Received March 30, 2020, accepted April 11, 2020, date of publication April 15, 2020, date of current version May 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988157

Late Acceptance Hill Climbing Based Social Ski Driver Algorithm for Feature Selection

BITANU CHATTERJEE[1], TRINAV BHATTACHARYYA[1], KUSHAL KANTI GHOSH[1], PAWAN KUMAR SINGH[2], (Member, IEEE), ZONG WOO GEEM[3], AND RAM SARKAR[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India
[2]Department of Information Technology, Jadavpur University, Kolkata 700106, India
[3]Department of Energy IT, Gachon University, Seongnam 13120, South Korea

Corresponding author: Zong Woo Geem (geem@gachon.ac.kr)

This work was supported by the Energy Cloud Research and Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT under Grant 2019M3F2A1073164.

ABSTRACT Feature selection (FS) is mainly used as a pre-processing tool to reduce dimensionality by eliminating irrelevant or redundant features to be used for a machine learning or data mining algorithm. In this paper, we have introduced binary variant of a recently proposed meta-heuristic algorithm called Social Ski Driver (SSD) optimization. To the best of our knowledge, SSD has not been used yet in the domain of FS. Two binary variants of SSD are proposed using S-shaped and V-shaped transfer functions. Besides, the exploitation ability of SSD is improved by using a local search method, *called* Late Acceptance Hill Climbing (LAHC). The hybrid meta-heuristic is then converted to binary version by using said transfer functions. The proposed methods are applied on 18 standard UCI datasets and compared with 15 state-of-the-art FS methods. Also to check the robustness of the proposed method, we have applied it to 3 high dimensional microarray datasets and compared with 6 state-of-the-art methods. Achieved results confirm the superiority of the proposed methods compared to other meta-heuristic wrapper based FS methods considered here. Source code of this work is available at https://github.com/consigliere19/SSD-LAHC.

INDEX TERMS Social ski driver optimization, feature selection, late acceptance hill climbing, UCI dataset, meta-heuristic optimization, microarray data.

I. INTRODUCTION

With the recent advances in technology, huge amount of data has become available in different domains of image processing, pattern recognition, and disease diagnosis system [1]. As a consequence, data dimensionality creates a huge impact on the performance of the various machine learning and data mining tasks, both in terms of time and storage needs of the computing devices. In this context, it can noted that there may be some redundancy in the data itself. For example, all the features developed by some means to represent a pattern or an image are not important for the classification or analysis of the same. Here, comes the role of a feature selection (FS) technique. FS is a data pre-processing step which attempts to eliminate all the irrelevant and redundant features [2] from the underlying dataset or feature vector, and thereby reduces required processing time and storage space. Due to the elimination of the non-informative features, FS technique also helps in enhancing the performance of the corresponding machine learning tasks [3].

Based on the usage of learning algorithm, FS techniques can broadly be divided into two categories [4]: filter and wrapper. Filter methods do not use any learning technique during elimination (selection) of the irrelevant (important) features, rather use different scoring criteria [5]–[8] to rank the features indicating their importance in order. Wrapper methods use learning techniques (such as classifiers) as a part of the selection and evaluate the subset of the selected features [9]–[14]. Filter methods are faster but wrapper methods, in general, perform much better [4]. Three factors must be considered while using a wrapper based FS model: choice of classifier, evaluation criteria of feature subsets (such as accuracy), and a searching (or optimization) technique to find the best subset of features [15].

FS is considered to be an NP-complete combinatorial optimization problem. Generating all possible subsets and

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

evaluating those are not suitable for large datasets. This is because for a dataset containing n features, 2^n feature subsets will be generated and evaluating all of those require a great computational cost. The search for the best performing feature subset can also be carried out randomly. A heuristic search strategy performs a guided search which does not always find the best solution but tries to obtain a near-optimal solution in computationally reasonable time. Heuristic approaches are classified into two types- specific heuristics which are designed for a particular problem, and general-purposed meta-heuristics which are designed to solve a wide range of problems [16].

In this paper, we have made an attempt to propose a FS method following a recently proposed meta-heuristic algorithm called Social Ski Driver (SSD) optimization algorithm [17], which has produced significantly good results as compared to other optimization algorithms in the literature. To enhance its exploitation ability, we have embedded a local search algorithm called Late Acceptance Hill Climbing (LAHC) into SSD algorithm. The contributions of this paper are highlighted below:

- Application of SSD for FS problem for the first time to the best of our knowledge.
- Enhancing the exploitation ability of SSD by using LAHC.
- Use of two different transfer functions: S-shaped and V-shaped for proposing the FS methods
- Validation of the proposed model on 18 standard UCI datasets and comparison with 15 state-of-the-art FS methods and on 3 standard microarray datasets and comparison with 6 state-of-the-art FS methods.

The rest of the paper is organized as follows: Section II provides a brief review of the past FS methods. Section III provides detailed description of the proposed FS methods. The results obtained by the FS versions of SSD are explained in Section IV. Section V provides the comparison of the proposed model with 15 state-of-the-art FS methods.Section VI shows the applicability of the proposed methods on high dimensional microarray datasets. Lastly, Section VII concludes our work and provides directions for future extension of this work.

II. LITERATURE SURVEY

FS is an optimization problem where the aim is to simultaneously maximize the classification accuracy and minimize the number of selected features. The role of FS is crucial because it helps us to gauge the performance of a machine learning technique. There are several research articles published in the literature which have tried to solve the FS problem, and some of those are described here.

Nature inspired algorithms are popular because of a number of factors: easy to adopt, flexible, not requiring very complex mathematical derivation, and their ability to avoid a local optima. Genetic Algorithm (GA), the oldest algorithm in this category, was used in [18] for the selection of features in automatic pattern classifier. It was further used in [9], and in [19]. Hybrid versions of the GA were subsequently utilized in [20]–[22], and [4]. In [23], a Histogram-Based Multi Objective GA (HMOGA) was proposed for finding informative features from higher-dimensional data. This idea was applied for two previously proposed feature sets for handwritten *Devanagari* numeral recognition problem. In [21], two stages of optimization were involved - the outer optimization stage completed the global search for the best subset of features in a wrapper way, while the inner optimization performed the local search in a filter manner. A tribe competition-based GA (TCbGA) to solve FS problems in pattern classification was proposed in [24]. A new Deluge based GA for FS was also proposed recently in [25].

PSO is a powerful optimization technique introduced in 1995 and further used in [10] for optimization of non-linear functions. A binary version of this algorithm was subsequently used in [26]. Six new transfer functions for binary PSO was introduced in [27]. A hybrid version of the PSO algorithm was used in [28] and [29]. The authors of [30] used a new algorithm called Sentiment Fitness Sum Binary PSO (SCO-FS-BPSO). This article overcame the drawbacks of the binary PSO and was used for sentiment classification.

Simulated Annealing (SA) for optimization problems was introduced in [31], and then used in [32]. The concept of SA was to mimic the annealing of solids. This algorithm was subsequently used by the authors of [33] which incorporated the Firefly Algorithm (FA) with SA to escape from the local optima and increase the quality of the solution. Another optimization technique called Binary Coordinate Ascent (BCA) was proposed in [15]. This method also used two popular feature subset selection (FSS) meta-heuristics namely, Sequential Forward Selection (SFS) and Sequential Floating Forward Selection (SFFS). The authors of [34] introduced a new algorithm called Grasshopper Optimization Algorithm (GOA) for solving optimization problems. This algorithm was mathematically modeled and was inspired by the behavior of grasshopper swarms in nature. The work reported in [35] proposed a hybrid method based on the GOA and used it to optimize the parameters of the SVM classifier and simultaneously find the best feature subset. In [36], the authors have proposed Harmony Search Algorithm (HSA) following the musical performance process. It has three operations: random search, pitch adjustment, and harmony memory. HSA has been applied to different domains [37], including FS [38].

The authors of [14] used the Ant Lion Optimizer (ALO) to solve the FS problem. This algorithm mimicked the hunting behavior of ant lions in nature. Then a hybrid binary ALO was used in [16]. This work used two incremental hill-climbing techniques which are QuickReduct and CEBARKCC. Different versions of the Grey Wolf Optimizer (GWO) were used in [11] and [39]. Authors in [39] used a new algorithm called BGWOPSO which was a binary version of a hybrid form of the GWO and PSO. This technique proved to be relatively better in comparison to its peers. After that, authors in [40] proposed a new GWO algorithm integrated on a two phase mutation to solve the FS problem based on a wrapper method. The work reported in [41] used the Whale Optimization Algorithm (WOA) which had hitherto not been used for FS problems. This algorithm was inspired by the social behavior of humpback whales. This was further used in [16]. In [42], authors proposed a hybrid version of the WOA with SA for FS, where SA was used to improve the best solution found after each iteration of WOA. Subsequently in [43], the authors proposed an enhanced meta-heuristic approach using the GWO and WOA to develop a wrapper-based FS. This method successfully removed the drawbacks of the GWO and WOA and hence was superior compared to both of them. A wrapper-based FS algorithm was designed and substantiated based on the binary variant of the Dragonfly Algorithm (BDA) in [44]. This algorithm mimicked the behavior of dragonflies in nature. The authors of [45] introduced the Ant Colony Optimization (ACO) algorithm as a new meta-heuristic. This algorithm was inspired by the foraging behavior of ant colonies. The work in [46] subsequently used a wrapper-filter FS technique based on the ACO algorithm. In [47], the authors introduced a binary variant of the Grasshopper Optimization Algorithm (BGOA) to solve the FS problem. This work proposed two approaches- the first approach used a V-shaped function and the sigmoid function as transfer functions while the second approach used the mutation operator to exploit the BGOA.

The authors in [48] proposed a method called Correlation based Feature Selection (CFS). This algorithm coupled the evaluation formula with an appropriate correlation measure and a heuristic search strategy. A correlation based Memetic framework was used in [49]. The work used a combination of GA and local search. The Spotted Hyena Optimization (SHO) algorithm was used in [2]. It used two different hybrid models of this algorithm. In the first model, SA was embedded in SHO to improve the best solution after each iteration, while the second model used SA to enhance the final solution obtained by the SHO algorithm. A novel technique called the Gravitational Search Algorithm (GSA) was used in [50]. This algorithm was based on the law of gravity and mass interactions. In the given algorithm, the searcher agents are a collection of masses which interact with each other based on Newtonian gravity and the laws of motion. In [51], a binary version of the GSA was used. In [52], the authors used a GSA-based algorithm with evolutionary crossover and mutation operators to solve the FS task. The work reported in [53] used the Salp Swarm Algorithm (SSA) and Multiple SSA (MSSA). The inspiration of these algorithms came from the swarming behavior of salps when navigating and foraging in oceans. In [1], the authors proposed a wrapper FS method which combined a time-varying number of leaders and followed a binary form of the SSA called TVBSSA, combined with a Random Weight Network (RWN). In [54], the authors proposed the Improved Salp Swarm Algorithm (ISSA) where Opposition Based Learning (OBL) was used to improve the population diversity in the search space and local search was used to enhance the exploitation ability of the algorithm. The work reported in [13] used the Moth Flame Optimization (MFO) algorithm. It was inspired by the navigation method of moths in nature called transverse orientation. Authors of [55] proposed a local search technique called LAHC through Redundancy and Relevancy (LAHCRR) and combined it with Memetic Algorithm (MA) to form Late Hill Climbing based MA (LHCMA) for FS, which reduced the feature dimension to a significant extent. The work reported in [56] proposed a new optimization algorithm called Equilibrium Optimizer (EO) inspired by control volume mass balance models used to estimate both dynamic and equilibrium states. A novel bio-inspired optimization algorithm called the Barnacles Mating Optimizer (BMO) algorithm was proposed in [57]. The inspiration of this algorithm came from the mating behaviour of barnacles in nature and the authors of this work used the Hardy-Weinberg principle for the generation of a new offspring. This principle states that allele and genotype frequencies in a population will remain constant from generation to generation in the absence of other evolutionary influences. In a recent work [58], a new optimization method called the Black Widow Optimization (BWO) algorithm was introduced which was inspired by the mating behavior of black widow spiders. The proposed method includes an exclusive stage, namely, cannibalism where species with inappropriate fitness were omitted, thus leading to early convergence.

Existence of so many meta-heuristic and hybrid meta-heuristic FS methods obviously raises the question about introducing another hybrid meta-heuristic FS method. However, according to No Free Lunch theorem [59] for optimization, there cannot be any single algorithm to solve all the optimization problems. With every new algorithm following any natural phenomena, mainly researchers target to give some new dimension to the algorithm where both exploration and exploitation will have a better trade-off, so that it eventually escapes the local optima and reaches to the global optima. But achieving these goals is not easy, specifically if one wants to propose an algorithm applicable to different domains. This practically motivates the researchers to come up with better methods than the past which, in turn, keeps the research alive in this domain. For our particular given problem, in order to find the best algorithm, NFL should remind us that we need to focus on the particular problem at hand, the assumptions, the priors (extra information), the data and the cost. If we consider an optimization problem, the multi-modal functions have large number of dimensions and finding an optimal value for all those dimensions simultaneously is next-to-impossible. That is why researchers try to solve these type of problems using some meta-heuristic methods where the aim is to get an optimal solution within a reasonable amount of time. Now FS is an optimization problem [42], and there may exist multiple optimum subsets i.e., with same dimension and same classification accuracy. Here also, it is used to find out optimal set of features keeping in mind the storage space and computation time along with the performance of the machine learning algorithm. Therefore, researchers are striving to provide algorithms which can meet these requirements.

This motivated us to propose a hybrid meta-heuristic FS method based on the SSD [17] algorithm.

The concept of the SSD algorithm was inspired by various evolutionary algorithms such as PSO [26], GWO [60], and Sine Cosine Algorithm (SCA) [12]. The SSD algorithm has been previously used for parameter optimization of SVM classifier in [17].

III. PRESENT WORK

A. SOCIAL SKI DRIVER OPTIMIZATION: AN OVERVIEW

SSD algorithm is a novel optimization algorithm recently proposed in [17]. It mimics the path taken by ski-drivers downhill. The components of SSD are briefly described as follows:

- Position of the agents ($X_i^{R_n}$): The position of the agents is used to calculate the objective function at that location in the n-dimensional search space.
- Personal best position (P_i): The fitness value for each agent is compared with the previously obtained best fitness value and the corresponding best position is stored as the personal best position of the agent.
- Mean global best position (M_i): The agents move towards the global best position which represents the mean of the best three solutions as follows:

$$M_i = \frac{X_\alpha + X_\beta + X_\gamma}{3} \tag{1}$$

  where X_α, X_β and X_γ are the three best solutions obtained so far.
- Velocity of the agents(V_i): The positions of the agents are updated using the equation:

$$X_i^{t+1} = X_i^t + V_i^t \tag{2}$$

$$V_i^{t+1} = \begin{cases} \text{if } r_2 \leq 0.5 \\ \quad c\sin(r_1)(P_i^t - X_i^t) + \sin(r_1)(M_i^t - X_i^t) \\ \text{else if } r_2 > 0.5 \\ \quad c\cos(r_1)(P_i^t - X_i^t) + \cos(r_1)(M_i^t - X_i^t) \end{cases} \tag{3}$$

  where V_i is the velocity of X_i, r_1 and r_2 are uniformly generated random numbers in the range [0,1], P_i is the personal best position of the i^{th} agent, M_i is the mean global best position for the entire population, and c is a parameter which is used to make a balance between exploration and exploitation and it is calculated as:

$$c^{t+1} = \alpha c^t \tag{4}$$

  where t is the current iteration and α is used to reduce the value of c.

Equation 3 indicates that the moving directions for the agents are not straightforward because of the sine and cosine functions which gives the algorithm a better exploration capability and diversifies the search space, but in a guided manner.

B. LATE ACCEPTANCE HILL CLIMBING

Hill climbing is a simple local search method where a new solution is selected only if it is better. Because of this greedy approach, Hill climbing often gets stuck in local optima. To overcome this, LAHC [61] is proposed. LAHC has the ability to choose a lower performing solution in order to overcome a local optima. Here a solution is immediately chosen if it has better performance and λ number of solutions with worse performance are stored to allow for the acceptance of slightly poor solutions in order to improve them into better ones. Here, neighbor is obtained *via* mutation. If the performance of the neighbor is better than the performance of the original solution, then the neighbor is chosen as the best solution. Else, we temporarily consider the neighbor which gives a poorer result. This consideration is performed only if the performance of the neighbor is worse than the performance of the best solution by a tolerance factor of δ per cent. After that, mutation is again performed on the neighbor to find its neighbor. If the performance of this newly found neighbor of neighbor is better than that of its best counterpart, then this solution is considered as the best solution; else the obtained neighbor is discarded.

C. PROPOSED BINARY SSD

In FS, searching for the best feature subset is a challenging problem, especially in the wrapper-based methods. This is because the selected subset needs to be evaluated by the learning algorithm at each iteration. FS is a binary optimization problem [42], where the solutions are limited to binary values {0, 1}. A solution or 'agent' is represented using a binary vector where 1 represents that the corresponding feature is selected and 0 represents that the corresponding feature is not selected. Solution vector size is equal to number of features in the dataset in consideration.

SSD is originally proposed for solving continuous optimization problems. To suit the FS problem, the position of an agent must be converted to a binary vector. This binary conversion is performed by using two different transfer function: S-shaped (Figure 1a) and V-shaped (Figure 1b). This transfer function gives the probability of updating the position of an agent from 0 to 1 and vice versa. An S-shaped function, depicted in Figure 1a, is given by Equation 5.

$$T(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

The agent position is updated as per Equation 6.

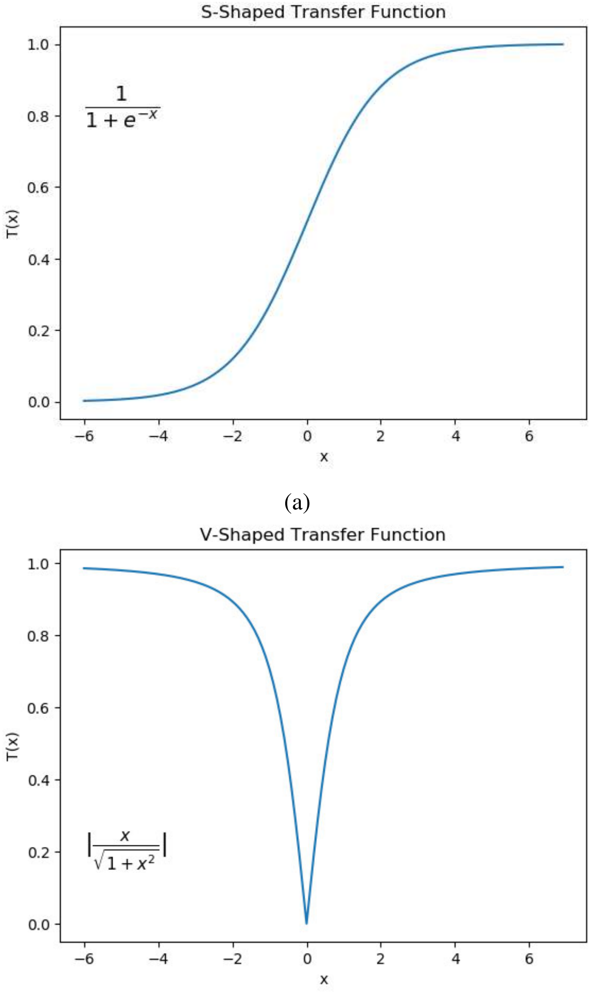
$$X_d^{t+1} = \begin{cases} 1 & \text{if } T(X_d^{t+1}) > rand \\ 0 & \text{if } T(X_d^{t+1}) \leq rand \end{cases} \tag{6}$$

where X_d^{t+1} is the updated position of the agent. V-shaped function, depicted in Figure 1b, is defined as Equation 7.

$$T(x) = \left| \frac{x}{\sqrt{1 + x^2}} \right| \tag{7}$$

Using the V-shaped function agent position is updated as per Equation 8:

$$X_j^{t+1} = \begin{cases} \sim X_j^t & \text{if } T(X_j^{t+1}) > rand \\ X_j^t & \text{if } T(X_j^{t+1}) \leq rand \end{cases} \tag{8}$$



(a)

(b)

FIGURE 1. (a) S-shaped and (b) V-shaped Transfer Functions used to convert continuous SSD search space to binary.

where X^{t+1} is the agent's updated position, X^t is the position of the agent at that instant of time, and $rand \in [0, 1]$.

After updating an agent's position in each iteration, a local search is performed using LAHC to optimize the position of the agent in order to obtain a better fitness value. The local search using LAHC increases the exploitation ability of the SSD algorithm, and by doing so it helps to escape the local optima, as described in Section III-B.

The analysis of our algorithm shows that its time complexity is $O(iter * psize * \lambda^2 * (t_{fitness} + dim))$ where iter is the maximum number of iterations, psize is the population size, λ is the parameter of LAHC, $t_{fitness}$ is the complexity of calculating the fitness of a particular agent using the classifier and dim is the dimension of the dataset. Algorithm 2 represents the pseudocode for the proposed method. The pseudocode of LAHC is provided in Algorithm 1.

Algorithm 1 Pseudocode for LAHC

```
Input: Agent $X = (x_1, x_2, \ldots, x_d)$
Output: Agent $X' = (x'_1, x'_2, \ldots, x'_d)$
  $OriginalFit \leftarrow fitness(X)$
  for $j \leftarrow 1$ to $\lambda$ do
    $mX = mutate(X)$
    $mFit = fitness(mX)$
    if $mFit < OriginalFit$ then
      $X \leftarrow mX$
      $OriginalFit \leftarrow mFit$
    else if $mFit < (1 + \delta) \times OriginalFit$ then
      for $k \leftarrow 1$ to $\lambda$ do
        $mX' = mutate(mX)$
        $mFit = fitness(mX')$
        if $mFit < OriginalFit$ then
          $X \leftarrow mX'$
          $OriginalFit \leftarrow mFit$
          $break$
        end if
      end for
    end if
  end for
```

Algorithm 2 Pseudocode for the Proposed FS Method

```
Input: MaxIter, PopSize
Output: Best Agent $X = (x_1, x_2, \ldots, x_d)$
  Generate initial $position$, $velocity$, $gbest$, $pbest$ randomly
  for $iter \leftarrow 1$ to $MaxIter$ do
    for $i \leftarrow 1$ to $PopSize$ do
      $agent \leftarrow position[i]$
      $fit \leftarrow fitness(agent)$
      Update $gbest$ by randomly choosing any of the best three positions
      Update $pbest$
      Generate $rand1$, $rand2$
      if $rand1 < 0.5$ then
        $velocity[i] = c * sin(rand2) * (pbest[i] - agent) + sin(rand2) * (gbest - agent)$
      else
        $velocity[i] = c * cos(r2) * (pbest[i] - agent) + cos(rand2) * (gbest - agent)$
      end if
      Update $position[i]$ using Equation 6 or Equation 8.
      $position[i] = LAHC(position[i])$
    end for
    $c = \alpha \times c$
  end for
```

D. FITNESS FUNCTION

In this part, we discuss how to assess the quality of a solution. Since the above algorithm is a wrapper based method, so a learning algorithm must be used for the evaluation. For this work, we have used the k-Nearest Neighbor (KNN) [62] classifier to evaluate the classification accuracy, following the works of [11], [42], [43]. Our proposed fitness function has two components: classification accuracy and number of features. Since these two components are opposing in nature,

TABLE 1. Description of the UCI datasets used in the present work.

Sl. No.	Dataset	No. of Attributes	No. of Samples	No. of Labels	Dataset Domain
1	Breastcancer	9	699	2	Biology
2	BreastEW	30	569	2	Biology
3	CongressEW	16	435	2	Politics
4	Exactly	13	1000	2	Biology
5	Exactly2	13	1000	2	Biology
6	HeartEW	13	270	2	Biology
7	IonosphereEW	34	351	2	Electromagnetic
8	KrvskpEW	36	3196	2	Game
9	Lymphography	18	148	2	Biology
10	M-of-n	13	1000	2	Biology
11	PenglungEW	325	73	2	Biology
12	SonarEW	60	208	2	Biology
13	SpectEW	22	267	2	Biology
14	Tic-tac-toe	9	958	2	Game
15	Vote	16	300	2	Politics
16	WaveformEW	40	5000	3	Physics
17	WineEW	13	178	3	Chemistry
18	Zoo	16	101	6	Artificial

i.e., we want to achieve higher classification accuracy with lower number of features, so, we have decided to use classification error. This is because a lower error *i.e.*, high accuracy would imply a low fitness value, so would a lower number of features. Equation 9 shows the fitness function which evaluates a given feature subset.

$$\downarrow Fitness = \omega \times \eta + (1 - \omega) \times \frac{|S|}{|D|} \qquad (9)$$

where $|S|$ is the number of features in the selected feature subset, $|D|$ is the total number of features of the dataset, η is the classification error of the feature subset, and $\omega \in [0, 1]$ [11] indicates the relative weight of the number of features and the classification error.

IV. EXPERIMENTAL RESULTS

This section reports the experimental observations found which help to prove the applicability of binary variants of SSD in FS domain. We have used KNN [62] classifier to measure classification accuracy. As per the recommendations in [11], [43], [44], we have set $K = 5$. For each dataset, 80% of the instances are used for training the classification model and rest 20% are used for testing. Proposed methods are implemented using Python3 [63] and obtained graphs are plotted using Matplotlib [64].

A. DATASET DESCRIPTION

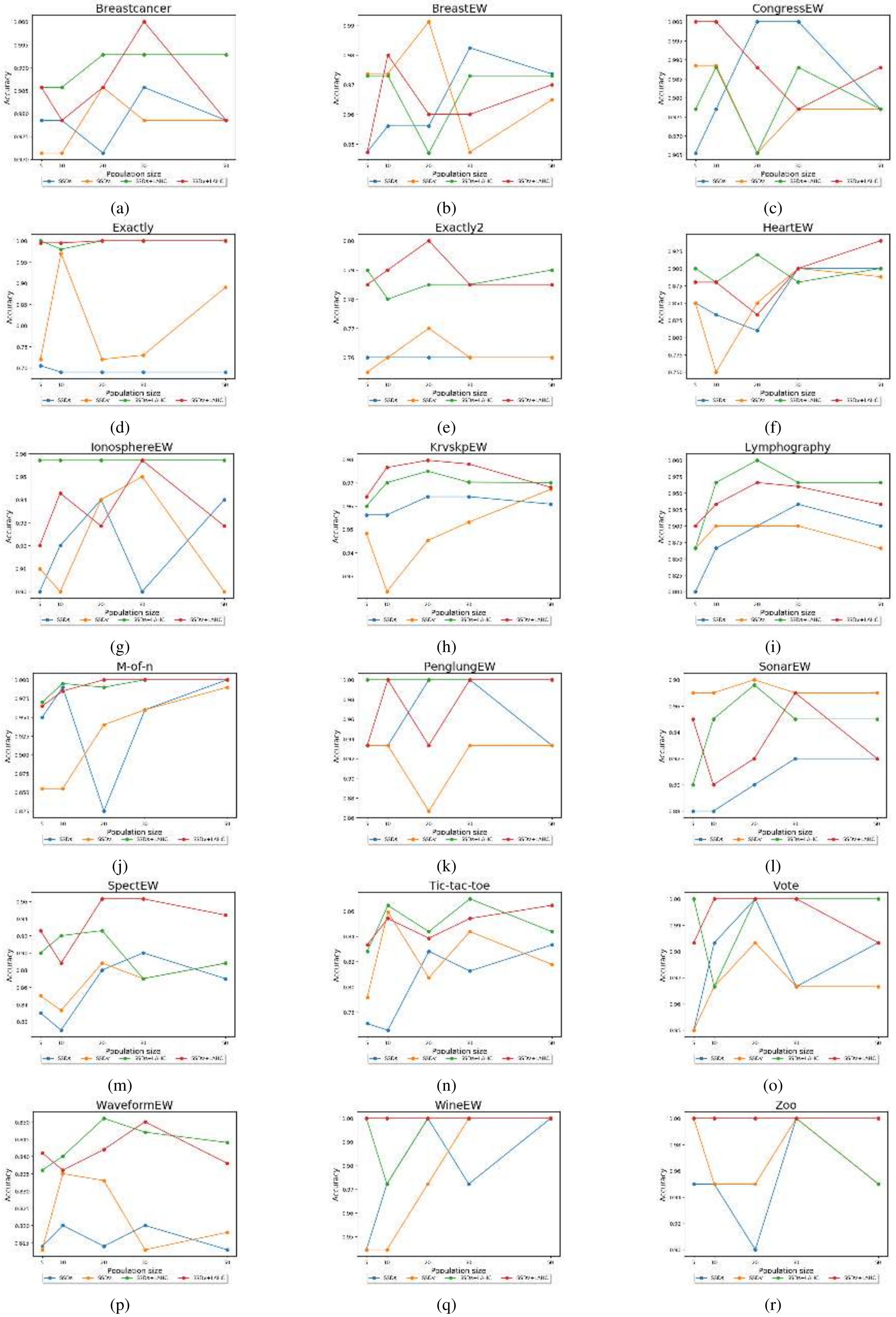
In this work, 18 different benchmark datasets from the UCI data repository [65] are used to assess the performance of the proposed methods.

Table 1 shows the details of the used datasets. There are 15 bi-class and 3 multi-class datasets.

B. PARAMETER TUNING

We have done various experiments to know the effect of population size on the performance of the proposed methods (in terms of classification accuracy). We have evaluated the proposed binary SSD methods for population sizes of [5, 10, 20, 30, 50]. Figure 2 shows the effect of changing population size on the classification accuracy of the algorithm for all datasets. Two effects have been noticed: first, increasing the population size does not always improve the classification accuracy, and second, for most of the datasets, not much variation has been seen in the classification accuracy for different population sizes. On the basis of the outcomes shown in Figure 2, we have set the population size to 30 and tried to improve the classification accuracy for the datasets used here.

In this work, our aim is to minimize the fitness function, i.e., minimizing the number of features as well as the classification error. If the value of ω is increased then the algorithm gives more focus on reducing the number of features; if ω is given a lower value then more focus is given on decreasing the classification error. Upon experimenting with several values of ω, the optimal value of ω is fixed at 0.2. Thus the algorithm under consideration assigns greater importance to minimize the classification error, i.e., maximizing the accuracy. During the course of the work, we have experimented with different values of c and α in the search space. As already mentioned in [17], our experiments also come to the conclusion that the classification accuracy improves upon increasing the value of c. The classification accuracy improves when the value of α is slowly reduced but then again decreases after reaching a maximum. This happens due to overfitting when the value of α is too low. Eventually, the maximum classification accuracy is achieved when the values of c and α are kept as 100 and 0.9 respectively. We have experimented with different values of λ which allows us to store solutions of slightly poorer performance. The dilemma is to either improve the classification accuracy or the execution time. Since both of these cannot be improved simultaneously so the value of λ is fixed at 15, which is the upper limit on the acceptable number of worse solutions.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

FIGURE 2. Effect of the population variation on classification accuracy of the four proposed binary variants of the SSD based FS methods for 18 UCI datasets.

TABLE 2. Performance of proposed methods in terms of classification accuracy.

Dataset	Original	SSDs	SSDv	SSDs+LAHC	SSDv+LAHC
Breastcancer	94.44	97.87	98.22	98.93	99.29
BreastEW	96.3	97.37	97.37	98.25	99.12
CongressEW	91.55	97.7	98.85	100	100
Exactly	67.5	91	98	100	100
Exactly2	74	77	77.5	79	80
HeartEW	80.92	88.89	87.97	91.67	92.59
IonosphereEW	91.4	94.28	93.58	96.43	95
KrvskpEW	96.3	96.32	96.79	97.81	97.55
Lymphography	68.3	93.33	95	96.67	96.67
M-of-n	84.9	98	100	100	100
PenglungEW	93.1	100	93.33	100	100
SonarEW	83.33	92.86	90.48	97.62	95.24
SpectEW	88.8	90.74	92.59	95.15	93.52
Tic-tac-toe	71.5	83.07	83.33	87.24	86.2
Vote	95	98.33	99.17	100	100
WaveformEW	76.8	82.4	82.8	84.4	83.7
WineEW	78.9	100	100	100	100
Zoo	95	100	100	100	100
Sum of ranks		52	47	22	24
Average rank		2.889	2.611	1.222	1.333
Final rank		**4**	**3**	**1**	**2**

Figure 3 shows how the fitness values of SSDs, SSDv, SSDs+LAHC and SSDs+LAHC change with increasing number of iterations.

From Figure 2 and Figure 3, we have decided to use population size as 30 and maximum number of iterations as 50.

C. DISCUSSION

In this section, we have provided the results obtained by the proposed approaches over the datasets mentioned in Section IV-A and analyzed them. From Table 2, it is clear that the proposed methods are able to perform FS efficiently. For all the datasets, classification accuracy has improved after applying the FS methods (with huge margins in multiple cases).

SSDs and SSDv both have achieved more than or equal to 90% accuracy for 14 datasets whereas for SSDs+LAHC and SSDv+LAHC the count is 15. Moreover, SSDs and SSDv have got 100% accuracy for three datasets each while SSDs+LAHC and SSDv+LAHC each has scored full for seven datasets which is quite remarkable.

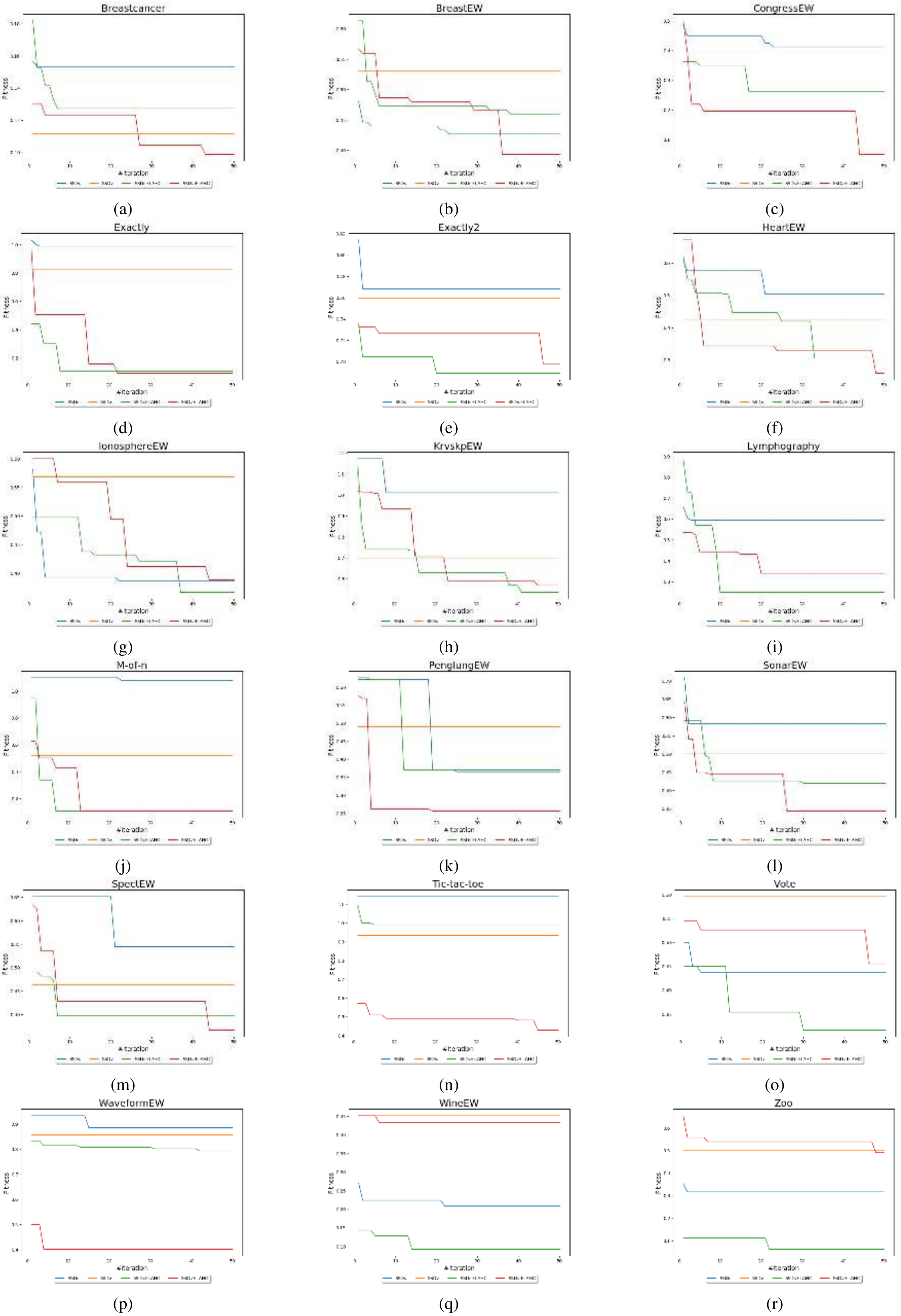
If we keep aside the classification accuracy and focus on the number features used to obtain these high accuracies, even then the proposed methods have performed significantly well. Almost on all the datasets, all the proposed variants have used only 50% or lesser number of features for classification.

In case of Exactly, Sonar, SpectEW, Tic-tac-toe accuracies have improved by quite a bit after using LAHC. For many cases like Lymphography, PenglungEW, SpectEW, Wine, Zoo, number of selected features has reduced after using LAHC. Observing the results in Table 2 and Table 3, it quite obvious that the use of LAHC is significantly helping the algorithm to explore different parts of the search space and achieve better results.

Now, let us compare SSDs and SSDv in terms of achieved classification accuracy. For BreastEW, WineEW and Zoo, SSDs and SSDv have achieved same classification accuracy. SSDs has performed better than SSDv in 4 cases: HeartEW, IonosphereEW, PenglungEW and SonarEW. For PenglungEW, the accuracy difference is quite significant. For rest 11 cases, SSDv has achieved higher classification accuracy. For, HeartEW, M-of-n and Zoo, SSDs and SSDv have selected same number of features. SSDs has selected more number of features in 4 cases: Exactly, KrvskpEW, PenglungEW, and SonarEW. For rest 11 cases, SSDv has selected lesser number of features than SSDs. In case of SSDs+LAHC and SSDv+LAHC in terms of achieved classification accuracy, for 8 datasets both the methods have achieved same classification accuracy. In five cases SSDs+LAHC has performed better and in 5 cases SSDv+LAHC has performed better. In case of Exactly, HeartEW, SonarEW, and Tic-tac-toe, SSDs+LAHC and SSDv+LAHC have selected same number of features. SSDs+LAHC has selected lesser number of features in 4 cases: Breastcancer, BreastEW, SpectEW, and Zoo. For rest 10 cases SSDv+LAHC has selected lesser number of features.

V. COMPARISON AND ANALYSIS

To establish the superiority of the proposed FS methods, we have compared these methods with 15 state-of-the-art methods: 8 popular meta-heuristic FS methods: GA, PSO, ALO, GSA, GWO, GOA, DA, and SSA; and 7 hybrid meta-heuristic FS methods. WOASAT-2 [44] is hybrid of WOA and SA. BGWOPSO [39] is developed by hybridizing GWO and PSO. Following three different strategies, GWO and WOA are hybridized [66]: serial grey-whale optimizer (HSGW), random switching grey-whale optimizer (RSGW),



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

FIGURE 3. Convergence graphs of the four proposed methods for 18 UCI datasets.

TABLE 3. Performance of proposed methods in terms of no. of selected features.

Dataset	Original	SSDs	SSDv	SSDs+LAHC	SSDv+LAHC
Breastcancer	9	3.5	4	2.5	3
BreastEW	30	9	12	9	9.5
CongressEW	16	2	5	5.5	4.5
Exactly	13	8	6.5	6	6
Exactly2	13	1.5	2	8	7.5
HeartEW	13	3.5	3.5	5	5
IonosphereEW	34	5	13.5	12	11.5
KrvskpEW	36	23	20.5	20	16.5
Lymphography	18	6.5	9.5	6.5	6
M-of-n	13	7	7	6	6.5
PenglungEW	325	167	143	140	139
SonarEW	60	28	27	23.5	23.5
SpectEW	22	10	12	9	11.5
Tic-tac-toe	9	7	9	9	9
Vote	16	3	7	4.5	4
WaveformEW	40	21	27	22.5	20
WineEW	13	4.5	6	3	2
Zoo	16	6	6	4.5	5
Sum of ranks		39	54	36	31
Average rank		2.167	3.000	2.000	1.722
Final rank		**3**	**4**	**2**	**1**

and adaptive switching grey-whale optimizer (ASGW). In WOA-CM [16], performance of WOA is enhanced by using crossover and mutation. We have obtained the results from the mentioned articles. The authors in the corresponding articles have got the results with the best parameters obtained through exhaustive experiments, and we have chosen the best results from those articles.

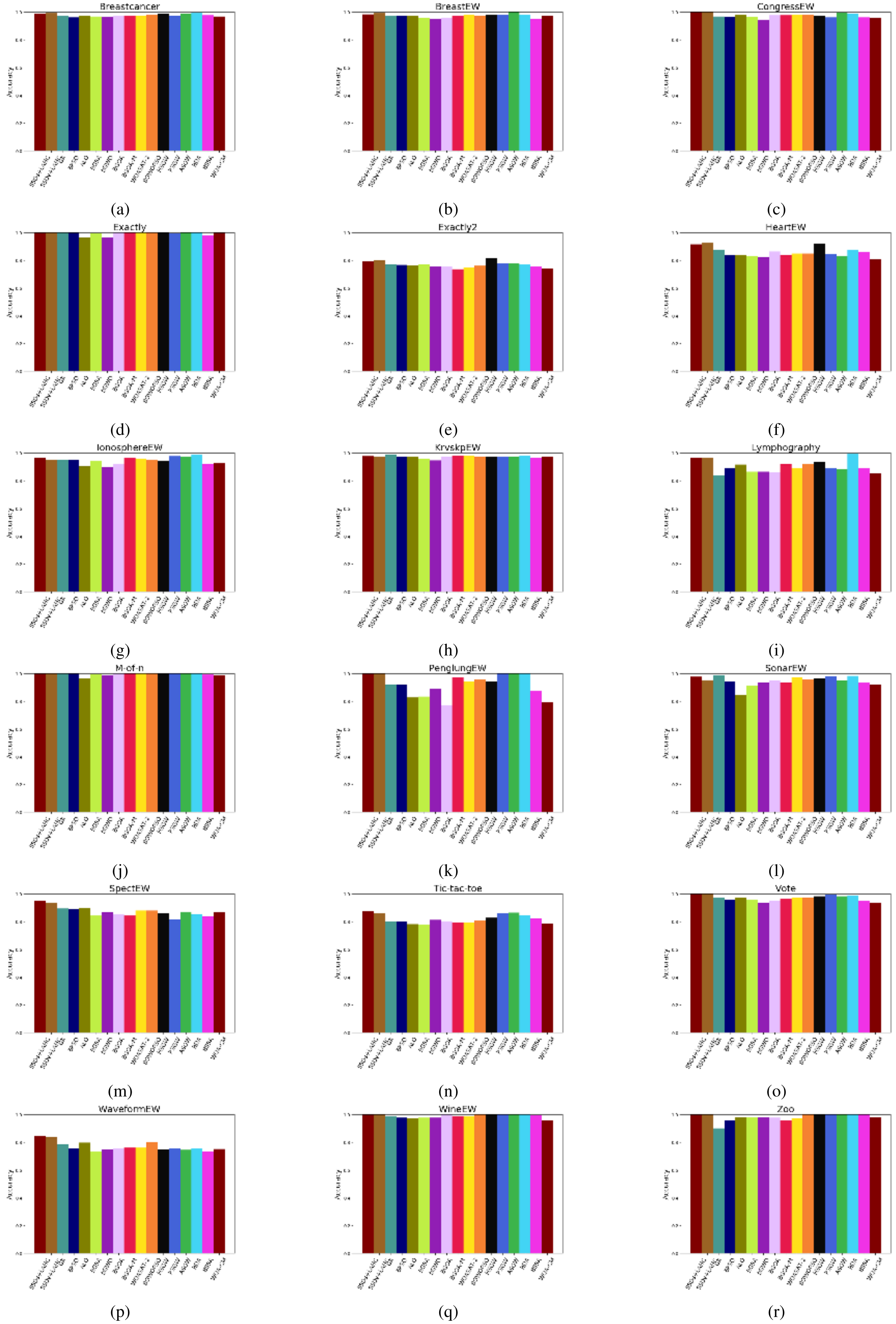
Figure 4 shows the performance of $SSDs + LAHC$ and $SSDv + LAHC$ in terms of achieved classification accuracy. From Figure 4, it can be observed that $SSDs + LAHC$ performs best in case of 10 datasets: CongressEW, Exactly, M-of-n, PenglungEW, SpectEW, Tic-tac-toe, Vote, WaveformEW, WineEW, and Zoo. $SSDv + LAHC$ performs best in 9 cases: Breastcancer, CongressEW, Exactly, HeartEW, M-of-n, PenglungEW, Vote, WineEW and Zoo. Now, for CongressEW, Exactly, M-of-n, PenglungEW, Vote, WineEW, and Zoo datasets, $SSDs + LAHC$ and $SSDv + LAHC$ have combinedly achieved the best accuracy. For Lymphography dataset, both $SSDs + LAHC$ and $SSDv + LAHC$ have achieved second best classification accuracy. For Breastcancer dataset, $SSDs + LAHC$ has achieved second highest accuracy. For BreastEW, Exactly2, and HeartEW, $SSDs + LAHC$ has achieved third highest accuracy. For Tic-tac-toe, $SSDv + LAHC$ has achieved third highest accuracy.

Now, both $SSDs+LAHC$ and $SSDv+LAHC$ have achieved same accuracy as GA for Exactly and M-of-n datasets. For remaining 16 datasets, both $SSDs + LAHC$ and $SSDv + LAHC$ have performed better than GA. Same can be stated if we compare $SSDs + LAHC$ and $SSDv + LAHC$ with PSO. Both $SSDs + LAHC$ and $SSDv + LAHC$ completely outperform ALO, GSA and GWO for all 18 datasets. $SSDs + LAHC$ has achieved same accuracy as DA in case of 5 datasets: Exactly, M-of-n, Penglung, WineEW, and Zoo. DA outperforms $SSDs + LAHC$ in 5 cases: Breastcancer, IonosphereEW, KrvskpEW, Lymphograophy, and SonarEW. $SSDs + LAHC$ wins in rest 8 cases. With DA, $SSDv + LAHC$ has 8 wins, 6 ties, and 4 losses. Both $SSDs + LAHC$ and $SSDv + LAHC$ completely outperforms SSA except for Zoo dataset, where both have achieved same accuracy.

Figure 5 shows the performance of $SSDs + LAHC$ and $SSDv + LAHC$ in terms of selected number of features. From Figure 5, it can be observed that $SSDs + LAHC$ performs best *i.e.*, selects lowest number of features in case of Breastcancer dataset. $SSDs + LAHC$ performs second best BreastEW, HeartEW, M-of-n, and WineEW datasets. $SSDv + LAHC$ performs best for WineEW. It performs as second best for Breastcancer, HeartEW, and Lymphography. With GA, both $SSDs+LAHC$ and $SSDv+LAHC$ have 3 wins and 2 ties. With PSO, $SSDs+LAHC$ has 3 ties and 3 wins, and $SSDv+LAHC$ has 2 ties and 2 wins. With ALO, $SSDs + LAHC$ has 9 wins and 1 tie, and $SSDv + LAHC$ has 11 wins and no tie. With GSA, $SSDs + LAHC$ has 5 wins and 1 tie and $SSDv + LAHC$ 6 wins and 2 ties. With GWO, $SSDs + LAHC$ 9 wins and no tie and $SSDv + LAHC$ has 8 wins and 2 ties. With GOA, $SSDs + LAHC$ has 9 wins and 1 tie, and $SSDv + LAHC$ has 11 wins and no tie.

Figure 6 and Figure 7 respectively shows the average accuracies achieved and average number of features selected over the 18 UCI datasets using the proposed methods and 15 state-of-the-art methods considered for comparison.

To determine the statistical significance of the proposed methods, Wilcoxon test has been performed. It is a non-parametric statistical test where pairwise comparison is performed [67]. Here the null hypothesis is, two sets of results have same distribution. If the distributions of two results are statistically different, then the generated p-value from the test statistics will be < 0.05, as we have performed the test at 0.05% significance level, resulting in the rejection of the null



(a) (b) (c)

(d) (e) (f)

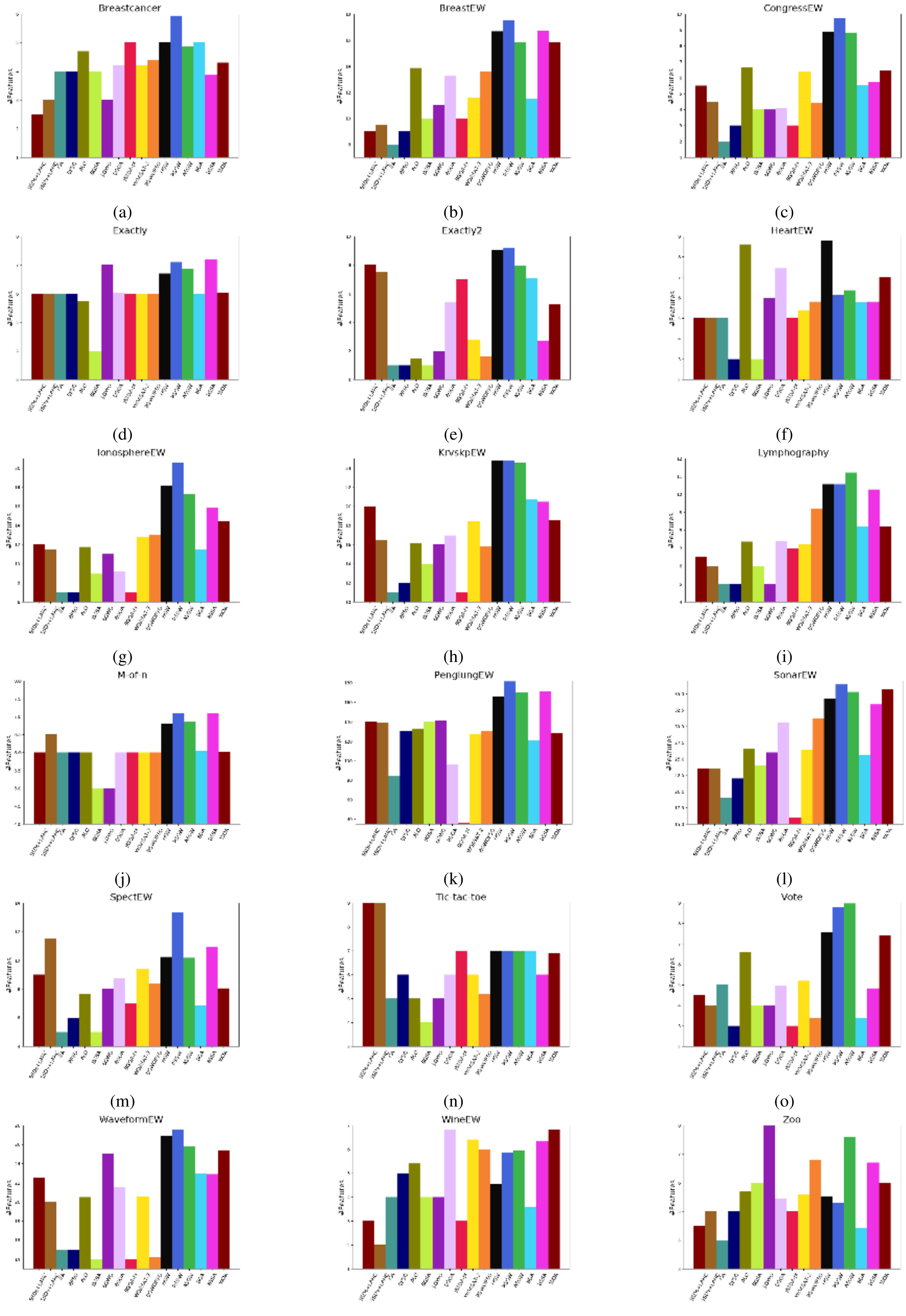
(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

FIGURE 4. Comparison of Classification Accuracy obtained by proposed FS method with some state-of-the-art FS methods for 18 UCI datasets.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

FIGURE 5. Comparison of no. of selected features by proposed FS method with some state-of-the-art FS methods for 18 UCI datasets.

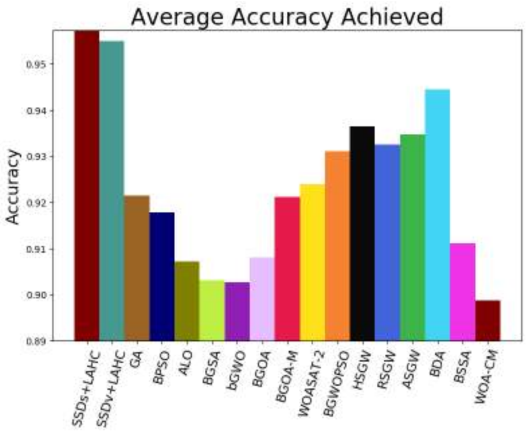


FIGURE 6. Average classification accuracy achieved for the UCI datasets using all the methods considered.

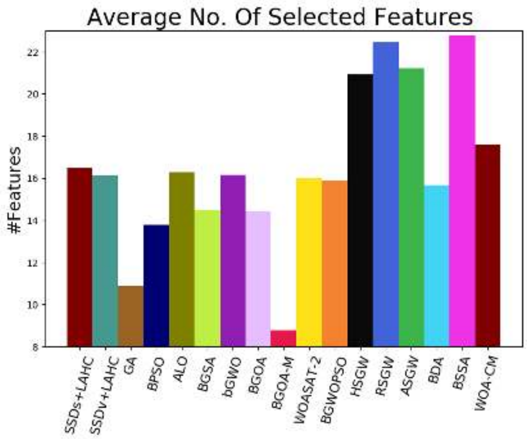


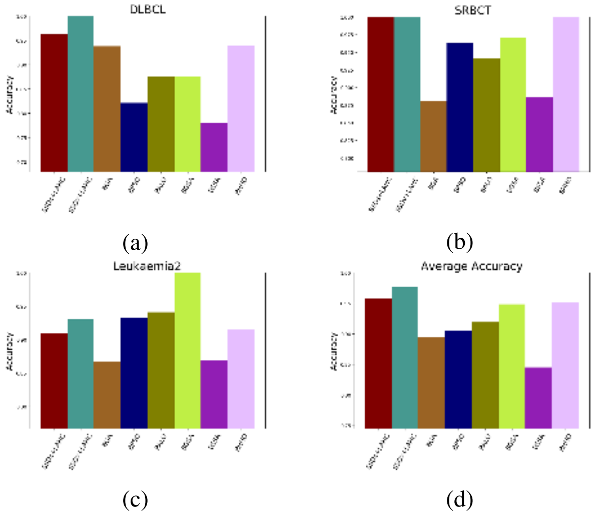
FIGURE 7. Average no. of features selected for the UCI datasets using all the methods considered.

hypothesis. From the test results, provided in Table 4, we can conclude that the proposed FS methods are statistically significant.

The main characteristic of the proposed FS method that separates it from other meta-heuristic algorithms is that the search direction is not straightforward because of the sine and cosine functions. This allows diversification of the search space and the parameter c in Equation 3 maintains a balance between exploration and exploitation, guiding it to finally converge to a solution. Furthermore, LAHC allows the algorithm to refine the solution, overcoming local optima in the process and leading to improved results.

VI. EXPERIMENTS ON HIGH DIMENSIONAL DATA

To show the robustness of the proposed FS methods, we have also applied these to high dimensional microarray datasets. It is to be noted that microarray datasets [68] are important medical diagnostic tools used for identifying or classifying different diseases including cancer. The main challenge of working with these datasets is that they tend to have small number of samples and large number of features. We have applied the proposed FS methods on three standard and publicly available microarray datasets and compared the obtained results with 6 state-of-the-art methods: GA, PSO, ALO, GSA, SSA, and Harris Hawks Optimizer (HHO) [68]. Table 5 shows the details of the microarray datasets used in this work. These datasets are high dimensional, with no. of features > 1000.



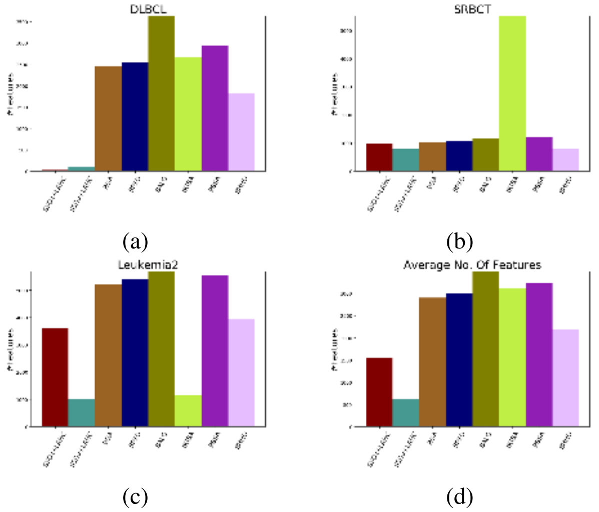
(a)

(b)

(c)

(d)

FIGURE 8. Comparison of classification accuracy achieved for microarray datasets by the proposed method with state-of-the-art methods mentioned.



(a)

(b)

(c)

(d)

FIGURE 9. Comparison of no. of features selected for microarray datasets by the proposed method with state-of-the-art methods mentioned.

Figure 8 shows the accuracies achieved for 3 microarray datasets using the proposed methods and mentioned state-of-the-art methods. Figure 8d shows the average accuracy achieved over the used 3 microarray datasets using the proposed methods and the state-of-the-art methods. Figure 9 shows the number of features selected using the proposed method and state-of-the-art methods. Figure 9d shows aver-

TABLE 4. P-values of the Wilcoxon rank-sum test for the classification accuracy of the proposed methods as compared to all the other methods considered here using for the UCI datasets.

	GA	BPSO	ALO	BGSA	bGWO	BGOA	BGO A-M	WOA SAT-2	BGWOPSO	HSGW	RSGW	ASGW	BDA	BSSA	WOA-CM
SSDs+LAHC	0.001	0	0	0	0	0	0.001	0.001	0.001	0.011	0.008	0.019	0.04	0	0
SSDv+LAHC	0.003	0	0	0	0	0	0.001	0.002	0.002	0.013	0.026	0.03	0.032	0	0

TABLE 5. Description of the microarray datasets used in the present work.

Sl. No.	Dataset	No. of Attributes	No. of Samples	No. of Labels	Dataset Domain
1	DLBCL	5469	77	2	Biology
2	SRBCT	2308	82	4	Biology
3	Leukaemia2	11225	72	3	Biology

age number of selected features over the 3 datasets. From the results it is quite evident that the proposed approaches are able to achieve higher classification accuracy with lower number of features in compared to the mentioned state-of-the-art methods. So, this proves the robustness and applicability of the proposed methods.

VII. CONCLUSION

In this work, we have introduced new meta-heuristic FS algorithms based on the recently proposed SSD optimization algorithm. To convert the continuous search space of the SSD to binary, we have used S-shaped and V-shaped transfer functions. To enhance the exploitation ability of SSD, we have applied a local search algorithm, *namely* LAHC. The FS problem is formulated as a multi-objective optimization task with a fitness function tending to achieve high classification accuracy with low number of selected features. The obtained results show that the incorporation of LAHC with SSD has significantly improved the results; both in terms of classification accuracy and number of selected features. We have applied the proposed methods on 18 well known UCI datasets and 3 microarray datasets, and compared the achieved results with some state-of-the-art FS methods. Comparison shows that the proposed methods are able to produce better results. Hence, it can be said that the proposed methods are able to effectively search the feature space and converge to (near) optimal solution better than other methods considered here for comparison. Moreover, amongst S-shaped and V-shaped transfer functions both SSD and hybrid SSD perform marginally better with V-shaped transfer function. The proper functioning of the proposed method depends on the parameter setting of SSD and LAHC. The optimal values for these parameters for a different problem may be different, which will require some experiments to determine, and is therefore considered as a limitation of this method. Also, having the same stochastic nature as other swarm-intelligence algorithms, as per No Free Lunch theorem [59], binary SSD is not guaranteed to produce outstanding results for all FS problems. For further studies, binary SSD can be applied to other standard datasets, real-world problems and it can also be employed with more classifiers like neural network, random forest, SVM etc. A comparative study can be made of the obtained results for microarray datasets with other hybrid meta-heuristics present in the literature. It would be interesting to hybridize SSD with other meta-heuristics or other local search algorithms.

REFERENCES

[1] H. Faris, A. A. Heidari, A. M. Al-Zoubi, M. Mafarja, I. Aljarah, M. Eshtay, and S. Mirjalili, "Time-varying hierarchical chains of salps with random weight networks for feature selection," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112898, doi: 10.1016/j.eswa.2019.112898.

[2] H. Jia, J. Li, W. Song, X. Peng, C. Lang, and Y. Li, "Spotted hyena optimization algorithm with simulated annealing for feature selection," *IEEE Access*, vol. 7, pp. 71943–71962, 2019, doi: 10.1109/access.2019.2919991.

[3] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[4] M. Ghosh, S. Adhikary, K. K. Ghosh, A. Sardar, S. Begum, and R. Sarkar, "Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods," *Med. Biol. Eng. Comput.*, vol. 57, no. 1, pp. 159–176, Jan. 2019, doi: 10.1007/s11517-018-1874-4.

[5] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 249–256, doi: 10.1016/b978-1-55860-247-2.50037-1.

[6] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Machine Learning*. Berlin, Germany: Springer, 1994, pp. 171–182, doi: 10.1007/3-540-57868-4_57.

[7] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/tpami.2005.159.

[8] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2005, pp. 507–514.

[9] R. Leardi, "Genetic algorithms in feature selection," in *Genetic Algorithms in Molecular Modeling*. Amsterdam, TheNetherlands: Elsevier, 1992, pp. 67–86, doi: 10.1016/b978-012213810-2/50004-9.

[10] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Cybern. Simulation*, Oct. 1997, pp. 4104–4108, doi: 10.1109/ICSMC.1997.637339.

[11] 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, doi: 10.1016/j.neucom.2015.06.083..

[12] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.

[13] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.

[14] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[15] A. Zarshenas and K. Suzuki, "Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning," *Knowl.-Based Syst.*, vol. 110, pp. 191–201, Oct. 2016, doi: 10.1016/j.knosys.2016.07.026.

[16] M. M. Mafarja and S. Mirjalili, ''Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection,'' *Soft Comput.*, vol. 23, no. 15, pp. 6249–6265, Aug. 2019, doi: 10.1007/s00500-018-3282-y.

[17] A. Tharwat and T. Gabel, ''Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm,'' *Neural Comput. Appl.*, Apr. 2019, doi: 10.1007/s00521-019-04159-z.

[18] W. Siedlecki and J. Sklansky, ''A note on genetic algorithms for large-scale feature selection,'' in *Handbook of Pattern Recognition and Computer Vision*. Singapore: World Scientific, Aug. 1993, pp. 88–107, doi: 10.1142/9789814343138_0005.

[19] J. Yang and V. Honavar, ''Feature subset selection using a genetic algorithm,'' in *Feature Extraction, Construction Selection*. Cham, Switzerland: Springer, 1998, pp. 117–136, doi: 10.1007/978-1-4615-5725-8_8.

[20] I.-S. Oh, J.-S. Lee, and B.-R. Moon, ''Hybrid genetic algorithms for feature selection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004, doi: 10.1109/tpami.2004.105.

[21] J. Huang, Y. Cai, and X. Xu, ''A hybrid genetic algorithm for feature selection wrapper based on mutual information,'' *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1825–1844, Oct. 2007, doi: 10.1016/j.patrec.2007.05.011.

[22] S. Jiang, K.-S. Chin, L. Wang, G. Qu, and K. L. Tsui, ''Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department,'' *Expert Syst. Appl.*, vol. 82, pp. 216–230, Oct. 2017, doi: 10.1016/j.eswa.2017.04.017.

[23] M. Ghosh, R. Guha, R. Mondal, P. K. Singh, R. Sarkar, and M. Nasipuri, ''Feature selection using histogram-based multi-objective GA for handwritten devanagari numeral recognition,'' in *Advances in Intelligent Systems and Computing*. Singapore: Springer, 2018, pp. 471–479, doi: 10.1007/978-981-10-7566-7_46.

[24] B. Ma and Y. Xia, ''A tribe competition-based genetic algorithm for feature selection in pattern classification,'' *Appl. Soft Comput.*, vol. 58, pp. 328–338, Sep. 2017, doi: 10.1016/j.asoc.2017.04.042.

[25] R. Guha, M. Ghosh, S. Kapri, S. Shaw, S. Mutsuddi, V. Bhateja, and R. Sarkar, ''Deluge based genetic algorithm for feature selection,'' *Evol. Intell.*, Mar. 2019, doi: 10.1007/s12065-019-00218-5.

[26] R. Poli, J. Kennedy, and T. Blackwell, ''Particle swarm optimization,'' *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007, doi: 10.1007/s11721-007-0002-0.

[27] S. Mirjalili and A. Lewis, ''S-shaped versus V-shaped transfer functions for binary particle swarm optimization,'' *Swarm Evol. Comput.*, vol. 9, pp. 1–14, Apr. 2013, doi: 10.1016/j.swevo.2012.09.002.

[28] H. H. Inbarani, A. T. Azar, and G. Jothi, ''Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis,'' *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 175–185, Jan. 2014, doi: 10.1016/j.cmpb.2013.10.007.

[29] P. Moradi and M. Gholampour, ''A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy,'' *Appl. Soft Comput.*, vol. 43, pp. 117–130, Jun. 2016, doi: 10.1016/j.asoc.2016.01.044.

[30] L. Shang, Z. Zhou, and X. Liu, ''Particle swarm optimization-based feature selection in sentiment classification,'' *Soft Comput.*, vol. 20, no. 10, pp. 3821–3834, Oct. 2016, doi: 10.1007/s00500-016-2093-2.

[31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, ''Optimization by simulated annealing,'' *Science*, vol. 220, no. 4598, pp. 671–680, 1983, doi: 10.1126/science.220.4598.671.

[32] P. J. M. van Laarhoven and E. H. L. Aarts, ''Simulated annealing,'' in *Simulated Annealing: Theory Applications*. Amsterdam, The Netherlands: Springer, 1987, pp. 7–15, doi: 10.1007/978-94-015-7744-1_2.

[33] L. Zhang, K. Mistry, C. P. Lim, and S. C. Neoh, ''Feature selection using firefly optimization for classification and regression models,'' *Decis. Support Syst.*, vol. 106, pp. 64–85, Feb. 2018, doi: 10.1016/j.dss.2017.12.001.

[34] S. Saremi, S. Mirjalili, and A. Lewis, ''Grasshopper optimisation algorithm: Theory and application,'' *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.

[35] I. Aljarah, A. M. Al-Zoubi, H. Faris, M. A. Hassonah, S. Mirjalili, and H. Saadeh, ''Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm,'' *Cognit. Comput.*, vol. 10, no. 3, pp. 478–495, Jun. 2018, doi: 10.1016/j advengsoft.2017.01.004.

[36] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, ''A new heuristic optimization algorithm: Harmony search,'' *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001, doi: 10.1177/003754970107600201.

[37] M. Saka, O. Hasançebi, and Z. Geem, ''Metaheuristics in structural optimization and discussions on harmony search algorithm,'' *Swarm Evol. Comput.*, vol. 28, pp. 88–97, Jun. 2016, doi: 10.1016/j.swevo.2016.01.005.

[38] L. Zheng, R. Diao, and Q. Shen, ''Self-adjusting harmony search-based feature selection,'' *Soft Comput.*, vol. 19, no. 6, pp. 1567–1579, Jun. 2015, doi: 10.1007/s00500-014-1307-8.

[39] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, ''Binary optimization using hybrid grey wolf optimization for feature selection,'' *IEEE Access*, vol. 7, pp. 39496–39508, 2019, doi: 10.1109/ACCESS.2019.2906757.

[40] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque, and S. Mirjalili, ''A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection,'' *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112824, doi: 10.1016/j.eswa.2019.112824.

[41] S. Mirjalili and A. Lewis, ''The whale optimization algorithm,'' *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

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

[43] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, ''Efficient hybrid nature-inspired binary optimizers for feature selection,'' *Cognit. Comput.*, vol. 12, no. 1, pp. 150–175, Jan. 2020, doi: 10.1007/s12559-019-09668-6.

[44] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, ''Binary dragonfly algorithm for feature selection,'' in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2017, doi: 10.1109/ictcs.2017.43.

[45] M. Dorigo and G. Di Caro, ''Ant colony optimization: A new meta-heuristic,'' in *Proc. Congr. Evol. Comput.*, 1999, pp. 12–17, doi: 10.1109/cec.1999.782657.

[46] M. Ghosh, R. Guha, R. Sarkar, and A. Abraham, ''A wrapper-filter feature selection technique based on ant colony optimization,'' *Neural Comput. Appl.*, pp. 1–19, Apr. 2019, doi: 10.1007/s00521-019-04171-3.

[47] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, ''Binary grasshopper optimisation algorithm approaches for feature selection problems,'' *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019, doi: 10.1016/j.eswa.2018.09.015.

[48] N. Gopika and A. M. kowshalaya M. E., ''Correlation based feature selection algorithm for machine learning,'' in *Proc. 3rd Int. Conf. Commun. Electron. Syst. (ICCES)*, Oct. 2018.

[49] S. Senthamarai Kannan and N. Ramaraj, ''A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm,'' *Knowl.-Based Syst.*, vol. 23, no. 6, pp. 580–585, Aug. 2010, doi: 10.1016/j.knosys.2010.03.016.

[50] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, ''GSA: A gravitational search algorithm,'' *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: 10.1016/j.ins.2009.03.004.

[51] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, ''BGSA: Binary gravitational search algorithm,'' *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010, doi: 10.1007/s11047-009-9175-3.

[52] M. Taradeh, M. Mafarja, A. A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, and H. Fujita, ''An evolutionary gravitational search-based feature selection,'' *Inf. Sci.*, vol. 497, pp. 219–239, Sep. 2019, doi: 10.1016j.ins.2019.05.038.

[53] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, ''Salp swarm algorithm: A bio-inspired optimizer for engineering design problems,'' *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: 10.1016/j.advengsoft.2017.07.002.

[54] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, ''Improved Salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection,'' *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113122, doi: 10.1016/j.eswa.2019.113122.

[55] M. Ghosh, T. Kundu, D. Ghosh, and R. Sarkar, ''Feature selection for facial emotion recognition using late hill-climbing based memetic algorithm,'' *Multimedia Tools Appl.*, vol. 78, no. 18, pp. 25753–25779, Sep. 2019, doi: 10.1007/s11042-019-07811-x.

[56] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, ''Equilibrium optimizer: A novel optimization algorithm,'' *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190, doi: 10.1016/j.knosys.2019.105190.

[57] M. H. Sulaiman, Z. Mustaffa, M. M. Saari, H. Daniyal, I. Musirin, and M. R. Daud, ''Barnacles mating optimizer: An evolutionary algorithm for solving optimization,'' in *Proc. IEEE Int. Conf. Autom. Control Intell. Syst.*, Oct. 2018, pp. 99–104, doi: 10.1109/i2cacis.2018.8603703.

[58] V. Hayyolalam and A. A. Pourhaji Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103249, doi: 10.1016/j.engappai.2019.103249.

[59] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[60] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[61] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *Eur. J. Oper. Res.*, vol. 258, no. 1, pp. 70–78, Apr. 2017, doi: 10.1016/j.ejor.2016.07.012.

[62] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992, doi: 10.1080/00031305.1992.10475879.

[63] G. van Rossum and F. L. Drake, *The Python Language Reference Manual*. Bristol, U.K.: Network Theory, 2011.

[64] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[65] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[66] M. Mafarja, A. A. Heidari, H. Faris, S. Mirjalili, and I. Aljarah, "Dragonfly algorithm: Theory, literature review, and application in feature selection," in *Nature-Inspired Optimizers*. Cham, Switzerland: Springer, Feb. 2019, pp. 47–67, doi: 10.1007/978-3-030-12127-3_4.

[67] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition based grey wolf optimization," *Expert Syst. Appl.*, vol. 151, Aug. 2020, Art. no. 113389, doi: 10.1016/j.eswa.2020.113389.

[68] R. Alanni, J. Hou, H. Azzawi, and Y. Xiang, "A novel gene selection algorithm for cancer classification using microarray datasets," *BMC Med. Genomics*, vol. 12, no. 1, Dec. 2019, Art. no. 10, doi: 10.1186/s12920-018-0447-6.

BITANU CHATTERJEE is currently an Undergraduate Student with the Computer Science and Engineering, Jadavpur University, Kolkata, India. His areas of interest include machine learning, optimization, and graph theory.

TRINAV BHATTACHARYYA is currently an Undergraduate Student with the Computer Science and Engineering, Jadavpur University, Kolkata, India. His areas of interest include machine learning, optimization, and graph theory.

KUSHAL KANTI GHOSH is currently a senior year Undergraduate Student with the Computer Science and Engineering, Jadavpur University, Kolkata, India. His areas of interest include machine learning, optimization, game theory, and image processing.

PAWAN KUMAR SINGH (Member, IEEE) received the B.Tech. degree in information technology from the West Bengal University of Technology, in 2010, and the M.Tech. degree in computer science and engineering and the Ph.D. (engineering) degree from Jadavpur University (J.U.), in 2013 and 2018, respectively. He is currently working as an Assistant Professor with the Department of Information Technology, J.U. He has published more than 50 research papers in peer-reviewed journals and international conferences. His areas of current research interests are computer vision, pattern recognition, handwritten document analysis, image processing, machine learning, and artificial intelligence. He is a member of The Institution of Engineers (India) and the Association for Computing Machinery (ACM), as well as a Life Member of the Indian Society for Technical Education (ISTE), New Delhi, and the Computer Society of India (CSI). He also received the RUSA 2.0 Fellowship for pursuing his post-doctoral research in J.U., in 2019.

ZONG WOO GEEM received the B.Eng. degree from Chung-Ang University, the M.Sc. degree from Johns Hopkins University, and the Ph.D. degree from Korea University. He has researched at Virginia Tech, the University of Maryland at College Park, and Johns Hopkins University. He is currently an Associate Professor with the Department of Energy IT, Gachon University, South Korea. He invented a music-inspired optimization algorithm and harmony search, which has been applied to various scientific and engineering problems. His research interest includes phenomenon-mimicking algorithms and their applications to energy, environment, and water fields. He has served for various journals as an Editor (an Associate Editor for *Engineering Optimization* and a Guest Editor for *Swarm and Evolutionary Computation*, the *International Journal of Bio-Inspired Computation*, the *Journal of Applied Mathematics*, *Applied Sciences*, and *Sustainability*).

RAM SARKAR (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering from the University of Calcutta, in 2003, and the M.E. degree in computer science and engineering and the Ph.D. degree in engineering from Jadavpur University, in 2005 and 2012, respectively. He joined in the Department of Computer Science and Engineering, Jadavpur University, as an Assistant Professor, in 2008, where he is currently working as an Associate Professor. His areas of current research interest are image processing, pattern recognition, machine learning, and bioinformatics. He received the Fulbright-Nehru Fellowship (USIEF) for postdoctoral research in the University of Maryland at College Park, College Park, MD, USA, in 2014 and 2015.