

Received April 7, 2020, accepted April 26, 2020, date of publication April 30, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991543

Improved Binary Sailfish Optimizer Based on Adaptive β -Hill Climbing for Feature Selection

KUSHAL KANTI GHOSH¹, SHAMEEM AHMED¹, PAWAN KUMAR SINGH², (Member, IEEE), ZONG WOO GEEM³, (Member, IEEE), AND RAM SARKAR¹, (Senior Member, IEEE)

¹Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

²Department of Information Technology, Jadavpur University, Kolkata 700106, India

³Department of Energy IT, Gachon University, Seongnam 13120, South Korea

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

This work was supported in part by the National Research Foundation of Korea (NRF) through the Korean Government (MSIT) under Grant 2020R1A2C1A01011131, and in part by the Energy Cloud Research and Development Program through the National Research Foundation of Korea (NRF), Ministry of Science, ICT, under Grant 2019M3F2A1073164.

ABSTRACT Feature selection (FS), an important pre-processing step in the fields of machine learning and data mining, has immense impact on the outcome of the corresponding learning models. Basically, it aims to remove all possible irrelevant as well as redundant features from a feature vector, thereby enhancing the performance of the overall prediction or classification model. Over the years, meta-heuristic optimization techniques have been applied for FS, as these are able to overcome the limitations of traditional optimization approaches. In this work, we introduce a binary variant of the recently-proposed Sailfish Optimizer (SFO), named as Binary Sailfish (BSF) optimizer, to solve FS problems. Sigmoid transfer function is utilized here to map the continuous search space of SFO to a binary one. In order to improve the exploitation ability of the BSF optimizer, we amalgamate another recently proposed meta-heuristic algorithm, *namely* adaptive β -hill climbing ($A\beta$ HC) with BSF optimizer. The proposed BSF and $A\beta$ BSF algorithms are applied on 18 standard UCI datasets and compared with 10 state-of-the-art meta-heuristic FS methods. The results demonstrate the superiority of both BSF and $A\beta$ BSF algorithms in solving FS problems. The source code of this work is available in <https://github.com/Rangerix/MetaheuristicOptimization>.

INDEX TERMS Binary sailfish optimizer, feature selection, adaptive β -hill climbing, hybrid optimization, UCI dataset.

I. INTRODUCTION

With the recent advancements of computing devices, huge amount of data has become available in the domain of image processing, pattern recognition, financial analysis, business management, and medical studies [1], [2] amongst others. As a consequence, data dimensionality has increased a lot, which has huge impact on the performance of machine learning and data mining algorithms, both in terms of time and storage requirements. However, not all the attributes or features are important for the corresponding learning model. In this context, feature selection (FS) method is used as a data pre-processing step which helps remove all such irrelevant and redundant features [3], and thereby reduces required processing time and storage space. This, in turn, increases the overall classification (or prediction)

The associate editor coordinating the review of this manuscript and approving it for publication was Ioannis Schizas¹.

accuracy of the corresponding machine learning or data mining algorithms [4]. Depending on the evaluation criteria of features, FS techniques are divided in two categories [3]: filter and wrapper. Filter method evaluates features based on pre-defined criteria, (*e.g.*, Information Gain [5], ReliefF [6], Chi-square [7], Fisher Score [8], Laplacian score [9] etc.), and thereby selects the most important features according to that criteria. On the other hand, wrapper method uses a learning algorithm to evaluate feature subsets and thereby selects the optimum feature subset for the corresponding task [10]. Due to non-requirement of learning algorithms, filter methods perform faster than wrapper methods but wrapper methods, in general, achieve higher accuracy [11].

In the last decade, meta-heuristic algorithms have become quite popular in solving various optimization problems due to their ability to avoid local optima, non-derivative mechanism, and flexibility [12]. Two major characteristics of a meta-heuristic algorithm are [10]: exploration or

diversification, which is the ability to search the whole solution space for best solution in each iteration 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 balance between exploration and exploitation.

The presence of a significant number of meta-heuristic and hybrid meta-heuristic FS strategies clearly brings up the issue about presenting another hybrid meta-heuristic FS algorithm. In any case, as indicated by *No Free Lunch* theorem [13] for optimization, there cannot be any single algorithm that can take care of all the improvement issues. With each new algorithm following any regular or natural phenomena, researchers primarily focus to give some new facet to the algorithm where both exploration and exploitation will have a superior trade-off, so that it ultimately gets away from the local optima and compasses to the global optima. Nevertheless, accomplishing these objectives are not simple, explicitly in the event that one needs to propose an algorithm that can be applicable to different domains. This is the motivation to the researchers to formulate better methods over the past which, thus, keeps the research area alive. For a specific problem, in order to discover the best algorithm, the *No Free Lunch* theorem ought to advise us that we have to concentrate on the particular problem at hand, the hypotheses, the priors (additional data), the information and the cost.

While considering an optimization problem, the multimodal functions suffer from huge dimensions and finding an ideal value for every dimension at the same time is almost next-to-impossible. This is the reason why researchers attempt to take care of these sorts of issues utilizing some meta-heuristic strategies where the aim is to get an optimal solution within a reasonable amount of time. As FS is considered as an optimization problem [10], there may exist numerous optimal subsets i.e., having same dimension and same precision. Here likewise, it would be extremely hard to discover an optimal feature subset keeping in mind the extra storage space and running time alongside the performance of the machine learning algorithm. In this regard, research is still going on in order to meet these prerequisites. This has inspired us to propose a hybrid meta-heuristic FS method based on Sailfish Optimizer (SFO) [14] algorithm. SFO is proposed by mimicking the behavior of sailfish group hunting that follows attack-alternation strategy to hunt a school of sardine. In this paper, we have hybridized the binary variant of SFO, known as Binary Sailfish (BSF) optimizer with another recently proposed meta-heuristic algorithm, adaptive β -hill climbing ($A\beta HC$) [15]. Generally, two different models are followed for hybridizing meta-heuristic algorithms [16]: low level and high level. In low level hybridization, a function in a meta-heuristic is replaced by another meta-heuristic. In its high level variant, the candidate meta-heuristics are executed in sequence. We have hybridized BSF and $A\beta HC$ in a high level fashion, following the pipeline model, where each meta-heuristic optimization algorithm works on the output of

previous optimization algorithm. To the best of our knowledge, this is the first time BSF is used in FS where it is hybridized with $A\beta HC$ algorithm. In a nutshell, the main contributions of this work are as follows:

- A new FS method known as BSF optimizer is introduced using the recently proposed meta-heuristic SFO.
- The newly proposed BSF optimizer is hybridized with another recently proposed meta-heuristic called adaptive β -hill climbing ($A\beta HC$) algorithm.
- The proposed hybrid FS approach is evaluated on 18 standard UCI datasets [17] using K-nearest Neighbors (KNN) classifier.
- The proposed FS approach is compared with 10 state-of-the-art meta-heuristic based FS methods.

The rest of the paper is organized as follows: Section II provides a brief review of meta-heuristic FS methods present in the literature. Section III provides detailed description of the proposed FS method. The results obtained by the proposed method are reported in Section IV. In Section V, the proposed method is compared with 10 state-of-the-art meta-heuristic and hybrid meta-heuristic FS methods. Lastly, Section VI concludes this work, discusses its limitations and provides direction of possible future work.

II. LITERATURE SURVEY

Recently, the field of optimization has gained much attention from researchers, especially the field of hybrid meta-heuristics. Meta-heuristic is a high-level procedure [18] designed to find, generate, or select a heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem. Meta-heuristic algorithms can be divided into different categories: single solution based and population based [19], nature inspired and non-nature inspired [20], metaphor based and non-metaphor based [21] etc. From the 'inspiration' point of view, these algorithms can roughly be divided into four categories [22]: 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 random population over the iterations and eliminates the worst solutions to obtain improved solution. Genetic algorithm (GA) [23] is a well-known method of this category which follows the Darwin's theory of evolution. Co-evolving algorithm [24], Cultural algorithm [25], Genetic programming [26], Grammatical evolution [27], Bio-geography based optimizer [28], Stochastic fractal search [29] 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) [30], developed

by following the behavior of flock of birds. Another notable method of this category is Ant colony optimizer (ACO) [31], inspired from the foraging method of some ant species. Some other methods belonging to this category are: Shuffled frog leaping algorithm [32], Bacterial foraging [33], Artificial bee colony (ABC) [34], Firefly algorithm [35], Grey wolf optimizer (GWO) [12], Crow search algorithm [36], Whale optimization algorithm (WOA) [37], Grasshopper optimization algorithm [38], Squirrel search algorithm [39] 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) [40], developed by following the annealing [41] process of metals studied in metallurgy and materials sciences. Another popular method of this category is Gravitational search algorithm (GSA) [42], developed by following gravity and mass interaction. Some other methods of this category are: Self propelled particles [43], Harmony search algorithm [44], Black hole optimization [45], Sine cosine algorithm [46], Multi-verse optimizer [47], Find-Fix-Finish-Exploit-Analyze [48] etc.
- Human related algorithms find global optima by following human behavior. Teaching-Learning-Based optimization [49] 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 [50], League championship algorithm [51], Fireworks algorithm [52], Tug of war optimization [53], Volleyball premier league algorithm [54].

FS, however, is a binary optimization problem and most of the above mentioned optimization algorithms have been applied for solving FS problems. Different applications of GA for FS can be found in [55]–[58]. PSO based FS methods can be found in [59]–[61]. ACO and GSA based FS methods can be found in [62] and [42] respectively.

In recent times, hybrid meta-heuristic algorithms have been used in a large margin for solving FS problems. These algorithms are reported to achieve better performance in various real-life problems [63]. In [64], the first hybrid meta-heuristic algorithm is proposed for FS by combining GA with local search algorithm. The hybrid of Markov chain and SA is proposed in [65]. Memetic algorithm and Late acceptance hill climbing have been hybridized and used for FS for facial emotion recognition [66]. Spotted hyena optimizer is combined with SA and used for FS on UCI datasets in [3]. Hybrid of GA and SA has been used for FS on UCI datasets in [67]. In [68], Salp swarm algorithm is hybridized with Opposition based learning (OBL) and a local search method which are applied on UCI datasets for FS. In [69], GA and PSO have been hybridized for FS and applied on Digital mammogram datasets. In [70], the hybrid of GWO

and PSO has been applied on UCI datasets for FS. Hybridization of PSO with GSA can be found in [71]. Hybrid combination of ACO and GA has been proposed in [72]. In [73], a hybrid version of Differential Evolution (DE) and ABC for FS has been proposed and applied on UCI datasets.

The recently developed SFO is inspired by a group of hunting sailfish. This method is previously applied on 20 uni-modal and multi-modal mathematical functions. It is also applied for solving five engineering design problems such as Circular antenna array design problem, Gear train design problem, I-beam design problem, Three-bar truss design problem, and Welded beam design problem.

III. PRESENT WORK

A. SAILFISH OPTIMIZER: AN OVERVIEW

SFO [14] is a population based meta-heuristic algorithm, which is inspired from the attack-alternation strategy of a group of hunting sailfishes which hunts a school of sardines. This hunting strategy gives an upper hand to the hunters by providing them the chance of saving their energy. It considers two populations: sailfish population and sardine population. The sailfishes are considered as the candidate solutions and the problem's variables are positions of sailfishes in the search space. The algorithm tries to randomize the movement of search agents (both sailfish and sardine) as much as possible. Sailfishes are considered to be scattered in the search space, whereas the positions of sardines help to find the best solution in the search space.

The sailfish with best fitness value is called 'elite' sailfish and its position at i^{th} iteration is given by P_{SfBest}^i . In case of sardine, the 'injured' is the one with the best fitness value and its position at i^{th} iteration is given by $P_{SrdInjured}^i$. At each iteration, the positions of sardines and sailfishes are updated. At $i + 1^{th}$ iteration, the new position P_{Sf}^{i+1} of a sailfish is updated using 'elite' sailfish and 'injured' sardine as per Equation 1.

$$P_{Sf}^{i+1} = P_{SfBest}^i - \mu_i \times \left(rnd \times \frac{P_{SfBest}^i + P_{SrdInjured}^i}{2} - P_{Sf}^i \right) \quad (1)$$

where P_{Sf}^i is the previous position of the Sf^{th} sailfish, rnd is a random number between 0 and 1 and μ_i is a coefficient which is generated as per Equation 2.

$$\mu_i = 2 \times rnd \times PrD - PrD \quad (2)$$

where PrD is prey density, which indicates the number of preys at each iteration. At each iteration, the value of PrD , calculated by Equation 3, decreases as the number of prey decreases during group hunting.

$$PrD = 1 - \frac{Num_{Sf}}{Num_{Sf} + Num_{Srd}} \quad (3)$$

where Num_{Sf} and Num_{Srd} are the number of sailfishes and sardines respectively.

$$Num_{Sf} = Num_{Srd} \times Prcnt \quad (4)$$

where $Prcnt$ denotes the percentage of sardine population that forms the initial sailfish population. Initial number of sardines is always considered to be larger than number of sailfishes.

The sardine positions are updated in each iteration as given by Equation 5.

$$P_{Srd}^{i+1} = rnd(0, 1) \times (P_{SLfBest}^i - P_{Srd}^i + ATK) \quad (5)$$

$$ATK = A \times (1 - (2 \times itr \times \kappa)) \quad (6)$$

where P_{Srd}^i and P_{Srd}^{i+1} denote the previous and updated positions of the sardine respectively and ATK represents the sailfish's attack power at iteration itr . Now, the number of sardines that updates their positions and the amount of displacement depend upon ATK . Reducing the ATK assists the convergence of search agents. Using the parameter ATK , the number of sardines that updates their position (γ) and the number of variables of them (δ) are calculated as follows:

$$\gamma = Num_{Srd} \times ATK \quad (7)$$

$$\delta = v \times ATK \quad (8)$$

where v is the number of variables and Num_{Srd} is the number of sardines. If any of the sardines becomes fitter than any sailfish, the sailfish updates its position following this sardine, and the sardine is eliminated from its population.

Random selection of sailfishes and sardines guarantees the exploration of search space. As the attack power of sailfishes decreases after every iteration, it provides a chance to a sardine to escape from the best sailfish, which assists in exploitation. The ATK parameter tries to find a balance between exploration and exploitation.

B. ADAPTIVE β -HILL CLIMBING

$A\beta HC$ [15] is a recently proposed meta-heuristic algorithm, an adaptive version of the βHC [74], which is, in fact, an improved version of Hill climbing (HC) algorithm. HC algorithm is a simplest form of local search method. But it often gets stuck in local optima. To overcome this limitation, βHC is proposed. Given a solution $\mathcal{R} = (r_1, r_2, \dots, r_D)$, βHC iteratively generates an improved solution $\mathcal{R}'' = (r_1'', r_2'', \dots, r_D'')$ based on two operators: \mathcal{N} - operator (Neighborhood operator) and β -operator. The \mathcal{N} - operator randomly chooses a neighbor $\mathcal{R}' = (r_1', r_2', \dots, r_D')$ of the solution \mathcal{R} , which is defined in Equation 9 as:

$$r_i' = r_i \pm U(0, 1) \times \mathcal{N} \quad \exists i \in [1, D] \quad (9)$$

where i is chosen randomly in the range $[1, D]$, D is the dimension of the problem, \mathcal{N} denotes the highest possible distance between current solution and its neighbor. β - operator is inspired from the uniform mutation operator of GA. New solution is assigned values either from the current solution or randomly from the corresponding range with probability value $\beta \in [0, 1]$.

$$r_i'' = \begin{cases} r_r & \text{if } rnd \leq \beta \\ r_i' & \text{else} \end{cases} \quad (10)$$

where rnd is a random number, $rnd \in [0, 1]$, r_r is another random number within the range of that particular dimension of the problem in consideration.

Now, the outcome of this βHC largely depends on the chosen values of \mathcal{N} and β . Setting values for these two parameters requires exhaustive experiments. To avoid this overhead, $A\beta HC$ is proposed. In $A\beta HC$, \mathcal{N} and β are expressed as a function of iteration number. $\mathcal{N}(t)$ is the value of \mathcal{N} in t^{th} iteration. $\mathcal{N}(t)$ is defined as Equation 11 following the work presented in [47].

$$\mathcal{N}(t) = 1 - \frac{t^{\frac{1}{K}}}{MaxIter^{\frac{1}{K}}} \quad (11)$$

where K is a constant, $MaxIter$ is maximum number of iterations.

The value of β in t^{th} iteration is denoted as $\beta(t)$. As per [75], it is adapted within a specific range $[\beta_{min}, \beta_{max}]$ and defined as Equation 12.

$$\beta(t) = \beta_{min} + (\beta_{max} - \beta_{min}) \times \frac{t}{MaxIter} \quad (12)$$

Now, if the generated neighbor \mathcal{R}'' is better than the solution in consideration \mathcal{R} , \mathcal{R} is replaced with \mathcal{R}'' .

C. PROPOSED BINARY SAILFISH OPTIMIZER

Let the original feature set be $\mathcal{F} = \{f_1, f_2, \dots, f_D\}$, where D is the total number of features or the dimension of the feature set and let the class label be $\mathcal{C} = \{c_1, \dots, c_l\}$, where l is the number of classes. FS method finds a subset $\mathcal{S} = \{s_1, \dots, s_m\}$, where $m < D$, $\mathcal{S} \subset \mathcal{F}$ and \mathcal{S} has lower classification error rate than any other subset of same size or any proper subset of \mathcal{S} . FS is a binary optimization problem [10], 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 corresponding feature is not selected. The size of this vector is equal to number of features in the original dataset. SFO is proposed to solve continuous optimization problems where a solution consists of real values. To map the continuous search space of the standard SFO to a binary one, we use a transfer function [76]. We have used Sigmoid transfer function, depicted in Figure 1 and expressed by Equation 13.

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

Now, using the probability values generated by Equation 13, the current position of the Sailfish will be updated as per Equation 14.

$$X^d(t) = \begin{cases} 1 & \text{if } rnd < T(X^d(t)) \\ 0 & \text{if } rnd \geq T(X^d(t)) \end{cases} \quad (14)$$

Generally, FS is a multi-objective problem, with two objectives: (a) to achieve highest classification accuracy (i.e. maximization problem), and (b) to select lowest number of features (i.e. minimization problem). Now, these two

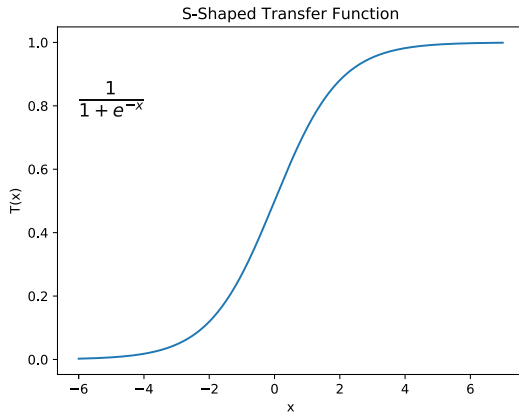


FIGURE 1. Utilized transfer function for converting continuous search space of SFO to binary.

objectives are contradictory in nature. To eliminate this contradiction, we have considered classification error rate. Using Equation 15, these two objectives are combined and the FS problem is converted to a single objective problem.

$$\downarrow \text{Fitness} = \omega\gamma(S) + (1 - \omega)\frac{|\mathcal{S}|}{D} \quad (15)$$

where \mathcal{S} represents the selected feature subset, $|\mathcal{S}|$ represents cardinality of the selected feature subset or number of selected features, $\gamma(S)$ represents classification error rate of \mathcal{S} , D is the original dimension of the dataset and $\omega \in [0, 1]$ represents weight.

Now, in SFO, the exploration is taken care of [14] by using random initialization of the sailfish and sardine population, and encircling strategy following hyper-sphere neighborhood. Exploitation is taken care of by using sardine population and movement of sardines around best sailfish and sardine. The parameter, ATK (defined in Equation 6) tries to find a balance between exploration and exploitation capabilities of the algorithm. Now, to find the optimal feature subset, the FS technique needs to find the global optima which requires proper exploration and exploitation of the search space. So, we have used $A\beta$ HC to enhance both the exploration and exploitation abilities of BSF.

The analysis of $A\beta$ BSF shows that, its worst case time complexity is $\mathcal{O}(\maxIter \times (N_{srd} \times t_{fitness} + D))$, where \maxIter is the maximum number of iterations, N_{srd} is the number of sardines, $t_{fitness}$ is the time requirement for calculating the fitness value of a particular agent using a given classifier and D is the dataset dimension.

IV. EXPERIMENTAL RESULTS

We have used KNN classifier [77] with Euclidean distance metric to measure the classification accuracy of the selected feature subset obtained by applying the proposed FS method on the entire (i.e. original) dataset. As per the recommendations provided in [10], [78], [79], we have set $K = 5$. For each dataset, 80% of the instances are used for training the model and rest 20% are used for testing. We have applied the

FS methods on the train data, and determined which features are to be included in the selected feature subset. From test data, only those features are selected and then classification accuracy is measured based on test data using KNN classifier. Proposed method is implemented using Python3 [80] and graphs are plotted using Matplotlib [81].

A. DATASET DESCRIPTION

In order to assess the performance of BSF and $A\beta$ BSF, 18 UCI datasets [17] have been considered. The datasets are selected from various backgrounds. The description of these datasets is presented in Table 1. Table 1 shows that there are 15 bi-class and 3 multi-class datasets. These datasets are diverse in terms of both number of attributes (features) and instances. These variances help in establishing the robustness of the proposed methods.

B. PARAMETER TUNING

There are two parameters which are always very important for any multi-agent evolutionary algorithm: population size and number of maximum iterations. Population size characterizes how a single agent learns from other agents' experiences, whereas iterations provide step-wise evolution of the agents. In order to find the proper values for these two parameters, experiments have been performed by varying one parameter *w.r.t.* the other.

We have experimented with different values of A , κ and $Prcnt$ mentioned in Equation 6 and Equation 4. Both A and κ are responsible for the decrease in attack power. ATK decreasing linearly from A to 0, intends to lead the method from exploration towards exploitation. As already mentioned in [14], we have also set $A = 4$ and $\kappa = 0.001$ through thorough experimentation. In case of $Prcnt$, mentioned in Equation 4, decrease in value implies more Sardines (larger N_{srd}), which in turn implies increased exploration. Hence, with decrease in $Prcnt$, the classification accuracy improves. But, as per the computational complexity of $A\beta$ BSF mentioned in Section III-C, the time requirement also increases with the increase of N_{srd} . So, in order to maintain a trade-off between the obtained classification accuracy and time requirement, we have decided to set the value of $Prcnt = 0.1$.

Figure 2 shows the effect of the size of the population on achieved classification accuracy using the proposed methods. Considering that the time requirement increases with increase in population size and the effect of the same on classification accuracy, we have decided to use population size as 20 for all the further experiments. Figure 3 shows the value of the fitness function in each iteration.

C. DISCUSSION

In this section, we have discussed about the results of the proposed BSF and $A\beta$ BSF methods for the datasets mentioned in Section IV-A. From Table 2, it is quite evident that the proposed methods have performed quite well for FS. BSF has achieved $\geq 90\%$ accuracy in case of 11 (61.1%)

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

Sl. No.	Dataset	No. of Attributes	No. of Samples	No. of Classes	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

datasets, whereas for $A\beta SF$, that count is 15 (83.33%). $A\beta SF$ has achieved 100% accuracy for 10 (55.5%) datasets such as Breastcancer, BreastEW, CongressEW, Lymphography, M-of-n, PenglungEW, Vote, WineEW, and Zoo, which is quite impressive. For KrvskpEW dataset, it has achieved 99.06% accuracy.

Now, observing Table 2, we can also conclude that $A\beta HC$ significantly helps the BSF algorithm to explore different parts of the search space and achieve better solution. In case of some datasets like Exactly, HeartEW, M-of-n, PenglungEW, SonarEW, and SpectEW, the classification accuracies have improved by a significant margin ($> 9\%$).

V. COMPARISON

To check the applicability of the proposed methods, we have compared it with 10 state-of-the-art methods: 4 popular meta-heuristic FS methods: GA, PSO, Ant Lion optimizer (ALO), and GSA and 6 hybrid meta-heuristic FS methods. GWO and WOA are hybridized following three different strategies [78]: serial grey-whale optimizer (HSGW), random switching grey-whale optimizer (RSGW), and adaptive switching grey-whale optimizer (ASGW). WOASAT-2 [10] is hybrid of WOA and SA. BGWOPSO [70] is developed by hybridizing GWO and PSO. In WOA-CM [82], the performance of WOA is enhanced by using crossover and mutation. The values of the control parameters of these methods are described in Table 3.

Table 4 shows the performance of $A\beta HC$ in terms of achieved classification accuracy. For each dataset, the methods are ranked as per their corresponding achieved classification accuracies. The Average rank is obtained by taking average over the utilized 18 UCI datasets. Assigned rank indicates the rank assigned to the methods as per their obtained average rank. From Table 4, it can be observed that $A\beta HC$ performs the best in 16 cases (88.9%), which is quite high. In case of Exactly2 dataset, it performs as the 2nd best

following HSGW, whereas for SonarEW dataset, it performs as the 3rd best following BGA and RSGW.

It is worth mentioning that $A\beta HC$ outperforms BGA completely in 15 cases, and ties in 2 cases. For SonarEW dataset, BGA performs better than $A\beta HC$. $A\beta HC$ outperforms PSO in 16 cases with ties in 2 cases. $A\beta HC$ completely outperforms both BALO and BGSA methods for all the 18 datasets, which is quite impressive.

The HSGW method outperforms $A\beta HC$ in case of Exactly2 dataset. Both HSGW and $A\beta HC$ methods ties in 4 cases. However, $A\beta HC$ outperforms HSGW method in 13 cases. With respect to RSGW method, $A\beta HC$ has 4 ties, 13 wins and 1 loss. The ASGW method is unable to outperform $A\beta HC$ for any of the datasets. In comparison with RSGW method, $A\beta HC$ has 5 ties and 13 wins. With respect to WOASAT2 method, $A\beta HC$ has 0 loss, 2 ties and 16 wins. Again, the BGWOPSO method fails to outperform $A\beta HC$ in any case. Moreover, both $A\beta HC$ and BGWOPSO methods tie in 4 cases whereas in 14 cases, $A\beta HC$ algorithm outperforms BGWOPSO. Both the methods namely WOA-CM and $A\beta HC$ tie in case of Exactly dataset and for the rest of 17 datasets, the proposed $A\beta HC$ algorithm wins.

Table 5 shows the performance of $A\beta HC$ w.r.t. number of selected features. Average rank is calculated by averaging the ranks obtained for each dataset based on the number of features selected. $A\beta HC$ has selected the lowest number of features in case of 6 datasets. For SonarEW dataset, though $A\beta HC$ algorithm fails to achieve the highest classification accuracy, it has selected lowest number of features and achieved 3rd best accuracy. So, considering both Table 4 and Table 5, we can say that $A\beta HC$ performs best with respect to the state-of-the-art methods considered here for comparison.

Figure 4 shows the average classification accuracies achieved by $A\beta BSF$ and 10 state-of-the-art methods used here for comparison. It clearly shows that $A\beta BSF$ has

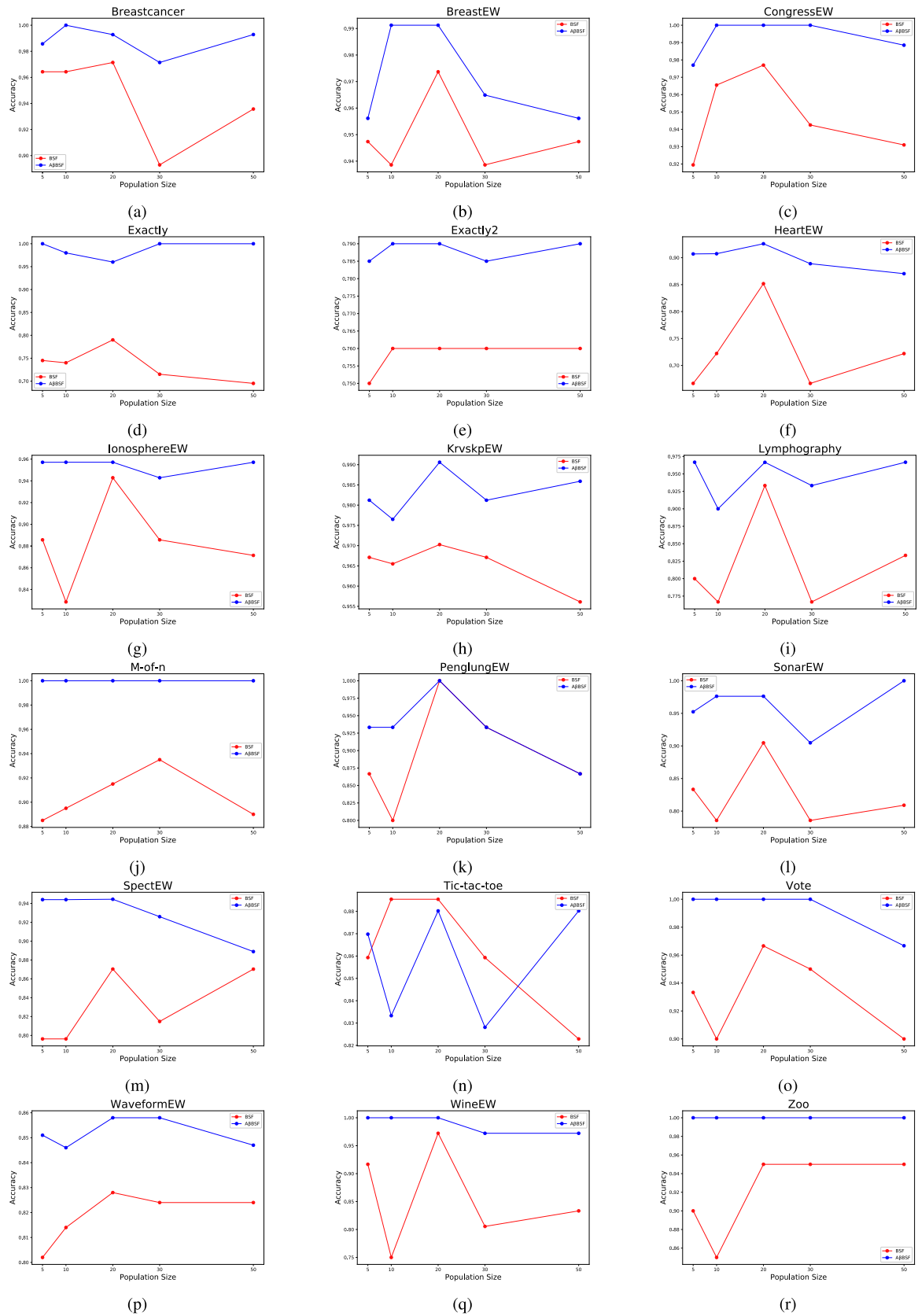


FIGURE 2. Effect of population size on classification accuracy for 18 UCI datasets using BSF and A β BSF.

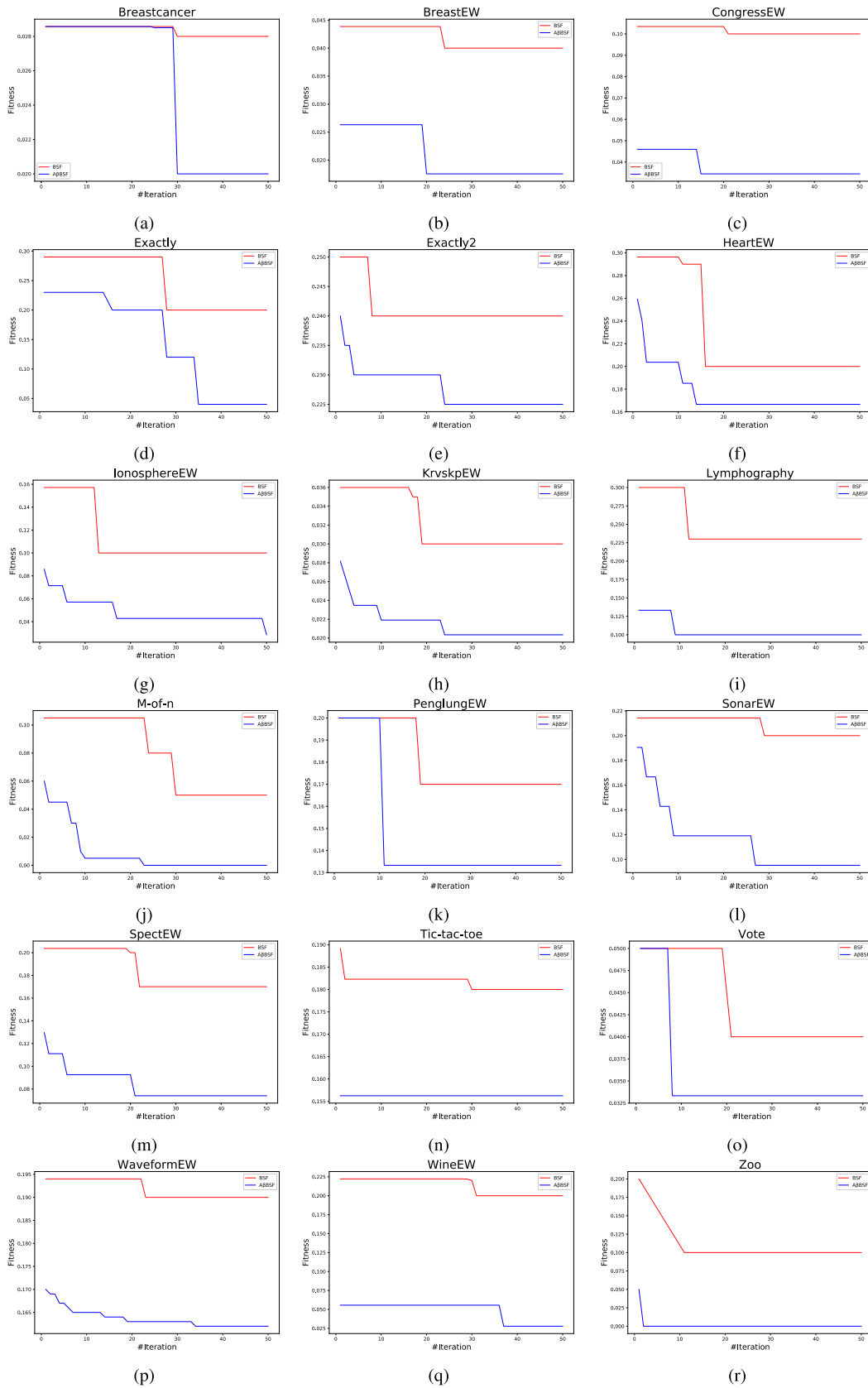


FIGURE 3. Best fitness value in each iteration for 18 UCI datasets using BSF and A β BSF.

TABLE 2. Performance of BSF and A β BSF in terms of classification accuracy and selected no. of features.

Sl. No.	Dataset	Original		BSF		A β BSF	
		Accuracy	#Features	Accuracy	#Features	Accuracy	#Features
1	Breastcancer	96	9	97.85	2	100	4
2	BreastEW	92.63	30	97.4	25	100	11
3	CongressEW	92.18	16	97.7	1	100	4
4	Exatly	72.3	13	79	13	100	8
5	Exactly2	73.3	13	76	1	81.5	11
6	HeartEW	68.15	13	85.18	6	94.44	5
7	IonosphereEW	83.43	34	94.27	34	98.51	7
8	KrVsKpEW	96.1	36	97.03	35	99.06	32
9	Lymphography	81.33	18	93.33	6	100	15
10	M-of-n	87.4	13	91.5	13	100	7
11	PenglungEW	81.33	325	90	52	100	39
12	SonarEW	80.95	60	88.47	50	97.62	17
13	SpectEW	82.22	22	85.04	20	94.44	8
14	Tic-tac-toe	81.1	9	81.1	9	88.54	7
15	Vote	92.33	16	95	5	100	3
16	WaveformEW	81.44	40	82.5	39	86	30
17	WineEW	66.67	13	97.22	5	100	3
18	Zoo	87	16	95	15	100	6

TABLE 3. Parameter setting of state-of-the-art methods used for comparison.

Algorithm	Parameters
BGA	pop_size = 8 max_iter = 70 Mutation and crossover rate = 0.8
BPSO	pop_size = 8 max_iter = 70 inertia factor = 0.1 individual-best acceleration factor = 0.1
BALO	pop_size = 8 max_iter = 200
BGSA	pop_size = 8 max_iter = 20 G=1 $\alpha = 20$
HSGW, RSGW, ASGW	pop_size = 20 max_iter = 100 a in GWO and WOA = [2 0] α in GSA = 20 G0 in GSA = 100
WOASAT-2	pop_size = 10 max_iter = 100 a in WOA = [2 0] selection pressure = 0.5
BGWOPSO	pop_size = 10 max_iter = 100 c1 = c2 = c3 = 0.5 w = 0.5+rand()/2 l \in [1, 0]
WOA-CM	pop_size = 10 max_iter = 100 a in WOA = [2 0]

achieved highest average classification accuracy as compared to state-of-the-art methods. Figure 5 shows the average number of features selected by A β BSF and the 10 state-of-the-art methods used. Now, from Figure 5, it can also be observed that the A β BSF algorithm has selected second lowest number of features.

To determine the statistical significance of A β BSF algorithm, Wilcoxon rank-sum test [83] has been performed. It is a non-parametric statistical test where pairwise comparison

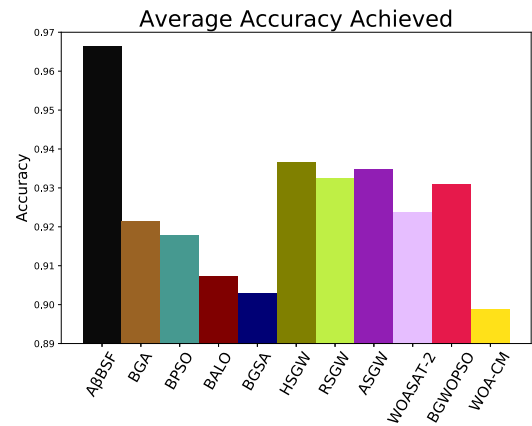


FIGURE 4. Average accuracies achieved by the proposed method and 10 state-of-the-art methods over the utilized 18 UCI datasets.

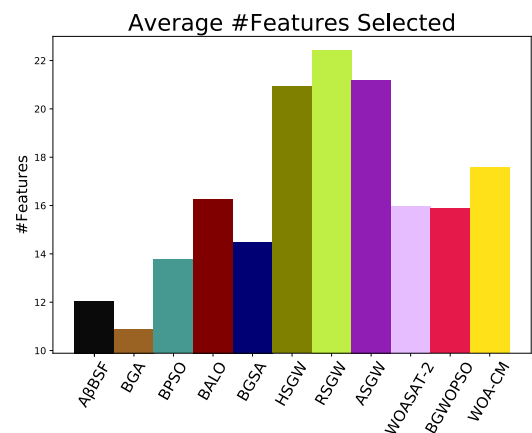


FIGURE 5. Average number of features selected by the proposed method and 10 state-of-the-art methods over the utilized 18 UCI datasets.

is performed. Here, the null hypothesis states that the two sets of results have same distribution. This implies that if the distribution of two results are statistically different, then the

TABLE 4. Comparison of classification accuracy obtained by proposed FS method with some state-of-the-art FS methods for 18 UCI datasets.

Dataset	A β BSF	BGA	BPSO	BALO	BGSA	HSGW	RSGW	ASGW	WOASAT-2	BGWOPSO	WOA-CM
Breastcancer	1	0.9743	0.9629	0.974	0.9686	0.986	0.971	0.985	0.97	0.98	0.968
BreastEW	1	0.9754	0.9719	0.974	0.9544	0.981	0.982	1	0.98	0.97	0.971
CongressEW	1	0.9679	0.9633	0.981	0.9633	0.975	0.961	0.994	0.98	0.98	0.956
Exactly	1	1	1	0.965	0.994	1	0.997	0.999	1	1	1
Exactly2	0.81	0.77	0.768	0.762	0.77	0.815	0.779	0.777	0.75	0.76	0.742
HeartEW	0.9444	0.8741	0.837	0.838	0.8296	0.923	0.848	0.831	0.85	0.85	0.807
IonosphereEW	0.9857	0.9489	0.9489	0.904	0.9432	0.944	0.978	0.972	0.96	0.95	0.926
KrvskpEW	0.9906	0.985	0.9731	0.973	0.9549	0.973	0.972	0.971	0.98	0.97	0.972
Lymphography	1	0.8378	0.8919	0.917	0.8649	0.934	0.893	0.884	0.89	0.92	0.852
M-of-n	1	1	1	0.967	0.994	1	1	1	1	1	0.991
PenglungEW	1	0.9189	0.9189	0.827	0.8333	0.942	1	1	0.94	0.96	0.792
SonarEW	0.9762	0.9904	0.9423	0.845	0.9135	0.964	0.979	0.948	0.97	0.96	0.919
SpectEW	0.9444	0.8955	0.8881	0.899	0.8433	0.862	0.815	0.87	0.88	0.88	0.866
Tic-tac-toe	0.8854	0.7996	0.7996	0.783	0.7766	0.828	0.859	0.865	0.79	0.81	0.785
Vote	1	0.9733	0.96	0.972	0.96	0.983	0.996	0.984	0.97	0.97	0.939
WaveformEW	0.86	0.7836	0.756	0.797	0.7344	0.748	0.757	0.746	0.76	0.8	0.753
WineEW	1	0.9888	0.9775	0.972	0.9775	1	1	1	0.99	1	0.959
Zoo	1	0.902	0.9608	0.98	0.9804	1	1	1	0.97	1	0.98
Average rank	1.16	4.47	5.84	5.84	7.47	3.47	3.79	3.84	4.68	4.26	7.63
Assigned rank	1	6	8	7	9	2	3	4	7	5	10

TABLE 5. Comparison of number of selected features obtained by proposed FS method with some state-of-the-art FS methods for 18 UCI datasets.

Dataset	A β BSF	BGA	BPSO	BALO	BGSA	HSGW	RSGW	ASGW	WOASAT-2	BGWOPSO	WOA-CM
Breastcancer	4	4	4	4.7	4	5	5.933	4.867	4.2	4.4	4.302
BreastEW	11	8	9	13.85	10	16.667	17.5	15.833	11.6	13.6	15.81
CongressEW	4	2	3	6.65	4	8.867	9.7	8.833	6.4	4.4	6.448
Exactly	8	6	6	5.75	4	6.7	7.1	6.867	6	6	6.045
Exactly2	11	1	1	1.5	1	9.033	9.2	7.933	2.8	1.6	5.252
HeartEW	5	5	3	8.6	3	8.767	6.133	6.367	5.4	5.8	6.995
IonosphereEW	7	7	7	11.75	9	18.167	20.5	17.3	12.8	13	14.416
KrvskpEW	32	11	12	16.15	14	24.8	24.8	24.5	18.4	15.8	18.54
Lymphography	15	5	5	7.35	6	10.567	10.567	11.2	7.2	9.2	8.208
M-of-n	7	6	6	6	5	6.8	7.1	6.867	6	6	6.006
PenglungEW	39	84	130	133.1	140	165.333	181.2	170.3	127.4	130.8	128.05
SonarEW	17	19	22	26.6	24	34.3	36.433	35.3	26.4	31.2	35.64
SpectEW	8	5	6	7.65	5	10.233	13.3	10.167	9.4	8.4	8.052
Tic-tac-toe	7	5	6	5	4	7	7	7	6	5.2	6.903
Vote	3	5	3	6.6	4	7.567	8.8	8.967	5.2	3.4	7.408
WaveformEW	30	15	15	20.5	14	26.933	27.533	25.833	20.6	14.2	25.4
WineEW	3	4	5	5.4	4	4.533	5.867	5.933	6.4	6	6.799
Zoo	6	4	5	5.7	6	5.533	5.3	7.6	5.6	6.8	6
Average rank	4.6	1.7	2.2	4.7	2.5	7.2	8.1	7.5	4.4	4.6	6
Assigned rank	5	1	2	6	3	7	9	8	4	5	7

TABLE 6. *p*-values obtained by the Wilcoxon rank-sum test for the classification accuracy of the proposed FS method as compared to 10 state-of-the-art FS methods considered here.

	BGA	BPSO	BALO	BGSA	HSGW	RSGW	ASGW	WOASAT-2	BGWOPSO	WOA-CM
A β BSF	0.001	0.000	0.000	0.000	0.001	0.001	0.001	0.000	0.001	0.000

generated *p*-value from the test statistics will be < 0.05 , when the test is performed at 0.05% significance level. This will result in the rejection of the null hypothesis. From the test results provided in Table 6, we can conclude that the results of the proposed A β BSF algorithm is found to be statistically significant.

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 gets stuck in local optima, or (ii) it fails to properly search the promising

areas discovered, and cannot converge, or (iii) both. We have tried to address both the issues in the proposed method. The proposed A β BSF method uses an encircling strategy [14], that provides a hyper-sphere neighborhood around the solutions. This ensures that all the possible solutions are examined properly, both in terms of exploration and exploitation: the two main aspects of any meta-heuristic algorithm.

In terms of the factors present in the proposed methods, exploration is controlled by sardine population and their diverse movement, and bandwidth of neighborhood (\mathcal{N}).

Exploitation is mainly taken care by the movement of sailfishes around best sardine, sailfish's attack power and, random-walk strategy in the β -operator.

VI. CONCLUSION

In this work, we have proposed binary version of SFO to solve FS problems. To convert continuous SFO to BSF (which is its binary variant), we have used the Sigmoid transfer function. Besides, we have hybridized SFO with $A\beta$ HC algorithm and proposed a binary hybrid meta-heuristic $A\beta$ BSF for FS. Both the proposed BSF and $A\beta$ BSF methods are applied on 18 standard UCI datasets. From the obtained results and comparison with 10 state-of-the-art meta-heuristic FS approaches, we can conclude that the proposed $A\beta$ BSF performs significantly well for solving FS problems.

Now, the proper functioning of the $A\beta$ BSF algorithm depends on the parameters of $A\beta$ HC and SFO. The optimal values of these parameters for a different set of problems may be completely different, which would require exhaustive experiments to determine. Therefore, it can be considered as a limitation of the proposed method. Also, as per *No Free Lunch* theorem [13], just like any other meta-heuristic optimization algorithm, the $A\beta$ BSF algorithm is not guaranteed to produce the best results for all FS problems. For future studies, we can apply the $A\beta$ BSF algorithm to different real world problems like facial emotion recognition, handwriting recognition, script recognition, etc. It would be interesting to hybridize BSF with other meta-heuristic algorithms as well as hybridize $A\beta$ HC with other 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. Hum. Comput.*, vol. 2019, Dec. 2019, doi: [10.1007/s12652-019-01624-4](https://doi.org/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](https://doi.org/10.1016/j.eswa.2019.112898).
- [3] 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](https://doi.org/10.1109/access.2019.2919991).
- [4] 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](https://doi.org/10.1016/j.compeleceng.2013.11.024).
- [5] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948, doi: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
- [6] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 249–256, doi: [10.1016/b978-1-55860-247-2.50037-1](https://doi.org/10.1016/b978-1-55860-247-2.50037-1).
- [7] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 80–89, Jun. 2004, doi: [10.1145/1007730.1007741](https://doi.org/10.1145/1007730.1007741).
- [8] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. 27th Conf. Uncertainty Artif. Intell.* Arlington, Virginia, USA: AUAI Press, 2011, pp. 266–273.
- [9] 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, p. 507–514.
- [10] 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](https://doi.org/10.1016/j.neucom.2017.04.053).
- [11] 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](https://doi.org/10.1007/s11517-018-1874-4).
- [12] 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](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [13] 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](https://doi.org/10.1109/4235.585893).
- [14] S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The sailfish optimizer: A novel nature-inspired Metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 20–34, Apr. 2019, doi: [10.1016/j.engappai.2019.01.001](https://doi.org/10.1016/j.engappai.2019.01.001).
- [15] M. A. Al-Betar, I. Aljarah, M. A. Awadallah, H. Faris, and S. Mirjalili, "Adaptive β -hill climbing for optimization," *Soft Comput.*, vol. 23, no. 24, p. 13 489–13 512, Mar. 2019, doi: [10.1007/s00500-019-03887-7](https://doi.org/10.1007/s00500-019-03887-7).
- [16] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Hoboken, NJ, USA: Wiley, 2009.
- [17] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [18] R. Balamurugan, A. M. Natarajan, and K. Premalatha, "Stellar-mass black hole optimization for biclustering microarray gene expression data," *Appl. Artif. Intell.*, vol. 29, no. 4, pp. 353–381, Apr. 2015, doi: [10.1080/08839514.2015.1016391](https://doi.org/10.1080/08839514.2015.1016391).
- [19] M. Gendreau and J.-Y. Potvin, "Metaheuristics in combinatorial optimization," *Ann. Operations Res.*, vol. 140, no. 1, pp. 189–213, Nov. 2005, doi: [10.1007/s10479-005-3971-7](https://doi.org/10.1007/s10479-005-3971-7).
- [20] 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>
- [21] 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](https://doi.org/10.1016/b978-0-12-813314-9.00010-4).
- [22] 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, Jan. 2020, doi: [10.1007/s00500-019-03949-w](https://doi.org/10.1007/s00500-019-03949-w).
- [23] L. Davis, *Handbook of Genetic Algorithms*. New York, NY, USA: Van Nostrand Reinhold, 1991.
- [24] 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, Jun. 1990, doi: [10.1016/0167-2789\(90\)90076-2](https://doi.org/10.1016/0167-2789(90)90076-2).
- [25] 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](https://doi.org/10.1007/978-3-642-25661-5_10).
- [26] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.
- [27] 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](https://doi.org/10.1007/bfb0055930).
- [28] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008, doi: [10.1109/tevc.2008.919004](https://doi.org/10.1109/tevc.2008.919004).
- [29] 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](https://doi.org/10.1016/j.knosys.2014.07.025).
- [30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4. Nov./Dec. 1995, pp. 1942–1948, doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [31] 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](https://doi.org/10.1109/mci.2006.329691).
- [32] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," *Eng. Optim.*, vol. 38, no. 2, pp. 129–154, Mar. 2006, doi: [10.1080/03052150500384759](https://doi.org/10.1080/03052150500384759).
- [33] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Jun. 2002, doi: [10.1109/MCS.2002.1004010](https://doi.org/10.1109/MCS.2002.1004010).
- [34] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007, doi: [10.1007/s10898-007-9149-x](https://doi.org/10.1007/s10898-007-9149-x).

- [35] 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](https://doi.org/10.1007/978-3-642-04944-6_14).
- [36] A. Askarzadeh, "A novel Metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016, doi: [10.1016/j.compstruc.2016.03.001](https://doi.org/10.1016/j.compstruc.2016.03.001).
- [37] 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](https://doi.org/10.1016/j.advengsoft.2016.01.008).
- [38] 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](https://doi.org/10.1016/j.advengsoft.2017.01.004).
- [39] 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](https://doi.org/10.1016/j.swevo.2018.02.013).
- [40] 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](https://doi.org/10.1126/science.220.4598.671).
- [41] 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](https://doi.org/10.1007/978-94-015-7744-1_2).
- [42] 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](https://doi.org/10.1016/j.ins.2009.03.004).
- [43] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, Aug. 1995, doi: [10.1103/physrevlett.75.1226](https://doi.org/10.1103/physrevlett.75.1226).
- [44] 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](https://doi.org/10.1177/003754970107600201).
- [45] 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](https://doi.org/10.1016/j.ins.2012.08.023).
- [46] 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](https://doi.org/10.1016/j.knosys.2015.12.022).
- [47] 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, Feb. 2016, doi: [10.1007/s00521-015-1870-7](https://doi.org/10.1007/s00521-015-1870-7).
- [48] 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](https://doi.org/10.1016/j.cie.2018.12.033).
- [49] 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](https://doi.org/10.1016/j.cad.2010.12.015).
- [50] 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](https://doi.org/10.1109/tevc.2003.814902).
- [51] A. Husseinzadeh 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](https://doi.org/10.1016/j.asoc.2013.12.005).
- [52] 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](https://doi.org/10.1007/978-3-642-13495-1_44).
- [53] 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](https://doi.org/10.1007/978-3-319-46173-1_15).
- [54] R. Moghdani and K. Salimifard, "Volleyball premier league algorithm," *Appl. Soft Comput.*, vol. 64, pp. 161–185, Mar. 2018, doi: [10.1016/j.asoc.2017.11.043](https://doi.org/10.1016/j.asoc.2017.11.043).
- [55] R. Leardi, "Application of genetic algorithm-PLS for feature selection in spectral data sets," *J. Chemometrics*, vol. 14, nos. 5–6, pp. 643–655, Sep. 2000, doi: [10.1002/1099-128x\(200009/12\)14:5/6<643::aid-cem621>3.0.co;2-e](https://doi.org/10.1002/1099-128x(200009/12)14:5/6<643::aid-cem621>3.0.co;2-e).
- [56] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, Feb. 2015, doi: [10.1109/lgrs.2014.2337320](https://doi.org/10.1109/lgrs.2014.2337320).
- [57] 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](https://doi.org/10.1016/j.patrec.2007.05.011).
- [58] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2052–2064, Mar. 2014, doi: [10.1016/j.eswa.2013.09.004](https://doi.org/10.1016/j.eswa.2013.09.004).
- [59] B. Chakraborty, "Feature subset selection by particle swarm optimization with fuzzy fitness function," in *Proc. 3rd Int. Conf. Intell. Syst. Knowl. Eng.*, Nov. 2008, pp. 1038–1042, doi: [10.1109/iske.2008.4731082](https://doi.org/10.1109/iske.2008.4731082).
- [60] S. Lee, S. Soak, S. Oh, W. Pedrycz, and M. Jeon, "Modified binary particle swarm optimization," *Prog. Natural Sci.*, vol. 18, no. 9, pp. 1161–1166, Sep. 2008, doi: [10.1016/j.pnsc.2008.03.018](https://doi.org/10.1016/j.pnsc.2008.03.018).
- [61] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, Mar. 2007, doi: [10.1016/j.patrec.2006.09.003](https://doi.org/10.1016/j.patrec.2006.09.003).
- [62] L. Ke, Z. Feng, and Z. Ren, "An efficient ant colony optimization approach to attribute reduction in rough set theory," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1351–1357, Jul. 2008, doi: [10.1016/j.patrec.2008.02.006](https://doi.org/10.1016/j.patrec.2008.02.006).
- [63] E.-G. Talbi, "Taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, 2002, doi: [10.1023/a:1016540724870](https://doi.org/10.1023/a:1016540724870).
- [64] 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](https://doi.org/10.1109/tpami.2004.105).
- [65] O. C. Martin and S. W. Otto, "Combining simulated annealing with local search heuristics," *Ann. Oper. Res.*, vol. 63, no. 1, pp. 57–75, Feb. 1996, doi: [10.1007/bf02601639](https://doi.org/10.1007/bf02601639).
- [66] 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](https://doi.org/10.1007/s11042-019-07811-x).
- [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](https://doi.org/10.1504/ijcat.2013.056915).
- [68] 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](https://doi.org/10.1016/j.eswa.2019.113122).
- [69] J. Jona and N. Nagaveni, "A hybrid swarm optimization approach for feature set reduction in digital mammograms," *WSEAS Trans. Inf. Sci. Appl.*, vol. 9, pp. 340–349, 2012.
- [70] 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](https://doi.org/10.1109/access.2019.2906757).
- [71] X. Lai and M. Zhang, "An efficient ensemble of GA and PSO for real function optimization," in *Proc. 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol.*, 2009, pp. 651–655, doi: [10.1109/iccsit.2009.5234780](https://doi.org/10.1109/iccsit.2009.5234780).
- [72] M. E. Basiri and S. Nemati, "A novel hybrid ACO-GA algorithm for text feature selection," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 2561–2568, doi: [10.1109/cec.2009.4983263](https://doi.org/10.1109/cec.2009.4983263).
- [73] E. Zorarpac and S. A. zel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016, doi: [10.1016/j.eswa.2016.06.004](https://doi.org/10.1016/j.eswa.2016.06.004).
- [74] M. A. Al-Betar, " β -Hill climbing: An exploratory local search," *Neural Comput. Appl.*, vol. 28, no. S1, pp. 153–168, Dec. 2017, doi: [10.1007/s00521-016-2328-2](https://doi.org/10.1007/s00521-016-2328-2).
- [75] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007, doi: [10.1016/j.amc.2006.11.033](https://doi.org/10.1016/j.amc.2006.11.033).
- [76] 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](https://doi.org/10.1016/j.swevo.2012.09.002).
- [77] 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](https://doi.org/10.2307/2685209).
- [78] 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](https://doi.org/10.1007/s12559-019-09668-6).
- [79] 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](https://doi.org/10.1016/j.neucom.2015.06.083).

- [80] G. van Rossum and F. L. Drake, *The Python Language Reference Manual*. Boston, MA, USA: Network Theory Ltd., 2011.
- [81] 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](https://doi.org/10.1109/MCSE.2007.55).
- [82] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018, doi: [10.1016/j.asoc.2017.11.006](https://doi.org/10.1016/j.asoc.2017.11.006).
- [83] F. Wilcoxon, *Individual Comparisons by Ranking Methods* (Springer Series in Statistics). New York, NY, USA: Springer, 1992, pp. 196–202, doi: [10.1007/978-1-4612-4380-9_16](https://doi.org/10.1007/978-1-4612-4380-9_16).



KUSHAL KANTI GHOSH is currently pursuing the undergraduate degree in computer science and engineering with Jadavpur University, Kolkata, India. His areas of interests include machine learning, optimization, game theory, and image processing.



SHAMEEM AHMED is currently pursuing the sophomore undergraduate degree in computer science and engineering with Jadavpur University, Kolkata, India. His areas of 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, the M.Tech. degree in computer science and engineering and the Ph.D. degree in engineering from Jadavpur University (J.U.), in 2013 and 2018, respectively. He also received RUSA 2.0 Fellowship for pursuing the Ph.D. degree with J.U., in 2019. He is currently working as an Assistant Professor with the Department of Information Technology, J.U. He has published more than 50 research articles 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 Association for Computing Machinery (ACM) as well as a Life Member of the Indian Society for Technical Education (ISTE, New Delhi) and Computer Society of India (CSI).



ZONG WOO GEEM (Member, IEEE) received the B.Eng. degree from Chung-Ang University, the Ph.D. degree from Korea University, and the M.Sc. degree from Johns Hopkins University. He was a Researcher with Virginia Tech, University of Maryland, 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 (an Associate Editor for the *Engineering Optimization*, a Guest Editor for the *Swarm and Evolutionary Computation*, the *International Journal of Bio-Inspired Computation*, the *Journal of Applied Mathematics, Applied Sciences, Complexity, and Sustainability*).



RAM SARKAR (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering from the University of Calcutta, in 2003, 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 the Department of Computer Science and Engineering with Jadavpur University as an Assistant Professor, in 2008, where he is currently working as an Associate Professor. He received the Fulbright-Nehru Fellowship (USIEF) for Postdoctoral Research with the University of Maryland, College Park, USA, from 2014 to 2015. His areas of current research interests are image processing, pattern recognition, machine learning, and bioinformatics.

• • •