

Received September 24, 2020, accepted October 5, 2020, date of publication October 16, 2020, date of current version November 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031718

# Mayfly in Harmony: A New Hybrid Meta-Heuristic Feature Selection Algorithm

TRINAV BHATTACHARYYA<sup>1</sup>, BITANU CHATTERJEE<sup>1</sup>,  
PAWAN KUMAR SINGH<sup>2</sup>, (Member, IEEE), JIN HEE YOON<sup>3</sup>, (Member, IEEE),  
ZONG WOO GEEM<sup>4</sup>, (Member, IEEE), AND RAM SARKAR<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

<sup>2</sup>Department of Information Technology, Jadavpur University, Kolkata 700106, India

<sup>3</sup>School of Mathematics and Statistics, Sejong University, Seoul 05006, South Korea

<sup>4</sup>Department of Energy IT, Gachon University, Seongnam 13120, South Korea

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

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant 2020R1A2C1A01011131.

**ABSTRACT** Feature selection is a process to reduce the dimension of a dataset by removing redundant features, and to use the optimal subset of features for machine learning or data mining algorithms. This helps to minimize the time requirement to train a learning algorithm as well as to lessen the storage requirement by ignoring the less-informative features. Feature selection can be considered as a combinatorial optimization problem. In this paper, the authors have presented a new feature selection algorithm called Mayfly-Harmony Search (MA-HS) based on two meta-heuristics namely Mayfly Algorithm and Harmony Search. Mayfly Algorithm has not hitherto been used for feature selection problems to the best of the author's knowledge. An S-shaped transfer function is incorporated for converting it into a binary version of Mayfly Algorithm. When different candidate solutions obtained from various regions of the search space using Mayfly Algorithm are taken into the harmony memory and processed by Harmony Search, a superior solution can be ensured. This is the primary reason for proposing a hybrid of Mayfly Algorithm and Harmony Search. Thus, combining harmony search with Mayfly Algorithm leads to an increased exploitation of the search space and an overall improvement in the performance of Mayfly-Harmony Search (MA-HS) algorithm. The proposed algorithm has been applied on 18 UCI datasets and compared with 12 other state-of-the-art meta-heuristic FS methods. Experiments have also been performed on three high-dimensional microarray datasets. The results obtained support the superior performance of the algorithm compared to the other methods. The source code of the proposed algorithm can be found using the link as follows: <https://github.com/trin07/MA-HS>.

**INDEX TERMS** MA-HS algorithm, mayfly optimization, harmony search, feature selection, meta-heuristic, hybrid method, UCI datasets.

## I. INTRODUCTION

In recent years, with the improvement in data collection methods in various fields, the amount of data available has increased dramatically. This has led to greater complexity in terms of both computational time and space required to execution of algorithms in the domains of machine learning and data mining. Much of this available data is redundant and decreases the efficiency of the algorithm. For classification

tasks, the irrelevant data causes a significant drop in accuracy and performance. This is why feature selection (FS) has gained importance in the scientific community in recent times.

### A. FEATURE SELECTION

FS aims to select only those features in the dataset which will be useful for classification and discard the irrelevant ones. FS has been used for various purposes such as neuro-imaging [1], gene prediction [2], text mining, image processing and fault diagnosis in industrial applications [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague<sup>id</sup>.

FS methods can be broadly classified into filters, wrappers and embedded. Filter techniques [4]–[6] use various scoring metrics to rank the features and choose the most important ones so no classifier is required to gauge the effectiveness of the filter methods. Wrapper techniques [7]–[9] use a classification algorithm to evaluate different feature subsets and the one which produces the best result is declared as the solution. This is why wrapper methods are generally considered to be superior as filter methods completely ignore the relation between the chosen subset and the inductive algorithm performance [10]. The optimal feature subset is influenced by the specific bias and heuristic of the inductive algorithm. Embedded techniques [11], [12] are similar to wrapper methods but instead use an intrinsic model building metric during learning.

FS is an NP-hard problem as there are  $2^n$  possible solutions for a dataset having  $n$  features in total. Evaluating each of these solutions is not a feasible option because of the high computational cost involved. A feature subset which is a possible solution to the problem at hand can be selected using random search. This can be done using a heuristic strategy which performs a guided search over the entire solution space to find a reasonably good feature subset which may not be the optimal solution but is acceptable within computational constraints. Higher-level heuristics or meta-heuristics have become quite popular in recent years to solve FS problems in different fields [13] such as handwriting recognition [14], [15], benchmark problems [16], [17], gene selection [18], medical diagnosis [19], financial problems, network intrusion and security.

## B. MOTIVATION

Numerous meta-heuristic algorithms have been proposed to solve the FS problem in various domains in the literature. Each of these algorithms has its own pros and cons in dealing with the particular problem at hand. The performance of an algorithm depends on its exploration and exploitation capabilities, as well as the inherent characteristics of the dataset used. Hybridized algorithms seek to combine two or more meta-heuristics so that the shortcomings of a particular method can be dealt with by another method. For instance, an algorithm with good exploration ability can be hybridized with one with good exploitation ability in order to set a good trade-off between exploration and exploitation capabilities of the overall system. This is an active area of research today. The No Free Lunch (NFL) theorem [20] states that the performance, averaged over all problems in the class, is the same for any solution method. This means that on a particular problem, different algorithms may obtain different results, but over all problems, they are equivalent. This has encouraged researchers to build on previous work and come up with new problem-specific approaches for improved performance. In this paper, the authors have proposed a hybrid algorithm based on Harmony Search (HS) algorithm [21] and Mayfly Algorithm (MA) [22] for solving the FS problem. HS algorithm has been applied for optimization in

various domains in the past. MA is a recently proposed meta-heuristic which has been shown to perform well in dealing with optimization problems. Despite being similar to PSO, MA is seen to have greater ability in finding out a more optimal solution than PSO and hence it has better chances of finding the globally optimal solution. However, in certain cases, premature convergence diminishes the quality of the final solution. As explained earlier, an algorithm with good exploration ability can be hybridized with another algorithm having high exploitation ability. MA is seen to effectively explore the search space while HS algorithm exploits and improves the existing feature subsets. This provides a compelling reason to hybridize MA with HS algorithm. If different solutions obtained from various regions of the search space using MA are put into the harmony memory, then a superior solution can be found by HS algorithm. Thus, combining HS algorithm with MA can lead to an increased exploitation of the search space and an overall improvement in the performance of the hybrid algorithm. This has been proven in this work which is evident from the results obtained.

## C. CONTRIBUTIONS

The main contributions of this work are as follows:

- Hybridization of MA with HS for the first time to the best of the authors' knowledge.
- Modification of MA to enhance its exploration ability.
- Use of HS algorithm to increase the exploitation ability of the hybrid MA-HS algorithm.
- Use of S-shaped transfer function to convert continuous search space into a binary search space.
- Validate the effectiveness of MA-HS by applying it on 18 standard UCI datasets and comparison with 12 state-of-the-art FS algorithms, and then providing a statistical analysis of the results obtained.
- Prove the robustness of the algorithm by applying it on 3 high dimensional microarray datasets.

## II. LITERATURE SURVEY

FS is essentially a combinatorial optimization problem. Often, in real-world optimization problems, it is not feasible to perform an exhaustive search due to the large amount of time required for processing. This is where heuristic methods are particularly useful. These methods aim to find good solutions, which may not be optimal, in reasonable computational time. Meta-heuristics are advanced heuristics which provide a set of guidelines or strategies to develop heuristic optimization algorithms [23]. The crux of a meta-heuristics algorithm is formed by the processes of intensification and diversification, which correspond to exploitation and exploration of the search space. Numerous meta-heuristic algorithms have been proposed in recent years to solve optimization problems in various domains.

Meta-heuristics can be nature inspired or non-nature inspired. Nature inspired meta-heuristic algorithms can be broadly classified into the four categories of evolutionary

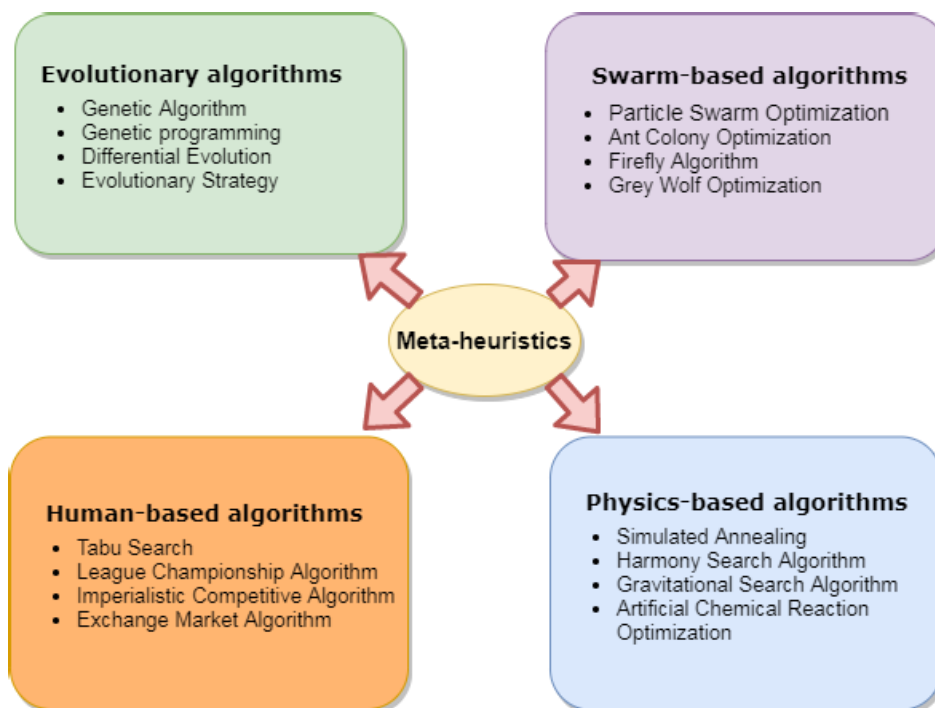


FIGURE 1. Different categories of meta-heuristic algorithms.

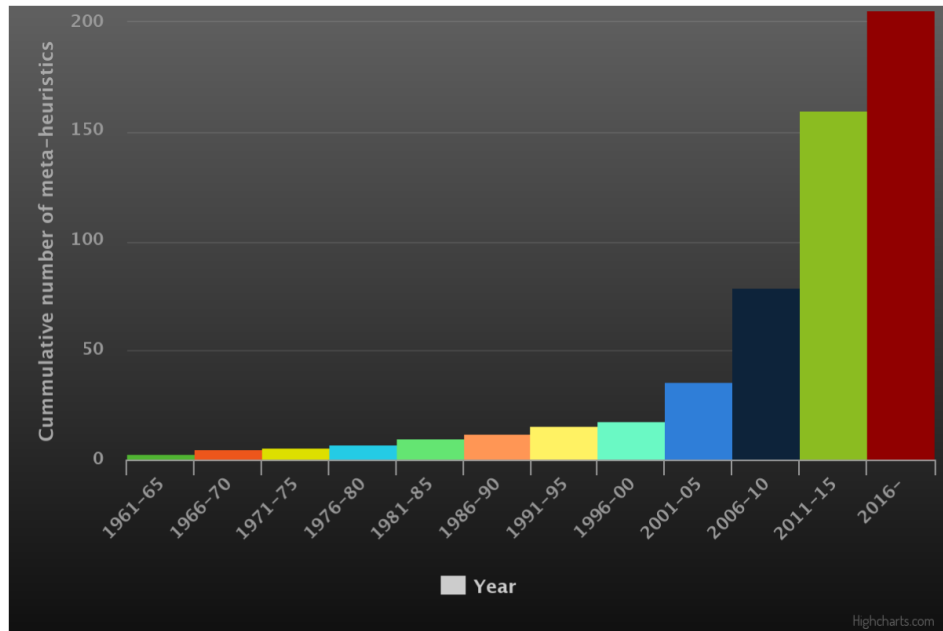
algorithms, physics-based algorithms, swarm-based algorithms and human-based algorithms, as shown in Figure 1.

- *Evolutionary algorithms*: These techniques are based on biological evolution involving genetic recombination, mutation and natural selection. Examples include Genetic Algorithm (GA) [24], Evolutionary Strategy (ES) [25] and Genetic Programming (GP) [26].
- *Swarm-based algorithms*: These algorithms are based on the behavior of a population of social organisms as a whole. Here, individuals seek to interact with one another and try to improve themselves based on the knowledge gained by the entire swarm. Examples include Particle Swarm Optimization (PSO) [27], Ant Colony Optimization (ACO) [28], Bat-inspired Algorithm (BA) [29] and Firefly Algorithm (FA) [30].
- *Physics-based algorithms*: These processes are based upon physical phenomenon like gravity, inertia and electromagnetic force. Some algorithms under this category are Simulated Annealing (SA) [31], Gravitational Search Algorithm (GSA) [32], Harmony Search (HS) [21], Ray Optimization (RO) [33] and Central Force Optimization (CFO) [34].
- *Human-based algorithms*: These are meta-heuristic techniques inspired by human behavior. Examples include Imperialist Competitive Algorithm (ICA) [35], League Championship Algorithm (LCA) [36] and Exchange Market Algorithm (EMA) [37].

Meta-heuristics can also be classified based on the number of solutions dealt with by the algorithm [38]. *Trajectory-based algorithms* involve a single solution and

work on improving it as it navigates through the search space. These are often hybridized with other diversification techniques to prevent the solution from being stuck in local optima. Examples include SA, TS, Greedy Randomized Adaptive Search Procedure (GRASP) [39] and Variable Neighborhood Search (VNS) [40]. *Population-based algorithms*, on the other hand, deal with a set of solutions simultaneously. These solutions (called agents) are allowed to improve using the information obtained from mutual interaction. New and better solutions are produced by merging different solutions. Examples include GA, PSO, ACO, Red Deer Algorithm (RDA) [41] and Shuffled Frog Leaping Algorithm (SFLA) [42].

The number of new meta-heuristic algorithms developed in recent years and used in the field of optimization has been visualized in Figure 2. Most of these methods have been used to solve the FS problem in the literature. Hybridized algorithms, where two or more independent methods are combined together, have grown in popularity over recent years. In combination, one method can effectively overcome the shortcomings of another, leading to improved results. GA was used for FS in automatic feature classifiers in [43], and hybridized in hybridized GA [19], [44], [45]. Binary PSO was used to solve the FS problem in [46]. It was further improved as Sentiment Fitness Sum Binary PSO (SCO-FS-BPSO) in [47]. A hybrid version of the Grasshopper Optimization Algorithm (GOA) was used in [48] for FS and parameter tuning of the SVM classifier. Mirjalili [49] developed Ant Lion Optimizer (ALO) for FS, based on the behavior of ant lions. A binary GSA, based on Newtonian laws of



**FIGURE 2.** A histogram showing the rise in the cumulative number of meta-heuristic algorithms developed in recent years.

gravity and motion, was proposed by Rashedi *et al.* [50]. The authors of [51] proposed a hybrid binary CRO with SA for FS in biomedical datasets. A hybrid Harris Hawks Optimizer (HHO) with SA was proposed by the authors of [52]. Zhang *et al.* [53] proposed a new approach for multi-objective FS called Binary differential evolution with self-learning (MOFS-BDE). An efficient method to prevent premature convergence called Return cost based binary firefly algorithm (Rc-BBFA) was proposed by the authors in [54]. Hegazy *et al.* proposed an improved version of the Salp Swarm Algorithm (SSA) [55], based on the swarming mechanism of salps, for FS. Other notable meta-heuristics include artificial bee colony [56], [57], modified PSO [58], [59].

The authors of [60] used PSO to select informative text features for improved text clustering. In [61] the authors proposed an enhanced Krill Herd Algorithm (KHA) for FS in the domain of text document clustering. A hybrid version of this method (MMKHA) was used in [62]. The methods proposed by the authors of [63]–[65] mimicked the inner workings of the brain for symbolic optimization and were used for object detection, categorization and visual tracking. The communication method used by honeybees inspired the work of [66] for 3D reconstruction which is robust to outliers. The authors of [67] used to swarm behaviour of the Honeybee Search Algorithm (HSA) for object tracking.

HS was proposed by Geem *et al.* [21]. Since its initial development in 2000, HS has been applied to a wide range of computational problems. Applications include water distribution networks [68]–[70]; structural design [71]–[73]; benchmark optimization [71], [74], [75]; soil stability

analysis [76], [77]; transport-related problems [78]; energy studies [79], [80], image enhancement [81], [82] and medical diagnosis [83], [84]. The MA [22] is inspired by the flight patterns and mating behavior of mayflies. It has been used for solving optimization problems which are encountered in the real world. effectively. In this paper, the authors have proposed a novel hybrid algorithm for FS named as MAHS algorithm, based on MA and HS algorithm.

### III. SOME PRELIMINARIES

#### A. MAYFLY OPTIMIZATION ALGORITHM

Mayflies are insects that belong to the order Ephemeroptera, part of a group of insects known as Palaeoptera. These insects appear mainly during the month of May in the UK, thus having the name Mayfly. Immature mayflies spend several years growing as aquatic nymphs until they are ready to go to the surface as adult mayflies. Most male adults assemble in swarms a few metres above the water to attract the females. They perform a nuptial dance which involves characteristic up and down movement generating a pattern. Female mayflies go to these swarms for mating. The mating process lasts only for a few seconds after which they drop the eggs in the water and the cycle continues. A detailed description of the above process is mentioned in the works of Allan and Flecker [85] Barbara and Peckarsky *et al.* [86].

MA has been developed by Zervoudakis and Tsafarakis [22] and is a new method for solving FS problems. It is a hybrid method combining the advantages of classical optimization methods such as PSO [87], GA [88] and FA. In the work of [89], it has been shown that PSO needed modifications as it is likely to get stuck in a local optimum, especially

for problems having a high dimension. The MA performs the necessary modifications, thereby enabling the algorithm to have a better performance across small and large scale feature sets. The components of MA are given as follows:

- **Movement of male mayflies:** The position of a male mayfly is updated as in Equation 1:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (1)$$

where  $x_i^t$  is the present position of the male mayfly and the new position  $x_i^{t+1}$  is obtained by adding the current position with the velocity  $v_i^{t+1}$ . The male mayflies are always a few meters above the surface of water and develop great speeds. A male mayfly's velocity is calculated as in Equation 2:

$$v_{kj}^{t+1} = g * v_{kj}^t + a_1 * e^{-\beta r_p^2} * (pbest_{kj} - x_{kj}^t) + a_2 * e^{-\beta r_g^2} * (gbest_j - x_{kj}^t) \quad (2)$$

where  $v_{kj}^t$  is the velocity of mayfly  $k$  in dimension  $j$  at time  $t$ ,  $x_{kj}^t$  is the position of the same mayfly at time  $t$ ,  $a_1$  and  $a_2$  are positive attraction constants which are used for measuring the contribution of the cognitive and social components respectively,  $g$  is a gravitational coefficient and  $\beta$  is a fixed visibility coefficient used to limit a mayfly's visibility to others.  $pbest_k$  is the most optimal position that the particular mayfly  $k$  ever visited and  $gbest_j$  is the  $j^{th}$  component of the position of the best male mayfly. Since this is a minimization problem so  $pbest_k$  is updated as follows:

$$pbest_k = \begin{cases} x_k^{t+1} \\ \text{if } fitness(x_k^{t+1}) < fitness(pbest_k) \end{cases} \quad (3)$$

where  $fitness(x_k^t)$  gives the fitness value of a position as given in subsection IV-B, i.e., the quality of a solution. Finally  $r_p$  is the Cartesian distance between  $x_k$  and  $pbest_k$ , while  $r_g$  is the Cartesian distance between  $x_k$  and  $gbest$ . These are calculated as shown:

$$|x_k - X_k| = \sqrt{\sum_{j=1}^n (x_{kj} - X_{kj})^2} \quad (4)$$

where  $x_{kj}$  represents the position of the  $j^{th}$  element of  $k^{th}$  mayfly and  $X_k$  either represents  $pbest_k$  or  $gbest$ . It is essential that the best mayflies at a particular time keep performing the nuptial dance which gives a stochastic element to the algorithm. Mathematical representation of this dance is given in Equation 5.

$$v_{kj}^{t+1} = g * v_{kj}^t + d * r \quad (5)$$

where  $d$  is the coefficient of nuptial dance and  $r$  is a random value  $\in [-1, 1]$ . The nuptial dance coefficient progressively reduces as  $d_{itr} = d_0 * \delta^{itr}$ . In this equation  $d_0$  is the initial value of the nuptial dance coefficient,  $itr$  is the current number of iterations and  $\delta$  is a random value  $\in [0, 1]$ .

- **Movement of female mayflies:** The female mayflies move towards the males for breeding. The position of a female mayfly is updated as follows:

$$y_i^{t+1} = y_i^t + v_i^{t+1} \quad (6)$$

where  $y_i^t$  is the current position of the female mayfly at time  $t$  and it is updated by adding its velocity  $v_i^{t+1}$ . The attraction process between males and females depends on the quality of the current solution, i.e., the best performing female is attracted to the best performing male and so on. The velocity of a female is updated as in Equation 7

$$v_{kj}^{t+1} = \begin{cases} \text{if } fitness(y_k) > fitness(x_k) \\ g * v_{kj}^t + a_2 * e^{-\beta r_{mf}^2} * (x_{kj}^t - y_{kj}^t) \\ \text{else if } fitness(y_k) \leq fitness(x_k) \\ g * v_{kj}^t + fl * r \end{cases} \quad (7)$$

where  $v_{kj}^t$  is the  $j^{th}$  component of the  $k^{th}$  female mayfly's velocity at time  $t$ ,  $y_{kj}^t$  is the position of female mayfly  $k$  in dimension  $j$  at time  $t$ ,  $x_{kj}^t$  is the  $j^{th}$  component of the position of male mayfly  $k$  at time  $t$ ,  $a_2$  and  $\beta$  are previously defined attraction constant and visibility coefficient respectively,  $g$  is the gravity coefficient defined before in Equation 2,  $r$  is a random value  $\in [-1, 1]$ , and  $r_{mf}$  is the Cartesian distance between male and female mayflies which is given in Equation 4.  $fl$  is a random walk coefficient in the instance of a female not being attracted with a male and  $fl_{itr} = fl_0 * \delta^{itr}$ . Here  $itr$  and  $\delta$  are two previously defined variables in Equation 5.

- **Crossover between mayflies:** The crossover operation is done by first selecting a male mayfly and then a female. This selection is done based on their fitness value, i.e., the best male breeds with the best female. Two offsprings are produced after a crossover as shown in Equation 9.

$$offspring1 = r_{of} * male + (1 - r_{of}) * female \quad (8)$$

$$offspring2 = r_{of} * female + (1 - r_{of}) * male \quad (9)$$

Here  $male$  is the parent male mayfly,  $female$  is the female parent and  $r_{of}$  is a stipulated value between 0 and 1. The initial velocities of the offspring are set as 0.

- **Mutation of mayflies:** The newly generated offspring are mutated to enhance the exploration ability of the algorithm. A normally distributed random number is added to the offspring's variable as described in

$$offspring'_n = offspring_n + \kappa \quad (10)$$

where  $\kappa$  is the normally distributed random value.

## B. HARMONY SEARCH

Harmony in nature is a special relationship between several sound waves that have different frequencies. Musical performances tend to have a soothing effect on humans. The amount of soothing effect is determined by its aesthetic estimation.

**Algorithm 1** Pseudo Code for Harmony SearchInput: MaxHS, NumVariables,  $r_{accept}$ ,  $r_{pa}$ ,  $b_{range}$ Output: Best Solution  $X = (x_1, x_2, \dots, x_d)$ 

```

Initialize HM
for itr ← 1 to MaxHS do
  for i ← 1 to NumVariables do
    if rand1 < raccept then
      Choose a value from HM for the ith variable
      if rand2 < rpa then
        Adjust the value by an amount of brange *
        rand3
      end if
    else
      Choose a random value
    end if
  end for
  Accept the new solution if better
end for
Return the current best solution

```

A musical performer strives to achieve the best aesthetic state which is analogous to a global optimum. Thus finding the best aesthetic state is an optimization problem determined by an objective function. Aesthetic estimation is calculated by the set of sounds played by simultaneously joined instruments. The aesthetic value of the sound generated can be improved by practice.

A novel method called HS was introduced by Geem et al. [21] to generate a set of harmonies which are improvised similar to an evolutionary algorithm. The algorithm initialises a Harmony Memory (HM) and subsequently modifies it to create new harmonies. The new harmonies replace the previous ones if they are found to be better with respect to an objective function. In this way HS performs improvisation upon existing musical harmonies and achieves an optimal combination of harmonies. The HS algorithm is a reliable tried-and-tested method and has been previously used for optimization in the work of [70], [72]. The pseudo code of HS is given in Algorithm 1.

**IV. PROPOSED WORK**

FS is a binary optimization problem having restricted solutions which are represented by either 0 or 1. Each agent or solution is represented by a binary vector where a value 1 shows that the chosen feature is selected and value of 0 represents that the feature is not selected. The size of this solution vector corresponds to the number of features in a particular dataset. In FS, the selected feature subset needs to be evaluated during every iteration of the algorithm. This is therefore a difficult task, especially the wrapper-based methods.

**A. PROPOSED MA-HS ALGORITHM**

The MA described in subsection III-A was originally developed for continuous optimization problems. The re-designed

**Algorithm 2** Pseudo Code for the Proposed MA-HS Algorithm

Input: PopSize, MaxIter

Output: Best Agent  $X = (x_1, x_2, \dots, x_d)$ 

```

Initiate population and velocity of male and female
mayflies randomly
Evaluate population and then find gbest
for itr ← 1 to MaxIter do
  for i ← 1 to PopSize do
    Update pbest
    Evaluate and update the velocities of male and
    female mayflies
  end for
  Sort the mayflies and rank them
  Perform crossover and generate male and female off-
  spring
  Mutate the offspring
  Replace worst mayflies with the best new offspring
  generated
  Perform HS on male mayflies
  Update gbest
end for

```

algorithm for FS problems, called **MA-HS** is described in Algorithm 2. Each solution vector in MA is converted into its binary form, i.e., consisting only of 0's and 1's and then evaluated. To achieve this conversion, the S-shaped transfer function is used. This function gives the probability of whether to choose a particular feature in a solution vector. It is a reliable function and has previously been used by many researchers [17], [90]. The S-shaped transfer function used in this algorithm is depicted in Figure 4 and shown in Equation 11.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (11)$$

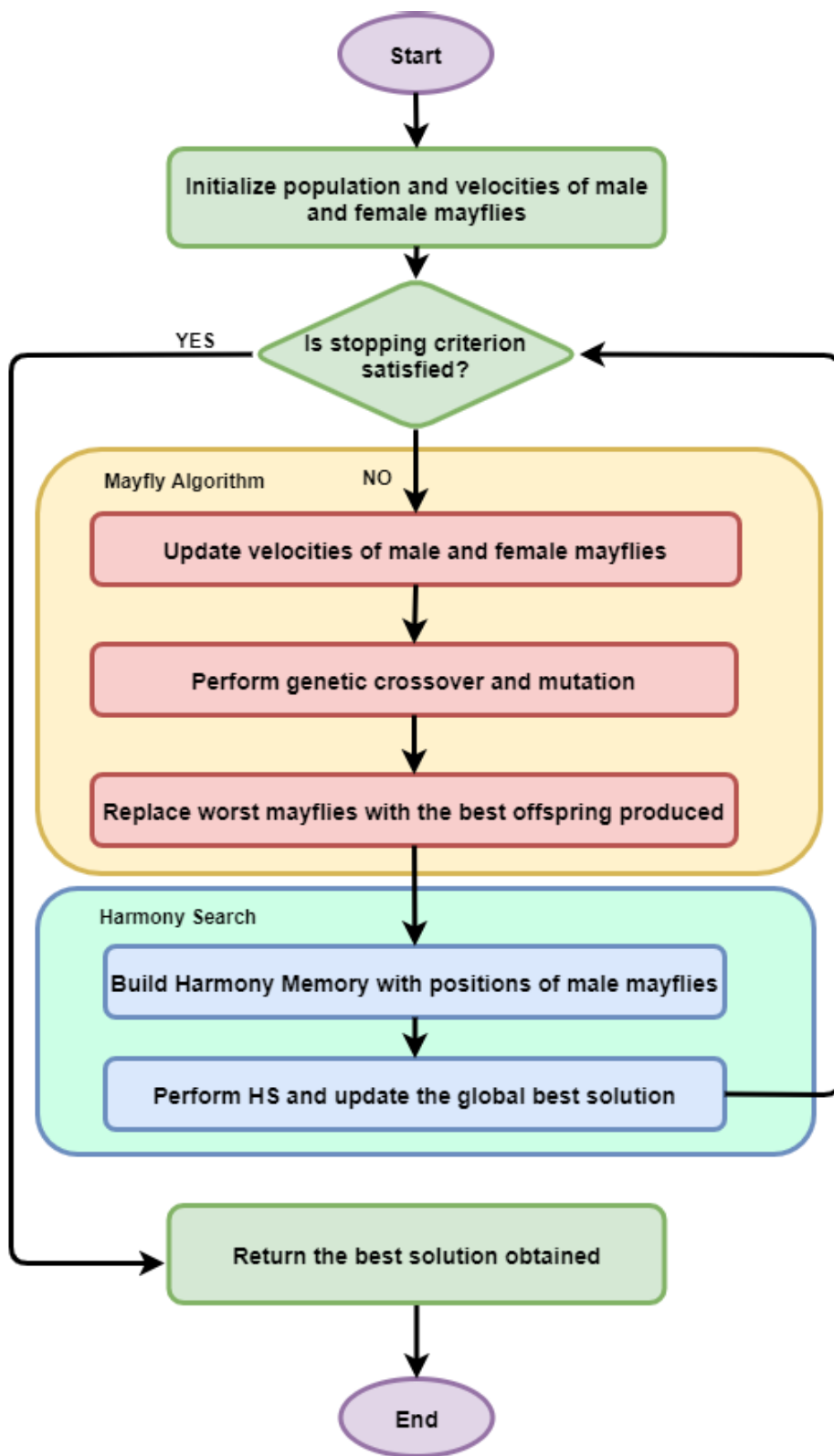
During the conversion process, the agent's feature is updated according to Equation 12.

$$P_d^{t+1} = \begin{cases} 1 & \text{if } S(P_d^{t+1}) > rand \\ 0 & \text{if } S(P_d^{t+1}) \leq rand \end{cases} \quad (12)$$

where  $P_d^{t+1}$  is the updated feature subset of the agent,  $rand$  is a random number between 0 and 1, and  $S(P_d^{t+1})$  is the S-shaped transfer function as previously defined in Equation 11. A broad outline of the proposed framework is given in Figure 3.

**B. FITNESS FUNCTION**

The algorithm evaluates the quality of a solution in this section. A learning algorithm has been used since it is a wrapper-based method. The K-Nearest Neighbor (KNN) classifier [91] is used for finding out the classification accuracy. The fitness function consists of the classification error and the number of features. In FS, the authors intend to



**FIGURE 3.** An outline of the proposed MA-HS algorithm used for solving FS problems.

increase the accuracy and simultaneously reduce the number of features. For this purpose, classification error is used instead of accuracy. This is because both the error and

number of features needs to be decreased. Combining these two, the fitness function will therefore be reduced to a single objective function. Equation 13 gives the function for

TABLE 1. Basic information related to the UCI datasets used in the present work.

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

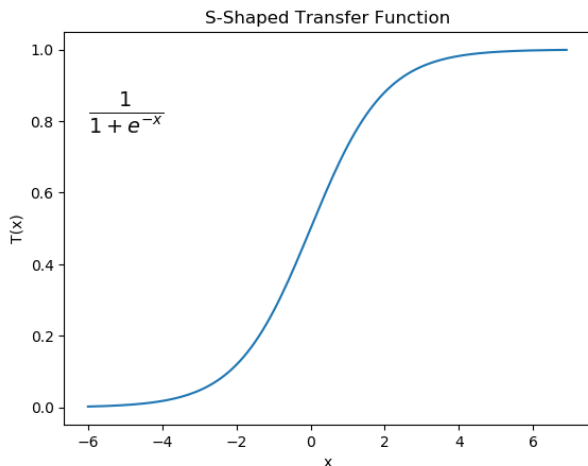


FIGURE 4. S-shaped transfer function for converting continuous Mayfly search space into binary.

evaluating a feature subset.

$$\downarrow Fitness = \gamma \times \lambda + (1 - \gamma) \times \frac{|f|}{|F|} \quad (13)$$

where  $|f|$  is the number of features in the feature subset,  $|F|$  is the number of features in the given dataset,  $\lambda$  is the classification error and  $\gamma \in [0, 1]$  is a parameter that gives a relative contribution between the classification error and the number of features.

### V. EXPERIMENTAL RESULTS

This section deals with the results which support the effectiveness of the MA-HS algorithm for solving the FS problem. KNN classifier [91] has been used for finding out the classification accuracy. The authors have set  $K = 5$  as per recommendations reported by Mafarja et al. [92]. In each dataset, 80% of the instances are used for training and rest 20% are used for testing the classification model. The method

proposed here is implemented using Python3 [93] and graphs are obtained using Matplotlib [94]. The experiments are conducted on a PC with 3.30 GHz Intel(R) Pentium(R) CPU G4400 and 8 GB Memory. The time complexity of the algorithm is found out to be  $O(MaxIter * PopSize * MaxHS * (time_{fitness} + d))$  where  $MaxIter$  is the maximum number of iterations,  $PopSize$  is the population size,  $MaxHS$  is the maximum number of iterations of HS,  $time_{fitness}$  is the complexity of calculating the fitness of an agent and  $d$  is the dimension of the dataset.

#### A. DATASET DESCRIPTION

For this work, the authors have chosen 18 different datasets from the UCI data repository [95]. They are utilized for evaluating the efficacy of the proposed method. The 18 datasets consist of 15 bi-class and 3 multi-class datasets as described in Table 1.

#### B. TUNING OF PARAMETERS

The aim of the proposed method is to maximize the classification accuracy and try to reduce the number of features at the same time. In other words both the classification error as well as the number of features must be reduced. That is why the fitness function includes components of classification error and the number of features. A single parameter, in this case  $\gamma$  can be used for determining the relative weightage given to error and the number of features. Hence the objective is to minimize the fitness function which in turn will minimize the classification error and the number of features. In the proposed work, greater importance has been given to increase the classification accuracy. So the value of  $\gamma$  is set as 0.9 in Equation 13. The gravity coefficient  $g$  is set as 0.98 to imitate the effect of gravity on mayflies. Furthermore, the values of  $d, fl, a_1$  and  $a_2$  have been set to a comparatively high values to allow the mayflies to move freely. This has been done to enhance the exploration ability of the algorithm. The final list of parameter values is given in Table 2.



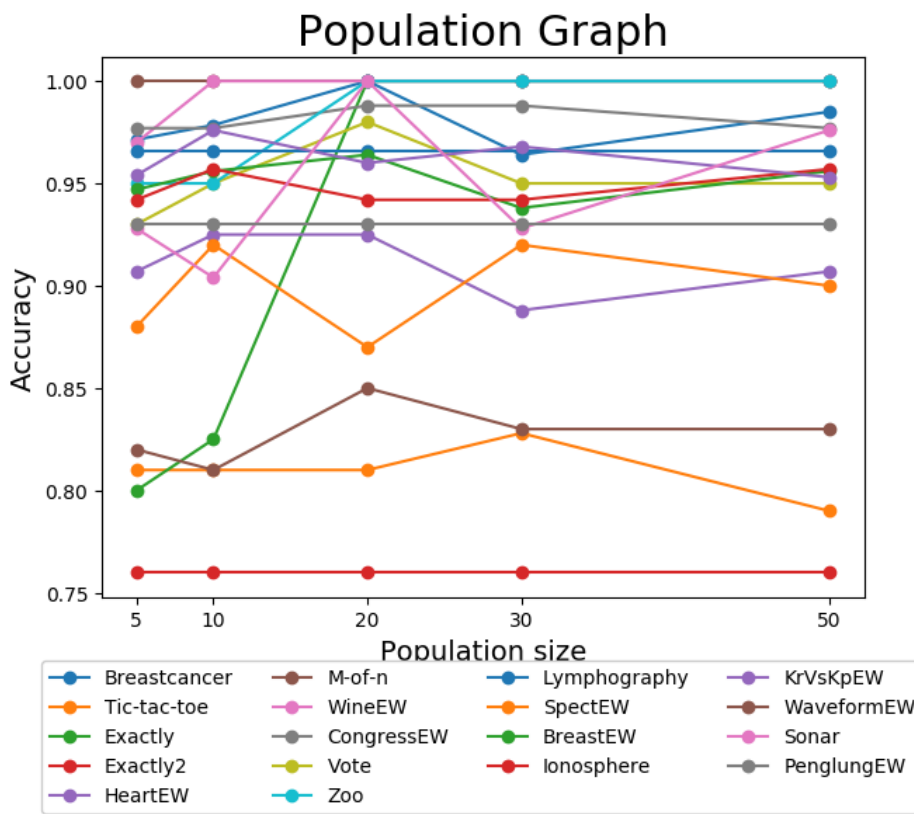


FIGURE 5. Graph showing the variation of population sizes on classification accuracy obtained by the proposed MA-HS algorithm for 18 UCI datasets.

TABLE 2. Parameters along with its corresponding values used in the proposed MA-HS algorithm.

Parameter	Meaning	Value
PopSize	Population size	20
MaxIter	Maximum number of iterations	20
$\gamma$	Relative weightage used for fitness value	0.9
$a_1$	Positive attraction constant	3
$a_2$	Positive attraction constant	3.5
$\beta$	Visibility coefficient	0.1
$d_0$	Initial nuptial dance coefficient	3
$fl_0$	Initial random walk coefficient	3
$g$	Gravitational coefficient	0.98
$r_{of}$	Random value for crossover	0.95
MaxHS	Maximum iterations in HS	30
$r_{accept}$	Rate of acceptance	0.8
$r_{pa}$	Rate of pitch adjustment	0.3
$b_{range}$	Range of pitch adjustment	2

Figure 5 provides a graph which demonstrates the effect of variation in *PopSize* on classification accuracy. Five different population sizes of [5, 10, 20, 30, 50] have been used with the number of iterations being 20 for each population size. With the exception of Ionosphere, KrVsKpEW, SpectEW and Tic-tac-toe, all the other remaining 14 datasets achieve their maximum value at 20 using MA-HS algorithm. Taking this into account, the authors have set the optimal population size as 20 for the MA-HS algorithm.

### C. ANALYSIS OF RESULTS

Convergence of the best fitness value with progressive increase in iterations is graphically shown in Figure 7. In several cases, the best fitness obtained by MA and HS is nowhere close to that of MA-HS. Moreover MA-HS achieves its best fitness value at near 10 iterations which is about 50% of the total number of iterations. MA and HS take longer time to achieve their optimal fitness values. Thus MA-HS is better than both MA and HS as it finds out the optimal fitness value and converges quicker. A similar conclusion can also be drawn from Figure 6 where it is evident that the accuracy converges quickly for 80% of the datasets. Here too the final accuracy is achieved by MA-HS within 10 iterations or 50% of the total. So MA and HS help each other thereby increasing the effectiveness of MA-HS over individual MA and HS methods.

A comparison has been given in Table 3 and Table 4 to show the effectiveness of the proposed MA-HS algorithm. Table 3 deals with the classification accuracy and Table 4 with the number of features selected. Both the tables show that the MA-HS method performs far better than the original MA as well as HS. MA-HS achieves the first rank in both the cases. In Table 3, the accuracy of MA-HS is found to be better than MA and HS in all cases giving MA-HS a substantial edge in terms of accuracy. Let us now look at the results of Table 4.

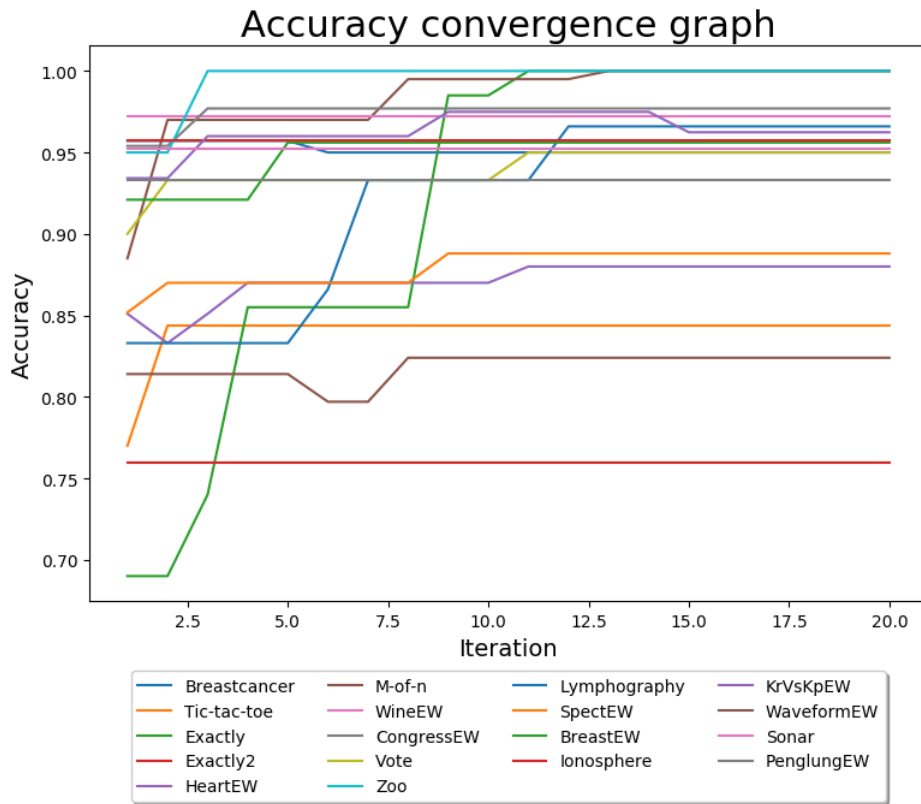


FIGURE 6. Graph showing the variation of classification accuracy obtained using the MA-HS algorithm (considering MaxIter=20) for 18 UCI datasets.

TABLE 3. Comparison of MA, HS and MA-HS algorithms with respect to classification accuracy.

Dataset	MA	HS	MA-HS
Breastcancer	0.9785	0.942	0.9929
Tic-tac-toe	0.82	0.776	0.8385
Exactly	0.99	0.685	1
Exactly2	0.76	0.76	0.76
HeartEW	0.87	0.851	0.926
M-of-n	1	0.75	1
WineEW	0.97	0.916	1
CongressEW	0.977	0.954	1
Vote	0.983	0.933	1
Zoo	1	1	1
Lymphography	0.86	0.833	0.9667
SpectEW	0.87	0.796	0.907
BreastEW	0.94	0.956	0.9824
Ionosphere	0.91	0.9	0.98
KrVsKpEW	0.971	0.938	0.974
WaveformEW	0.813	0.788	0.835
Sonar	0.928	0.857	1
PenglungEW	1	1	1
Average rank	1.77	2.72	1
Final rank	2	3	1

TABLE 4. Comparison of MA, HS and MA-HS algorithms with respect to the number of selected features.

Dataset	MA	HS	MA-HS
Breastcancer	2	3	3
Tic-tac-toe	5	4	4
Exactly	7	3	6
Exactly2	1	3	1
HeartEW	4	5	4
M-of-n	6	3	6
WineEW	4	3	2
CongressEW	3	6	3
Vote	5	5	3
Zoo	10	10	5
Lymphography	8	7	6
SpectEW	9	5	5
BreastEW	6	10	5
Ionosphere	12	12	3
KrVsKpEW	21	17	13
WaveformEW	17	16	13
Sonar	28	20	16
PenglungEW	150	165	63
Average rank	2.16	2	1.22
Final rank	3	2	1

MA-HS gives least number of features for 15 out of 18 or 83% of the datasets. Overall, the average rank of MA-HS is by far better than both MA and HS. Comparing MA with HS, it is seen that MA outperforms HS in terms of accuracy but

the opposite is true after analyzing the number of features selected. It can be safely inferred that in MA-HS, both MA and HS mutually benefit from each other and ultimately it improves the classification accuracy as well as the number of

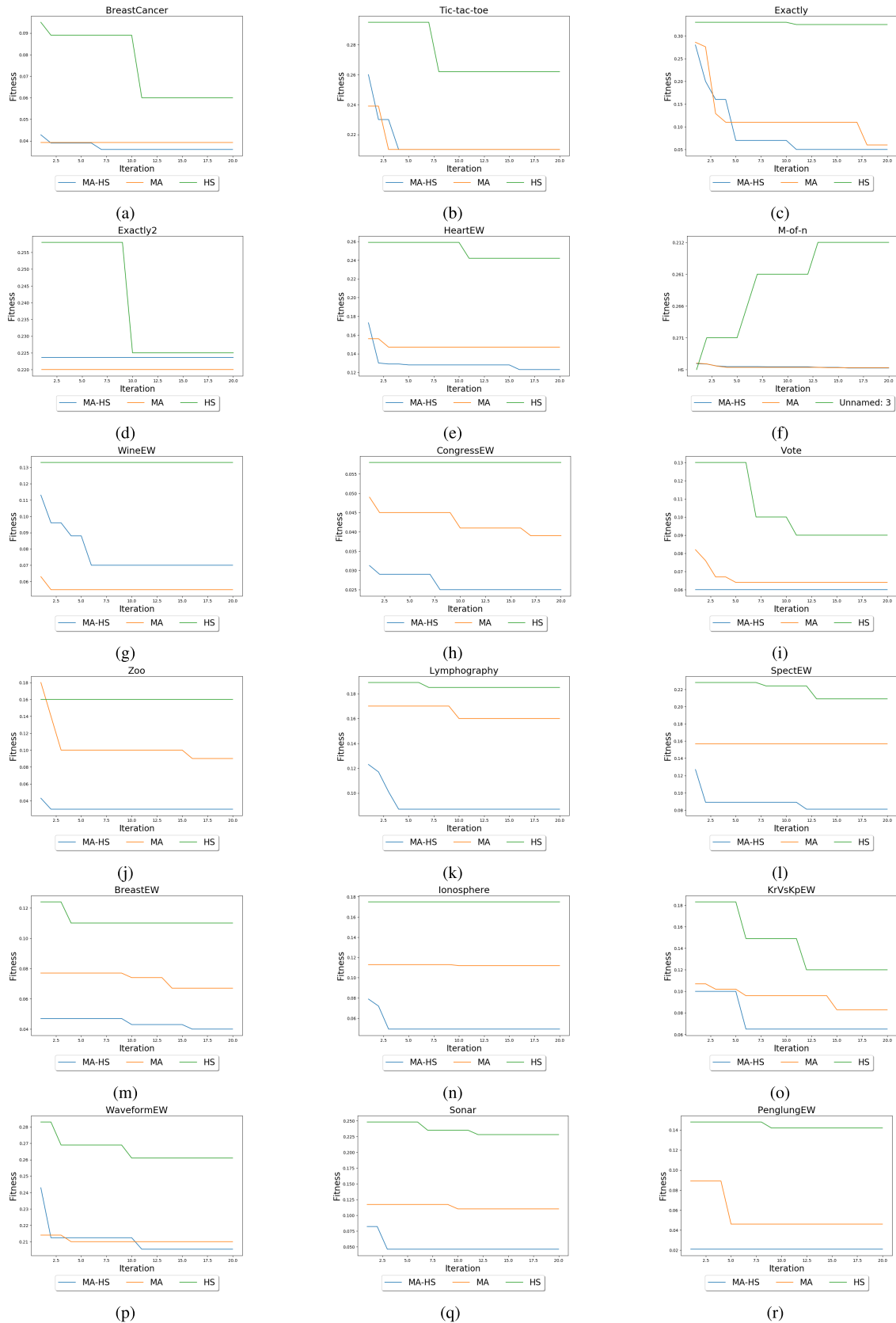


FIGURE 7. Convergence graphs obtained using MA, HS and MA-HS algorithms for 18 UCI datasets.

**TABLE 5.** Parameter settings of 12 state-of-the-art meta-heuristic FS methods used for comparison.

Algorithm	PopSize	MaxIter
GA	8	70
BPSO	8	70
ALO	8	200
BGSA	8	20
BDA	8	20
BSSA	10	100
WOA	10	100
BGWOPSO	10	100
ECWSA	10	100
BGOA	8	20
WOASAT	10	100
bALO-QR	10	100

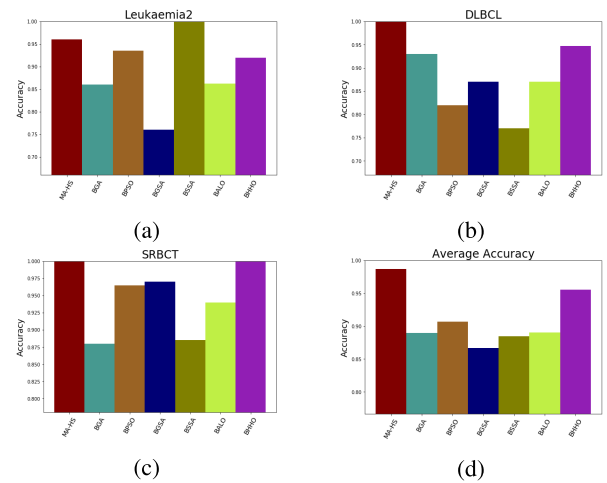
features. The sole MA method is strengthened by incorporating HS and vice versa, and the combined MA-HS algorithm is an effective technique for solving FS problems.

**VI. COMPARISON AND DISCUSSION**

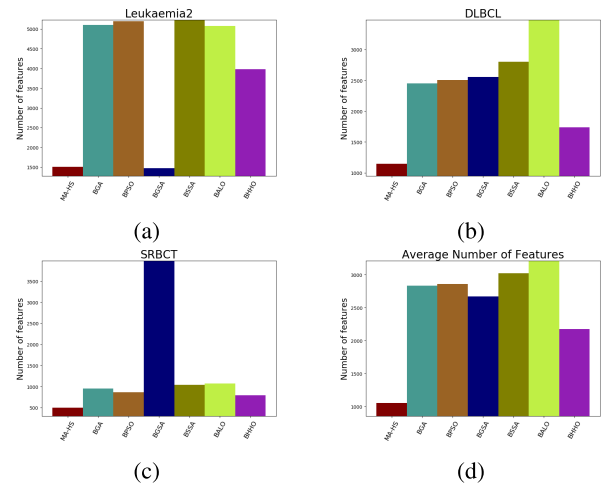
The proposed method is compared with 12 state-of-the-art meta-heuristic FS methods for demonstrating its performance. The other methods described below are tried-and-tested algorithms and have previously been used for the 18 benchmark UCI datasets. The other given meta-heuristics are GA, BPSO [96], ALO, BGSA, binary Dragonfly Algorithm (BDA) [92], binary Salp Swarm Algorithm (BSSA) [97], Whale Optimization Algorithm (WOA) [98], BGWOPSO [99], Embedded Chaotic Whale Survival Algorithm (ECWSA) [100], binary Grasshopper Optimization Algorithm (BGOA) [101], WOASAT [102] and bALO-QR [101]. Among these, BGWOPSO is a hybrid of Grey Wolf Optimization (GWO) [9] and PSO, WOASAT stands for WOA with SA, and bALO-QR is binary ALO hybridized with QuickReduct (QR) [103]. The parameter settings of the other algorithms used for comparison is given in Table 5. Let us first look at the results of Table 6. It can be seen that the MA-HS algorithm performs way better than its peers. The average rank of the proposed algorithm is better than the second algorithm i.e., BDA by a considerable margin. Its average rank is 1.6 while BDA has an average rank of 2.4. In 13 out of 18 or 72% of the datasets, the proposed MA-HS algorithm gives the best accuracy. For the remaining 5, it achieves the second rank for 3 of them. Only Exactly2 and KrVsKpEW datasets have a worse rank.

Now let us analyze the results of Table 7. For 11 out of 18 datasets or 61% of the datasets, the proposed MA-HS method gives the best result, i.e., least number of features. It obtains the second rank for 6 of the remaining 7 datasets and gets third rank for the other KrVsKpEW dataset. Therefore MA-HS comes within second rank for 17 out of 18 or 95% of the datasets. As a whole, MA-HS algorithm outperforms all the compared algorithms by a wide margin and is assigned the first rank.

From both Table 6 and Table 7, it can be seen that the proposed MA-HS algorithm is successfully able to increase



**FIGURE 8.** Comparison of the classification accuracies obtained for microarray datasets by the proposed method and the mentioned state-of-the-art methods.



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

the classification accuracy as well as reduce the number of features. The versatility of the method is evident as it outperforms many state-of-the-art meta-heuristic FS algorithms for all kinds of datasets, be it small or large. The MA-HS algorithm is able to achieve this performance because of the combination of its underlying characteristics through hybridization. The exploratory ability of MA coupled with exploitative ability of HS algorithm gives it an edge over the others. In fact, the average rank obtained by the MA-HS algorithm in Table 7 is substantially less making it a highly suitable algorithm for FS.

**A. STATISTICAL SIGNIFICANCE TEST**

In order to find out the statistical significance of the results, Friedman test [104], [105] has been performed. It is a non-parametric statistical test and the procedure involves ranking each row, and then considering the values of ranks by columns. The final result gives a probability or *p*-value. The

**TABLE 6. Comparison of the proposed MA-HS based FS algorithm with some state-of-the-art FS methods in terms of classification accuracy.**

Dataset	MA-HS	GA	BPSO	ALO	BGSA	BDA	BSSA	WOA	BGWOPSO	ECWSA	BGOA	WOASAT	bALO-QR
Breastcancer	<b>0.9929</b>	0.974	0.963	0.974	0.9686	0.9928	0.9768	0.9571	0.98	0.9518	0.9737	0.97	0.974
Tic-tac-toe	0.8385	0.7996	0.7996	0.783	0.7766	<b>0.8469</b>	0.8205	0.7511	0.81	0.7878	0.8038	0.79	0.8
Exactly	<b>1</b>	<b>1</b>	<b>1</b>	0.965	0.994	<b>1</b>	0.9803	0.75	<b>1</b>	0.7811	0.9999	<b>1</b>	0.912
Exactly2	0.76	0.77	0.768	0.762	0.77	0.7725	0.7582	0.69	0.76	0.7912	0.7565	0.75	0.76
HeartEW	<b>0.926</b>	0.8741	0.837	0.838	0.8296	0.8759	0.8605	0.76	0.85	0.8556	0.8635	0.85	0.884
M-of-n	<b>1</b>	<b>1</b>	<b>1</b>	0.967	0.994	<b>1</b>	0.9918	0.85	<b>1</b>	0.9213	<b>1</b>	<b>1</b>	<b>1</b>
WineEW	<b>1</b>	0.9888	0.9775	0.972	0.9775	<b>1</b>	0.9933	0.928	<b>1</b>	0.9802	0.9985	0.99	<b>1</b>
CongressEW	<b>1</b>	0.9679	0.9633	0.981	0.9633	0.9866	0.9628	0.929	0.98	0.9619	0.9772	0.98	0.972
Vote	<b>1</b>	0.9733	0.96	0.972	0.96	0.9894	0.9511	0.9387	0.97	0.95	0.9484	0.97	0.948
Zoo	<b>1</b>	0.902	0.9608	0.98	0.9804	<b>1</b>	<b>1</b>	0.9647	<b>1</b>	0.98	0.9778	0.97	0.961
Lymphography	0.9667	0.8378	0.8919	0.917	0.8649	<b>0.9922</b>	0.89	0.785	0.92	0.8739	0.8586	0.89	0.886
SpectEW	<b>0.907</b>	0.8955	0.8881	0.899	0.8433	0.8519	0.8361	0.787	0.88	0.7988	0.8565	0.88	0.9
BreastEW	<b>0.9824</b>	0.9754	0.9719	0.974	0.9544	0.9792	0.9484	0.9553	0.97	0.9733	0.9606	0.98	0.962
Ionosphere	0.9857	0.9489	0.9489	0.904	0.9432	<b>0.9911</b>	0.9182	0.89	0.95	0.8672	0.922	0.96	0.869
KrVsKpEW	0.974	<b>0.985</b>	0.9731	0.973	0.9549	0.9794	0.9644	0.9151	0.97	0.9392	0.9708	0.98	0.975
WaveformEW	<b>0.835</b>	0.7836	0.756	0.797	0.7344	0.758	0.7335	0.712	0.8	0.7985	0.7562	0.76	0.894
SonarEW	<b>1</b>	0.9904	0.9423	0.845	0.9135	0.9841	0.9372	0.8543	0.96	0.7638	0.9497	0.97	0.84
PenglungEW	<b>1</b>	0.9189	0.9189	0.827	0.8333	<b>1</b>	0.8775	0.729	0.96	0.8766	0.7712	0.94	0.665
Average Rank	<b>1.6</b>	4.5	6.1	6	7.5	2.4	6.6	10.3	4	7.7	6.3	4.7	6.2
Assigned Rank	<b>1</b>	4	6	5	10	2	9	12	3	11	8	5	7

**TABLE 7. Comparison of the proposed MA-HS based FS algorithm with some state-of-the-art FS methods in terms of number of features selected.**

Dataset	MA-HS	GA	BPSO	ALO	BGSA	BDA	BSSA	WOA	BGWOPSO	ECWSA	BGOA	WOASAT	bALO-QR
Breastcancer	<b>3</b>	4	4	4.7	4	5	3.8	5.4	4.4	4.5	4.2	4.2	4.05
Tic-tac-toe	<b>4</b>	5	6	5	<b>4</b>	7	6	10.8	5.2	7.74	6	5.2	6.48
Exactly	6	6	6	<b>5.75</b>	4	6	7.2	6	6	7.1	6	6	5.85
Exactly2	<b>1</b>	<b>1</b>	<b>1</b>	1.5	<b>1</b>	7.1	2.7	5.7	1.6	9	5.4	2.8	5.85
HeartEW	4	5	<b>3</b>	8.6	<b>3</b>	5.7	5.8	8.6	5.8	9.4	7.4	5.4	5.8
M-of-n	6	6	6	6	<b>5</b>	6	7.1	9.7	6	<b>5</b>	6	6	5.85
WineEW	<b>2</b>	4	5	5.4	4	3.6	6.3	8.8	6	6	6.8	6.4	5.46
CongressEW	3	<b>2</b>	3	6.6	4	5.5	5.7	10.3	4.4	5.6	4.1	6.4	4.32
Vote	<b>3</b>	5	<b>3</b>	6.6	4	3.4	4.8	7.6	3.4	6	4.9	5.2	5.6
Zoo	5	<b>4</b>	5	5.7	6	4.4	6.7	9.9	6.8	8	5.4	5.6	5.92
Lymphography	6	<b>5</b>	<b>5</b>	7.3	6	8.2	10.3	10.5	9.2	7.7	7.4	7.2	9
SpectEW	<b>5</b>	<b>5</b>	6	7.6	<b>5</b>	6.8	10.9	11.2	8.4	7.8	8.7	9.4	7.7
BreastEW	<b>5</b>	8	9	13.8	10	11.5	16.7	21	13.6	15	13.2	11.6	12.6
IonosphereEW	<b>3</b>	7	7	11.7	9	11.5	15.8	21.4	13	9.52	9.1	12.8	13.6
KrVsKpEW	13	<b>11</b>	12	16.1	14	20.7	20.4	28	15.8	13	16.9	18.4	14.4
WaveformEW	<b>13</b>	15	15	20.5	14	23	22.9	33.2	14.2	14	21.5	20.6	20.8
SonarEW	<b>16</b>	19	22	26.6	24	25.6	33.4	43.4	31.2	20	30.5	26.4	24
PenglungEW	<b>63</b>	84	130	133.1	140	121.2	171.6	144.3	130.8	65	95.7	127.4	130
Average Rank	<b>1.6</b>	2.4	2.8	6.2	3.1	5.7	8	10.2	6.2	6.3	6	6.1	6.05
Assigned Rank	<b>1</b>	2	3	9	4	5	11	12	9	10	6	8	7

**TABLE 8. p-values obtained using Friedman statistical test considering classification accuracy of the proposed MA-HS algorithm over all other FS methods for 18 UCI datasets.**

MA-HS	MA	HS	GA	BPSO	ALO	BGSA	BDA	BSSA	WOA	BGWOPSO	ECWSA	BGOA	WOASAT	bALO-QR
MA-HS	0.001	0	0.004	0.001	0	0	0.04	0	0	0.002	0	0	0.001	0

null hypothesis states that the two sets of results have the same distribution. If the two results are statistically significant, then the *p*-value will be less than 0.05 which is a universal threshold. If this comes out to be true then the null hypothesis will be rejected. From the results given in Table 8, it can be concluded that the results are statistically significant as the *p*-values are less than 0.05 for each and every case.

**VII. EXPERIMENTS ON HIGH DIMENSIONAL DATA**

The authors have also performed experiments on high dimensional microarray datasets to demonstrate the effectiveness of the proposed method. Microarray data represent the states of a cell at the molecular level and is used in the field of medical research for cancer classification. These datasets typically

have a huge number of features (greater than 1000). The proposed method has been used on three publicly available microarray datasets, the details of which have been given in Table 9. The obtained results have been compared with state-of-the-art methods: GA, PSO, GSA, SSA, ALO and HHO.

Table 10 shows the classification accuracy obtained by the proposed method and the state-of-the-art algorithms on the three microarray datasets and the average accuracy obtained. Table 11 demonstrates the comparison of the number of selected features for each of the three datasets and the average number of selected features. It is quite evident that on average the proposed method outperforms the other algorithms in terms of classification accuracy, while selecting fewer features, as shown in Figure 8 and Figure 9. For the

**TABLE 9.** Description of the microarray datasets used in the present work.

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

**TABLE 10.** Comparison of the proposed MA-HS based FS algorithm with some state-of-the-art FS methods in terms of classification accuracy on microarray datasets.

Dataset	MA-HS	BGA	BPSO	BGSA	BSSA	BALO	BHHO
Leukaemia2	0.9613	0.9352	0.8605	0.7611	<b>1</b>	0.8623	0.9223
DLBCL	<b>1</b>	0.93	0.82	0.87	0.77	0.87	0.947
SRBCT	<b>1</b>	0.88	0.9652	0.97	0.885	0.94	<b>1</b>
Average Rank	<b>1.33</b>	4.67	4	4.33	4.33	4.67	2.33
Assigned Rank	<b>1</b>	3	5	4	4	3	2

**TABLE 11.** Comparison of the proposed MA-HS based FS algorithm with some state-of-the-art FS methods in terms of number of selected features on microarray datasets.

Dataset	MA-HS	BGA	BPSO	BGSA	BSSA	BALO	BHHO
Leukaemia2	1508	5111	5198	<b>1472</b>	5228	5073	3979
DLBCL	<b>1147</b>	2452	2505	2558	2802	3476	1741
SRBCT	<b>497</b>	949	865	3872	1042	1071	798
Average Rank	<b>1.33</b>	4	4.33	4.33	6	6.67	2.33
Assigned Rank	<b>1</b>	3	4	4	5	6	2

Leukaemia2 dataset, binary SSA achieves a slightly higher accuracy than MA-HS, but selects a much greater number of features. Binary GSA selects a slightly less number of features but with far lower accuracy. In all other cases, the MA-HS algorithm proves to be clearly superior to the rest of the methods. These experiments on high dimensional data further establish the robustness of the proposed method.

### VIII. CONCLUSION AND FUTURE DIRECTION

In this work, a novel meta-heuristic feature selection (FS) algorithm called Mayfly-Harmony Search (MA-HS) has been developed which is based on Mayfly Algorithm (MA) [22] and Harmony Search (HS) [21]. To the best of the author's knowledge, the proposed hybridization is a new one, and has never been used for solving FS problems in the past. The authors have taken the help of S-shaped transfer function (Equation 12) to change the continuous search space to a binary one consisting of only 0's and 1's. MA has been chosen due to its high exploration ability and enhanced exploitation ability is achieved by including HS as a reliable hybrid method. FS is a binary optimization problem and the fitness function is designed as such to maximize the classification accuracy and minimize the number of features selected. Furthermore, MA-HS has been applied on 18 UCI datasets and 3 high-dimensional microarray datasets, and compared with many well-known meta-heuristics. It proves that MA-HS is able to obtain a desired high classification accuracy and low number of features in comparison to the other methods. MA-HS converges quickly to an optimal value as shown in Figure 7 and Figure 6. It may be noted that the algorithm can perform even better by optimizing the set of values of

the parameters. The parameter values have been tuned by experimenting with a range of values and these have also been discussed in subsection V-B. The set of parameters can be further optimized but will require more experimentation. According to the No Free Lunch Theorem [20], the algorithm may not give the best results for all types of problems. This is evident as the MA-HS algorithm does not give the best results for all the datasets. The algorithm is sometimes prone to premature convergence for datasets like Exactly2. As a future scope of this work, more advanced classifiers may be used for determining the fitness value. The mutation process may be strengthened which will increase the exploration ability of the algorithm. Other transfer functions such as V-shaped or U-shaped can be used here. The meta-heuristic can be combined with other state-of-the-art methods. The proposed work can be used in various fields such as text mining, bio-informatics such as genomics and neuro-imaging, image processing and for industrial purposes such as fault diagnosis.

### REFERENCES

- [1] K. Kampa, S. Mehta, C. A. Chou, W. A. Chaovaitongse, and T. J. Grabowski, "Sparse optimization in feature selection: Application in neuroimaging," *J. Global Optim.*, vol. 59, nos. 2–3, pp. 439–457, Jul. 2014.
- [2] M. L. Bermingham, R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan, H. Campbell, A. F. Wright, J. F. Wilson, F. Agakov, P. Navarro, and C. S. Haley, "Application of high-dimensional feature selection: Evaluation for genomic prediction in man," *Sci. Rep.*, vol. 5, no. 1, p. 10312, Sep. 2015.
- [3] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2015, pp. 1200–1205.
- [4] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Machine Learning: ECML-94*. Berlin, Germany: Springer, 1994, pp. 171–182.

- [5] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 249–256.
- [6] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
- [7] R. Leardi, "Genetic algorithms in feature selection," in *Genetic Algorithms in Molecular Modeling*. Amsterdam, The Netherlands: Elsevier, 1996, pp. 67–86.
- [8] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Cybern. Simulation*, Oct. 1997, pp. 4104–4108.
- [9] 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.
- [10] I. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza, "Filter versus wrapper gene selection approaches in DNA microarray domains," *Artif. Intell. Med.*, vol. 31, no. 2, pp. 91–103, Jun. 2004.
- [11] S. Maldonado and J. López, "Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification," *Appl. Soft Comput.*, vol. 67, pp. 94–105, Jun. 2018.
- [12] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 882–895, Oct. 2020.
- [13] Q. Al-Tashii, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," *IEEE Access*, vol. 8, pp. 125076–125096, 2020.
- [14] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "A genetic algorithm based approach for selection of local features in handwritten digit recognition application," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1592–1606, May 2012.
- [15] M. Ghosh, R. Guha, R. Mondal, P. K. Singh, R. Sarkar, and M. Nasipuri, "Feature selection using histogram-based multi-objective Ga for handwritten Devanagari numeral recognition," in *Intelligent Engineering Informatics*. Singapore: Springer, 2018, pp. 471–479.
- [16] R. Guha, M. Ghosh, S. Kapri, S. Shaw, S. Mutsuddi, V. Bhateja, and R. Sarkar, "Deluge based genetic algorithm for feature selection," *Evol. Intell.*, pp. 1–11, Mar. 2019, doi: [10.1007/s12065-019-00218-5](https://doi.org/10.1007/s12065-019-00218-5).
- [17] 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.
- [18] M. Ghosh, S. Begum, R. Sarkar, D. Chakraborty, and U. Maulik, "Recursive memetic algorithm for gene selection in microarray data," *Expert Syst. Appl.*, vol. 116, pp. 172–185, Feb. 2019.
- [19] 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.
- [20] 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.
- [21] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *J. Simul.*, vol. 76, no. 2, pp. 60–68, Feb. 2001.
- [22] K. Zervoudakis and S. Tsafarakis, "A mayfly optimization algorithm," *Comput. Ind. Eng.*, vol. 145, Jul. 2020, Art. no. 106559, doi: [10.1016/j.cie.2020.106559](https://doi.org/10.1016/j.cie.2020.106559).
- [23] K. Sörensen and F. Glover, "Metaheuristics," in *Encyclopedia of Operations Research and Management Science*, vol. 62. New York, NY, USA: Springer, 2013, pp. 960–970.
- [24] E. Bonabeau, M. Dorigo, D. D. Marco, G. Theraulaz, and G. Théraulaz, *Swarm Intelligence: From Natural To Artificial Systems*, no. 1. Oxford, U.K.: Oxford Univ. Press, 1999.
- [25] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002, doi: [10.1023/a:1015059928466](https://doi.org/10.1023/a:1015059928466).
- [26] J. R. Koza and J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, vol. 1. Cambridge, MA, USA: MIT Press, 1992.
- [27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4. Nov./Dec. 1995, pp. 1942–1948.
- [28] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
- [29] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*. Berlin, Germany: Springer, 2010, pp. 65–74.
- [30] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.
- [31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [32] E. Rashedi, H. Nezamabadi-Pour, and S. Saryzadi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
- [33] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vols. 112–113, pp. 283–294, Dec. 2012.
- [34] R. A. Formato, "Central force optimization," *Prog. Electromagn. Res.*, vol. 77, pp. 425–491, Aug. 2007.
- [35] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 4661–4667.
- [36] A. H. Kashan, "League championship algorithm: A new algorithm for numerical function optimization," in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, 2009, pp. 43–48.
- [37] N. Ghorbani and E. Babaei, "Exchange market algorithm," *Appl. Soft Comput.*, vol. 19, pp. 177–187, Jun. 2014.
- [38] M. Baghel, S. Agrawal, and S. Silakari, "Survey of Metaheuristic algorithms for combinatorial optimization," *Int. J. Comput. Appl.*, vol. 58, no. 19, pp. 21–31, Nov. 2012.
- [39] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *J. Global Optim.*, vol. 6, no. 2, pp. 109–133, Mar. 1995.
- [40] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.
- [41] A. F. Fard and M. Hajiaghvaei-Keshтели, "Red deer algorithm (RDA); a new optimization algorithm inspired by red deer's mating," in *Proc. Int. Conf. Ind. Eng.*, 2016, pp. 33–34.
- [42] 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.
- [43] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," in *Handbook of Pattern Recognition and Computer Vision*. Singapore: World Scientific, Aug. 1993, pp. 88–107.
- [44] 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.
- [45] S. Jiang, K.-S. Chin, L. Wang, G. Qu, and K. L. Tsui, "Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department," *Expert Syst. Appl.*, vol. 82, pp. 216–230, Oct. 2017.
- [46] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
- [47] L. Shang, Z. Zhou, and X. Liu, "Particle swarm optimization-based feature selection in sentiment classification," *Soft Comput.*, vol. 20, no. 10, pp. 3821–3834, Mar. 2016.
- [48] I. Aljarah, A. M. Al-Zoubi, H. Faris, M. A. Hassonah, S. Mirjalili, and H. Saadeh, "Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm," *Cognit. Comput.*, vol. 10, no. 3, pp. 478–495, Jan. 2018.
- [49] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
- [50] E. Rashedi, H. Nezamabadi-pour, and S. Saryzadi, "BGSA: Binary gravitational search algorithm," *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010.
- [51] C. Yan, J. Ma, H. Luo, and A. Patel, "Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets," *Chemometric Intell. Lab. Syst.*, vol. 184, pp. 102–111, Jan. 2019.
- [52] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A hybrid harris hawks optimization algorithm with simulated annealing for feature selection," *Artif. Intell. Rev.*, pp. 1–45, Jun. 2020, doi: [10.1007/s10462-020-09860-3](https://doi.org/10.1007/s10462-020-09860-3).
- [53] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
- [54] Y. Zhang, X.-F. Song, and D.-W. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Inf. Sci.*, vols. 418–419, pp. 561–574, Dec. 2017.

- [55] A. E. Hegazy, M. A. Makhlof, and G. S. El-Tawel, "Improved salp swarm algorithm for feature selection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 3, pp. 335–344, Mar. 2020, doi: [10.1016/j.jksuci.2018.06.003](https://doi.org/10.1016/j.jksuci.2018.06.003).
- [56] X.-H. Wang, Y. Zhang, X.-Y. Sun, Y.-L. Wang, and C.-H. Du, "Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106041.
- [57] Y. Zhang, S. Cheng, Y. Shi, D.-W. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Syst. Appl.*, vol. 137, pp. 46–58, Dec. 2019.
- [58] Y. Zhang, H.-G. Li, Q. Wang, and C. Peng, "A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection," *Int. J. Speech Technol.*, vol. 49, no. 8, pp. 2889–2898, Aug. 2019.
- [59] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017.
- [60] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *J. Comput. Sci.*, vol. 25, pp. 456–466, Mar. 2018, doi: [10.1016/j.jocs.2017.07.018](https://doi.org/10.1016/j.jocs.2017.07.018).
- [61] L. M. Q. Abualigah, *Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering*. Berlin, Germany: Springer, 2019.
- [62] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm," *Appl. Intell.*, vol. 48, no. 11, pp. 4047–4071, 2018.
- [63] D. E. Hernández, E. Clemente, G. Olague, and J. L. Briseño, "Evolutionary multi-objective visual cortex for object classification in natural images," *J. Comput. Sci.*, vol. 17, pp. 216–233, Nov. 2016, doi: [10.1016/j.jocs.2015.10.011](https://doi.org/10.1016/j.jocs.2015.10.011).
- [64] G. Olague, E. Clemente, D. E. Hernandez, A. Barrera, M. Chan-Ley, and S. Bakshi, "Artificial visual cortex and random search for object categorization," *IEEE Access*, vol. 7, pp. 54054–54072, 2019.
- [65] G. Olague, D. E. Hernandez, E. Clemente, and M. Chan-Ley, "Evolving head tracking routines with brain programming," *IEEE Access*, vol. 6, pp. 26254–26270, 2018.
- [66] G. Olague and C. Puente, "Honeybees as an intelligent based approach for 3D reconstruction," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, Aug. 2006, pp. 1116–1119.
- [67] O. E. Perez-Cham, C. Puente, C. Soubervielle-Montalvo, G. Olague, C. A. Aguirre-Salado, and A. S. Nuñez-Varela, "Parallelization of the honeybee search algorithm for object tracking," *Appl. Sci.*, vol. 10, no. 6, p. 2122, Mar. 2020, doi: [10.3390/app10062122](https://doi.org/10.3390/app10062122).
- [68] S. Kim and H. Lee, "The leak detection method using GA and HS algorithm," in *Proc. World Water Environ. Resour. Congr.*, Jun. 2003, pp. 1–10.
- [69] Z. W. Geem, "Harmony search in water pump switching problem," in *Proc. Int. Conf. Natural Comput.* Berlin, Germany: Springer, 2005, pp. 751–760.
- [70] Z. W. Geem, "Optimal scheduling of multiple dam system using harmony search algorithm," in *Proc. Int. Work-Confer. Artif. Neural Netw.* Berlin, Germany: Springer, 2007, pp. 316–323.
- [71] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Comput. Methods Appl. Mech. Eng.*, vol. 194, nos. 36–38, pp. 3902–3933, Sep. 2005.
- [72] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, nos. 9–10, pp. 781–798, Apr. 2004.
- [73] F. Erdal and M. Saka, "Effect of beam spacing in the harmony search based optimum design of grillages," *Asian J. Civil Eng.*, vol. 9, no. 3, pp. 215–228, Jan. 2008.
- [74] A. Mukhopadhyay, A. Roy, S. Das, S. Das, and A. Abraham, "Population-variance and explorative power of harmony search: An analysis," in *Proc. 3rd Int. Conf. Digit. Inf. Manage.*, Nov. 2008, pp. 775–781.
- [75] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Appl. Math. Comput.*, vol. 198, no. 2, pp. 643–656, May 2008.
- [76] L. Li, Y. Wang, Q. Wang, and P. Sun, "New procedure for simulating arbitrary slip surface f soil slope in stability analysis," *J. Hydraulic Eng.*, vol. 16, no. 4, pp. 535–541, 2008.
- [77] L. Li, S.-C. Chi, and X.-S. Chu, "Location of non-circular slip surface using the modified harmony search method based on correcting strategy," *Yantu Lixue (Rock Soil Mech.)*, vol. 27, no. 10, pp. 1714–1718, 2006.
- [78] Z. W. Geem, K. S. Lee, and Y. Park, "Application of harmony search to vehicle routing," *Amer. J. Appl. Sci.*, vol. 2, no. 12, pp. 1552–1557, Dec. 2005.
- [79] M. Fesanghary, E. Damangir, and I. Soleimani, "Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm," *Appl. Thermal Eng.*, vol. 29, nos. 5–6, pp. 1026–1031, Apr. 2009.
- [80] Z. W. Geem and H. Hwangbo, "Application of harmony search to multi-objective optimization for satellite heat pipe design," in *Proc. of*, 2006, pp. 1–3.
- [81] M. A. Al-Betar, Z. A. A. Alyasseri, A. T. Khader, A. L. Bolaji, and M. A. Awadallah, "Gray image enhancement using harmony search," *Int. J. Comput. Intell. Syst.*, vol. 9, no. 5, pp. 932–944, Sep. 2016.
- [82] Y. A. Z. A. Alkareem, I. Venkat, M. A. Al-Betar, and A. T. Khader, "Edge preserving image enhancement via harmony search algorithm," in *Proc. 4th Conf. Data Mining Optim. (DMO)*, Sep. 2012, pp. 47–52.
- [83] A. Panchal and B. Tom, "Harmony search optimization for HDR prostate brachytherapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 75, no. 3, pp. S720–S721, Nov. 2009.
- [84] H. Dong, Y. Bo, M. Gao, and T. Zhu, "Improved harmony search for detection with photon density wave," *Proc. SPIE*, vol. 6625, Feb. 2008, Art. no. 662523.
- [85] J. Allan and A. Flecker, "The mating biology of a mass-swarming mayfly," *Animal Behav.*, vol. 37, pp. 361–371, Mar. 1989, doi: [10.1016/0003-3472\(89\)90084-5](https://doi.org/10.1016/0003-3472(89)90084-5).
- [86] B. L. Peckarsky, A. R. McIntosh, B. W. Taylor, and J. Dahl, "Predator chemicals induce changes in mayfly life history traits: A whole-stream manipulation," *Ecology*, vol. 83, no. 3, pp. 612–618, Mar. 2002, doi: [10.1890/0012-9658\(2002\)083\[0612:pcimj\]2.0.co;2](https://doi.org/10.1890/0012-9658(2002)083[0612:pcimj]2.0.co;2).
- [87] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43, doi: [10.1109/mhs.1995.494215](https://doi.org/10.1109/mhs.1995.494215).
- [88] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, pp. 95–99, 1988.
- [89] M. Li, H. Chen, X. Shi, S. Liu, M. Zhang, and S. Lu, "A multi-information fusion 'triple variables with iteration' inertia weight PSO algorithm and its application," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105677.
- [90] W.-Q. Zhang, Y. Zhang, and C. Peng, "Brain storm optimization for feature selection using new individual clustering and updating mechanism," *Appl. Intell.*, vol. 49, no. 12, pp. 4294–4302, 2019.
- [91] N. S. Altman, "An introduction to kernel and nearest-neighbor non-parametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.
- [92] 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.
- [93] G. van Rossum and F. L. Drake, *The Python Language Reference Manual*. Surrey, U.K.: Network Theory Ltd., 2011.
- [94] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.
- [95] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [96] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–38, Feb. 2008.
- [97] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.
- [98] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [99] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496–39508, 2019.
- [100] R. Guha, M. Ghosh, S. Mutsuddi, R. Sarker, and S. Mirjalili, "Embedded chaotic whale survival algorithm for filter-wrapper feature selection," 2020, *arXiv:2005.04593*. [Online]. Available: <http://arxiv.org/abs/2005.04593>
- [101] 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.



- [102] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.
- [103] J. R. Anaraki and M. Eftekhari, "Improving fuzzy-rough quick reduct for feature selection," in *Proc. 19th Iranian Conf. Electr. Eng.*, May 2011, pp. 1–6.
- [104] P. K. Singh, R. Sarkar, and M. Nasipuri, "Statistical validation of multiple classifiers over multiple datasets in the field of pattern recognition," *Int. J. Appl. Pattern Recognit.*, vol. 2, no. 1, pp. 1–23, 2015.
- [105] P. Singh, R. Sarkar, and M. Nasipuri, "Significance of non-parametric statistical tests for comparison of classifiers over multiple datasets," *Int. J. Comput. Sci. Math.*, vol. 7, no. 5, pp. 410–442, 2016.



**JIN HEE YOON** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mathematics from Yonsei University, South Korea. She is currently a Faculty with the School of Mathematics and Statistics, Sejong University, Seoul, South Korea. Her research interests include fuzzy regression analysis, fuzzy time series, optimizations, intelligent systems, and deep learning. She is a Board Member of the Korean Institute of Intelligent Systems (KIIS). She has been regularly working as an organizer and a committee member of several international conferences. She has been working as an Associate Editor, a Guest Editor, and an Editorial Board Member of several journals, including SCI and SCIE journals.



**TRINAV BHATTACHARYYA** is currently pursuing the bachelor's degree in computer science and engineering with Jadavpur University, Kolkata, India. His research interests include machine learning, optimization, and graph theory.



**BITANU CHATTERJEE** is currently pursuing the bachelor's degree in computer science and engineering with Jadavpur University, Kolkata, India. His research interests include machine learning, optimization, and graph theory.



**PAWAN KUMAR SINGH** (Member, IEEE) received the B.Tech. degree in information technology from the West Bengal University of Technology, in 2010, and the M.Tech. degree in computer science and engineering and the Ph.D. (engineering) degree from Jadavpur University (JU), in 2013 and 2018, respectively. He also received RUSA 2.0 Fellowship for pursuing his postdoctoral research with JU, in 2019. He is currently working as an Assistant Professor with

the Department of Information Technology, JU. He has published more than 50 research papers in peer-reviewed journals and international conferences. His current research interests include computer vision, pattern recognition, handwritten document analysis, image and video processing, feature optimization, machine learning, and artificial intelligence. He is a member of the Institution of Engineers (India) and the Association for Computing Machinery (ACM) as well as a Life Member of the Indian Society for Technical Education (ISTE, New Delhi) and the Computer Society of India (CSI).



**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, 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 interests include phenomenon-mimicking algorithms and their applications to energy, environment, and water fields. He has served for various journals as an Editor. He served as an Associate Editor for *Engineering Optimization*, a Guest Editor for *Swarm & 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. (Engineering) degree from Jadavpur University, in 2005 and 2012, respectively. He joined the Department of Computer Science and Engineering, Jadavpur University, as an Assistant Professor, in 2008, where he is currently working as an Associate Professor. He received

Fulbright-Nehru Fellowship (USIEF) for postdoctoral research with the University of Maryland, College Park, MA, USA, during 2014–15. His current research interests include image processing, pattern recognition, machine learning, and bioinformatics.

...