Binary Social Mimic Optimization Algorithm With X-Shaped Transfer Function for Feature Selection

KUSHAL KANTI GHOSH[1], **PAWAN KUMAR SINGH**[2], (Member, IEEE),
JUNHEE HONG[3], (Member, IEEE), **ZONG WOO GEEM**[3], (Member, IEEE),
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)

ABSTRACT Definitive optimization algorithms are not able to solve high dimensional optimization problems when the search space grows exponentially with the problem size, and an exhaustive search also becomes impractical. To encounter this problem, researchers use approximation algorithms. A category of approximation algorithms is meta-heuristic algorithms which have shown an acceptable degree of efficiency to solve this kind of problems. Social Mimic Optimization (SMO) algorithm is a recently proposed meta-heuristic algorithm which is used to optimize problems with continuous solution space. It is proposed by following the behavior of people in society. SMO can efficiently explore the solution space for obtaining optimal or near-optimal solution by minimizing a given fitness function. Feature selection is a binary optimization problem where the aim is to maximize the classification accuracy of a learning algorithm using minimum the number of features. To convert the continuous search space to a binary one, a proper transfer function is required. The effect a transfer function has on the binary variant of an optimization algorithm is very important since selecting a particular subset of features based on the solution values attained by the algorithm in continuous search space depends on the considered transfer function. To this end, we have proposed a new transfer function, *namely* X-shaped transfer function, to enhance the exploration and exploitation ability of binary SMO. The proposed X-shaped transfer function utilizes two components and crossover operation to obtain a new solution. Effect of the proposed X-shaped transfer function is compared with the effect of four S-shaped and four V-shaped transfer functions on SMO in terms of achieved classification accuracy, rate of convergence, and number of features selected over 18 standard UCI datasets. The proposed algorithm is also compared with state-of-the-art meta-heuristic feature selection (FS) algorithms. Experimental results confirm the efficiency of the proposed approach in improving the classification accuracy compared to other meta-heuristic algorithms, and the superiority of X-shaped transfer function over commonly used S-shaped and V-shaped transfer functions. The source code of the proposed method along with the datasets used can be found at https://github.com/Rangerix/SocialMimic.

INDEX TERMS Social mimic optimization, transfer function, meta-heuristic, feature selection, UCI.

I. INTRODUCTION

In this era of computer and technology, with every advancement in the field of image processing, pattern recognition, financial analysis, business management, medical studies [1]–[4] and others, we are bound to deal with huge

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

amount of data, whose dimensions are increasing everyday. Two most important categories of the methods in the field of data mining are classification and clustering, which work on the features or attributes representing the dataset to make some prediction or to extract some useful information from such datasets. However, when the dimensions *i.e.*, number of features of the datasets are increased then the performance of these methods gets affected

considerably [5]. Again, high dimensional datasets have various disadvantages such as larger time requirement for construction of learning model, possible existence of irrelevant and redundant attributes or features, and degraded performance due to redundancy of features which make analysis or classification of the data very difficult. Here comes the importance of feature selection (FS) methods. FS is a data pre-processing step which aims to remove all possible irrelevant and redundant features [6] from the underlying dataset or feature vector, and thereby reducing the storage and time requirement to process the data.

FS is considered as an NP-complete combinatorial optimization problem. Generating all possible subsets of features and evaluating those are not feasible for large datasets. This is because, for a dataset containing n features, 2^n feature subsets will be generated and evaluating all of those requires a huge computational cost. There are randomized algorithms that attempt to search for the optimum feature subset in a randomized manner. On the other hand, a heuristic search strategy performs a guided search which may not always find the optimum solution but tries to produce a near-optimal solution in terms of computational time. Heuristic approaches are classified into two categories - specific heuristics which are designed for a particular problem, and general purposed meta-heuristics which are designed to solve a wide range of problems [7].

Based on the usage of learning algorithm, FS methods can broadly be divided into two categories [8]: filter and wrapper. Filter methods do not use any learning algorithm during elimination (selection) of the irrelevant (important) features, rather use different pre-defined scoring criteria to rank the features indicating their importance in terms of classification ability. Wrapper methods use learning algorithms (such as classifiers) as a part of the selection as well as evaluation of the subset of the selected features in each step of the algorithm. Filter methods are faster but wrapper methods, in general, perform much better [8]. Meta-heuristic methods are mostly wrapper based, since they require a classification algorithm for evaluation of a selected feature subset. In the last decade, meta-heuristic algorithms have become quite popular in solving FS problems also due to their ability to obtain an optimal or near-optimal solution in a reasonable time [9]. Two main characteristics of these algorithms are: exploration or diversification, which is the ability to search the whole solution space when looking for new solution in each iteration by avoiding local optima, and exploitation or intensification, which implies finding a better solution in the neighborhood of the obtained solution, leading to faster convergence. A good meta-heuristic algorithm tries to find a proper balance between exploration and exploitation.

In this work, we have proposed a meta-heuristic FS algorithm. Here, we have introduced a new transfer function and applied this transfer function to a recently proposed meta-heuristic optimization algorithm called Social Mimic Optimization (SMO) algorithm for the purpose of FS. Main contributions of this work are as follows:

- A new FS technique is developed following a recently proposed optimization algorithm called SMO.
- A novel X-shaped transfer function is introduced.
- The performance of the new transfer function in combination with SMO is compared with widely used S-shaped and V-shaped transfer functions.
- The proposed FS method is evaluated on 18 standard UCI datasets [10].
- It is also compared with five classical and five recently proposed meta-heuristic based FS methods.
- The performance of the proposed FS method is statistically validated using Wilcoxon rank-sum test [11].

The rest of this paper is organized as follows: Section II provides a brief review about the FS methods and transfer functions found in the literature. Section III provides detailed description of the proposed FS method. The results obtained by the FS versions of SMO are reported in Section IV. Section V provides the comparison results of the proposed model with state-of-the-art FS methods. Lastly, Section VI concludes this work and provides the possible future extension of this work.

II. LITERATURE SURVEY

FS is an optimization problem where the aim is to maximize the classification accuracy of a learning algorithm using minimum the number of features. The role of FS is crucial because it helps us gauge the performance of different machine learning and data mining techniques.

In the past two decades, nature-inspired meta-heuristic algorithms are at the forefront due to number of important factors of these algorithms: easy to adopt, flexible, usage of less mathematical derivation, their ability to avoid local optima. These algorithms have the ability to exploit the information of the population in order to find the optimal solutions. Meta-heuristic algorithms can also be divided into different categories based on different criteria: single solution based and population based [12], nature inspired and non-nature inspired [13], metaphor based and non-metaphor based [14]. From the 'inspiration' point of view, these algorithms can roughly be divided into four categories [15]: Evolutionary, Swarm inspired, Physics based, and Human related.

- **Evolutionary algorithms** are basically inspired from biology. It utilizes crossover and mutation operators to evolve the initial population, usually selected in a random fashion, over the iterations and eliminates the worst solutions in order to obtain the improved solution. Genetic algorithm (GA) [16] is a well-known method of this category which follows the Darwin's theory of evolution. Co-evolving algorithm [17], Cultural algorithm [18], Genetic programming [19], Grammatical evolution [20], Bio-geography based optimizer [21], Stochastic fractal search [22], Salp swarm algorithm [23], Black widow optimization [24], Barnacles mating optimizer [25] etc. are some well-known evolutionary algorithms.

- **Swarm inspired algorithms** imitate individual and social behavior of swarms, herds, schools, teams or any group of animals. Every individual has its own behavior, but the behavior of the accumulated individuals helps to solve complex optimization problems. One of the most popular algorithms of this category is Particle swarm optimization (PSO) [26], developed by following the behavior of flock of birds. Another notable method of this category is Ant colony optimizer (ACO) [27], inspired from the foraging method of some ant species. Some other methods belonging to this category are: Bacterial foraging [28], Firefly algorithm [29], Grey Wolf optimizer (GWO) [9], Ant Lion optimizer (ALO) [30], Whale optimization algorithm [31], Grasshopper optimization algorithm (GOA) [32], Squirrel search algorithm [33], Harris Hawks optimization (HHO) [34] etc.
- **Physics based algorithms** are inspired by the rules governing a physical process. The inspiring physical process ranges from music, metallurgy to mathematics, physics, chemistry, and complex dynamic systems. One of the oldest algorithms of this category is Simulated Annealing (SA) [35], developed by following the annealing [36] process of metals present in metallurgy and materials sciences. Another popular method of this category is Gravitational search algorithm (GSA) [37], developed by following gravity and mass interaction. Some other methods of this category are Harmony search (HS) algorithm [38], Black hole optimization [39], Sine Cosine algorithm [40], Multi-verse optimizer [41], Find-Fix-Finish-Exploit-Analyze [42], Atom search optimization [43], Equilibrium optimizer [44] etc.
- **Human related algorithms** searches for the global optima by following human behavior. Teaching-Learning-Based optimization [45] is one such popular method belonging to this category, developed by following the enhancing procedure of class grade. Some other methods of this category are: Society and civilization [46], League championship algorithm [47], Fireworks algorithm [48], Tug of war optimization [49], Volleyball Premier League algorithm [50], Political optimizer [51].

FS is a binary optimization problem, and transfer functions are required to convert the search space of a continuous optimization algorithm to a binary one. Transfer function generates a probability value based on the position/velocity of a solution and with this probability value, real valued solution is converted to a binary one. Kennedy and Eberhart have proposed binary PSO (BPSO) algorithm, using a sigmoid transfer function [52]. GA is used in [53] for the selection of features in automatic pattern classifier. In [54], the authors have proposed V-shaped transfer function. In [37], binary GSA (BGSA) is proposed using V-shaped transfer function ($|\tanh(x)|$). In [55], the authors have proposed eight binary variants of PSO using four S-shaped and four V-shaped transfer functions. These transfer functions are given in Table 1.

In [56], the authors have proposed six binary variants of ALO using three S-shaped S2, S3, and S4 (as mentioned in

TABLE 1. Popular S-Shaped and V-Shaped transfer functions [55] (used for comparison with X-Shaped transfer function).

S-shaped transfer functions		V-shaped transfer functions	
S1	$T(x) = \frac{1}{1+e^{-x}}$	V1	$T(x) = |tanh(x)|$
S2	$T(x) = \frac{1}{1+e^{-2x}}$	V2	$T(x) = \left|erf(\frac{\sqrt{\pi}}{2}x)\right|$
S3	$T(x) = \frac{1}{1+e^{-3x}}$	V3	$T(x) = \left|\frac{x}{\sqrt{1+x^2}}\right|$
S4	$T(x) = \frac{1}{1+e^{-x/2}}$	V4	$T(x) = \left|\frac{2}{\pi}\arctan(\frac{\pi}{2}x)\right|$

Table 1) and three V-shaped V2, V3, V4 (as mentioned in Table 1) transfer functions. In [57], Dragonfly algorithm is used for FS by utilizing V3 transfer function and applied on 18 standard UCI datasets. In [58], binary variants of GOA is proposed using S1 and V1 transfer functions. HHO [59] is converted to its binary version using S1 and V1 transfer functions and applied on microarray datasets. In [60], the authors have proposed binary variants of Butterfly optimization algorithm using S1 and V2 transfer functions and applied on 21 UCI datasets. In [61], four V-shaped transfer functions V1, V2, V3, and V4 are used to convert GWO into its binary variant for solving FS problems.

Presence of such a significant number of meta-heuristic FS algorithms along with transfer functions, clearly raises the question about the need for (i) another meta-heuristic FS method, and (ii) another transfer function. However, as indicated by *No Free Lunch* [62] theorem for optimization, there cannot be any single algorithm which will be equally applicable for all the optimization problems desiring optimal solutions. With each new algorithm following any regular or natural phenomenon, researchers primarily aim to provide some new facet to the algorithm where both exploration and exploitation will have a superior trade-off, thereby trying to get away from the local optima and eventually compass to the global optima. Nevertheless, accomplishing these objectives are not straightforward, hence motivating researchers to propose new algorithm that can be applicable to different problem domains. In summary, this is the key reason to the researchers to make an attempt in order to formulate better methods in comparison with the past methods which, thus keeps the research alive in this domain. For a specific problem, in order to discover the best algorithm, the *No Free Lunch* theorem ought to guide researchers that they have to concentrate on the particular problem at hand, the hypotheses, the priors (additional data), the information and the cost.

For the complex optimization problems, the multi-modal functions are having huge number of dimensions and finding an ideal value for all those dimensions at the same time is almost next-to-impossible. This challenging aspect of the optimization problem prompts researchers to plunge into the field of meta-heuristic strategies where the aim is to get an optimal solution within a reasonable amount of time. FS is considered as an optimization problem - there may exist numerous optimal feature subsets *i.e.*, having same dimension and same precision. Here likewise, it would be extremely

hard to discover an optimal feature set where burden of the extra storage space and running time alongside the performance of the machine learning algorithm would be lessen. In this way, research is still going on by developing new algorithms which can meet these requirements. This has also inspired us to propose a new meta-heuristic FS methodology based on the SMO [63] algorithm.

III. PRESENT WORK

A. SOCIAL MIMIC OPTIMIZATION: AN OVERVIEW

SMO algorithm [63] is proposed by following the human behavior. Each individual tries to 'mimic' or assimilate himself/herself to someone more esteemed, more intelligent and more powerful. Accordingly, each solution (analogous to an individual) in an optimization problem moves towards the global optima reached so far by imitating the parameters of that global optima. In this algorithm, *Follower* represents the population, *Follower*$_i$ represents i^{th} solution in the population, *Leader* represents the global optima obtained so far. During an iteration, each *Follower*$_i$ calculates the difference between its fitness value and the fitness value of the global optimal using Equation 1.

$$Difference = \frac{fitness(Leader) - fitness(Follower_i)}{fitness(Follower_i)} \quad (1)$$

$$if \, (Difference == 0) \, Difference = random(0, 1] \quad (2)$$

In next step, each *follower*$_i$ updates itself using Equation 3.

$$Follower_i = Follower_i + Difference \times Follower_i \quad (3)$$

The fitness value of each *Follower*$_i$ is calculated and *Leader* is updated accordingly.

A brief overview of the SMO algorithm is represented in Figure 1. The reason we have chosen this optimization method is because SMO is simple to implement but can produce effective results. Besides, it does not require any inherent parameter in contrary to other popular meta-heuristic algorithms, except only the population size and maximum number of iterations. As a result of this, no parameter tuning is required which itself requires exhaustive experiments to get the optimal values for the parameters of any algorithm.

B. PROPOSED METHOD

Let the original feature set be $\mathcal{F} = \{f_1, f_2, \ldots, f_D\}$, where D is the total number of features or the dimension of the feature set and let the class label be $\mathcal{C} = \{c_1, \ldots, c_l\}$, where l is the number of classes. The FS method tries to find out a subset $\mathcal{S} = \{s_1, \ldots, s_m\}$, where $m < D$, $\mathcal{S} \subset \mathcal{F}$ and $\mathcal{S}$ has lower classification error than any other subset of same size or any proper subset of $\mathcal{S}$.

It has already been mentioned that FS is a binary optimization problem [64], where the solution is limited to binary values {0, 1}. Here, a solution is represented using a binary vector where 1 indicates that corresponding feature is selected and 0 indicates otherwise. The size of this vector is equal to number of features in the original dataset. The SMO algorithm is proposed to solve continuous optimization
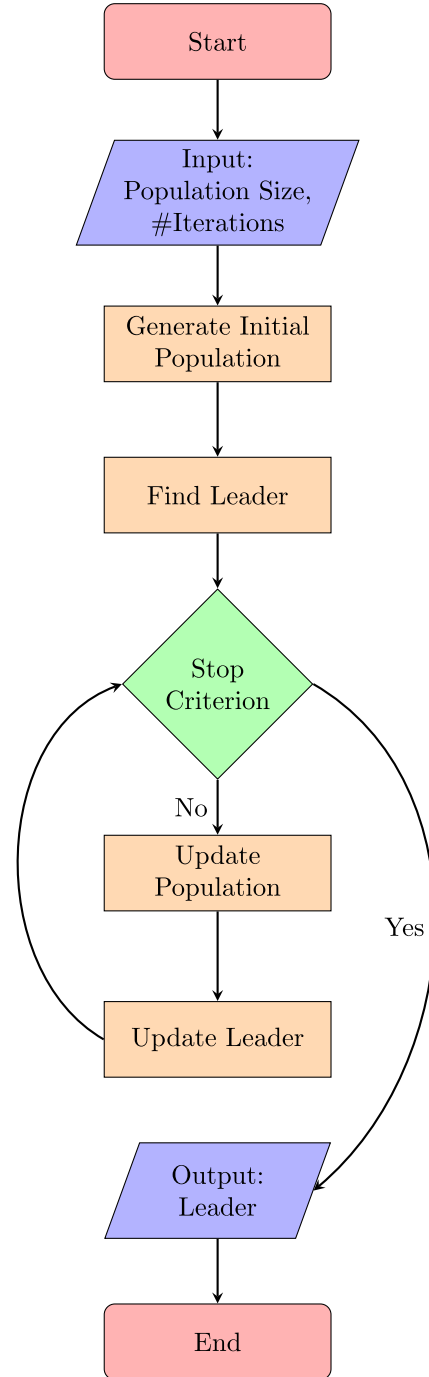

FIGURE 1. Flowchart of the SMO algorithm.

problems where a solution consists of real values. To map the continuous search space of the standard SMO algorithm to a binary one, a transfer function is required [55]. In the literature, there are mainly two types of transfer functions commonly used, which are S-shaped and V-shaped.

In case of S-shaped transfer functions, the solutions are updated based on Equation 4.

$$F_i^d(t+1) = \begin{cases} 1 & \text{if } rnd < SFunction(F_i^d(t+1)) \\ 0 & \text{if } rnd \geq SFunction(F_i^d(t+1)) \end{cases} \quad (4)$$

where $rnd \in [0, 1]$ is a random number, $F_i^d(t + 1)$ represents the dth dimension of the ith solution (*Follower*) in $(t + 1)^{th}$ iteration.

In case of V-shaped transfer functions, the solutions are updated based on Equation 5.

$$F_i^d(t + 1) = \begin{cases} F_i^d(t) & \text{if } rnd < VFunction(F_i^d(t + 1)) \\ \sim F_i^d(t) & \text{if } rnd \geq VFunction(F_i^d(t + 1)) \end{cases} \quad (5)$$

where $rnd \in [0, 1]$ is a random number, $F_i^d(t + 1)$ represents the dth dimension of the ith solution in $(t + 1)^{th}$ iteration, $F_i^d(t)$ represents the dth dimension of the ith solution in t^{th} iteration, $\sim F_i^d(t)$ represents the complement of $F_i^d(t)$, *i.e.*, if $F_i^d(t) = 0$, then $\sim F_i^d(t) = 1$ and *vice-versa*.

Now, in case of S-shaped transfer function, solution in the next $((t + 1)^{th})$ iteration is modified without considering the impact of solution in the current (t^{th}) iteration. This may diverge the agents, leading to slower convergence of the algorithm. In swarm inspired algorithms, where the agents are updated based on their velocity values, a big value of velocity in the positive or negative direction shows that the agents should have large movements to reach the optimum position. In contrast, a small value of the velocity indicates insignificant movement. Again, the zero velocity means that the new position should not be changed [54]. Now, these concepts are changed by using the S-shaped transfer function. The value of velocity in the negative and the positive directions creates different values for the new position. Moreover, the zero value of velocity generates either zero or one with probability 0.5 for the new position [54]. Whereas, with V-shaped transfer function, the solution may get stuck in local optima since if low velocities are associated with a particular solution, in next iteration the solution remains the same with high probability. Transfer function performs a key role in helping a binary optimization algorithm to find the optimum solution [55]. In early steps, the exploration is very important to search promising regions and avoid getting trapped in local optima but during the later steps, the exploitation is more essential so that the probability of finding better solutions gets increased. In other words, a balance between exploration and exploitation is essential in order to achieve a good result. In the literature, we have found many such cases where the meta-heuristic strategies need to be enhanced by a local or global search which would able to find the optimal solution [65]–[69].

Considering the limitations of the commonly used transfer functions found in the literature, we have introduced a new transfer function which is X-shaped. Two components, as shown in Figure 2, are used to generate two different results. The best result is chosen and compared with the previous solution. If the new solution is better than the previous one, it will be selected as the next position; otherwise, a crossover operator is applied on the new and previous solution. In this case, the best result of crossover operator is chosen as the new position. Due to crossover, there is a chance
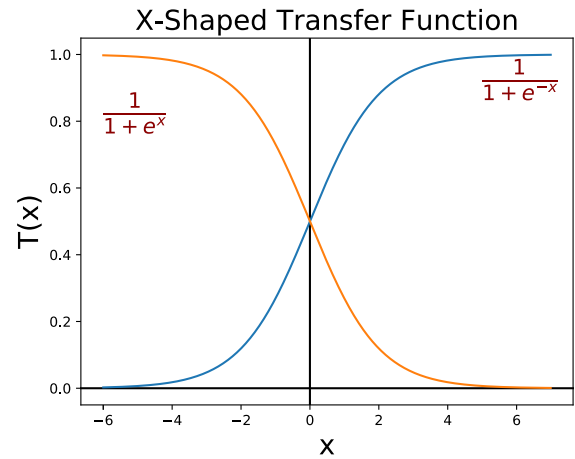

FIGURE 2. The proposed X-shaped transfer function.

for the new solution to retain the good characteristics of the solution of previous iteration.

To improve both the exploration and exploitation abilities of the optimization algorithm (SMO here), the two components are utilized using both Equation 6 and Equation 8, where Equation 8 is a mirror image of Equation 6 *w.r.t.* the line $y = 0$.

$$X_1(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

$$y_i^d(t + 1) = \begin{cases} 1 & \text{if } rnd1 < X_1(y_i^d(t + 1)) \\ 0 & \text{if } rnd1 \geq X_1(y_i^d(t + 1)) \end{cases} \quad (7)$$

$$X_2(x) = \frac{1}{1 + e^x} \quad (8)$$

$$z_i^d(t + 1) = \begin{cases} 1 & \text{if } rnd2 > X_2(z_i^d(t + 1)) \\ 0 & \text{if } rnd2 \leq X_2(z_i^d(t + 1)) \end{cases} \quad (9)$$

where, y_i and z_i are the binary versions of *Follower*$_i$ generated by Equation 6 and Equation 8 respectively, and $rnd1, rnd2 \in [0, 1]$ are random numbers.

$$F_i'(t + 1) = \begin{cases} y_i & \text{if } fitness(y_i) < fitness(z_i) \\ z_i & \text{if } fitness(y_i) \geq fitness(z_i) \end{cases} \quad (10)$$

Now, if $fitness(F_i'(t+1)) < fitness(F_i(t))$, then $F_i(t+1) := F_i'(t + 1)$. Otherwise, crossover operation is performed on $F_i'(t + 1)$ and $F_i(t)$. The crossover results in two children where the best one is chosen as the next solution. In this case, the child has a chance to retain the good qualities of the parent $F_i(t)$. Uniform crossover [70] has been chosen for crossover operation. This part is summed up in Equation 11.

if $fitness(F_i'(t + 1)) < fitness(F_i(t))$
    $F_i(t + 1) := F_i'(t + 1)$
else:
    $[child1, child2] = crossover(F_i'(t + 1), F_i(t))$
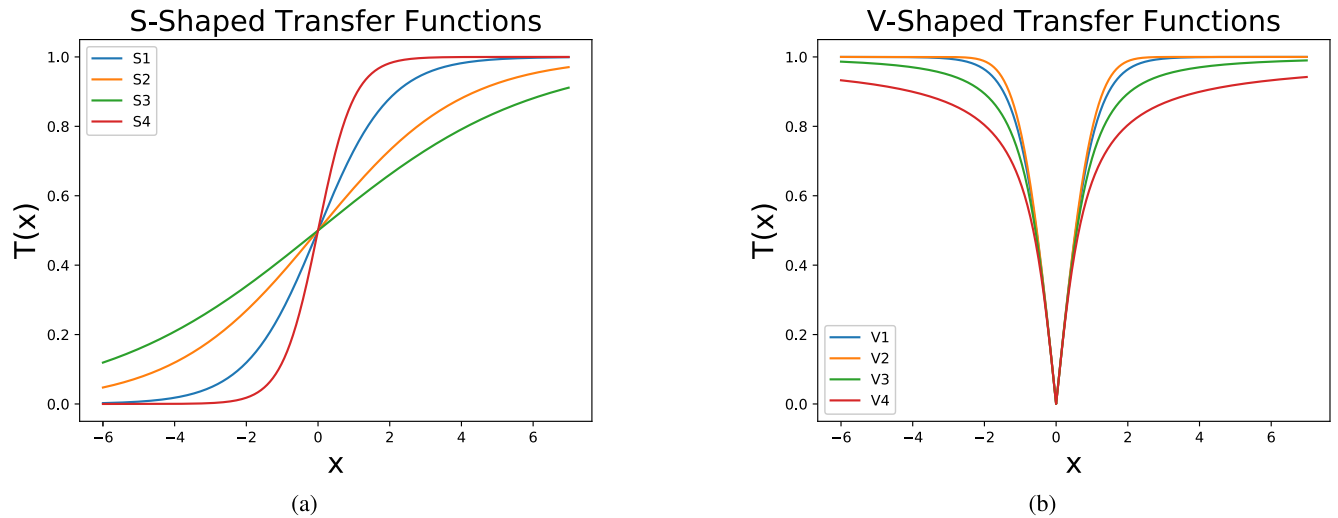   if $fitness(child1) < fitness(child2)$

FIGURE 3. (a) S-shaped and (b) V-shaped transfer functions used to compare the performance of proposed X-shaped transfer function.

TABLE 2. Description of the 18 UCI datasets utilized in the present work.

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

$$F_i(t+1) := child\,1$$

$$\text{else}$$

$$F_i(t+1) := child\,2$$

$$\text{endif}$$

$$\text{endif} \qquad (11)$$

In this work, we have compared the performance of the introduced X-shaped transfer function with the performance of eight different transfer functions (four S-Shaped and four V-Shaped transfer functions) when these are used with SMO algorithm. Table 1 shows the mathematical formulas of the eight transfer functions considered here whereas Figure 3 shows their corresponding graphs.

Now, FS is a multi-objective optimization problem with two main objectives: achieving maximum classification accuracy and selecting minimum number of features. Since these two goals are opposite in nature, we have considered classification error rate instead of accuracy. These two objectives are then combined into a single one and used as the fitness function, given in Equation 12. Each follower (solution) is assessed by the proposed fitness function which relies on the performance of the K-Nearest Neighbor (KNN) classifier [71] in order to determine the classification error rate and on the number of features selected.

$$\downarrow Fitness = \omega\gamma(\mathcal{F}') + (1-\omega)\frac{|\mathcal{F}'|}{|\mathcal{F}|} \qquad (12)$$

where $|\mathcal{F}|$ represents total number of features in the original dataset, $|\mathcal{F}'|$ represents the number of features in the selected subset, $\gamma(\mathcal{F}')$ denotes the classification error rate of $\mathcal{F}'$ using KNN classifier. $\omega \in [0, 1]$ denotes the importance of classification quality and selected subset dimension.
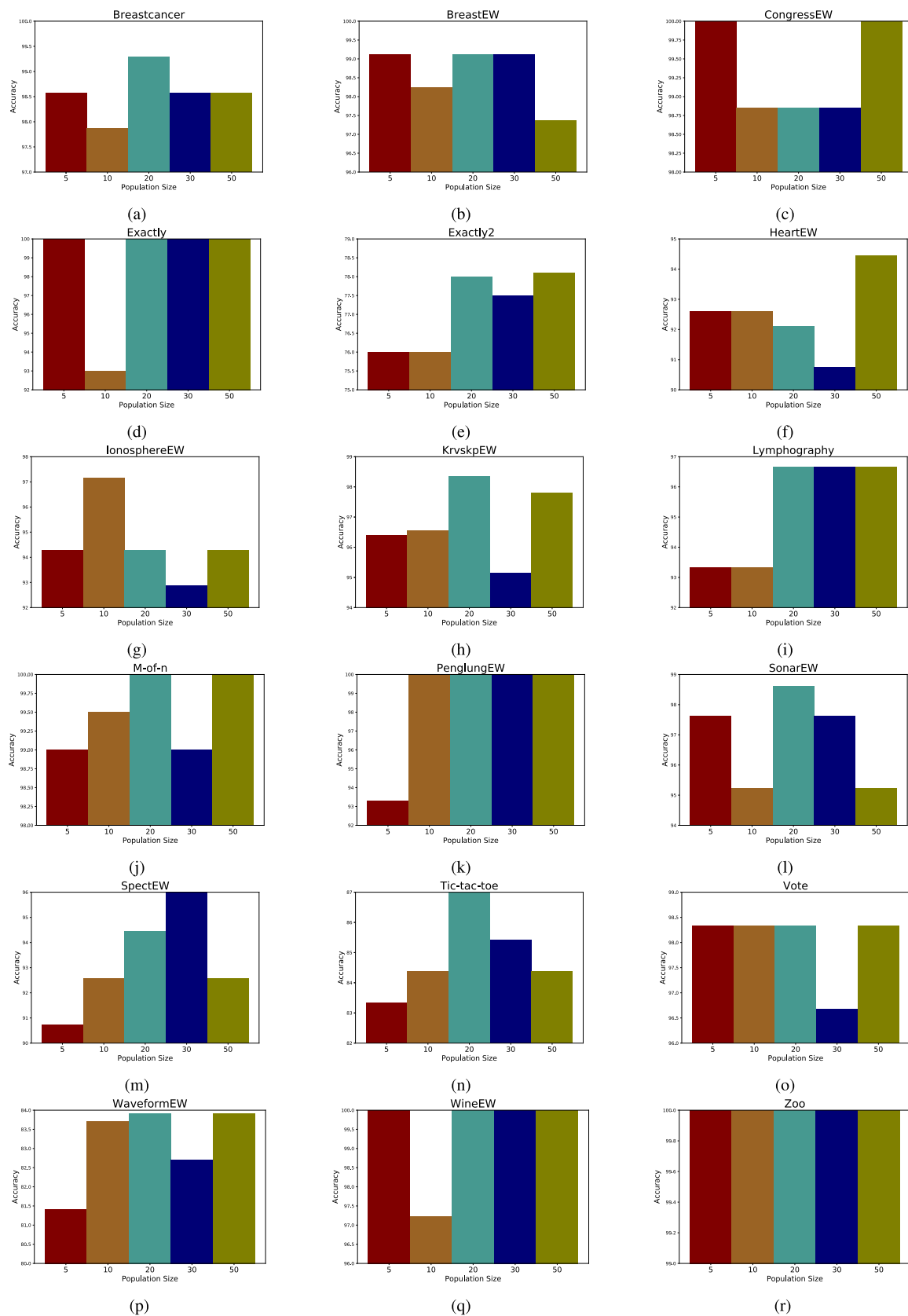
FIGURE 4. Effect of population sizes on classification accuracy using the proposed SMOX algorithm for 18 UCI datasets.
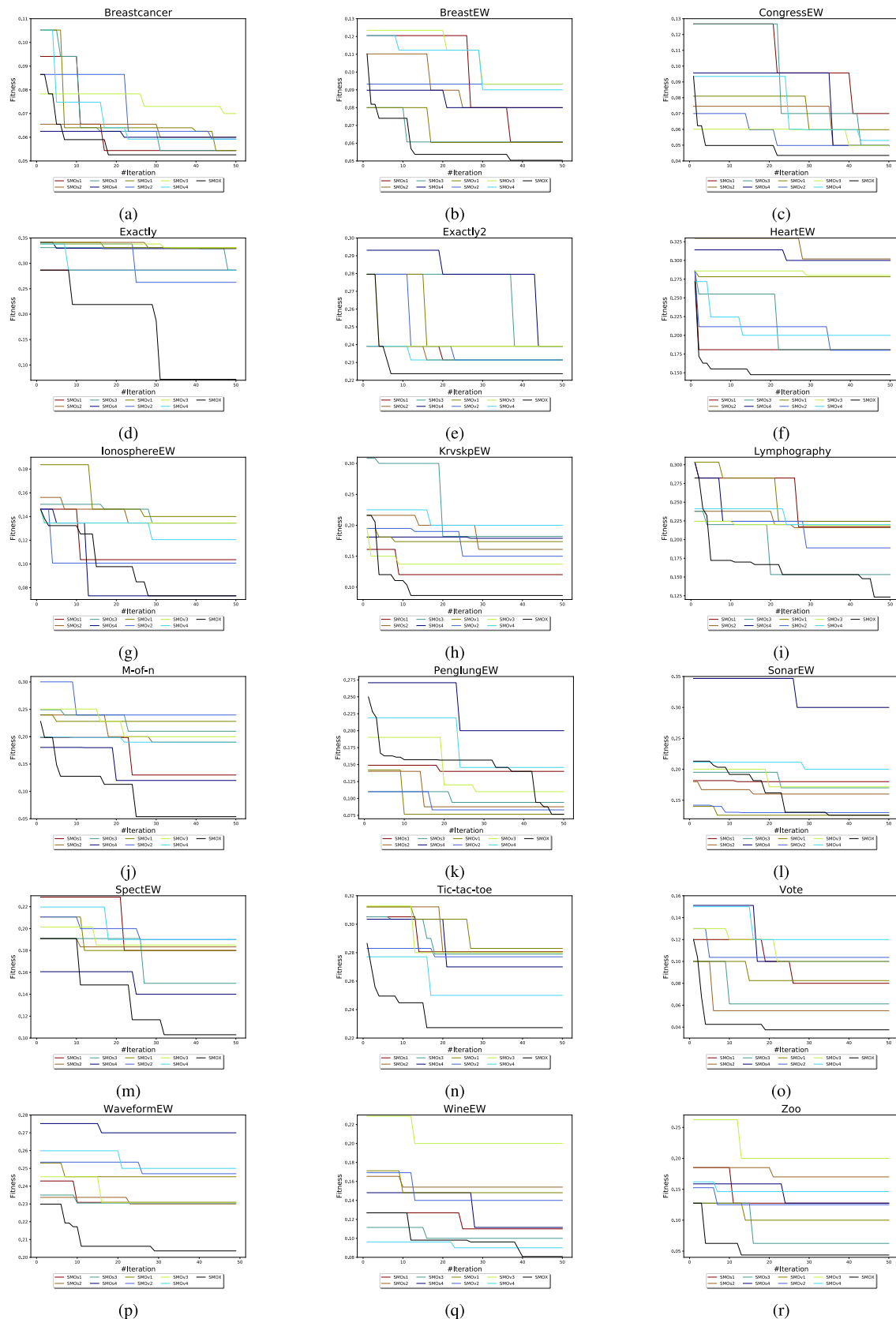
FIGURE 5. Best fitness values obtained in each iteration by SMO algorithm using X-shaped, four S-shaped and four V-shaped transfer functions on 18 UCI datasets.

—

TABLE 3. Comparison of the classification accuracies achieved by SMO algorithm using four S-shaped, four V-shaped and the proposed X-shaped transfer functions (the highest classification accuracies are highlighted in bold).

Dataset	SMOs1	SMOs2	SMOs3	SMOs4	SMOv1	SMOv2	SMOv3	SMOv4	SMOX
Breastcancer	98.57	99.29	98.57	97.85	97.14	97.86	98.57	98.57	**100**
BreastEW	97.37	98.25	96.49	97.36	96.49	96.49	98.25	96.49	**99.12**
CongressEW	97.7	96.55	97.7	96.55	98.85	97.7	96.55	98.85	**100**
Exactly	69	87	82	75.5	84	75	70	84.5	**100**
Exactly2	76	77	76	76	76	76	76	76	**80.5**
HeartEW	88.89	83.33	85.19	81.48	77.78	79.63	81.48	85.19	**92.59**
IonosphereEW	92.86	**95.71**	91.43	92.85	94.29	91.43	92.86	94.29	**95.71**
KrvskpEW	96.09	96.87	96.71	97.183	94.68	93.58	96.87	93.58	**98.6**
Lymphography	90	90	80	86.66	86.67	93.33	83.33	83.33	**96.7**
M-of-n	92.5	97	95.5	98	88	84	**100**	99.5	**100**
PenglungEW	93.33	**100**	93.33	**100**	**100**	93.33	**100**	**100**	**100**
SonarEW	90.48	95.24	95.24	95.23	90.48	85.71	92.86	88.1	**100**
SpectEW	85.19	**96.3**	88.89	92.59	79.63	87.04	94.44	83.33	**96.3**
Tic-tac-toe	76.56	81.25	78.65	81.25	77.6	78.65	77.08	77.08	**82.5**
Vote	**100**	**100**	96.67	**100**	96.67	98.33	98.33	**100**	**100**
WaveformEW	83.6	83.8	81	83.2	81.7	80.1	83.3	81.7	**84.5**
WineEW	97.22	**100**	**100**	**100**	97.22	97.22	**100**	**100**	**100**
Zoo	**100**	**100**	95	**100**	95	**100**	**100**	**100**	**100**

TABLE 4. Comparison of the number of features selected by SMO algorithm using four S-shaped, four V-shaped and the proposed X-shaped transfer function (bold values signify minimum number of features selected).

Dataset	SMOs1	SMOs2	SMOs3	SMOs4	SMOv1	SMOv2	SMOv3	SMOv4	SMOX
Breastcancer	4	4	4	**2**	3	3	4	3	3
BreastEW	6	3	7	**4**	10	10	13	4	9
CongressEW	2	8	3	5	10	6	4	5	**2**
Exactly	**1**	9	9	10	9	10	5	**1**	6
Exactly2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	11
HeartEW	4	**1**	6	6	9	2	7	9	4
IonosphereEW	7	13	9	**4**	10	12	9	10	8
KrvskpEW	27	15	27	28	26	21	24	25	**11**
Lymphography	11	10	9	7	5	12	**6**	9	7
M-of-n	10	10	10	8	8	8	7	9	**6**
PenglungEW	30	108	61	43	104	32	113	64	**20**
SonarEW	23	13	**9**	34	13	22	20	13	16
SpectEW	4	7	11	6	**1**	13	5	12	7
Tic-tac-toe	6	7	**5**	6	7	6	7	**5**	**5**
Vote	10	2	5	3	2	8	2	8	**1**
WaveformEW	31	27	26	24	30	**17**	30	31	19
WineEW	8	5	4	4	8	6	5	9	**3**
Zoo	5	10	10	**4**	10	7	8	9	4

The time complexity of the proposed method is $\mathcal{O}(maxIter \times popSize \times D \times t_{fitness})$, where $maxIter$ is the maximum number of iterations, $popSize$ represents the number of followers (individuals), D represents the dimension of the problem in consideration, and $t_{fitness}$ denotes the time requirement for calculating the fitness value of a particular individual using a given classifier. It is to be noted that the usage of X-shaped transfer function instead of S-shaped or V-shaped transfer functions, does not alter the time complexity.

IV. RESULTS AND DISCUSSION

We have used KNN [71] classifier with Euclidean distance metric to measure classification accuracy of the optimal feature subset selected by SMO algorithm. As per the recommendation found in the works described in [64], [72], [73], we have set $K = 5$. For each dataset, five fold cross-validation scheme is used for the evaluation purpose. Fundamentally, in k-fold cross-validation, the dataset is divided into k equal partitions (folds) where $k - 1$ folds are utilized for training and the remaining fold is utilized for testing the classification model. This procedure is iterated for M times. We have applied the FS methods on the train folds and determined which features are to be included in the selected feature subset. From test fold, only those features are selected and test classification accuracy is measured using the KNN classifier. Test fold is completely hidden from the FS method and used for the final evaluation purpose only. This work is implemented using Python3 [74] and graphs are plotted using Matplotlib [75].

TABLE 5. *p*-values obtained by the Wilcoxon rank-sum test for 18 UCI datasets using the proposed X-shaped transfer function as compared to four S-shaped and four V-shaped transfer functions.

Dataset	SMOs1	SMOs2	SMOs3	SMOs4	SMOv1	SMOv2	SMOv3	SMOv4
Breastcancer	0.001	0	0.001	0	0	0.002	0	0
BreastEW	0.001	0.001	0.002	0	0.003	0	0	0
CongressEW	0.041	0.038	0.02	0.007	0.017	0.084	0.017	0.01
Exactly	0	0	0	0	0	0	0	0
Exactly2	0.317	0.414	0.317	0.144	1	0.18	0.317	0.068
HeartEW	0	0	0	0	0	0	0	0
IonosphereEW	0	0	0	0	0	0	0	0
KrvskpEW	0	0.013	0	0	0	0.002	0.314	0.014
Lymphography	0	0	0	0	0	0	0	0
M-of-n	0	0	0	0	0	0	0	0
PenglungEW	0.005	0.008	0.025	0.157	0.046	0.046	0.014	0.011
SonarEW	0	0	0	0	0	0	0	0
SpectEW	0.001	0	0	0	0	0	0	0.001
Tic-tac-toe	0	0	0	0	0	0	0	0
Vote	0.002	0.001	0	0.001	0.001	0.001	0.001	0.001
WaveformEW	0.321	0.21	0.001	0	0.092	0	0.011	0
WineEW	0	0.001	0	0	0.001	0	0	0.001
Zoo	0.001	0.001	0.001	0	0	0	0.001	0
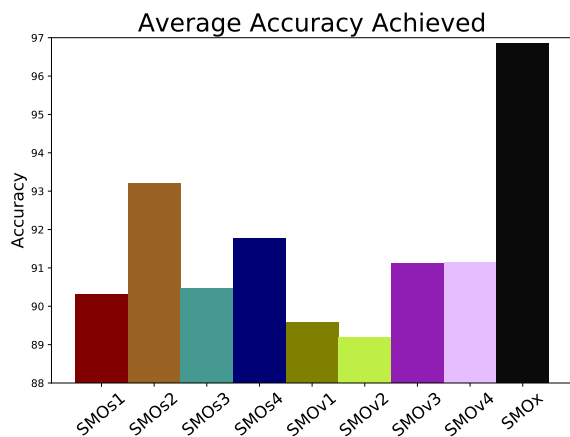
FIGURE 6. Average accuracies achieved by SMO algorithm using four S-shaped and four V-shaped and the proposed X-shaped transfer functions on 18 UCI datasets.
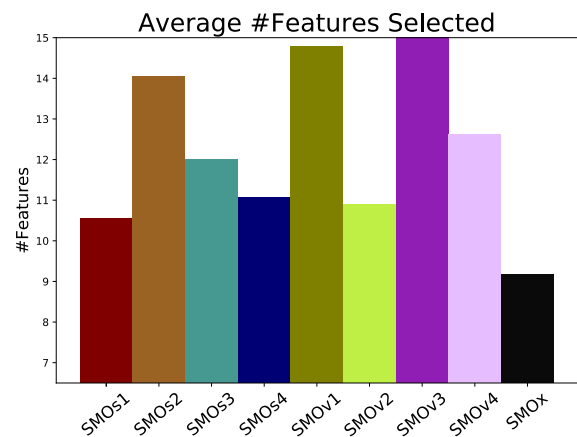
FIGURE 7. Average number of features selected by SMO algorithm using four S-shaped and four V-shaped and the proposed X-shaped transfer functions on 18 UCI datasets.

A. DATASET DESCRIPTION

For assessing the performance of the proposed FS method, 18 standard UCI datasets [10] are considered. The datasets are selected from various backgrounds. The underlying reason for selecting these datasets is that they are diverse in terms of number of attributes and instances present [76]. The description of these datasets is presented in Table 2. These variances help in establishing the robustness of the proposed method.

B. PARAMETER TUNING

There are two parameters which are very important for any multi-agent evolutionary algorithm: (a) population size and (b) maximum number of iterations to be used to run the algorithm. Population size characterizes how a single agent learns from other agents' experience, and iterations provide step-wise evolution of the agents. In order to find the optimal

values for these two parameters, exhaustive experiments have been performed by varying one parameter *w.r.t.* the other.

Figure 4 shows the effect of different population sizes on achieved classification accuracy using SMO algorithm with the proposed X-shaped transfer function. We have decided to set population size as 20 because (i) it is consistent, and (ii) it is able to achieve highest classification accuracy for most of the datasets. Figure 5 shows the values of the fitness function in each iteration using the proposed X-shaped, and the commonly used S-shaped and V-shaped transfer functions. Now, from the computational complexity of the SMO algorithm, mentioned in Section III, it can be observed that either increase in population size or maximum number of iterations, increases the time requirement. Considering both Figure 4 and Figure 5, it has been decided to set the values of

TABLE 6. Parameter values of the classic meta-heuristic FS methods used for comparison.

Method	Parameters
BGA	popSize = 20 maxIter = 30 Mutation and Crossover rate = 0.8
BPSO	popSize = 20 maxIter = 30 inertia factor = 0.1 individual-best acceleration factor = 0.1
BALO	popSize = 20 maxIter = 30
BGSA	popSize = 20 maxIter = 30 G=1 $\alpha = 20$
BHS	popsize = 20 maxIter = 30 HMCR = 0.8 PAR=0.2

population size as 20 and the maximum number of iterations as 30 for further experiments.

C. EXPERIMENTAL RESULTS

In this section, we have discussed about the results achieved by binary SMO algorithm using the proposed X-shaped transfer function and four S-shaped and four V-shaped transfer functions. The details related to these transfer functions are already mentioned in Table 1. We have denoted the binary SMO algorithm with i^{th} S-shaped and j^{th} V-shaped transfer functions (as mentioned in Table 1) as SMO*i* and SMO*v* respectively. The proposed binary SMO algorithm with X-shaped transfer function is abbreviated as SMOX.

Table 3 displays the classification accuracies achieved by the SMO*i*, SMO*vj*, and SMOX methods. Now, from Table 3, it can be observed that the SMOX algorithm has achieved the highest accuracy for all the utilized 18 UCI datasets. The SMOX algorithm is able to achieve 100% classification accuracy for nine cases (50%) which are: Breastcancer, CongressEW, Exactly, M-of-n, PenglungEW, SonarEW, Vote, WineEW, and Zoo. For BreastEW dataset, it has achieved the second best classification accuracy of 99.12%. In case of Exactly2, Tic-tac-toe, and WaveformEW datasets, the SMOX algorithm has achieved 80.5%, 82%, and 84.4% classification accuracies respectively.

Table 4 displays the number of features selected by the SMO*i*, SMO*vj*, and SMOX algorithms. From Table 4, it can be observed that the proposed SMOX algorithm has selected the minimum number of features for eight datasets which are: CongessEW, KrvskpEW, M-of-n, PenglungEW, Tic-Tac-Toe, Vote, WineEW and Zoo. However, the second best performing algorithm is found to be SMOs4 algorithm which selects the minimum number of features for five datasets: BreastCancer, BreastEW, Exactly2, IonosphereEW, and Zoo.

Figure 6 displays the average accuracies achieved by the nine (four SMO*i*, four SMO*vj*, SMOX) binary variants of

SMO algorithm over the utilized 18 UCI datasets. It can be clearly seen that the SMOX algorithm has achieved the highest classification accuracy among other binary variants. On an average, the SMOX algorithm has achieved about 96% classification accuracy. Figure 7 shows the average number of features selected by the nine binary variants of the SMO algorithm. From Figure 7, it can be observed that the SMOX algorithm has selected the lowest number of features in most of the cases. Upon averaging over the utilized 18 UCI datasets, it can be said that the proposed SMOX algorithm has selected < 10 features.

D. STATISTICAL ANALYSIS

To determine the statistical significance of the proposed SMOX algorithm, a non-parametric statistical test, known as Wilcoxon rank-sum test [11], has been performed. This is done in order to check whether the results of an algorithm are statistically different from other algorithms [77]. The null hypothesis states that the two sets of results are from the same distribution, therefore any difference in the two mean ranks comes only from sampling error. If the distributions of two results are statistically different, then the generated *p*-value from the test statistics will be < 0.05 (level of significance), as we have performed the test at 0.05% significance level, resulting in the rejection of the null hypothesis.

Here, we have deployed Wilcoxon test to prove that the obtained results by the proposed SMOX algorithm is statistically different from the obtained results by both the SMO*i* and SMO*vj* methods. For every datasets, each of the binary variants has been made to run 20 times and the accuracies obtained by the SMOX algorithm is compared with each of the SMO*i* and SMO*vj* methods via Wilcoxon test. The *p*-values obtained for pair-wise comparison of the SMOX, SMO*i* and SMO*vj* algorithms on 18 UCI datasets are provided in Table 5.

V. COMPARISON

In section IV, the proposed X-shaped transfer function has already proved its superiority in comparison to other transfer functions. In this section, we have compared the proposed SMOX algorithm with some popular meta-heuristic FS methods present in literature.

A. COMPARISON WITH CLASSIC META-HEURISTIC FS METHODS

Here, we have compared the results obtained by the SMOX algorithm with five traditional state-of-the-art approaches which are widely applied to solve FS problems in the literature. These approaches are GA, PSO, ALO, GSA, and HS. The values of the control parameters considered for these five methods are mentioned in Table 6.

Table 7 shows the performance of the SMO algorithm as compared to the above mentioned five methods both in terms of classification accuracies achieved and number of features selected. The SMOX algorithm has achieved better classification accuracy than BGA method for 15 cases as well

TABLE 7. Comparison of classification accuracies achieved and number of features selected by the proposed SMOX algorithm with five classic meta-heuristic FS methods (highest accuracy and lowest #features are highlighted in bold).

| Dataset	BGA		BPSO		BALO		BGSA		BHS		SMOX	
	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features
Breastcancer	0.9743	4	0.9629	4	0.974	4.7	0.9686	4	0.9928	5	**1**	**3**
BreastEW	0.9754	8	0.9719	9	0.974	13.85	0.9544	10	0.9561	**6**	**0.9912**	9
CongressEW	0.9679	**2**	0.9633	3	0.981	6.65	0.9633	4	0.9885	4	**1**	**2**
Exactly	1	6	1	6	0.965	5.75	0.994	**4**	0.69	1	**1**	6
Exactly2	0.77	**1**	0.768	**1**	0.762	1.5	0.77	**1**	0.76	**1**	**0.805**	6
HeartEW	0.8741	5	0.837	**3**	0.838	8.6	0.8296	**3**	0.8889	5	**0.9259**	4
IonosphereEW	0.9489	7	0.9489	7	0.904	11.75	0.9432	9	0.9285	**3**	**0.9571**	8
KrvskpEW	**0.985**	11	0.9731	12	0.973	16.15	0.9549	14	0.8529	**10**	**0.985**	11
Lymphography	0.8378	**5**	0.8919	**5**	0.917	7.35	0.8649	6	0.9333	10	**0.9667**	7
M-of-n	1	6	1	6	0.967	6	0.994	**5**	0.845	6	**1**	6
PenglungEW	0.9189	84	0.9189	130	0.827	133.1	0.8333	140	**1**	77	**1**	**20**
SonarEW	0.9904	19	0.9423	22	0.845	26.6	0.9135	24	0.9523	**16**	**1**	**16**
SpectEW	0.8955	**5**	0.8881	6	0.899	7.65	0.8433	**5**	0.8703	8	**0.963**	7
Tic-tac-toe	0.7996	5	0.7996	6	0.783	5	0.7766	**4**	0.7656	3	**0.82**	5
Vote	0.9733	5	0.96	3	0.972	6.6	0.96	4	0.9833	3	**1**	**1**
WaveformEW	0.7836	15	0.756	15	0.797	20.5	0.7344	**14**	0.806	16	**0.844**	19
WineEW	0.9888	4	0.9775	5	0.972	5.4	0.9775	4	0.9722	**3**	**1**	**3**
Zoo	0.902	**4**	0.9608	5	0.98	5.7	0.9804	6	0.95	6	**1**	**4**

as achieved same accuracy in 3 cases. In comparison to BGA, the SMOX algorithm has selected lowest number of features in 6 cases and same number of features in 6 cases. The SMOX algorithm has achieved better classification accuracy than BPSO in 16 cases and achieved same accuracy in 2 cases. Considering selected number of features, the SMOX algorithm has 9 wins and 3 ties with BPSO method. As compared to both BALO and BGSA methods, the SMOX has achieved better accuracy for all the 18 cases. In terms of selected number of features, the SMOX algorithm has 14 wins and 2 ties with BALO method and 10 wins with BGSA method. In terms of classification accuracy, the SMOX algorithm outperforms Binary HS algorithm in 17 cases and achieved same classification accuracy for only PenglungEW dataset.

Figure 8 illustrates the average accuracies achieved by the proposed SMOX algorithm and five state-of-the-art FS methods considered here. From Figure 8, it can be observed that the SMOX algorithm has achieved the highest classification accuracy. Considering all the 18 UCI datasets, the SMOX algorithm has achieved > 95% classification accuracy. Figure 9 provides the average number of features selected by SMOX and five state-of-the-art FS methods. It can be seen from Figure 9 that the proposed SMOX algorithm has selected the lowest number of features w.r.t. all the methods considered. It can also be observed that the SMOX algorithm has selected < 8 features. This proves the robustness of the proposed SMOX algorithm.

To prove the statistical significance of the results obtained by SMOX as compared to the state-of-the-art FS methods, we have also performed Wilcoxon rank-sum test for pairwise comparison of the proposed SMOX with other methods. In Table 8, the obtained p-values for each pair of methods are provided, with $p < 0.05$ marked bold.

B. COMPARISON WITH RECENT META-HEURISTIC FS METHODS

In this section, we have compared the results obtained by the proposed SMOX algorithm with five recently proposed
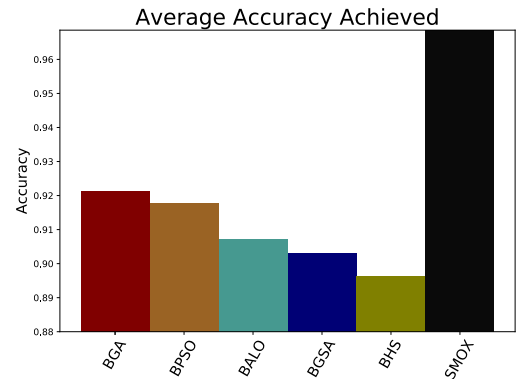
FIGURE 8. Average accuracies achieved by the proposed SMOX algorithm and five classic meta-heuristic methods over 18 UCI datasets.
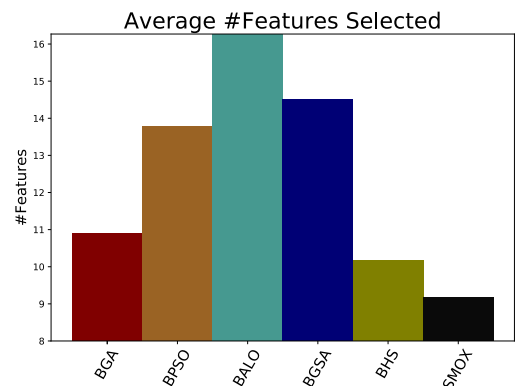
FIGURE 9. Average number of features selected by the proposed SMOX algorithm and five classic meta-heuristic methods over 18 UCI datasets.

meta-heuristic FS methods such as SSDs+LAHC, SSDv+LAHC, $A\beta$BSF, bBOA-S and BGWOPSO. SSDs+LAHC and SSDv+LAHC [69] are proposed by hybridizing the social ski driver (SSD) algorithm and late acceptance hill climbing (LAHC), and using a S-shaped transfer function S1 (as referred in Table 1) and a V-shaped

TABLE 8. Pairwise *p*-values of the Wilcoxon rank-sum test for the classification accuracy of the SMOX and five classic meta-heuristic FS methods considered here.

	BGA	BPSO	BALO	BGSA	BHS	SMOX
BGA	-	0.152	0.122	**0.017**	0.446	**0.001**
BPSO	0.152	-	0.616	**0.008**	0.948	**0.000**
BALO	0.122	0.616	-	0.528	0.879	**0.000**
BGSA	**0.017**	**0.008**	0.528	-	0.586	**0.000**
BHS	0.446	0.948	0.879	0.586	-	**0.000**
SMOX	**0.001**	**0.000**	**0.000**	**0.000**	**0.000**	-

TABLE 9. Parameter values of state-of-the-art methods used for comparison.

Method	Parameters
SSDs+LAHC SSDv+LAHC	popSize = 20 maxIter = 30 c = 100 $\alpha = 0.9$
AβBSF	popSize = 20 maxIter = 30 $\beta_{min}, \beta_{max} = 0.1, 1$ A = 4
bBOA-S	popSize = 20 maxIter = 30 $c_{min}, c_{max} = 0.01, 0.25$ a = 0.1
BGWOPSO	popSize = 20 maxIter = 30 c1=c2=c3=0.5 $l \in [0, 1]$

transfer function V3 (as referred in Table 1). AβBSF [78] is proposed by hybridizing sailfish optimizer with adaptive β-hill climbing algorithm. bBOA-S [60] is developed by following the recently proposed butterfly optimization algorithm (BOA) [79]. BGWOPSO [80] is developed by hybridizing both PSO and GWO methods. The parameter details of these methods considered for experimentation are mentioned in Table 9.

Table 10 shows the performance of the SMOX algorithm as compared to the above mentioned five FS methods both in terms of classification accuracies achieved and number of features selected. In terms of classification accuracy achieved, the SMOX algorithm is able to perform the best for almost 17 datasets. In case of BreastEW dataset, it performs the second best, following AβBSF method. For 11 datasets (61.11%), the SMOX algorithm has selected the lowest number of features. Figure 10 shows the average classification accuracies achieved by SMOX and the five recent meta-heuristic FS methods considered here. It clearly shows that the SMOX algorithm has achieved the highest average classification accuracy over all the 18 UCI datasets. Figure 11 shows the average number of features selected by SMOX and the five recent meta-heuristic FS methods considered. Now, from Figure 11, it can also be observed that the SMOX algorithm has selected the lowest number of features over all the 18 UCI datasets.
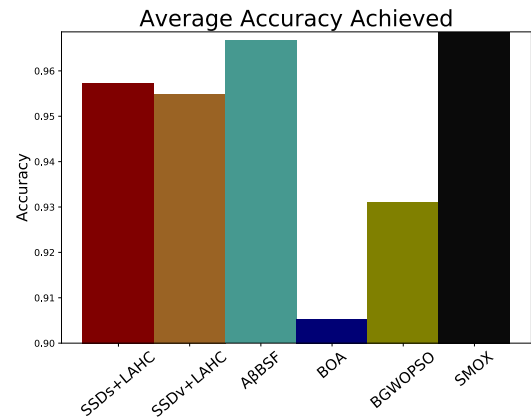

FIGURE 10. Average accuracies achieved by the proposed SMOX algorithm and five recent meta-heuristic FS methods over 18 UCI datasets.
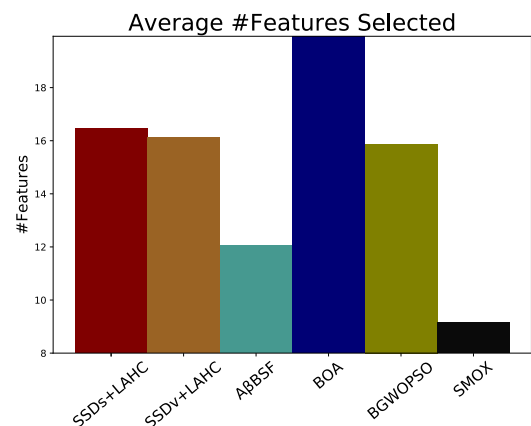

FIGURE 11. Average number of features selected by the proposed SMOX algorithm and five recent meta-heuristic FS methods over 18 UCI datasets.

To prove the statistical significance of the results obtained by the SMOX algorithm in comparison to the recently proposed meta-heuristic FS methods considered here, we have again performed Wilcoxon rank-sum test for pair-wise comparison of the SMOX with five recent meta-heuristic FS methods. In Table 11, the *p*-values obtained for each pair of methods are provided, with $p < 0.05$ marked bold. Table 11 clearly proves the statistical significance of the proposed SMOX algorithm.

A meta-heuristic algorithm can fail to find the optimal subset if (i) it cannot find the 'promising' area where the optimal solution (global optima) may lie, and converges to local optima, or (ii) it is unable to properly search the promising areas discovered, and fails to converge, or (iii) both. We have tried to address both these issues in the proposed SMOX algorithm. The proposed X-shaped transfer function utilizes two different components as well as crossover operation, thereby enhancing the search ability of the SMOX algorithm.

TABLE 10. Comparison of classification accuracies achieved and number of features selected by the proposed SMOX algorithm with five recent meta-heuristic FS methods (highest accuracy and lowest #features are highlighted in bold).

Dataset	SSDs+LAHC		SSDv+LAHC		AβBSF		bBOA-S		BGWOPSO		SMOX	
	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy	#Features
Breastcancer	0.9893	**2.5**	0.9929	3	**1**	4	0.9686	5.6	0.98	4.4	**1**	3
BreastEW	0.9825	**9**	0.9912	9.5	**1**	11	0.9709	16.8	0.97	13.6	0.9912	**9**
CongressEW	**1**	5.5	**1**	4.5	**1**	4	0.9593	6.4	0.98	4.4	**1**	**2**
Exactly	**1**	**6**	**1**	**6**	**1**	8	0.9724	7.6	**1**	**6**	**1**	**6**
Exactly2	0.79	8	0.8	7.5	**0.815**	11	0.7596	4.8	0.76	**1.6**	**0.815**	11
HeartEW	0.9167	5	0.9259	5	0.9444	5	0.8237	5.8	0.85	5.8	**0.9444**	**4**
IonosphereEW	0.9643	12	0.95	11.5	**0.9851**	**7**	0.907	16.2	0.95	13	**0.9851**	**7**
KrvskpEW	0.9781	20	0.9755	16.5	**0.9906**	32	0.966	17.6	0.97	**15.8**	**0.9906**	29
Lymphography	0.9667	6.5	0.9667	**6**	**1**	15	0.8676	8.4	0.92	9.2	**1**	13
M-of-n	**1**	**6**	**1**	6.5	**1**	7	0.972	6.8	**1**	**6**	**1**	**6**
PenglungEW	**1**	140	**1**	139	**1**	39	0.8775	172	0.96	130.8	**1**	20
SonarEW	0.9762	23.5	0.9524	23.5	0.9762	17	0.9362	32.8	0.96	31.2	**1**	**16**
SpectEW	0.9515	9	0.9352	11.5	0.9444	8	0.8463	10.8	0.88	8.4	**0.963**	**7**
Tic-tac-toe	0.8724	9	0.862	9	**0.8854**	7	0.7983	5.6	0.81	5.2	**0.8854**	**5**
Vote	**1**	4.5	**1**	4	**1**	3	0.9653	5.2	0.97	3.4	**1**	**1**
WaveformEW	0.844	22.5	0.837	20	**0.86**	30	0.7429	25	0.8	**14.2**	**0.86**	19
WineEW	**1**	3	**1**	**2**	**1**	3	0.9843	6.2	**1**	6	**1**	3
Zoo	**1**	4.5	**1**	5	**1**	6	0.9775	5.2	**1**	6.8	**1**	**4**

TABLE 11. Pairwise *p*-values obtained by the Wilcoxon rank-sum test for the classification accuracies achieved by the SMOX and five recent meta-heuristic FS methods.

	SSDs+ LAHC	SSDv+ LAHC	AβBSF	bBOA-S	BGW-OPSO	SMOX
SSDs+ LAHC	-	0.285	**0.007**	**0.000**	**0.001**	**0.003**
SSDv+ LAHC	0.285	-	**0.003**	**0.000**	**0.002**	**0.005**
AβBSF	**0.007**	**0.003**	-	**0.000**	**0.001**	**0.028**
BOA	**0.000**	**0.000**	**0.000**	-	**0.000**	**0.000**
BGW-OPSO	**0.001**	**0.002**	**0.001**	**0.000**	-	**0.001**
SMOX	**0.003**	**0.005**	**0.028**	**0.000**	**0.001**	-

VI. CONCLUSION AND FUTURE DIRECTIONS

In this work, we have proposed a new transfer function which inherently utilizes crossover operation thus helping the optimization algorithm to properly find any region where the global optima may lie. We have chosen a competent meta-heuristic algorithm called SMO, which is proposed recently following the human behavior of mimicking/copying more esteemed individuals. The SMO algorithm itself requires no such parameter to tune, since the agents simply follow the best agent found so far. We have compared the effect of the proposed X-shaped transfer function with four S-shaped and four V-shaped transfer functions commonly used in the literature while converting the continuous search space of SMO algorithm to a binary one. Publicly available 18 standard UCI datasets have been considered to assess the performance of our proposed algorithm. The comparison clearly displays the superiority of X-shaped transfer function both in terms of achieved classification accuracy and reduction of feature dimension. Hence, it can be concluded that the X-shaped transfer function aids SMO algorithm to search for the possible region towards achieving global optima. Finally, the proposed FS algorithm, SMOX (SMO with X-shaped transfer function) is compared with both five state-of-the-art FS methods and five recently proposed meta-heuristic FS methods. The experimental results show that the SMOX is able to achieve higher classification accuracy with lower number of features in both the cases. This, in turn, indicates that the SMOX is able to effectively search the feature space and find the optimal solution better than other FS methods. Statistical significance of the obtained results is also performed using Wilcoxon rank-sum test.

However, having the same stochastic nature as other meta-heuristic FS algorithms, as per *No Free Lunch* theorem [62], the SMOX is not guaranteed to produce outstanding results for all FS problems. As future scope of this work, we can apply the proposed X-shaped transfer function on different state-of-the-art FS methods. We can also apply SMOX on different real world problems, like musical symbol recognition, facial emotion recognition, handwritten digit/character/word recognition, etc. It would be interesting to investigate the performance of SMOX on high-dimensional datasets such as Microarray datasets. Enhanced initialization techniques can be thought of where the algorithm starts with an initial population closer to the global optima. We can also hybridize this algorithm with other population based meta-heuristic algorithms.

REFERENCES

[1] L. Xiong, R.-S. Chen, X. Zhou, and C. Jing, "Multi-feature fusion and selection method for an improved particle swarm optimization," *J. Ambient Intell. Humanized Comput.*, pp. 1–10, Dec. 2019, doi: 10.1007/s12652-019-01624-4.

[2] 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.

[3] X. Zhao, X. Zhang, Z. Cai, X. Tian, X. Wang, Y. Huang, H. Chen, and L. Hu, "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Comput. Biol. Chem.*, vol. 78, pp. 481–490, Feb. 2019, doi: 10.1016/j.compbiolchem.2018.11.017.

[4] M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai, H. Huang, and C. Tong, "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses," *Neurocomputing*, vol. 267, pp. 69–84, Dec. 2017, doi: 10.1016/j.neucom.2017.04.060.

[5] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015, doi: 10.1016/j.eswa.2015.07.007.

[6] 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.

[7] 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.

[8] 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, Aug. 2018, doi: 10.1007/s11517-018-1874-4.

[9] 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.

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

[11] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics* (Springer Series in Statistics). New York, NY, USA: Springer, 1992, pp. 196–202, doi: 10.1007/978-1-4612-4380-9_16.

[12] M. Gendreau and J.-Y. Potvin, "Metaheuristics in combinatorial optimization," *Ann. Oper. Res.*, vol. 140, no. 1, pp. 189–213, Nov. 2005, doi: 10.1007/s10479-005-3971-7.

[13] I. F. Jr, X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," 2013, *arXiv:1307.4186*. [Online]. Available: https://arxiv.org/abs/1307.4186

[14] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 185–231, doi: 10.1016/b978-0-12-813314-9.00010-4.

[15] A. F. Nematollahi, A. Rahiminejad, and B. Vahidi, "A novel meta-heuristic optimization method based on golden ratio in nature," *Soft Comput.*, vol. 24, no. 2, pp. 1117–1151, Apr. 2019, doi: 10.1007/s00500-019-03949-w.

[16] L. Davis, *Handbook of Genetic Algorithms*. New York, NY, USA: Van Nostrand Reinhold, 1991.

[17] W. D. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," *Phys. D, Nonlinear Phenomena*, vol. 42, nos. 1–3, pp. 228–234, 1990, doi: 10.1016/0167-2789(90)90076-2.

[18] X. Xue, M. Yao, and R. Cheng, "A novel selection operator of cultural algorithm," in *Advances in Intelligent and Soft Computing*. Berlin, Germany: Springer, 2011, pp. 71–77, doi: 10.1007/978-3-642-25661-5_10.

[19] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[20] C. Ryan, J. Collins, and M. O. Neill, "Grammatical evolution: Evolving programs for an arbitrary language," in *Genetic Programming* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 1998, pp. 83–96, doi: 10.1007/bfb0055930.

[21] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008, doi: 10.1109/tevc.2008.919004.

[22] H. Salimi, "Stochastic fractal search: A powerful Metaheuristic algorithm," *Knowl.-Based Syst.*, vol. 75, pp. 1–18, Feb. 2015, doi: 10.1016/j.knosys.2014.07.025.

[23] 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.

[24] 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.

[25] M. H. Sulaiman, Z. Mustaffa, M. M. Saari, and H. Daniyal, "Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103330, doi: 10.1016/j.engappai.2019.103330.

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. ICNN*, Nov. 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[27] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/mci.2006.329691.

[28] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002, doi: 10.1109/MCS.2002.1004010.

[29] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*. Berlin, Germany: Springer, 2009, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.

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

[31] 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.

[32] 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.

[33] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm Evol. Comput.*, vol. 44, pp. 148–175, Feb. 2019, doi: 10.1016/j.swevo.2018.02.013.

[34] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

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

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

[37] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Dec. 2009, doi: 10.1007/s11047-009-9175-3.

[38] 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.

[39] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2013, doi: 10.1016/j.ins.2012.08.023.

[40] 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.

[41] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Mar. 2015, doi: 10.1007/s00521-015-1870-7.

[42] A. Husseinzadeh Kashan, R. Tavakkoli-Moghaddam, and M. Gen, "Find-fix-finish-exploit-analyze (F3EA) meta-heuristic algorithm: An effective algorithm with new evolutionary operators for global optimization," *Comput. Ind. Eng.*, vol. 128, pp. 192–218, Feb. 2019, doi: 10.1016/j.cie.2018.12.033.

[43] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowl.-Based Syst.*, vol. 163, pp. 283–304, Jan. 2019, doi: 10.1016/j.knosys.2018.08.030.

[44] 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.

[45] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput. Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011, doi: 10.1016/j.cad.2010.12.015.

[46] T. Ray and K. M. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 386–396, Aug. 2003, doi: 10.1109/tevc.2003.814902.

[47] A. H. Kashan, "League championship algorithm (LCA): An algorithm for global optimization inspired by sport championships," *Appl. Soft Comput.*, vol. 16, pp. 171–200, Mar. 2014, doi: 10.1016/j.asoc.2013.12.005.

[48] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2010, pp. 355–364, doi: 10.1007/978-3-642-13495-1_44.

[49] A. Kaveh, "Tug of war optimization," in *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Cham, Switzerland: Springer, Nov. 2016, pp. 451–487, doi: 10.1007/978-3-319-46173-1_15.

[50] R. Moghdani, K. Salimifard, E. Demir, and A. Benyettou, "Multi-objective volleyball premier league algorithm," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105781, doi: 10.1016/j.knosys.2020.105781.

[51] Q. Askari, I. Younas, and M. Saeed, "Political optimizer: A novel socio-inspired meta-heuristic for global optimization," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105709, doi: 10.1016/j.knosys.2020.105709.

[52] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Simulation*, vol. 5, Oct. 1997, pp. 4104–4108.

[53] W. Siedledlecki and J. Skansky, "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.

[54] H. Nezamabadi-pour, M. Rostami-Shahrbabaki, and M. Farsangi, "Binary particle swarm optimization: Challenges and new solutions," *J. Comput. Soc. Iran Comput. Sci. Eng.*, vol. 6, no. 1, pp. 21–32, 2008.

[55] 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.

[56] M. Mafarja, D. Eleyan, S. Abdullah, and S. Mirjalili, "S-shaped vs. V-Shaped transfer functions for ant lion optimization algorithm in feature selection problem," in *Proc. Int. Conf. Future Netw. Distrib. Syst. ICFNDS*, 2017, pp. 1–7, doi: 10.1145/3102304.3102325.

[57] 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, pp. 12–17, doi: 10.1109/ictcs.2017.43.

[58] 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.

[59] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary harris hawks optimizer for high-dimensional, low sample size feature selection," in *Algorithms for Intelligent Systems*. Singapore Springer, Nov. 2019, pp. 251–272, doi: 10.1007/978-981-32-9990-0_12.

[60] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019, doi: 10.1016/j.eswa.2018.08.051.

[61] 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, doi: 10.1016/j.knosys.2020.105746.

[62] 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.

[63] S. Balochian and H. Baloochian, "Social mimic optimization algorithm and engineering applications," *Expert Syst. Appl.*, vol. 134, pp. 178–191, Nov. 2019, doi: 10.1016/j.eswa.2019.05.035.

[64] 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.

[65] 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.

[66] 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.

[67] M. Mafarja and S. Abdullah, "Investigating memetic algorithm in solving rough set attribute reduction," *Int. J. Comput. Appl. Technol.*, vol. 48, no. 3, p. 195, 2013, doi: 10.1504/ijcat.2013.056915.

[68] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105884, doi: 10.1016/j.asoc.2019.105884.

[69] B. Chatterjee, T. Bhattacharyya, K. K. Ghosh, P. K. Singh, Z. W. Geem, and R. Sarkar, "Late acceptance hill climbing based social ski driver algorithm for feature selection," *IEEE Access*, vol. 8, pp. 75393–75408, 2020, doi: 10.1109/access.2020.2988157.

[70] G. Syswerda, "Simulated crossover in genetic algorithms," in *Found. Genetic Algorithms*. Amsterdam, The Netherlands: Elsevier, 1993, pp. 239–255, doi: 10.1016/b978-0-08-094832-4.50021-0.

[71] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, p. 175, Aug. 1992, doi: 10.2307/2685209.

[72] 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, Jul. 2019, doi: 10.1007/s12559-019-09668-6.

[73] 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.

[74] G. van Rossum and F. L. Drake, *The Python Language Reference Manual*. Boston, MA, USA: Network Theory Ltd., 2011.

[75] 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.

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

[77] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/j.swevo.2011.02.002.

[78] K. K. Ghosh, S. Ahmed, P. K. Singh, Z. W. Geem, and R. Sarkar, "Improved binary sailfish optimizer based on adaptive β-Hill climbing for feature selection," *IEEE Access*, vol. 8, pp. 83548–83560, 2020, doi: 10.1109/access.2020.2991543.

[79] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Mar. 2018, doi: 10.1007/s00500-018-3102-4.

[80] Q. Al-Tashi, S. J. Abdul 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.

KUSHAL KANTI GHOSH is currently a senior year Undergraduate Student of computer science and engineering with Jadavpur University, Kolkata, India. His research interests 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. degree in engineering from Jadavpur University (JU), in 2013 and 2018, respectively. He is currently working as an Assistant Professor at the Department of Information Technology, JU. He has published more than 50 research articles in peer-reviewed journals and international conferences. His current research interests include 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) and a Life Member of the Indian Society for Technical Education (ISTE), New Delhi, and the Computer Society of India (CSI). He received the RUSA 2.0 Fellowship for pursuing his Postdoctoral Research in JU, in 2019.

JUNHEE HONG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 1987, 1989, and 1995, respectively. Since 1995, he has been a Professor with the Department of Energy IT, Gachon University, Seongnam, South Korea. His research interests include smart grids, supergrid, renewable energy, digitalization of electric power, and energy policy.

ZONG WOO GEEM (Member, IEEE) 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 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, 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 (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. In 2008, he joined the Department of Computer Science and Engineering, Jadavpur University, as an Assistant Professor, where he is currently working as an Associate Professor. His current research interests include 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, from 2014 to 2015.

● ● ●