

RESEARCH ARTICLE

Binary Drone Squadron Optimization Approaches for Feature Selection

HARPREET SINGH¹, SUCHITA SHARMA¹, MANJU KHURANA¹,
MANJIT KAUR², (Senior Member, IEEE), AND HEUNG-NO LEE², (Senior Member, IEEE)

¹Department of Computer Science and Engineering, Thapar Institute of Engineering & Technology, Patiala 147004, India

²School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Heung-No Lee (heungno@gist.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant through the Korean Government (MSIP) under Grant NRF-2021R1A2B5B03002118; and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program Supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2021-0-01835.

ABSTRACT A great amount of data is being created these days, which is kept in massive datasets with different irrelevant attributes that are unrelated to the goal notion. Feature selection deals with the selection of the most pertinent features that also aid to increase the classification accuracy. The topic of feature selection is viewed as a multiobjective optimization problem with two goals: improving the classification accuracy and reducing the number of features used. Drone Squadron Optimization (DSO) is one of the most recent artifact-inspired optimization algorithms; having two key components: semi-autonomous drones that hover over a terrain and a command center that manages the drones. In this paper, two binary variants of the DSO are proposed to deal with the feature selection problem. The proposed binary algorithms are applied on 21 different benchmark datasets with five state-of-the-art algorithms, i.e., Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), Flower Pollination Algorithm (FPA), Genetic Algorithm (GA) and Ant Lion Optimization (ALO). Different assessment indicators are used to assess the diversification and intensification of the optimization algorithms. When compared to current state-of-the-art wrapper-based algorithms, the suggested binary techniques are more efficient in scanning the dimension space and picking the most useful characteristics for categorization tasks, resulting in the lowest classification error rate.

INDEX TERMS Binary drone squadron optimization, drone squadron optimization, feature selection, wrapper algorithm, artifact inspired evolutionary algorithm.

I. INTRODUCTION

In computer science, a dataset includes significant, insignificant, or superfluous features that critically affect the performance of the classification due to a large number of features [1]. Choosing the important qualities or properties of the data is a perplexing issue. Feature selection is a technique which aims to eliminate superfluous variables within a dataset to better understand the data. The goal of feature selection approaches is to improve classifier performance and achieve a classification error rate that is almost comparable to, if not identical to, that of using the entire feature set. [2]. Wrappers

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li¹.

and filters are the two basic categories of feature selection methods [3]. The filter-based algorithms utilize statistical methods [4] to evaluate a feature subset, whereas learning algorithms are used in wrapper-based algorithms to search through the universe of potential solutions to find the best feature subset [5].

Filter based methods generally demonstrate quicker performance in comparison to wrapper-based methods since they calculate the distance between features, information gain and feature dependency, which is computationally less expensive than estimating a classifier precision [3]. Nevertheless, wrappers have been shown to be useful in locating the best feature subsets for a given classifier, hence they are commonly investigated for classification error rate [6]. When using a wrapper

feature selection technique, three factors should be decided: classifier, feature subset evaluation criteria, and a searching algorithm to find the best subset of features [4].

Finding a near optimum subset from the original set is a difficult task. Moreover, the search space size will increase exponentially with the amplification of the number of dimensions in a data set. Practically speaking, the exhaustive search techniques are not able to achieve the desired optimal solution and still experience the ill effects of stagnation in local optima [7]. In recent decades, metaheuristic algorithms inspired by nature have grown in prominence because of their ability to deal with complex real-world situations in a powerful and effective manner [8]. These algorithms are capable of utilizing the population's relevant knowledge in order to identify the best solutions. These resilient and efficient methods are used to solve a wide range of optimization problems, including financial time series prediction [9], economic dispatch optimization [10], neural networks [11], [12], wireless sensor networks [13] and engineering design problems [14], [15]. Many researchers have proposed metaheuristic based wrapper techniques for feature selection problem where metaheuristics are used to make subsets of features [16]. In practice, for the dataset with n number of attributes, the number of possible subsets are 2^n , which makes it impossible for the exhaustive search methods to evaluate all possible combinations due to time consumption [17]–[19]. Therefore, metaheuristic based wrapper techniques have shown better performance in comparison to the exhaustive search methods [1].

The feature subset generation stage of the metaheuristic algorithm produces the random solution primarily, and the fitness of the random solutions is assessed using some fitness function by means of a subset assessment stage for every solution corresponding to which the dataset is partitioned into training and testing sets. The classification algorithm is trained by means of the training set and evaluated using the testing set, and the fitness is calculated to get the best solution in each iteration. The best solution in turn is used in the consequent iterations for updating the solutions. The training and testing set is divided using a fivefold cross-validation procedure. This rudimentary process of feature selection is identical for the entire metaheuristic algorithm. Figure 1 shows the feature selection process. The remaining sections of the paper are organized as follows: Next section presents the related work while the proposed DSO algorithm for feature selection problem is described in Section 3. Section 4 discusses how DSO is used for feature selection, and Section 5 reports the findings and comments. Section 6 concludes with findings and recommendations for further work.

II. RELATED WORK

Emary *et al.* [20] used the Grey Wolf Optimization (GWO) strategy to tackle the feature selection problem. To determine the location of the grey wolf, the approach employed a stochastic crossover. Recently, Ant Lion Optimizer (ALO) [21] and Flower Pollination Algorithm (FPA) based

technique were employed to tackle the feature selection challenge [22]. In the realm of medical diagnostics, an enhanced GWO method with ELM has recently been presented to discover the best feature subset [23]. In the realm of text classification, a genetic algorithm combined with chaos theory for dimension reduction has been developed [24]. To determine the optimal feature subset, Xue *et al.* [25] presented novel starting strategies in the Particle Swarm Optimization. Gu *et al.* [26] addressed the feature selection problem by employing a competitive swarm optimizer, a variation of the PSO. Recently, a hybrid approach based on ABC and the differential evolution technique for picking an optimal collection of features for the classification issue has been developed [27].

Simulated annealing has recently been combined with a whale optimization approach to improve the exploitation of simulated annealing [28]. To increase the efficiency of the native crow search for feature selection, Sayed *et al.* [29] added chaotic maps into the crow search algorithm. As a method of feature selection for classification, a binary ant lion optimizer has also been presented in [21]. In ant lion optimization, Zawbaa *et al.* [30] used chaotic maps to minimize the attributes in high-dimensional data sets. In wrapper mode, another solution based on the whale optimization algorithm was developed to handle the feature selection problem [18]. A strategy for finding the best feature set was successfully implemented using a grasshopper optimization algorithm [31] and evolutionary population dynamics [32] in [33].

Hans and Kaur [34] proposed a binary multi verse optimization method for selecting the optimal feature set by incorporating the use of various transfer functions for position updation. Hu *et al.* [35] proposed an improved grey wolf optimizer so that the problem of discretization can easily be solved. Zhang *et al.* [36] integrated salp swarm algorithm (SSA) into the Harris Hawk Optimizer so that a balance between exploration and exploitation can be maintained. Neggaz *et al.* [37] proposed the used of sine cosine algorithm boosted by salp swarm algorithm. The method improves the exploration and avoids local minima. Emine and Ülker [38] proposed the use of social spider algorithm by using a transfer function which can be utilized for mapping from a continuous to a binary search space. Agrawal *et al.* [39] proposed the use of quantum concepts and the whale optimization algorithm to increase exploration and exploitation. The individual solutions can be represented in the form of bits and modified mutation and crossover operators have been introduced.

In spite of the many metaheuristic methodologies suggested for the feature selection problems, still, many difficulties remain unanswered, such as reduced accuracy in the case of datasets with higher number of attributes, high processing time, and solution subsets with inappropriate attributes. To address these issues, this study explores the use of DSO algorithm for feature selection as this is claimed to be robust, adaptive, and flexible and has powerful exploration and exploitation capabilities. In this context, this study

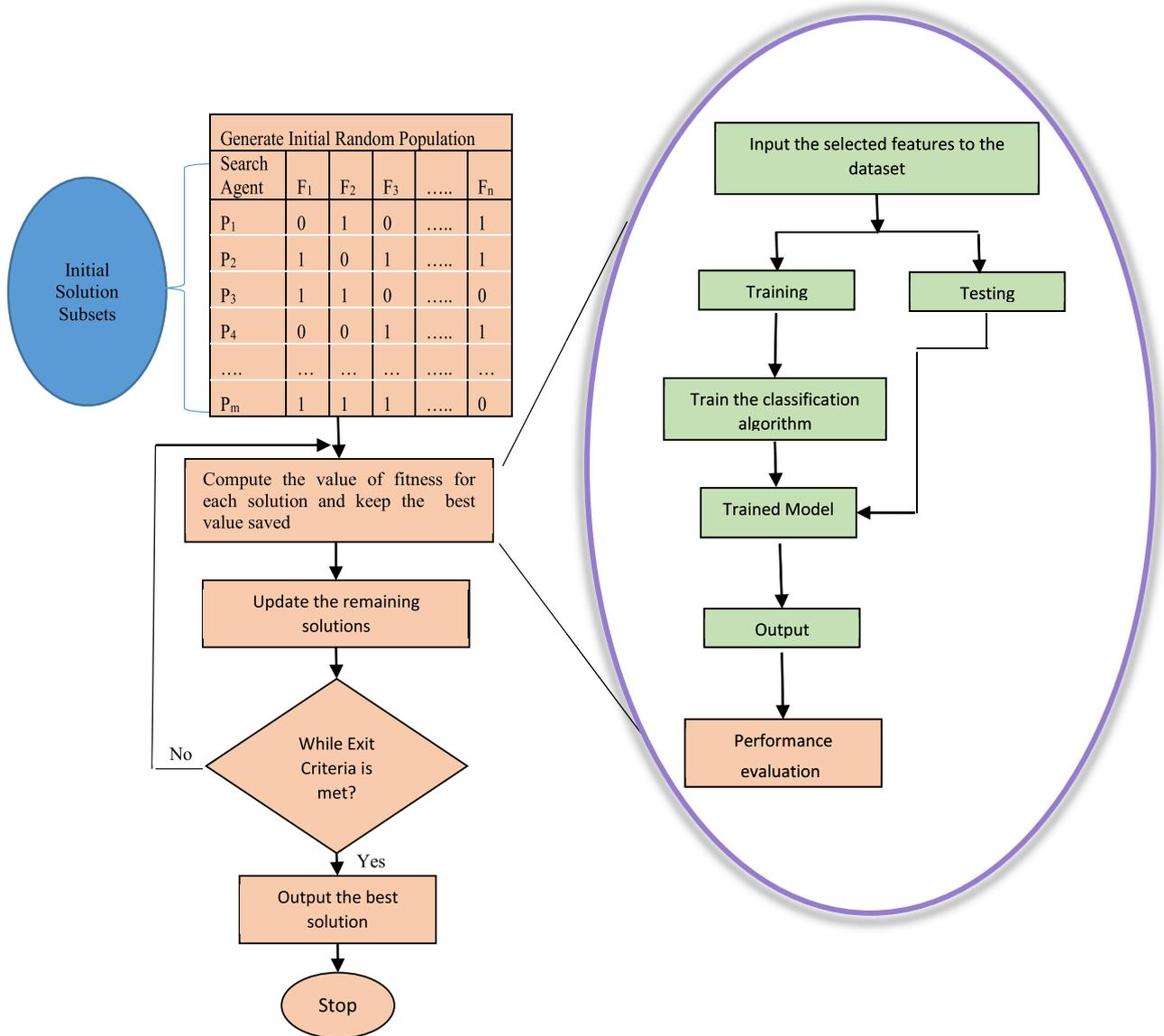


FIGURE 1. Feature selection process.

aims to propose two variants of the binary DSO algorithm for selecting a minimal number of features and acquiring a comparable or perhaps lower classification error rates than traditional feature selection methods.

III. PRELIMINARIES

The Drone Squadron Optimization is a non-nature-motivated evolutionary metaheuristic that imitates the drones’ behaviour which fly over a geographical area to explore and these drones are controlled by a command center [40]. Figure 2 shows the important terms used in the DSO and Figure 3 shows the characteristics of DSO.

A. DRONE SQUADRON OPTIMIZATION

1) INSPIRATION

DSO is inspired by the movement of drones, which are entities that make movements to explore the geographical

search space area, to locate something, or to complete a particular task. There are different teams of drones which are being controlled by the command centers. A drone tries to move closer to the destination location. The command center performs two tasks: manage the search operations of the drones partially and control the drones by making new firmware which contains the modules to explore the search space. The drones are divided into certain groups where each group is controlled by the firmware to control the movement of the drones. Each team is controlled by its own firmware. There are perturbation functions that are used to create the trial solution locations for the movement of the drones. The perturbation scheme is used to generate the new firmware using the following equations:

$$P = \text{Departure} + \text{offset}(), \tag{1}$$

$$TC = \text{calculate}(P), \tag{2}$$

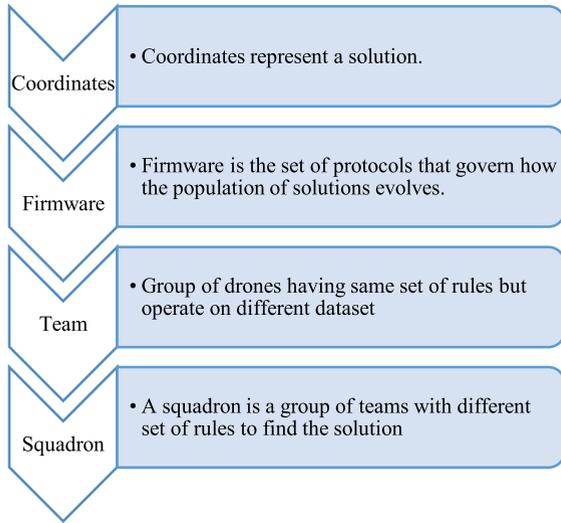


FIGURE 2. Important terms of DSO.



FIGURE 3. Characteristics of DSO.

where *departure* is a coordinate, *offset()* represents the perturbation movement, and Equation 1 and 2 show the formulas for finding the trial coordinates *TC*.

2) MOVEMENT OF DRONES

The drones move to the target positions calculated by the perturbation step and return the information to the command center. To calculate the target locations, the DSO may use any of the methods used in various optimization techniques. For example, consider the two following perturbations for team1 and team2:

$$P_1 : \overrightarrow{GBC} + (C_1 \times (\overrightarrow{GBC} - \overrightarrow{CBC}_{drone})) \quad (3)$$

$$P_2 : \overrightarrow{CBC}_{drone} + (G(0, 1) \times \sqrt{U(0, 1)_D - \overrightarrow{CBC}_{drone}}) \quad (4)$$

where $G(0, 1)$ are the values generated from Gaussian distribution, C_1 is a constant whose value is defined by the user, and $U(0, 1)_D$ represents an array generated from a uniform distribution ranging from 0 to 1.

It's worthwhile to mention that \overrightarrow{GBC} represents the global best coordinates and $\overrightarrow{CBC}_{drone}$ represents the current best coordinates.

Each drone performs the recombination with the best solution coordinates after the perturbation step, where all recombination methods have an equal probability of getting selected. The drones are permitted to move only in a certain search space region. If the calculated position of the drone comes out to be outside this region, then it is considered as a violation which needs to be corrected. The following equation is used to correct the violations:

$$violation_{team} = \sum_{drone=1}^n \sum_{j=1}^D \{ |T_m C_{team,drone,j} - UB_j| + |LB_j - T_m C_{team,drone,j}| \} \quad (5)$$

The number of drones in each team is represented by N , D represents the dimension, and the upper bounds array is UB , while the lower bounds array is LB . This calculation takes care of the violation cases where $T_m C_{team,drone,j} > UB_j$ or $T_m C_{team,drone,j} < LB_j$.

The symbol $T_m C_{team,drone}$ denotes the target position of each drone of the team. The quality of teams is measured by using the number of solutions generated out of bounds and the distance from the objective function. The corrections of the teams are performed at every iteration. The firmware is updated when a criterion set in the beginning is reached and in that process, the firmwares of the team with the worst results are replaced by the firmwares of the best performing team. The command center and drones are depicted in Figure 4. Figure 5 shows the pseudocode of DSO.

IV. THE PROPOSED BINARY DRONE SQUADRON OPTIMIZATION (BDSO)

A. BINARY DRONE SQUADRON OPTIMIZATION -METHOD 1 (S-BDSO)

As previously stated, because the new drone locations are values in continuous form, the continuous values must be translated to binary values that correspond to each other. A Sigmoidal (S-shaped) transfer function is used to accomplish this task of compressing continuous solutions in each dimension [41], which forces the drones to discretize their movement. The S-shaped function, as seen in Eq. (6) and Figure 6, is a typical transfer function.

$$S(P_i^k(t)) = \frac{1}{1 + e^{-P_i^k(t)}} \quad (6)$$

where P_i^k is the continuous-valued perturbation of the i th drone in k th dimension at iteration t .

The Eq. (7) is used to convert the S-shape function output into a binary values in the case of sigmoidal function.

$$x_i^k(t+1) = \begin{cases} 0, & \text{if } rand < S(P_i^k(t)) \\ 1, & \text{if } rand > S(P_i^k(t)) \end{cases} \quad (7)$$

where $x_i^k(t)$ and $P_i^k(t)$ indicate the position and perturbation of the i th drone at iteration t in the k th dimension.

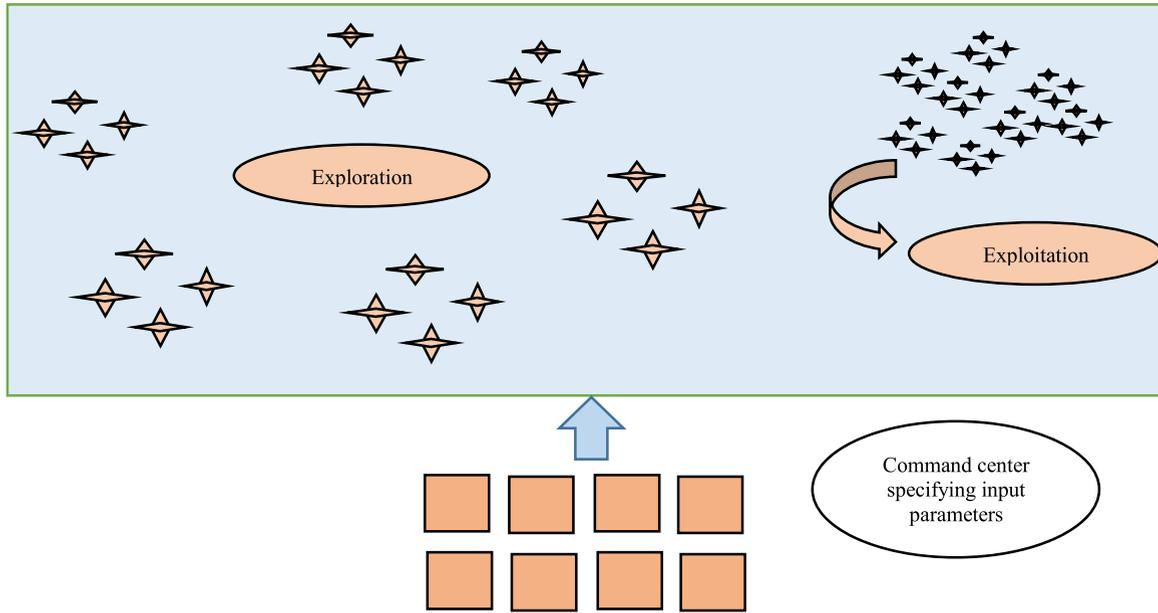


FIGURE 4. Depiction of the command center and drones in DSO.

```

Input: the fitness function, the upper and lower bound, number of teams
(n_team), number of drones in each team (n_drone), total number of
iterations (N), firmware options, number of iterations to be allowed
without result improvement (w)

Output: the best solution
Initialize the drones and teams, the commandcentre. Put
all the drones at random locations.
Choose the n_drone best solutions to be the current best solutions.
while stopping condition is not true do
  for each team do // procedure applicable on teams
    for each drone do
      Apply perturbation method to generate the trial coordinates Apply
      a recombination procedure to find the location where drone has to
    for each drone do // drone with same id of each team
      Calculate the rank of the drone using objective function
      Calculate the violations for each team
      if the current best solution is greater than the solution found
      by best drone in the present loop then
        The current best solution is replaced by the best drone's
        solution
        if the global best solution is greater than the solution found
        by best drone in the present loop then
          The global best solution is replaced by the best drone's
          solution
        end if end if
    end for
    Check for stagnation
    Calculate the average rank of each team and also the violations
    According to the quality of teams the firmware is updated
  
```

FIGURE 5. Pseudo-code of DSO.

B. BINARY DRONE SQUADRON OPTIMIZATION - METHOD 2 (V-BDSO)

This approach presents a V-shaped transfer function rather than an S-shaped transfer function and Eqs. (8) and (10) are

used to achieve so. The transfer functions of force drones to travel in a discrete search space are shown in Figure 7.

$$V(P_i^k(t)) = \left| \operatorname{erf}\left(\frac{\sqrt{\pi}}{2} P_i^k(t)\right) \right| \tag{8}$$

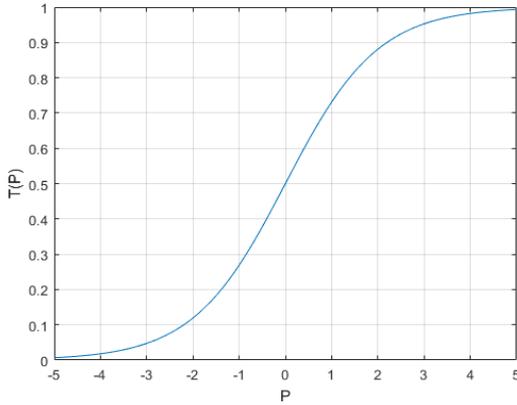


FIGURE 6. Sigmoid transfer function.

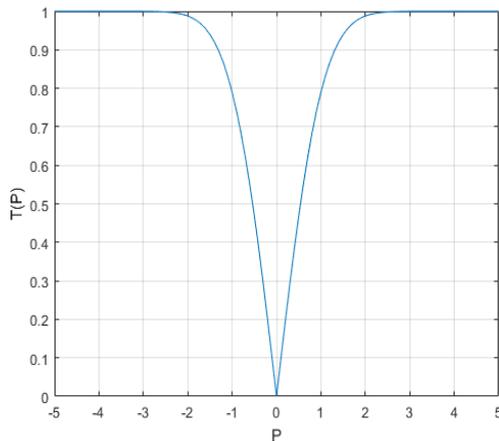


FIGURE 7. V-shaped transfer function.

$$V \left(P_i^k(t) \right) = \left| \frac{\sqrt{\pi}}{2} \int_0^{\frac{\sqrt{\pi}}{2} P_i^k(t)} e^{-t^2} dt \right| \quad (9)$$

The Eq. (9) can be rewritten as:

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1}, & \text{if } rand < V \left(P_i^k(t) \right) \\ x_i^k(t), & \text{if } rand > V \left(P_i^k(t) \right) \end{cases} \quad (10)$$

where $x_i^k(t)$ and $P_i^k(t)$ specify the location and perturbation of the i th drone in the k th dimension at iteration t , and $x_i^k(t)^{-1}$ represents the complement of $x_i^k(t)$.

A large $P_i^k(t)$ indicates a high likelihood of shifting site, which means the search agent is significantly away from the best solution, therefore it should change the position vector to hit possible search space regions on the road. Lesser values of $P_i^k(t)$ on the other hand, imply that a search agent should stay put and try to find anything close [44].

C. FITNESS FUNCTION

As mentioned earlier, the DSO algorithm is employed to search through the large subset of features to find the best subset which should have a minimal number of features and maximum accuracy. To evaluate the subset of features, the

following fitness function is used:

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \quad (11)$$

where $\gamma_R(D)$ is the KNN classifier’s classification error rate. In addition, R denotes the cardinality of the chosen feature subset, whereas N is the total number of features in the original dataset. The parameters $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ are taken from [21], [28] and relate to the relevance of classification quality and subset length, respectively.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. DATA DESCRIPTION

The experimental analysis has been performed on twenty-one datasets shown in Table 1 and are retrieved from UCI repository [42]. The datasets were chosen to reflect a wide range of characteristics and tuples on which the suggested technique must be evaluated [20], [21]. Especially, the chosen datasets offer a large amount of search space, allowing for a proper optimization method’s testing. Every dataset is partitioned in the same way as done in cross-validation procedures [43].

TABLE 1. Data sets that were utilized.

S. No.	Title	Symbol	Features	Samples
1	Breastcancer	Z1	9	699
2	BreastEW	Z2	30	569
3	Clean1	Z3	166	476
4	Clean2	Z4	166	6598
5	CongressEW	Z5	16	435
6	Exactly	Z6	13	1000
7	Exactly2	Z7	13	1000
8	HeartEW	Z8	13	270
9	IonosphereEW	Z9	34	351
10	KrvskpEW	Z10	36	3196
11	Lymphography	Z11	18	148
12	M-of-n	Z12	13	1000
13	PenglungEW	Z13	325	73
14	Semeion	Z14	265	1593
15	SonarEW	Z15	60	208
16	SpectEW	Z16	22	267
17	Tic-tac-toe	Z17	9	958
18	Vote	Z18	16	300
19	WaveformEW	Z19	40	5000
20	WineEW	Z20	13	178
21	Zoo	Z21	16	101

The training is done using $K-1$ folds, while the validation is done on the k th fold as in K fold cross-validation. For each dataset, the evaluation is repeated $K \times M$ times. The dataset’s training section is used to teach the classifier, and then its performance is evaluated using the dataset’s validation section. Finally, the testing dataset is used to evaluate the characteristics that have been chosen. Each drone is relocated during the training process to choose a feature subset. PSO, FPA, GA, ALO, and GWO were used to compare the suggested feature selection methods.

Before employing these methods for feature selection, certain parameters must be preset. Table 2 shows the parameters values used in this investigation. These parameter values were chosen based on the values found in the literature [20], [21].

TABLE 2. Experimental parameters.

Parameter	Value
GA Crossover ratio	0.9
p in FPA	0.8
No. of search agents	7
No. of iterations	100
Search domain	{0, 1}
GA (parent selection)	Roulette wheel
PSO constants	[2, 2]
GA mutation	0.1
GWO parameters	Max = 2 and min = 0
PSO (inertia)	[0.9, 0.6]
M (runs)	20
K (cross validation)	5

B. EVALUATION CRITERIA

In each run of the individual optimization method, the following metrics are applied to the data:

1) CLASSIFICATION ACCURACY

It is the average of classification accuracy values obtained from N runs. It can be defined as the fraction of tuples that are erroneously categorized.

$$accuracy = \frac{1}{N} \sum_{j=1}^N \frac{1}{M} \sum_{i=1}^M match(C_i, L_i) \quad (12)$$

The number of optimization algorithm runs is N , and the number of tuples in the testing data set is M . C_i is the data point i 's classifier output label, and L_i is the tuple i 's reference class label. When two separate labels are the same, the match returns 0; when they are different, the match returns 1.

2) STATISTICAL MEAN

This value represents the average of the fitness values obtained when all runs of the optimization method are exhausted, as shown in the equation below.

$$Mean = \frac{1}{M} \sum_{i=1}^M g^{*i} \quad (13)$$

here g^{*i} is the best value obtained during the i th run.

3) STATISTICAL BEST

This value represents the smallest of all fitness values accumulated across the iterations. The formula can be stated as:

$$min_{1 < i \leq M} \{g^i\} \quad (14)$$

4) STATISTICAL WORST

It is the maximum (worst) value obtained from all fitness values obtained throughout the iterations, as shown in the

following equation:

$$max_{1 < i \leq M} \{g^i\} \quad (15)$$

5) STATISTICAL STANDARD DEVIATION

The standard deviation indicates the variation between the best answers found throughout the model's major iterations. It describes the stability and robustness of an optimization technique and may be expressed as follows.

$$std.dev. = \sqrt{\frac{1}{M-1} \sum (g^i - Means)^2} \quad (16)$$

6) AVERAGE SELECTION SIZE

It is calculated by dividing the total number of features by the average number of picked features after each run. The formula is given as follows:

$$Mean = \frac{1}{M} \sum_{i=1}^M size(g^*) \quad (17)$$

Here, the number of features in the testing data set that were chosen represented by $size()$.

7) PROCESSING TIME

It is described as the optimization algorithm's running time averaged on all runs. The processing time can be determined using:

$$Mean = \frac{1}{M} \sum_{i=1}^M RunTime_{o,i} \quad (18)$$

where M specifies the iterations for optimization method o , and $RunTime_{o,i}$, the optimization algorithm o 's real calculation time at run number i .

8) NON-PARAMETRIC TESTING

This study uses Wilcoxon's signed rank test that seeks to discover substantial differences between the means of two samples. The test produces a p-value parameter, which checks the significance level of the two algorithms.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) COMPARITIVE ANALYSIS OF DSO, S-BDSO, AND V-BDSO

Table 3 demonstrates the comparison between the proposed binary DSOs approaches in the context of error rate in classification. It is evident from Table 3 that V-BDSO method outperforms the native one. Over all datasets utilized in this investigation, the native DSO does not outperform any binary DSO. On the other side, S-BDSO outperforms the native DSO across all datasets.

On all datasets, the suggested V-BDSO surpassed the native DSO in terms of average selection size, and the performance of S-BDSO is likewise competitive with V-BDSO, as shown in Table 4. The V-BDSO performed statistically better on almost all datasets in comparison to S-BDSO and in the penglungEW dataset, S-BDSO outperformed V-BDSO by providing 163.73 average selection size whereas V-BDSO demonstrates 167.60 average selection size.

TABLE 3. Classification accuracy obtained from DSO, S-BDSO, and V-BDSO on different data sets.

S. No	Dataset	DSO	S-BDSO	V- BDSO
1	Z1	0.962	0.970	0.971
2	Z2	0.93	0.955	0.962
3	Z3	0.852	0.899	0.906
4	Z4	0.954	0.959	0.965
5	Z5	0.93	0.953	0.968
6	Z6	0.729	0.731	0.863
7	Z7	0.671	0.713	0.73
8	Z8	0.792	0.822	0.829
9	Z9	0.886	0.919	0.958
10	Z10	0.923	0.966	0.979
11	Z11	0.8	0.838	0.888
12	Z12	0.828	0.852	0.956
13	Z13	0.854	0.897	0.942
14	Z14	0.967	0.978	0.987
15	Z15	0.86	0.923	0.941
16	Z16	0.768	0.780	0.882
17	Z17	0.763	0.800	0.803
18	Z18	0.924	0.952	0.961
19	Z19	0.718	0.741	0.773
20	Z20	0.954	0.988	0.977
21	Z21	0.854	0.941	0.97
Average		0.853	0.885	0.915

TABLE 4. Average selection size of DSO, S-BDSO, and V-BDSO on all data sets.

Dataset	DSO	S-BDSO	V- BDSO
Z1	6.08	3.80	3.00
Z2	22.45	16.13	11.00
Z3	134.27	89.87	86.40
Z4	111.67	90.20	79.80
Z5	11.67	5.27	3.53
Z6	11.80	7.60	7.58
Z7	8.60	7.40	6.20
Z8	10.40	8.67	6.00
Z9	23.00	14.67	10.40
Z10	31.53	19.40	14.75
Z11	11.67	9.73	7.67
Z12	12.33	9.09	7.47
Z13	234.53	163.73	167.60
Z14	214.20	145.07	85.78
Z15	43.93	31.40	16.73
Z16	17.87	9.93	7.27
Z17	7.54	7.00	5.13
Z18	10.80	6.53	5.13
Z19	37.00	21.80	14.50
Z20	11.40	7.87	5.00
Z21	11.13	7.00	5.00

Tables 5-8 provide the results of execution on all data sets and these results are normalized in the range 0 and 1. According to the results shown in Table 5, DSO demonstrated superior performance on 7 datasets on the mean fitness measure, while V-BDSO outperformed DSO and S-BDSO on 14 datasets. As may be seen from the statistical best fitness results in Table 6, DSO has the best results on 13 datasets,

TABLE 5. For the different runs for DSO, S-BDSO, and V-BDSO, Statistical mean fitness on the different data sets.

S. No	Dataset	DSO	S-BDSO	V- BDSO
1	Z1	0.044	0.041	0.035
2	Z2	0.045	0.042	0.036
3	Z3	0.118	0.115	0.109
4	Z4	0.044	0.053	0.047
5	Z5	0.065	0.062	0.056
6	Z6	0.141	0.156	0.150
7	Z7	0.267	0.273	0.271
8	Z8	0.175	0.182	0.176
9	Z9	0.107	0.104	0.098
10	Z10	0.075	0.072	0.066
11	Z11	0.135	0.132	0.126
12	Z12	0.032	0.042	0.039
13	Z13	0.098	0.095	0.089
14	Z14	0.027	0.024	0.018
15	Z15	0.091	0.088	0.082
16	Z16	0.161	0.158	0.152
17	Z17	0.115	0.112	0.106
18	Z18	0.085	0.094	0.091
19	Z19	0.085	0.082	0.076
20	Z20	0.043	0.037	0.031
21	Z21	0.041	0.038	0.032

TABLE 6. Statistical best fitness for DSO, S-BDSO, and V-BDSO on different data sets.

S. No	Dataset	DSO	S-BDSO	V- BDSO
1	Z1	0.031	0.024	0.018
2	Z2	0.032	0.025	0.019
3	Z3	0.092	0.105	0.098
4	Z4	0.031	0.044	0.037
5	Z5	0.039	0.052	0.045
6	Z6	0.133	0.142	0.139
7	Z7	0.251	0.255	0.257
8	Z8	0.159	0.172	0.165
9	Z9	0.094	0.087	0.081
10	Z10	0.062	0.055	0.049
11	Z11	0.109	0.117	0.115
12	Z12	0.015	0.024	0.021
13	Z13	0.072	0.075	0.078
14	Z14	0.014	0.007	0.001
15	Z15	0.078	0.071	0.065
16	Z16	0.148	0.141	0.135
17	Z17	0.089	0.092	0.095
18	Z18	0.068	0.075	0.074
19	Z19	0.072	0.065	0.059
20	Z20	0.027	0.020	0.014
21	Z21	0.015	0.038	0.025

whereas V-BDSO performed better on 8 datasets. The results on the statistical worst fitness measures are shown in Table 7, where it can be analyzed that V-BDSO performed significantly better than S-BDSO and DSO on 13 datasets, whereas S-BDSO outperformed other algorithms on 6 datasets only. Table 8 illustrates the statistical standard deviation fitness measure. As it can be seen from these results, V-BDSO

TABLE 7. Statistical worst for DSO, S-BDSO, and V-BDSO on different data sets.

S. No	Dataset	DSO	S-BDSO	V- BDSO
1	Z1	0.061	0.056	0.047
2	Z2	0.062	0.057	0.048
3	Z3	0.135	0.121	0.126
4	Z4	0.053	0.068	0.059
5	Z5	0.082	0.077	0.068
6	Z6	0.162	0.176	0.171
7	Z7	0.279	0.287	0.288
8	Z8	0.188	0.199	0.197
9	Z9	0.124	0.119	0.110
10	Z10	0.092	0.087	0.078
11	Z11	0.152	0.147	0.138
12	Z12	0.044	0.058	0.052
13	Z13	0.115	0.109	0.101
14	Z14	0.044	0.039	0.030
15	Z15	0.108	0.103	0.094
16	Z16	0.178	0.173	0.164
17	Z17	0.128	0.118	0.122
18	Z18	0.097	0.106	0.102
19	Z19	0.102	0.097	0.088
20	Z20	0.057	0.052	0.043
21	Z21	0.064	0.054	0.052

TABLE 8. Statistical standard deviation measures for DSO, S-BDSO, and V-BDSO on different data sets.

S. No	Dataset	DSO	S-BDSO	V- BDSO
1	Z1	0.017	0.012	0.005
2	Z2	0.015	0.01	0.003
3	Z3	0.023	0.018	0.011
4	Z4	0.033	0.028	0.021
5	Z5	0.031	0.026	0.019
6	Z6	0.018	0.025	0.030
7	Z7	0.039	0.034	0.027
8	Z8	0.021	0.016	0.009
9	Z9	0.023	0.015	0.008
10	Z10	0.018	0.013	0.006
11	Z11	0.029	0.024	0.017
12	Z12	0.033	0.042	0.045
13	Z13	0.021	0.016	0.009
14	Z14	0.025	0.022	0.013
15	Z15	0.014	0.009	0.002
16	Z16	0.02	0.015	0.008
17	Z17	0.054	0.049	0.042
18	Z18	0.027	0.022	0.015
19	Z19	0.013	0.008	0.001
20	Z20	0.046	0.041	0.034
21	Z21	0.016	0.006	0.012

outperformed S-BDSO and DSO on 16 datasets, whereas S-BDSO performed better on 2 datasets only.

Table 9 shows the average computing time of DSOs’ technique, which is the time it takes to arrive at a near-optimal solution. The results of this table demonstrate that the computational speed of S-BDSO is competitive to DSO, whereas compared to DSO, V-BDSO takes longer to compute than S-BDSO.

TABLE 9. Average computational time of DSO, S-BDSO and V-BDSO.

S. No	Dataset	DSO	S-BDSO	V- BDSO
1	Z1	50.52	39.81	51.19
2	Z2	57.39	39.69	57.11
3	Z3	61.06	48.44	66.56
4	Z4	4486.79	3334.25	3331.92
5	Z5	46.69	35.62	42.80
6	Z6	56.71	47.38	58.00
7	Z7	47.36	47.55	64.15
8	Z8	33.72	35.10	39.75
9	Z9	36.98	39.14	45.48
10	Z10	255.07	486.46	296.53
11	Z11	33.71	30.13	42.77
12	Z12	48.31	45.83	51.91
13	Z13	48.61	50.98	45.80
14	Z14	402.78	551.56	404.23
15	Z15	33.86	35.58	42.45
16	Z16	35.57	35.43	40.73
17	Z17	50.78	48.25	56.06
18	Z18	34.11	40.53	46.55
19	Z19	639.10	747.25	665.05
20	Z20	34.89	42.51	40.40
21	Z21	34.04	31.50	41.34

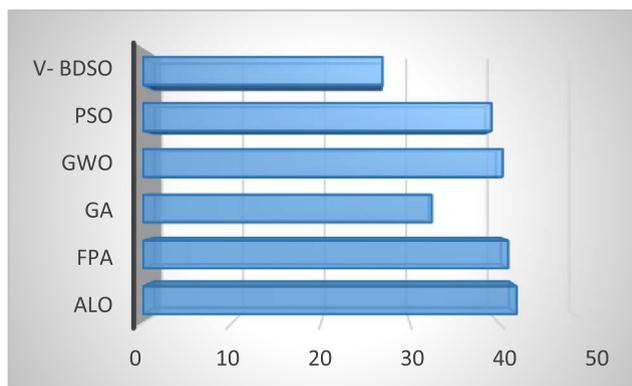


FIGURE 8. Average of the number of features selected on all datasets.

2) COMPARATIVE ANALYSIS OF RESULTS WITH STANDARD METHODOLOGIES

From the last section, it can be observed that V- BDSO provided the lowest classification error rate as well as the average selection size in comparison to DSO and S-BDSO. In this section, the performance of the best strategy, V-BDSO, is compared to the performance of several state-of-the-art ways that are frequently used to tackle the feature selection issue in the literature [18], [21], [28]. In terms of classification error rate, table 10 shows the findings of V-BDSO, ALO, FPA, GA, GWO, and PSO. As it may be observed in this table, V-BDSO outperforms ALO, FPA, PSO, GWO, and GA in terms of classification error rate on all datasets. This higher performance demonstrates the suggested approach’s ability to efficiently discover the search space’s optima.

TABLE 10. Comparison of V-BDSO's classification accuracy with standard methods on different datas sets.

Dataset	ALO	FPA	GA	GWO	PSO	V- BDSO
Z1	0.959	0.955	0.960	0.958	0.961	0.971
Z2	0.939	0.938	0.949	0.937	0.938	0.962
Z3	0.846	0.87	0.870	0.858	0.855	0.906
Z4	0.950	0.949	0.942	0.946	0.946	0.965
Z5	0.937	0.932	0.941	0.933	0.923	0.968
Z6	0.706	0.687	0.731	0.725	0.747	0.863
Z7	0.698	0.702	0.694	0.693	0.696	0.730
Z8	0.777	0.788	0.787	0.777	0.779	0.829
Z9	0.859	0.89	0.894	0.868	0.88	0.958
Z10	0.901	0.911	0.921	0.914	0.92	0.979
Z11	0.786	0.784	0.816	0.763	0.791	0.888
Z12	0.818	0.811	0.799	0.827	0.842	0.956
Z13	0.807	0.835	0.672	0.834	0.814	0.942
Z14	0.971	0.972	0.976	0.967	0.968	0.987
Z15	0.849	0.883	0.875	0.862	0.867	0.941
Z16	0.788	0.812	0.810	0.785	0.784	0.882
Z17	0.759	0.757	0.761	0.754	0.752	0.803
Z18	0.926	0.913	0.933	0.920	0.926	0.961
Z19	0.707	0.719	0.692	0.710	0.719	0.773
Z20	0.954	0.948	0.954	0.948	0.952	0.977
Z21	0.922	0.953	0.929	0.952	0.945	0.970
0.850	0.857	0.852	0.853	0.857	0.915	

TABLE 11. On different datas sets, the average selection size of V-BDSO in comparison to the standard methods.

Dataset	FPA	ALO	GA	GWO	PSO	V- BDSO
Z1	6.40	7.00	6.10	6.90	5.70	3.00
Z2	21.60	24.27	12.20	19.00	18.33	11.00
Z3	125.80	132.00	98.90	109.60	104.93	86.40
Z4	109.00	95.00	94.10	106.00	109.40	79.80
Z5	10.80	9.87	7.10	9.80	10.80	4.50
Z6	11.80	12.87	8.10	12.07	9.00	7.58
Z7	9.00	8.40	7.10	7.53	9.40	6.20
Z8	10.00	10.40	6.60	8.80	9.07	6.00
Z9	19.00	20.13	13.50	17.33	19.20	10.40
Z10	28.80	35.80	18.00	31.60	25.60	14.75
Z11	14.60	13.33	8.90	11.80	11.73	7.60
Z12	11.20	11.27	7.68	11.27	10.87	7.40
Z13	196.40	172.07	153.00	162.80	183.33	167.60
Z14	164.80	187.80	149.40	203.60	171.60	85.70
Z15	42.00	48.00	30.30	41.60	37.60	16.70
Z16	12.20	13.87	7.00	13.20	12.07	7.20
Z17	8.00	8.80	5.80	7.53	6.73	5.20
Z18	8.60	8.40	5.80	8.47	9.33	5.20
Z19	30.20	39.60	30.40	36.60	35.80	23.50
Z20	8.60	11.07	6.73	10.73	10.07	5.00
Z21	13.00	11.67	5.35	12.40	11.80	5.00

Table 11 shows the average selection size using V-BDSO and other approaches. In comparison to the other techniques used in this study, V-BDSO performs substantially better by picking a smaller number of features. V-BDSO performed better on all datasets, according to the findings provided in this table except penglungEW and spectEW, where GA

showed better performance by selecting less number of attributes. The suggested strength of V-BDSO resides in its increased exploration and exploitation capability, which enables it to reduce redundant characteristics before intensely searching the high-performance regions of the feature space.

TABLE 12. In comparison with state-of-the-art approaches on different data sets, statistical mean fitness was measured for the proposed approaches.

Dataset	ALO	FPA	GA	GWO	PSO	V- BDSO
Z1	0.048	0.086	0.045	0.047	0.049	0.035
Z2	0.067	0.120	0.055	0.067	0.069	0.036
Z3	0.160	0.188	0.133	0.151	0.152	0.109
Z4	0.055	0.095	0.062	0.061	0.062	0.047
Z5	0.069	0.178	0.064	0.081	0.084	0.056
Z6	0.302	0.412	0.280	0.256	0.255	0.150
Z7	0.304	0.358	0.308	0.306	0.316	0.267
Z8	0.227	0.338	0.215	0.224	0.225	0.176
Z9	0.144	0.195	0.108	0.125	0.125	0.098
Z10	0.107	0.221	0.084	0.083	0.087	0.066
Z11	0.218	0.316	0.187	0.214	0.218	0.126
Z12	0.188	0.362	0.204	0.166	0.163	0.032
Z13	0.197	0.335	0.129	0.192	0.191	0.089
Z14	0.034	0.084	0.028	0.038	0.041	0.018
Z15	0.156	0.214	0.127	0.136	0.139	0.082
Z16	0.217	0.257	0.191	0.217	0.217	0.152
Z17	0.248	0.344	0.242	0.252	0.251	0.106
Z18	0.079	0.153	0.072	0.077	0.077	0.085
Z19	0.300	0.372	0.320	0.286	0.285	0.076
Z20	0.055	0.145	0.052	0.053	0.052	0.031
Z21	0.086	0.188	0.074	0.061	0.064	0.032

TABLE 13. Statistical best fitness measure for the proposed approach on different data sets.

Dataset	ALO	FPA	GA	GWO	PSO	V- BDSO
Z1	0.039	0.075	0.034	0.039	0.041	0.018
Z2	0.058	0.109	0.044	0.059	0.061	0.019
Z3	0.151	0.177	0.122	0.143	0.144	0.092
Z4	0.046	0.084	0.051	0.053	0.054	0.030
Z5	0.06	0.167	0.053	0.073	0.076	0.039
Z6	0.293	0.401	0.269	0.248	0.247	0.133
Z7	0.251	0.347	0.297	0.298	0.308	0.265
Z8	0.159	0.327	0.204	0.216	0.217	0.197
Z9	0.135	0.184	0.097	0.117	0.117	0.081
Z10	0.098	0.21	0.073	0.075	0.079	0.049
Z11	0.209	0.305	0.176	0.206	0.21	0.109
Z12	0.179	0.351	0.193	0.158	0.155	0.015
Z13	0.188	0.324	0.118	0.184	0.183	0.072
Z14	0.025	0.073	0.017	0.03	0.033	0.001
Z15	0.147	0.203	0.116	0.128	0.131	0.065
Z16	0.208	0.246	0.18	0.209	0.209	0.135
Z17	0.239	0.333	0.231	0.244	0.243	0.089
Z18	0.07	0.142	0.061	0.069	0.069	0.068
Z19	0.291	0.361	0.309	0.278	0.277	0.059
Z20	0.046	0.134	0.041	0.045	0.044	0.014
Z21	0.077	0.177	0.063	0.053	0.056	0.055

In Tables 12-15, the statistical measurements acquired from several runs of the algorithms on all data sets are shown, and these results are standardized in the range 0 to 1. It can be observed from Tables 12 and 14 that V-BDSO outperformed ALO, FPA, PSO, GWO, and GA in the mean and worst fitness measure on all datasets whereas in the best fitness criteria, V-BDSO outperformed

other algorithms on most (eighteen out of twenty-one) of the datasets as shown in Table 13. Table 15 shows an overview of the statistical standard deviation measure findings obtained for all datasets. The performance of V-BDSO is competitive in comparison to state-of-the-art techniques (ALO, FPA, GA, GWO, and PSO), as seen in this table.

TABLE 14. Statistical worst fitness for the proposed approach in comparison with other approaches on different data sets.

Dataset	ALO	FPA	GA	GWO	PSO	V- BDSO
Z1	0.062	0.105	0.062	0.063	0.076	0.047
Z2	0.081	0.139	0.072	0.083	0.096	0.048
Z3	0.174	0.207	0.152	0.167	0.179	0.121
Z4	0.069	0.114	0.079	0.077	0.089	0.059
Z5	0.083	0.197	0.081	0.097	0.111	0.068
Z6	0.316	0.431	0.297	0.272	0.282	0.162
Z7	0.318	0.377	0.325	0.322	0.343	0.279
Z8	0.241	0.357	0.232	0.247	0.252	0.188
Z9	0.158	0.214	0.125	0.141	0.152	0.116
Z10	0.121	0.244	0.101	0.099	0.114	0.078
Z11	0.232	0.335	0.204	0.231	0.245	0.138
Z12	0.202	0.381	0.221	0.182	0.191	0.044
Z13	0.211	0.354	0.146	0.208	0.218	0.101
Z14	0.048	0.103	0.045	0.054	0.068	0.036
Z15	0.173	0.233	0.144	0.152	0.166	0.094
Z16	0.231	0.276	0.208	0.233	0.244	0.164
Z17	0.262	0.363	0.259	0.268	0.278	0.118
Z18	0.093	0.172	0.089	0.093	0.104	0.097
Z19	0.314	0.391	0.337	0.302	0.312	0.088
Z20	0.069	0.164	0.069	0.069	0.079	0.043
Z21	0.122	0.207	0.091	0.077	0.091	0.044

TABLE 15. Statistical standard deviation measures for the proposed approach in comparison with the standard approaches on different data sets.

Dataset	ALO	FPA	GA	GWO	PSO	V- BDSO
Z1	0.009	0.024	0.007	0.012	0.011	0.005
Z2	0.006	0.043	0.006	0.007	0.007	0.003
Z3	0.019	0.049	0.007	0.018	0.027	0.011
Z4	0.022	0.021	0.134	0.050	0.065	0.031
Z5	0.031	0.058	0.013	0.012	0.018	0.019
Z6	0.030	0.052	0.021	0.014	0.028	0.018
Z7	0.016	0.024	0.013	0.021	0.024	0.027
Z8	0.011	0.095	0.039	0.004	0.046	0.009
Z9	0.024	0.048	0.018	0.031	0.027	0.008
Z10	0.022	0.075	0.055	0.033	0.059	0.006
Z11	0.027	0.034	0.014	0.002	0.028	0.017
Z12	0.034	0.017	0.035	0.033	0.029	0.033
Z13	0.020	0.056	0.007	0.024	0.036	0.009
Z14	0.017	0.055	0.009	0.028	0.021	0.013
Z15	0.005	0.036	0.004	0.013	0.013	0.002
Z16	0.014	0.067	0.012	0.034	0.014	0.008
Z17	0.036	0.126	0.017	0.048	0.051	0.042
Z18	0.023	0.027	0.010	0.021	0.027	0.015
Z19	0.005	0.007	0.008	0.003	0.002	0.001
Z20	0.073	0.109	0.017	0.045	0.076	0.036
Z21	0.006	0.003	0.002	0.007	0.005	0.004

The results of V-BDSO and other algorithms in terms of computing time are shown in Table 16. On thirteen datasets, FPA has the fastest computational time, whereas GA and

GWO each have the fastest computational time on four datasets. In compared to the other techniques, V-BDSO needs much more processing time because of the V-shaped transfer

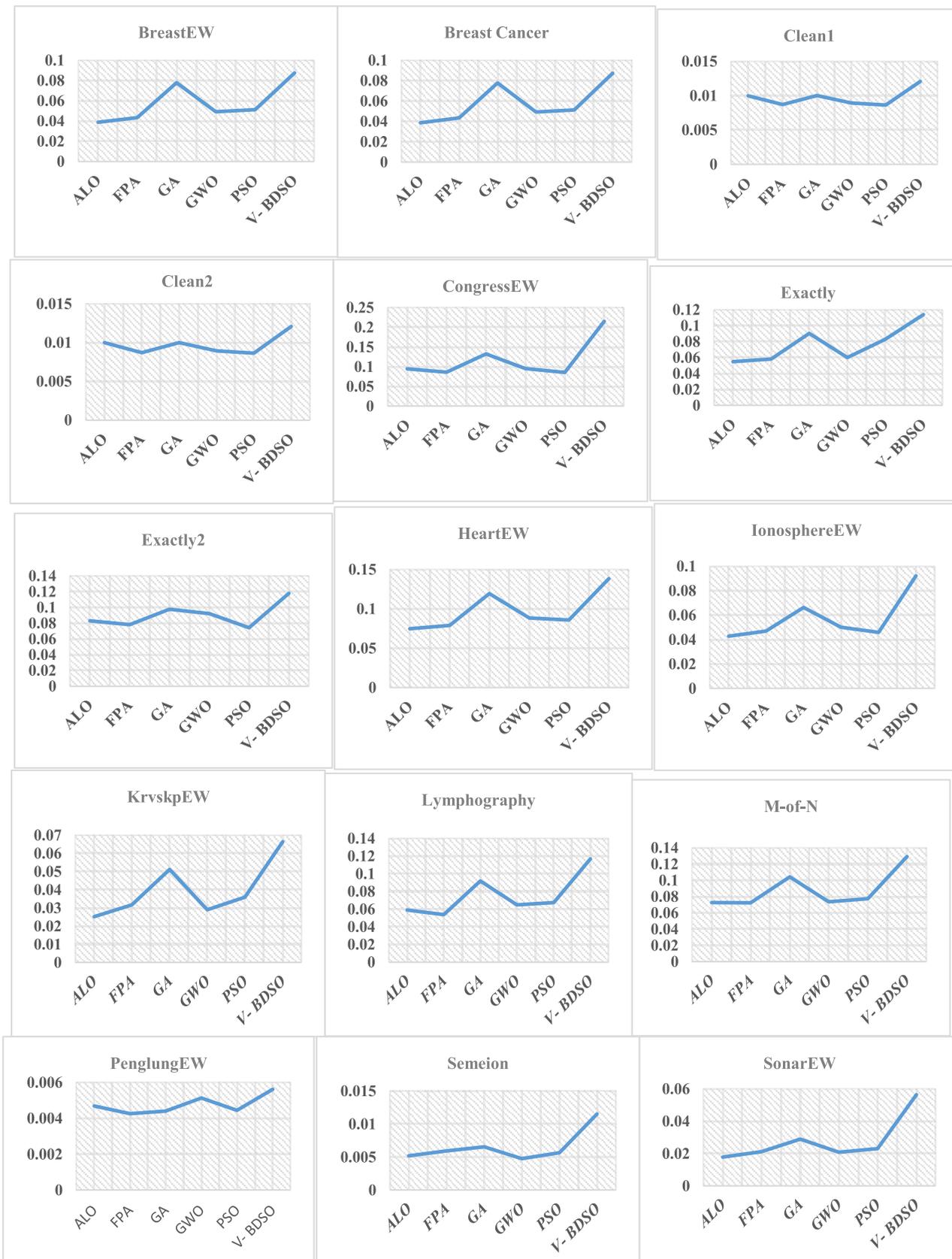


FIGURE 9. Accuracy to feature selected ratio for all 21 datasets.

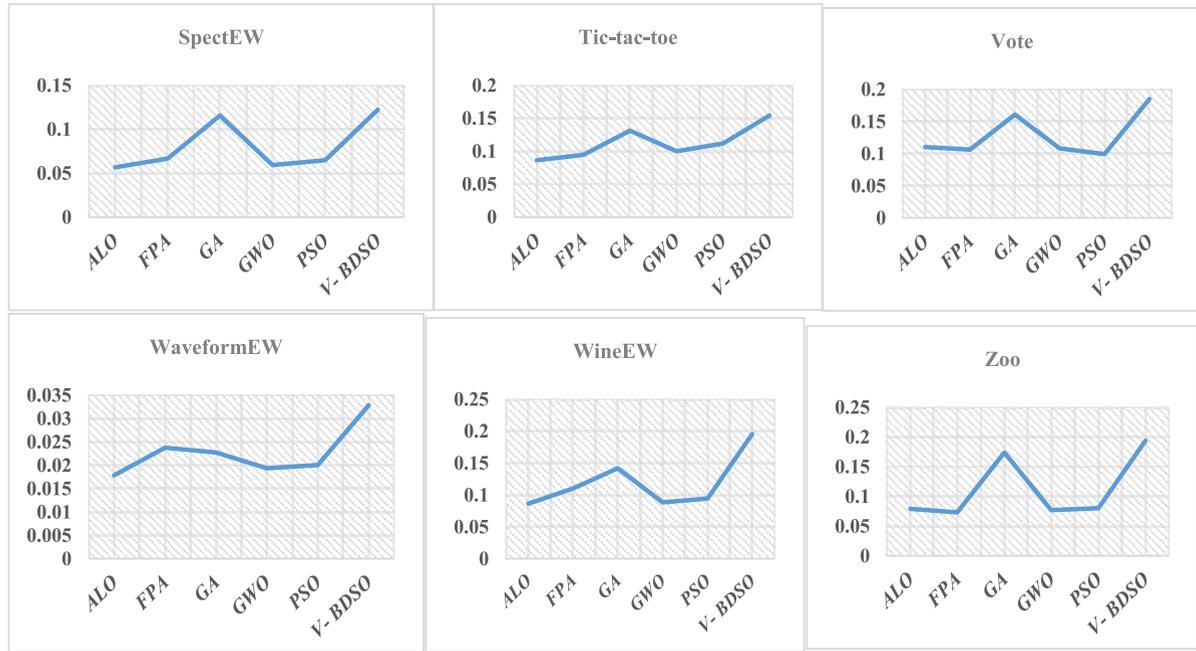


FIGURE 9. (Continued.) Accuracy to feature selected ratio for all 21 datasets.

TABLE 16. Comparison of computational time on different data sets.

Dataset	ALO	FPA	GA	GWO	PSO	V- BDSO
Z1	4.9	2.77	2.34	3.45	2.39	51.19
Z2	2.87	2.89	2.37	3.61	2.43	57.11
Z3	5.31	4.66	4.07	3.39	3.66	66.56
Z4	610.83	500.17	279.32	158.67	227.16	3331.92
Z5	2.88	2.26	2.81	3.32	2.64	42.8
Z6	3.92	3.34	3.85	4.04	4.53	58
Z7	4.22	3.32	4.15	4.52	4.82	64.15
Z8	2.69	2.06	2.46	3.09	2.49	39.75
Z9	3.14	2.31	2.58	3.25	2.64	45.48
Z10	18.16	32	10.42	9.53	17.16	296.53
Z11	2.68	2.03	2.46	2.98	3.04	42.77
Z12	4.08	3.32	3.53	3.39	4.56	51.91
Z13	7.65	2.28	2.51	2.49	2.56	45.8
Z14	28.41	46.84	13.1	31.67	24.51	404.23
Z15	3.3	2.17	2.39	2.97	2.62	42.45
Z16	2.88	2.07	2.45	3	2.4	40.73
Z17	4.36	3.81	3.28	4.38	4.49	56.06
Z18	2.89	2.09	2.62	3.25	2.57	46.55
Z19	40.51	82.17	23.48	20.63	90.09	665.05
Z20	2.68	2.02	2.46	3.13	2.45	40.4
Z21	2.79	2.08	2.33	3.25	2.24	41.34

function, which allows V-BDSO to map continuous values to binary values and requires considerable computational time. Table 17 shows the p values at 5% significance level found in most cases; the results are significant at $p \leq 0.05$ in the vast majority of cases.

Figure 8 shows the average number of features selected from all datasets. It can be clearly observed that V-DSDO selects the lowest number of features. Figure 9 depicts the accuracy to feature selection ratio on individual data sets, the results clearly indicate that the proposed V-BDSO algorithm

TABLE 17. Wilcoxon test for the average fitness.

	ALO	FPA	GA	GWO	PSO	DSO	S-DSO	V-DSO
DSO	2E-04	1E-05	3E-04	2E-04	1E-04	--	1E-05	1E-05
S-DSO	1E-04	1E-05	2E-04	1E-04	1E-04	1E-05	--	1E-05
V-DSO	6E-05	1E-05	1E-04	6E-05	6E-05	1E-05	1E-05	--

has achieved higher accuracy on feature selected ratio value on all datasets.

VI. CONCLUSION

In this research, binary variations of the Drone Squadron Optimization (DSO) are suggested and used to choose features in wrapper mode. The continuous version of DSO is transformed into its binary variant utilising V-shaped and S- shaped transfer functions. The new binary techniques were tested against well-known nature-inspired algorithms such as Ant Lion Optimization (ALO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Flower Pollination Algorithm (FPA), and Grey Wolf Optimizer (GWO) on 21 benchmark datasets from the UCI library. To examine different areas of performance, the evaluation uses a set of assessment criteria. The presented findings suggest that the proposed binary DSO technique can optimally explore the space of features and converge to the best solution faster than other algorithms. Moreover, the results prove that the binary algorithm proposed based on V-shaped function achieves an average classification accuracy of 91.5% and an average selection size of 26.4, whereas the binary algorithm proposed based on S-shaped function achieves an average classification accuracy of 88.5% and an average selection size of 32.4. Hence, V shaped function performs better than the S-shaped one.

It might be interesting to combine the DSO method with another population-based metaheuristic algorithm in future investigations. Furthermore, examining the performance of the DSO approach when used on considerably higher-dimensional datasets will be a great contribution.

REFERENCES

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
 [2] D. Y. Harvey and M. D. Todd, "Automated feature design for numeric sequence classification by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 474–489, Aug. 2015.
 [3] H. Liu and H. Motoda, *Feature Extraction. Construction Selection: A Data Mining Perspective*, vol. 453. Cham, Switzerland: Springer, 1998.
 [4] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-Filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man Cybern.*, vol. 37, no. 1, pp. 70–76, Feb. 2007.
 [5] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.
 [6] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.
 [7] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[8] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2010.
 [9] R. Dash, P. K. Dash, and R. Bisoi, "A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction," *Swarm Evol. Comput.*, vol. 19, pp. 25–42, Dec. 2014.
 [10] L. D. S. Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 989–996, May 2006.
 [11] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov. 1995, pp. 1942–1948.
 [12] J. Ilonen, J.-K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," *Neural Process. Lett.*, vol. 17, no. 1, pp. 93–105, 2003.
 [13] S. Arora and S. Singh, "Node localization in wireless sensor networks using butterfly optimization algorithm," *Arabian J. Sci. Eng.*, vol. 42, no. 8, pp. 3325–3335, Aug. 2017.
 [14] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2018.
 [15] J.-F. Fu, R. G. Fenton, and W. L. Cleghorn, "A mixed integer-discrete-continuous programming method and its application to engineering design optimization," *Eng. Optim.*, vol. 17, no. 4, pp. 263–280, 1991.
 [16] S. C. Yusta, "Different metaheuristic strategies to solve the feature selection problem," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 525–534, 2009.
 [17] I. A. Gheys and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, 2010.
 [18] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.
 [19] H. M. Zawbaa, E. Emary, and B. Parv, "Feature selection based on antlion optimization algorithm," in *Proc. 3rd World Conf. Complex Syst. (WCCS)*, Nov. 2015, pp. 1–7.
 [20] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
 [21] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.
 [22] S. A.-F. Sayed, E. Nabil, and A. Badr, "A binary clonal flower pollination algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 77, pp. 21–27, Jul. 2016.
 [23] Q. Li, H. Chen, H. Huang, X. Zhao, Z. Cai, C. Tong, W. Liu, and X. Tian, "An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," *Comput. Math. Methods Med.*, vol. 2017, pp. 1–15, Jan. 2017.
 [24] H. Chen, W. Jiang, C. Li, and R. Li, "A heuristic feature selection approach for text categorization by using chaos optimization and genetic algorithm," *Math. Problems Eng.*, vol. 2013, Dec. 2013, Art. no. 524017.
 [25] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.
 [26] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, Feb. 2018.
 [27] E. Zorarpac and S. A. Ozel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Exp. Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016.
 [28] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.
 [29] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 171–188, Jan. 2019.

- [30] H. M. Zawbaa, E. Emary, and C. Grosan, "Feature selection via chaotic antlion optimization," *PLOS ONE*, vol. 11, no. 3, 2016, Art. no. e0150652.
- [31] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.
- [32] A. Lewis, S. Mostaghim, and M. Randall, "Evolutionary population dynamics and multi-objective optimisation problems," *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. Hershey, PA, USA: IGI Global, 2008, pp. 185–206.
- [33] M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A.-Z. Alamm, and S. Mirjalili, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowl.-Based Syst.*, vol. 145, pp. 25–45, Apr. 2018.
- [34] R. Hans and H. Kaur, "Binary multi-verse optimization (BMVO) approaches for feature selection," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 6, no. 1, p. 91, 2020.
- [35] P. Hu, J. S. Pan, and S. C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105746.
- [36] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," *Eng. Comput.*, vol. 37, pp. 1–30, May 2020.
- [37] N. Neggaz, A. A. Ewees, E. M. Abd, and M. Mafarja, "Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection," *Exp. Syst. Appl.*, vol. 145, May 2020, Art. no. 113103.
- [38] B. A. Ş. Emine and E. Ülker, "An efficient binary social spider algorithm for feature selection problem," *Exp. Syst. Appl.*, vol. 146, May 2020, Art. no. 113185.
- [39] R. K. Agrawal, B. Kaur, and S. Sharma, "Quantum based whale optimization algorithm for wrapper feature selection," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106092.
- [40] V. V. de Melo and W. Banzhaf, "Drone squadron optimization: A novel self adaptive algorithm for global numerical optimization," *Neural Comput. Appl.*, vol. 30, pp. 1–28, Feb. 2017.
- [41] S. Mirjalili and S. Z. M. Hashim, "BMOA: Binary magnetic optimization algorithm," *Int. J. Mach. Learn. Comput.*, vol. 2, no. 3, p. 204, 2012.
- [42] K. Bache and M. Lichman, "UCI machine learning repository," *School Inf. Comput. Sci.*, Univ. California, Irvine, CA, USA, Tech. Rep., 2013, pp. 1–20. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [43] J. Franklin, "The elements of statistical learning: Data mining, inference and prediction," *Math. Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.
- [44] S. Saremi, S. Mirjalili, and A. Lewis, "How important is a transfer function in discrete heuristic algorithms," *Neural Comput. Appl.*, vol. 26, no. 3, pp. 625–640, 2015.



MANJU KHURANA was born in Hansi, India, in 1987. She received the B.E. degree in computer science from MDU, Rohtak, in 2008, the M.Tech. degree in computer science from Banasthali Vidyapith, Rajasthan, in 2010, and the Ph.D. degree in engineering from the Department of Computer Science, College of Engineering and Technology, Mody University of Science and Technology, Lakshmanagarh, Rajasthan, in 2016. Thereafter she worked with BRCM College of Engineering and Technology, Bahal, Bhiwani, Haryana, as an Assistant Professor, until June 2012. In July 2012, she joined the College of Engineering and Technology, Mody University of Science and Technology, as an Assistant Professor with the Department of Computer Science. In July 2017, she joined the Thapar Institute of Engineering & Technology, Patiala, Punjab, as an Assistant Professor. Her has authored several research papers in reputed international/national journals and conference/seminar proceedings and supervised M.Tech. dissertations. Her research interest includes energy efficient wireless sensor networks.



MANJIT KAUR (Senior Member, IEEE) received the M.E. degree in information technology from Punjab University, Chandigarh, India, in 2011, and the Ph.D. degree from the Thapar Institute of Engineering & Technology, Patiala, India, in 2019. She worked as an Assistant Professor in three well-known universities of India, such as Chandigarh University, Mohali, India, Manipal University Jaipur, Jaipur, India, and Bennett University, Greater Noida, India. In 2021, she moved to the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea, where she is currently affiliated. She has published more than 60 SCI/SCIE indexed articles so far. Her research interests include wireless sensor networks, digital image processing, and metaheuristic techniques. She was in the top 2% list issues by "World Ranking of Top 2% Scientists," in 2021. She was part of the 11 Web of Science/Scopus indexed conferences.



HARPREET SINGH received the B.Tech. degree in computer science and engineering from Guru Nanak Dev University, Amritsar, the M.Tech. degree in computer science and engineering from the National Institute of Technology, Jalandhar, and the Ph.D. degree from Guru Nanak Dev University. His research interests include machine learning and optimization algorithms.



SUCHITA SHARMA received the B.Tech. and M.E. degrees in computer science and engineering from the Green Hills Engineering College, Solan under the affiliation of Himachal Pradesh Technical University (HPTU), Hamirpur, India, and TIET, Patiala, India, respectively. Her research interests include medical image retrieval and biomedical image denoising.



HEUNG-NO LEE (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Los Angeles, CA, USA, in 1993, 1994, and 1999, respectively. From 1999 to 2002, he was with HRL Laboratories, LLC, Malibu, CA, USA, as a Research Staff Member. From 2002 to 2008, he was an Assistant Professor with the University of Pittsburgh, Pittsburgh, PA, USA. In 2009, he moved to the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea, where he is currently affiliated. His research interests include information theory, signal processing theory, blockchain, communications/networking theory, and their application to wireless communications and networking, compressive sensing, future internet, and brain-computer interface. He has received several prestigious national awards, including the Top 100 National Research and Development Award, in 2012, the Top 50 Achievements of Fundamental Researches Award, in 2013, and Science/Engineer of the Month, January 2014.