	49
Average		0.1404	0.1206	0.8668	0.8857	18.75	17.15

Table 4. The outcomes of comparing the classification accuracy of various other algorithms.

NO	Dataset	BinGWO [41]	BinEO[28]	BinMFO[42]	BinPSO[43]	BinSSA[44]	BinHOA[13]	BinDA
1	Breast Cancer	0.9391	0.9419	0.9289	0.940318	0.920075	0.9242	0.9917
2	Fri_c0_500_10	0.7942	0.8245	0.7892	0.81965	0.786125	0.8468	0.9087
3	Fri_c0_1000_10	0.8451	0.8643	0.8342	0.86233	0.837425	0.8600	0.9407
4	Vowel	0.8875	0.9143	0.8910	0.904719	0.883025	0.8902	0.9535
5	Australian	0.6437	0.6405	0.6327	0.643207	0.63099	0.6394	0.6942
6	Zoo	0.9300	0.9318	0.9225	0.93217	0.910575	0.9192	0.9903
7	Hepatitis	0.8254	0.8285	0.4636	0.825082	0.80807	0.8280	0.9000
8	Parkinsons	0.8645	0.8643	0.8524	0.866016	0.848635	0.8840	0.9579
9	IonosphereEW	0.8578	0.8647	0.8437	0.861748	0.84246	0.9042	0.9703
10	Satellite	0.9264	0.9456	0.9195	0.936729	0.91086	0.9297	0.9897
11	WaveForm	0.7653	0.7970	0.7645	0.780074	0.75601	0.8104	0.8859
12	Lung Cancer	0.6168	0.6184	0.6134	0.634962	0.611515	0.6234	0.6674
13	Spambase	0.8579	0.8833	0.8556	0.875134	0.84455	0.8743	0.9508
14	SonarEW	0.8131	0.8151	0.8076	0.815576	0.79097	0.8332	0.8921
15	Splice	0.6784	0.6888	0.6695	0.688603	0.665665	0.7663	0.8462
16	Movementlibras	0.7474	0.7533	0.7438	0.756018	0.73834	0.7439	0.8013
17	Robot1	0.8266	0.8366	0.8268	0.838274	0.81776	0.8388	0.9086
18	Hillvalley	0.6243	0.6263	0.6148	0.625262	0.613415	0.6690	0.7297
19	Clean1	0.5298	0.5325	0.5264	0.532142	0.527345	0.6987	0.7647
20	Leukemia	0.8551	0.8678	0.8535	0.861748	0.83657	0.9436	0.9709
Average		0.7914	0.8020	0.7677	0.7999	0.7790	0.8214	0.8857

Table 5. Results of comparing Average Fitness to various other recent algorithms.

NO	Dataset	BinGWO [41]	BinEO[28]	BinMFO[42]	BinPSO[43]	BinSSA[44]	BinHOA[13]	BinDA
1	Breast Cancer	0.0345	0.0318	0.0342	0.0328	0.0342	0.0302	0.033
2	Fri_c0_500_10	0.1801	0.1472	0.1768	0.1532	0.1747	0.1087	0.1036
3	Fri_c0_1000_10	0.1288	0.1114	0.1315	0.1104	0.1182	0.0929	0.1007
4	Vowel	0.0825	0.0571	0.0757	0.0677	0.0724	0.0635	0.0629
5	Australian	0.3238	0.3278	0.3261	0.3232	0.3225	0.3135	0.311
6	Zoo	0.0417	0.0404	0.0405	0.0389	0.0411	0.0336	0.0345
7	Hepatitis	0.1393	0.1383	0.1394	0.1372	0.1396	0.1198	0.1287
8	Parkinsons	0.0993	0.0967	0.1014	0.0965	0.0987	0.0623	0.0671
9	IonosphereEW	0.1107	0.1045	0.1143	0.1083	0.1081	0.0451	0.0477
10	Satellite	0.0454	0.0290	0.0431	0.0360	0.0430	0.0235	0.0241
11	WaveForm	0.2082	0.1815	0.2031	0.1943	0.2019	0.1487	0.1342
12	Lung Cancer	0.3064	0.2864	0.3007	0.2778	0.2941	0.2710	0.2687
13	Spambase	0.1164	0.0928	0.1098	0.0992	0.1114	0.0830	0.0535
14	SonarEW	0.1446	0.1436	0.1449	0.1435	0.1479	0.1102	0.1003
15	Splice	0.2928	0.2839	0.2979	0.2830	0.2913	0.1777	0.1332
16	Movementlibras	0.2071	0.2066	0.2115	0.2030	0.2050	0.2038	0.2124
17	Robot1	0.1319	0.1241	0.1297	0.1214	0.1284	0.1157	0.1247
18	Hillvalley	0.3375	0.3356	0.3393	0.3334	0.3372	0.2712	0.2308
19	Clean1	0.4298	0.4301	0.4330	0.4270	0.4272	0.2276	0.2318
20	Leukemia	0.1094	0.0996	0.1051	0.1020	0.1083	0.0074	0.0085
Average		0.1735	0.1634	0.1729	0.1644	0.1703	0.1255	0.1206

Table 6. Comparing selected features with other recent methods across all datasets.

NO	Dataset	BinGWO [41]	BinEO [28]	BinMFO [42]	BinPSO [43]	BinSSA [44]	BinHOA [13]	BinDA
1	Breast Cancer	5.6	7	5.6	6	4.0	6	5
2	Fri_c0_500_10	8.8	6	8.3	6.4	7.2	6	5
3	Fri_c0_1000_10	6	5.4	5.6	4.0	4.8	3	3
4	Vowel	9.6	8.04	8.8	8.0	7.2	8	7
5	Australian	8.72	8.5	8.4	6.4	6.5	6	5
6	Zoo	12.8	6.7	11.6	8.0	10.0	5.5	6
7	Hepatitis	10.8	13	9.6	8.8	9.5	7.4	8
8	Parkinsons	9.8	6.1	8.8	8.8	8.8	7	6
9	IonosphereEW	14.4	14	14.3	12.0	12.8	13	10
10	Satellite	15.2	14.5	15.6	14.4	15.2	17	14
11	WaveForm	24.8	19.2	25.2	20.8	22.4	20	19
12	Lung Cancer	21.6	14.4	20.5	18.4	19.2	15	16
13	Spambase	32.5	28	31.7	28.0	29.6	28	27
14	SonarEW	26.8	20	24.4	22.4	22.8	22	21
15	Splice	24.8	24	23.6	21.6	22.3	18	18
16	Movementlibras	35.2	33	34.4	30.4	32.0	31	30
17	Robot1	31.2	24.8	28.8	27.2	28.0	27	24
18	Hillvalley	29.6	30	26.4	24.0	25.2	32	25
19	Clean1	39.2	39	38.4	36.0	36.8	38	35
20	Leukemia	52	50	50.4	46.4	47.2	51	45
Average		20.9	18.6	20	17.9	18.6	17.8	16.5

With respect to each dataset, table 6 details the total number of characteristics that were chosen. With regard to seventeen out of 20 datasets, the suggested BinDA accomplishes a minimal number of features. The BinDA achieved the smallest average selection size, demonstrating its great size reduction capabilities. Fig. 7 shows that the HOA algorithm ranks second with a val

4.4 Discussion

This paper presents Binary Drawer Algorithm (BinDA) for feature selection. The Drawer Algorithm (DA) is a novel metaheuristic algorithm inspired by the process of selecting objects from several drawers to build an optimal combination. The standard DA has been enhanced with major features to increase its overall performance. The local search algorithm is a novel addition to the DA algorithm that improves its performance. The proposed methods' performance is categorized into two sections. The first is comparison analyses between the proposed BinDA and the standard DA in terms of average fitness, classification accuracy, and number of selected characteristics as represented in Table 3. The TAable 3 demonstrates that BinDA consistently achieves

higher average fitness than the standard DA in all 20 datasets. The second section is comparison analyses with other existing algorithm in the classification accuracy, Average Fitness, and selected features as presented in table 4, table5, table6, respectively.

The empirical results demonstrate outperform of the proposed algorithm on the other existing algorithms in the all metrics. The proposed method exhibits superiority in all metrics for several noteworthy reasons. Using The Drawer Algorithm (DA) in binary version by Transformation function which enhances the convergence speed of the algorithm and compensates for its limited search performance. The implementation of the local search algorithm has been found to enhance the accuracy of optimization to a certain degree. Additionally, it improves the algorithm's ability to utilize the search space and leads to a more extensive exploration of potential solutions.

5. Conclusion

A novel binary drawer algorithm (BinDA) proposed for feature selection problems in artificial intelligence applications are presented in this paper. The standard DA is improved by incorporating a

Local search algorithm, leading to improved performance. When used with the BinDA algorithm, the kNN or SVM classifier yields excellent results. On top of that, these classifiers have demonstrated excellent performance when trained using the given data. To start the population out on the right foot, we implement the Local search algorithm's strategy, which seeks to increase variety and traversal. Furthermore, it causes the algorithm to explore possible solutions more thoroughly and improves its capacity to use the search space. We assess the results of our proposed approach against six other feature selection methods across twenty datasets for evaluation. Our findings indicate that BinDA surpasses recently developed feature selection algorithms. Statistical analysis confirms the superiority of our proposed algorithm over existing techniques.

Notation list

D	the population matrix of the DA
lb	lower bounded
ub	upper bounded
$rand$	a random integer uniformly in the interval $[0, 1]$
Fun	represents the objective function
P	the matrix of the drawer
\vec{p}_i	the vector of j-th drawer
$\lceil \cdot \rceil$	is the usual mathematical ceiling function
T	represents the overall iterations
$N_D(t)$	the drawer number in the t-th iteration
$\vec{D}_{rand(N)}$	the corresponding element of the i-th column

Conflicts of Interest

The authors declare no conflict of interest.

References

- [1] K. Huang, and S. Aviyente, "Wavelet feature selection for image classification", *IEEE Trans. Image Process*, Vol. 17, No. 9, pp. 1709–1720, 2008.
- [2] G. Bissan, and J. Naoum, "High dimensional data classification and feature selection using support vector machines", *European Journal of Operational Research*, Vol. 265, No. 3, pp. 993-1004, 2018.
- [3] F. Hossam, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems", *Knowledge-Based Systems*, Vol. 154, pp. 43-67, 2018.
- [4] J. K. Robert, and J. Mao, "Statistical pattern recognition: A review", *IEEE Transactions on*
- [5] V. Belur, and Dasarathy, "Nearest neighbor (NN) norms: NN pattern classification techniques", *IEEE Computer Society Tutorial*, 1991.
- [6] K. Hosny, and A Mohamed, "BCOVIDOA: a novel binary coronavirus disease optimization algorithm for feature selection", *Knowledge-based systems*, Vol. 248, p. 108789, 2022.
- [7] E. Eid, and A. Hassanien, "Binary grey wolf optimization approaches for feature selection", *Neurocomputing*, Vol. 172, pp. 371-381, 2013.
- [8] K. Ali, and N. Farhoudi, "A new optimization method: Dolphin echolocation", *Advances in Engineering Software*, Vol. 59, pp. 53-70, 2023.
- [9] R. Venkata, and D. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems", *Computer-aided design*, Vol. 43, No. 3, pp. 303-315, 2011.
- [10] R. Esmat, and S. Saryazdi, "GSA: a gravitational search algorithm", *Information sciences*, Vol. 179, No. 13, pp. 2232-2248, 2009.
- [11] H. Nikolaus, and D. Müller, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)", *Evolutionary computation*, Vol. 11, No. 1, pp. 1-18, 2003.
- [12] R. Douglas, "Binary flower pollination algorithm and its application to feature selection", *Recent Advances in Swarm Intelligence and Evolutionary Computation*, pp. 85-100, 2015.
- [13] E. Dina, "BinHOA: Efficient binary horse herd optimization method for feature selection: Analysis and validations", *IEEE Access*, Vol. 10, pp. 26795-26816, 2002.
- [14] M. Majdi, "Binary dragonfly algorithm for feature selection", *International conference on new trends in computing sciences (ICTCS)*, IEEE, 2017.
- [15] K. Ali, and N. Farhoudi, "A new optimization method: Dolphin echolocation", *Advances in Engineering Software*, Vol. 59, pp. 53-70, 2013.
- [16] X. Bing, and N. Browne, "Particle swarm optimization for feature selection in classification: Novel initialization and updating mechanisms", *Applied soft computing*, Vol. 18, pp. 261-276, 2014.
- [17] A. Prachi, and A. Mohamed, "A novel binary gaining-sharing knowledge-based optimization algorithm for feature selection", *Neural*

- Computing and Applications*, Vol. 33, No 11, pp. 5989-6008, 2021.
- [18] H. Seyedmohsen, and A. Khaled, "A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research", *Applied Soft Computing*, Vol. 24, pp. 1078-1094, 2024.
- [19] K. Cio, and C. Lin, "Cultural evolution algorithm for global optimizations and its applications", *Journal of applied research and technology*, Vol. 11, No. 4, pp. 510-522, 2013.
- [20] M. Reza, and K. Salimifard, "Volleyball premier league algorithm", *Applied Soft Computing*, Vol. 64, pp. 161-185, 2018.
- [21] R. Venkata, and D. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems", *Computer-aided design*, Vol. 43, No. 3, pp. 303-315, 2011.
- [22] S. Ibrahim, and A. Mutlag, "Lightning search algorithm", *Applied Soft Computing*, Vol. 36, pp. 315-333, 2015.
- [23] M. Seyedali, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization", *Neural Computing and Applications*, Vol. 27, pp. 495-513, 2016.
- [24] H. Fatma, "Henry gas solubility optimization: A novel physics-based algorithm", *Future Generation Computer Systems*, Vol. 101, pp. 646-667, 2019.
- [25] T. Mohammad, "An evolutionary gravitational search-based feature selection", *Information Sciences*, Vol. 497, pp. 219-239, 2019.
- [26] H. Sajedi, "Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: Application of the simulated annealing feature selection method", *Science of the Total Environment*, Vol. 711, p. 135161, 2020.
- [27] A. Shameem, "AIEOU: Automata-based improved equilibrium optimizer with U-shaped transfer function for feature selection", *Knowledge-Based Systems*, Vol. 228, p. 107283, 2021.
- [28] E. Dina, and E. Rashed, "An improved equilibrium optimizer algorithm for features selection: methods and analysis", *IEEE Access*, Vol. 9, pp. 120309-120327, 2021.
- [29] K. Manoj, "Genetic algorithm: Review and application", *Available at SSRN*, p. 3529843, 2010.
- [30] Z. Yong, "Binary differential evolution with self-learning for multi-objective feature selection", *Information Sciences*, Vol. 507, pp. 67-85, 2020.
- [31] S. Dan, "Biogeography-based optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, pp. 702-713, 2008.
- [32] K. Soheyl, and S. Khalilpourazary, "Multi-objective stochastic fractal search: A powerful algorithm for solving complex multi-objective optimization problems", *Soft Computing* Vol. 24, pp. 3037-3066, 2020.
- [33] K. Purba, and A. Dinimaharawati, "Extended stochastic coati optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No 3, pp. 482-494, 2023, doi: 10.22266/ijies2023.0630.38.
- [34] A. O. Riyadh, and R. Nuijaa, "Malware cyberattacks detection using a novel feature selection method based on a modified whale optimization algorithm", *Wireless Networks*, Vol. 1, No. 17, 2023.
- [35] D. Sajjad, "Ring toss game-based optimization algorithm for solving various optimization problems", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 545-554, 2021., doi: 10.22266/ijies2021.0630.46.
- [36] K. Purba, and A. Dinimaharawati, "Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms", *International Journal of Intelligent Engineering & Systems*, Vol. 17, No. 2, 2024., doi: 10.22266/ijies2024.0430.31.
- [37] T. Eva, and V. Leiva, "Drawer Algorithm: A New Metaheuristic Approach for Solving Optimization Problems in Engineering", *Biomimetics*, Vol. 8, No. 2, p. 239, 2023.
- [38] M. Majdi, and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection", *Neurocomputing*, Vol. 260, pp. 302-312, 2017.
- [39] P. Duin, and J. Mao, "Statistical pattern recognition: A review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4-37, 2000.
- [40] P. Franz, "Bayesian network classifiers versus selective k-NN classifier", *Pattern Recognition*, Vol. 38, No. 1, pp. 1-10, 2005.
- [41] M. El-Kenawy, and A. Ibrahim, "MbGWOSFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection", *IEEE Access*, Vol. 8, pp. 107635-107649, 2020.
- [42] R. A. Khurma, I. Aljarah, and A. Sharieh, "An efficient moth flame optimization algorithm using chaotic maps for feature selection in the

- medical applications”, In: *Proc. of ICPRAM*, pp. 175–182, 2020.
- [43] S. B. Sakri, and Z. M. Zain, “Particle swarm optimization feature selection for breast cancer recurrence prediction”, *IEEE Access*, Vol. 6, pp. 29637–29647, 2018.
- [44] S. S. Shekhawat, and B. Qureshi, “BSSA: Binary salp swarm algorithm with hybrid data transformation for feature selection”, *IEEE Access*, Vol. 9, pp. 14867–14882, 2021.
- [45] C. Blake, C. Merz, “UCI Repository of Machine Learning Databases”, *Ph.D. dissertations*, Dept. Inform. Comput. Sci. Univ. California, Irvine, CA, USA, 1998.