

# Hybrid of Harmony Search Algorithm and Ring Theory based Evolutionary Algorithm for Feature Selection

**SHAMEEM AHMED**  <sup>1</sup>, **KUSHAL KANTI GHOSH**  <sup>1</sup>, **PAWAN KUMAR SINGH**  <sup>2</sup>  
**(Member, IEEE)**, **ZONG WOO GEEM**  <sup>3</sup>, **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>Department of Energy IT, Gachon University, Seongnam 13120, South Korea

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

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2020R1A2C1A01011131). This research was also supported by the Energy Cloud R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (2019M3F2A1073164).

**ABSTRACT** Feature Selection (FS) is an important pre-processing step in the fields of machine learning and data mining, which has a major impact on the performance of the corresponding learning models. The main goal of FS is to remove the irrelevant and redundant features, resulting in optimized time and space requirements along with enhanced performance of the learning model under consideration. Many meta-heuristic optimization techniques have been applied to solve FS problems because of its superiority over the traditional optimization approaches. Here, we have introduced a new hybrid meta-heuristic FS model based on a well-known meta-heuristic Harmony Search (HS) algorithm and a recently proposed Ring Theory based Evolutionary Algorithm (RTEA), which we have named as Ring Theory based Harmony Search (RTHS). Effectiveness of RTHS has been evaluated by applying it on 18 standard UCI datasets and comparing it with 10 state-of-the-art meta-heuristic FS methods. Obtained results prove the superiority of RTHS over the state-of-the-art methods considered here for comparison.

**INDEX TERMS** Ring Theory based Harmony Search, Feature Selection, Harmony Search, Ring Theory based Evolutionary Algorithm, Meta-heuristic, Hybrid Optimization, UCI datasets

## I. INTRODUCTION

In this era of computer and technology, with the advancements in the fields of image processing, pattern recognition, financial analysis, business management, medical studies [1, 2] and many more, we have to deal with huge amount of data, whose dimensions are increasing everyday. This has a great impact on the performances of different algorithms used in the field of machine learning and data mining in terms of time and space requirements. There may be numerous features in a dataset, but not all of which are useful or important for a particular task. Feature selection (FS), a data pre-processing step, can be used to remove the irrelevant and redundant features [3], resulting in optimized time and space requirements. Basically, these redundant features act as noise, and removal of those results in better performing ability of the corresponding machine learning or data mining algorithm [4]. There are two different categories of FS

techniques based on evaluation criteria of features [3]: Filter and Wrapper. A filter method evaluates features based on pre-defined mathematical or statistical criteria, (e.g., Relief [5], Information Gain [6], Laplacian Score [7], Chi-Square [8], Fisher Score [9], etc.) and selects most important features according to that. Whereas, a wrapper method uses a learning algorithm to evaluate feature subsets and selects the optimum subset for the corresponding task [10]. Filter methods are relatively faster than wrapper methods as the former do not use learning algorithm, whereas the later, in general, achieves higher accuracy [11].

In recent times, the meta-heuristic methods have become popular in solving various optimization problems due its advantages over traditional optimization methods, such as avoidance of local optima, non-derivative mechanism and flexibility [12]. Two important aspects of a meta-heuristic algorithm are [13]: exploration and exploitation. Exploration

means the ability to search the solution space for new potential solution in each iteration avoiding local optima, and exploitation means finding a better solution in the neighborhood of the solution obtained so far. A good meta-heuristic algorithm has a characteristic of maintaining the balance between both exploration and exploitation phases.

In this work we have introduced a new hybrid meta-heuristic based on a well-known meta heuristic Harmony Search (HS) algorithm [14] and a recently proposed Ring Theory based Evolutionary Algorithm (RTEA) [15]. HS is inspired from an artificial phenomena, musical harmony. Just like musical performances seek a best state (fantastic harmony) which is determined by aesthetic estimation, HS seeks a best state (global optimum) determined by fitness or objective function. RTEA draws inspiration from algebraic theory on evolution process. Generally, there are two models 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 high level version, the base meta-heuristics are executed in sequence. We have hybridized HS and RTEA in a high level fashion, which follows 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 HS is hybridized with RTEA for solving FS problems. In a nutshell, the main contributions of this work are as follows:

- RTHS: A new FS method named Ring Theory based Harmony Search is introduced using a popular meta-heuristic HS and recently proposed RTEA.
- The proposed hybrid FS approach is evaluated on 18 standard UCI datasets [17] using K-nearest Neighbors (KNN), Random Forest, and Naive Bayes classifiers.
- The proposed FS approach is compared with 10 state-of-the-art meta-heuristic FS methods.

## II. LITERATURE REVIEW

Meta-heuristic algorithms are categorized differently in the literature: single solution based and population based [18], nature inspired and non-nature inspired [19], metaphor based and non-metaphor based [20]. These algorithms can also be divided into four different categories from ‘inspiration’ point of view [21]: Evolutionary, Swarm inspired, Physics based and Human behavior related.

FS is a binary optimization problem and most of the standard and popular optimization algorithms introduced so far in the literature have been applied for solving FS problems. Different applications of Genetic Algorithm (GA) for FS can be found in [22, 23, 24, 25]. Particle Swarm Optimization (PSO) based FS methods can be found in [26, 27, 28]. Ant Colony Optimization (ACO) and Gravitational Search Algorithm (GSA) based FS methods can be found in [29] and [30] respectively. Different types of meta-heuristics algorithms based on the source of inspiration are briefed here.

- Evolutionary algorithms are inspired from biological science. It uses the concept of mutation and crossover

to evolve the randomly generated initial population over iterations and eliminates the worst solution in order to obtain a better solution. GA [31] is inspired from the biological evolution. *Mutation* and *crossover* are two of the most common operators used in GA. Mutation operates on a single solution and generally changes a feature randomly or following some pre-defined criterion. Crossover, on the other hand, operates on two parent solutions to produce two offspring, resulting in new and better solutions. Co-evolving algorithm [32], Differential evolution (DE) [33], Genetic Programming [34], Evolutionary programming [35], Bio-geography based optimizer [36], Stochastic fractal search [37] etc. are some of the well-known evolutionary algorithms.

- Swarm inspired algorithms mimic 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 us to solve complex optimization problems. One of the most popular Swarm inspired algorithms is PSO [38], which is proposed by simulating social behavior, as representation of the movements of organisms in a bird flock or fish schools. This method performs the search to obtain the optimal solution through agents, referred to as particles. The movement of these particles is influenced by local optima in search space, and are updated if a better solution is found. Another approach of this category is ACO [39], inspired from the foraging method of ant species. Grey Wolf Optimizer (GWO) [12] is inspired by the leadership hierarchy and hunting mechanism of grey wolves. Four types of grey wolves such as  $\alpha$ ,  $\beta$ ,  $\delta$  and  $\omega$  are employed for replicating the leadership ranking. The three main steps of hunting, which are searching for prey, surrounding a prey, and attacking the prey are used as evolution. Ant Lion Optimizer (ALO) [40] proposed mimicking the hunting strategy of antlions in nature. The main steps of the hunting strategy includes the random walk of ants, building traps for ants by antlions, entrapment of ants in traps, catching preys, and re-building the traps. Some other methods belonging to this category are: Shuffled frog-leaping algorithm [41], Bacterial foraging [42], Artificial bee colony (ABC) [43], Firefly algorithm [44], Cuckoo search algorithm [45], Whale optimization algorithm (WOA) [46], Grasshopper optimization [47], Dolphin echolocation [48], Squirrel search algorithm [49] etc.
- Physics based algorithms are inspired by the physical processes in nature. The inspiring physical processes include music, metallurgy to mathematics, physics, chemistry, and complex dynamic systems. One of the oldest algorithms of this category is Simulated Annealing (SA) [50], developed by following the annealing [51] process of metals present in metallurgy and materials sciences. Another popular method of this category is GSA [30], developed by following gravity and mass interaction. The search agents are considered as collection

of masses, which interact with each other based on the Newton's gravitational law and the laws of motion. The used HS in the present work belongs to this category. Some other methods of this category are Self propelled particles [52], Black hole optimization [53], Charged system search [54], Sine Cosine algorithm [55], Multi-verser optimizer [56] etc.

- Human related algorithms are inspired from human behavior and interactions. Teaching-Learning-Based optimization [57] is a very popular method of this category, developed by following the enhancing procedure of class grade. Imperialistic competitive algorithm [58] is inspired from the human socio-political evolution process. Here, the populations are divided into two categories: colonies and imperialists states. The idea of this algorithm stands upon the competition among imperialists to take control of the colonies. At the end of the competition, only one imperialist stands out as victor and takes control over all the colonies, and the weak empires collapse. Some other methods of this category are: Society and civilization [59], League championship algorithm [60], Tug of war optimization [61], Volleyball premier league algorithm [62].

Nowadays, hybrid meta-heuristics algorithms have been used frequently for solving FS problems. Hybrid meta-heuristics have been proven to be an efficient approach to achieve better performance in various real-life problems [63]. In [64], the first hybrid meta-heuristic method is proposed for FS by combining GA with local search algorithm. The hybrid combination 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 and applied on UCI datasets in [67]. In [68], Salp Swarm algorithm (SSA) is hybridized with Opposition Based Learning (OBL) and a local search method which is then 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. Hybrid of PSO and GSA can be found in [71]. Hybrid of ACO and GA has been proposed in [72]. In [73], a hybrid version of DE and ABC for FS has been proposed and applied on UCI datasets.

Therefore, it is a time to raise a question. Why do we even need any new hybrid meta-heuristic algorithm for solving FS problems, as we have abundant of such algorithms? This question is quite logical and obvious. The question is best answered by a work reported in [74], which proposes *No Free Lunch* theorem and summarizes that there is not a single optimization algorithm which is capable for solving every type of optimization problem. With each new optimization algorithm following any regular phenomena, researchers primarily focus to give some new facet to the algorithm where both exploration and exploitation will have a superior trade-

off, so it ultimately gets away from the local optima and compasses to the global optima. Nevertheless, accomplishing these objectives are not simple, particularly in the event for which one needs to propose an algorithm that can be applicable to different domains. This practically motivates the researchers to come up with better optimization algorithms in comparison with the previously proposed algorithms. This is the inspiration which keeps the research active in the field of FS and motivates us to propose a new hybrid FS algorithm called RTHS algorithm based on HS method [14] and recently proposed RTEA [15].

### III. PRELIMINARIES

#### A. HARMONY SEARCH: AN OVERVIEW

HS algorithm [14] is inspired from an artificial phenomena, musical harmony. It transforms the qualitative improvisation process into quantitative optimization process with some well defined rules and thus turning the beauty and harmony of music into solution for several optimization problems. Just like the musical performers seek a fantastic harmony, this algorithm seeks the best state determined by fitness (objective) function. Just like the music can be improved for better aesthetic estimation, the fitness value can be improved in every iteration on order to find a better solution. Three important components of this algorithm are: harmony memory, pitch adjustment and randomization. HS algorithm does not require differential gradients, thus it can consider continuous as well as discontinuous functions. The basic steps used in HS algorithm are as follows:

- Step 1: Randomly initialize initial population, here addressed as Harmony Memory (HM).
- Step 2: Improvise a new harmony from it.
- Step 3: If the new harmony is better than the worst harmony in HM, replace it.
- Step 4: If stopping criterion is not met, goto Step 2.

Now, it assumes that all the parts of a global solution exist initially in HM, which is not necessarily the case always. So, to bring diversity, it utilizes Harmony Memory Considering Rate (HMCR),  $HMCR \in [0, 1]$ . If this rate is too low, then very few elite harmonies are selected and it may converge too slowly. On the other hand, if this rate is extremely high (near 1), the pitches in the HM are mostly used, and other ones are not explored well, not leading to good solutions. Therefore, typically, we use  $HMCR = [0.7, 0.95]$  [75]. Another factor called Pitch Adjustment Rate (PAR) is introduced by mimicking pitch adjustment procedure. This produces a new pitch by adding small random amount to the existing pitch.

Below, we have discussed application of HS algorithm as a meta-heuristic FS method in different domains as well as some of the modifications of it found in the literature. In [76], the authors have tried to generate a new solution vector that improves accuracy and convergence rate of HS algorithm and applied it on constrained functions (minimization of the weight of the spring, Pressure vessel design, welded beam design etc.) and unconstrained functions. This work mainly

**Algorithm 1** Pseudocode of HS algorithm**Input:**  $popSize, maxIter, HMCR, PAR$ **Output:** Best harmony  $H = (h_1, h_2, \dots, h_D)$ 

```

Randomly generate initial population  $HM(0)$ 
for  $iter \leftarrow 1 \dots maxIter$  do
    if  $random(0, 1) \geq HMCR$  then
         $h^j(t+1) \leftarrow random(lb, ub) \forall j \in [1, D]$ 
        // lb= lower bound, ub= upper bound
    else
        for  $j \leftarrow 1 \dots D$  do
             $i \leftarrow random(1, popSize)$ 
             $h^j(t+1) \leftarrow HM_i^j(t)$ 
        end for
    end if
     $w$  be index of the worst harmony in  $HM(t)$ 
    if  $fitness(h(t+1)) < fitness(HM_w(t))$  then
         $HM_w(t) \leftarrow h(t+1)$ 
    end if
    for  $i \leftarrow 1 \dots popSize$  do
        if  $random(0, 1) < PAR$  then
             $HM_i^j(t+1) \leftarrow HM_i^j(t) + random(-1, 1) \times$ 
             $(ub - lb) \forall j \in [1, D]$ 
        end if
    end for
end for

```

discusses about the effect of constant parameters of HS algorithm. Another work reported in [77] describes a new version of HS algorithm for engineering optimization problems with continuous design variables. This algorithm has been applied on unconstrained function minimization problems (Rosenbrock function, Eason and Fenton's gear train inertia function, Wood function, Powerwell quartic function etc.), constrained function minimization problems, structural engineering optimization problems (Pressure vessel design, Welded beam design etc.) and more. In [78], a new variant of HS algorithm has been proposed, which is known as Global-best Harmony Search (GHS). This algorithm takes the help of swarm intelligence to improve the performance of HS algorithm. It has been applied on Sphere function, Schwefel's problem, Step function, Rosenbrock function, Rastrigin function etc. In [79] the authors present a cost minimization model for the the design of water distribution network. It is applied on five water distribution networks, which are Two-loop water distribution network, Hanoi water distribution network, New York City water distribution network, GoYang water distribution network, and BakRyun water distribution network. The work proposed in [75] reviews and analyzes the HS algorithm in the context of meta-heuristic algorithms.

**B. RING THEORY BASED EVOLUTIONARY ALGORITHM**

RTEA [15] is an recently proposed approach to solve combinatorial problems using algebraic theory. A global exploration operator (R-GEO) and local development operator (R-LDO) are proposed using the addition, multiplication and

inverse operation of direct product of the rings. Then by using R-GEO and R-LDO, new individuals are generated following a greedy strategy.

## 1) Some properties of Ring

**Definition of Ring** [80]: A ring  $(\mathcal{R}, +, \cdot)$  is a nonempty set  $\mathcal{R}$  together with two binary operations,  $+$  and  $\cdot$ , defined on  $\mathcal{R}$  such that:

- $\forall x, y \in \mathcal{R}, x + y = y + x$
- $\forall x, y, z \in \mathcal{R}, (x + y) + z = x + (y + z)$
- $\exists 0 \in \mathcal{R}$  such that  $\forall x \in \mathcal{R}, x + 0 = 0 + x = x$ , this 0 is called an additive identity
- $\forall x \in \mathcal{R} \exists y \in \mathcal{R}$  such that  $x + y = y + x = 0$ , y is called the inverse of x, written as  $-x$
- $\forall x, y, z \in \mathcal{R}, (x \cdot y) \cdot z = x \cdot (y \cdot z)$  and
- $\forall x, y, z \in \mathcal{R}, x \cdot (y + z) = x \cdot y + x \cdot z; (y + z) \cdot x = y \cdot x + z \cdot x$

Let  $\mathbb{Z}_n = \{[0], [1], \dots, [n-1]\}$  be a collection of residue classes of modulo n, where  $[i] = \{x \in \mathbb{Z} \mid x \equiv i \pmod{n}\}$ ,  $0 \leq i \leq n-1$ , where  $n \geq 2$  and  $\mathbb{Z}$  is the set of all integers. We define two binary operations as following:

$$[i] \oplus [j] = [(i + j) \pmod{n}] \text{ and } [i] \odot [j] = [(i \cdot j) \pmod{n}] \forall [i], [j] \in \mathbb{Z}_n.$$

Hence,  $\mathbb{Z}_n$  is a ring with operations  $\oplus$  and  $\odot$ .

**Direct product of rings** [80]: The direct product of rings is another ring, whose every element is an ordered m-tuple. If  $\mathcal{R}_i$  are rings,  $i \in \mathcal{I} = 1, 2, 3, \dots, m$ , then  $\Pi_{i \in \mathcal{I}} \mathcal{R}_i = \mathcal{R}_1 \times \mathcal{R}_2 \times \dots \times \mathcal{R}_m$  is a ring with operations  $(a_1, a_2, \dots, a_m) \oplus (b_1, b_2, \dots, b_m) = (a_1 + b_1, a_2 + b_2, \dots, a_m + b_m)$  and  $(a_1, a_2, \dots, a_m) \odot (b_1, b_2, \dots, b_m) = (a_1 b_1, a_2 b_2, \dots, a_m b_m)$ .  $\Pi_{i \in \mathcal{I}} \mathcal{R}_i$  is called as the direct product of  $\mathcal{R}_i, i \in \mathcal{I}$ .

## 2) RTEA Overview

So, we have  $\mathbb{Z}[r_1, r_2, \dots, r_D] = \{0, 1, \dots, r_1 - 1\} \times \{0, 1, \dots, r_2 - 1\} \times \dots \times \{0, 1, \dots, r_D - 1\}$ .

Now, let  $P_1 = (p_{11}, p_{12}, \dots, p_{1D})$ ,  $P_2 = (p_{21}, p_{22}, \dots, p_{2D})$ ,  $P_3 = (p_{31}, p_{32}, \dots, p_{3D})$  and  $P_4 = (p_{41}, p_{42}, \dots, p_{4D})$  be four different D-dimensional vector selected randomly  $\in \mathbb{Z}[r_1, r_2, \dots, r_D]$ . A new D-dimensional vector  $P = (p_1, p_2, \dots, p_D) \in \mathbb{Z}[r_1, r_2, \dots, r_D]$  is created from  $P_1, P_2, P_3, P_4$  using R-GEO, given by Equation 1.

$$p_k = \begin{cases} \{p_{1k} + p_{4k} \times [p_{2k} + (r_k - p_{3k})]\} (\pmod{r_k}) & \text{if } rndm(k) \leq 0.5; \\ \{p_{1k} + [p_{2k} + (r_k - p_{3k})]\} (\pmod{r_k}) & \text{otherwise.} \end{cases} \quad (1)$$

R-GEO reflects global exploration ability and R-LDO which acts as the local search operator is given by Algorithm 2.

Conciseley R-LDO and R-GEO are used to generate new individuals, and a greedy strategy is used to select individuals to form the new generation.

**IV. PROPOSED RTHS ALGORITHM**

Let us consider that we have the original feature set as  $\mathcal{F} = \{f_1, f_2, \dots, f_D\}$ , where  $n$  is the total number of attributes,

**Algorithm 2** Pseudocode of R-LDO

---

**Input**  $X = (x_1, x_2, \dots, x_D) \in \mathbb{Z}[r_1, r_2, \dots, r_D]$   
local search probability  $Prb_m \in (0, 0.5]$   
**Output**  $X = (x_1, x_2, \dots, x_D)$

```

for  $i = 1 \dots D$  do
  if  $rndm1 < Prb_m$  then
    if  $rndm2 < 0.5$  and  $x_i \neq 0$  then
       $x_i \leftarrow r_i - x_i;$ 
    else
       $x_i \leftarrow rndm(\{0, 1, \dots, r_i - 1\} - \{x_i\});$ 
    end if
  end if
end for

```

---

$\mathcal{C} = \{c_1, c_2, \dots, c_l\}$  be the class set, where  $l$  is the number of classes. Our target is to find a subset  $\mathcal{F}' = \{f'_1, f'_2, \dots, f'_o\}$ , where  $o < D$ ,  $\mathcal{F}' \subset \mathcal{F}$  and  $\mathcal{F}'$  has lower classification error rate than any other subset having same size or any proper subset of  $\mathcal{F}'$ . FS is a binary optimization problem, where 1 is indicates that the corresponding feature is selected whereas 0 indicates that the corresponding feature is discarded. So, we select the features having value 1 and discard those features having value 0. Our main goal is to decrease the number of 1's along with increasing the classification accuracy. In RTEA, which has been applied for solving Knapsack problem (KP) problem in the original paper, there are two cases. First, it is considered that the problem has a feasible solution, and second, it considers the opposite i.e. there is no feasible solution of the problem under consideration. For us, as we assume that there are potential solutions in the search space, so we consider this particular case [15] which is a binary problem in itself.  $\mathbb{Z}[r_1, r_2, \dots, r_D]$  in the previous section becomes  $\mathbb{Z}[2, 2, \dots, 2]$ , so we do not have to use any transfer function. Now, we need to consider another major aspect of FS, that is whenever we talk about achieving higher classification accuracy with lowest number of features by the algorithm, it can be observed that these two objectives are contradictory in nature. To get rid of this issue, classification error rate has been considered here. Using Equation 2, these two driving candidates have been combined.

$$\downarrow \text{Fitness} = \omega \zeta(\mathcal{F}') + (1 - \omega) \frac{|\mathcal{F}'|}{|\mathcal{F}|} \quad (2)$$

where  $\mathcal{F}'$  represents the set consisting of selected features,  $|\mathcal{F}'|$  represents number of selected features  $\zeta(\mathcal{F}')$  represents classification error rate of  $\mathcal{F}'$ ,  $|\mathcal{F}|$  is the original dimension of the dataset and  $\omega$  represents weight  $\in [0, 1]$ . HS algorithm finds the global optima by initiating HMCR and uses PAR to escape local optima. RTEA uses R-LDO for local exploration and R-GEO for global search. As FS is a binary optimization problem, R-LDO has been modified slightly as the population contains binary values. This modified version of R-LDO is given by Equation 3.

```

for  $d \leftarrow 1 \dots D$ 
   $X^d(t+1) = 1 - X^d(t)$  if  $random[0, 1] \leq Prb_m$ 
endfor

```

(3)

where  $X^d(t)$  represents the  $d^{th}$  dimension in  $t^{th}$  iteration of the solution  $X$ .

The direct product of rings is a ring too. As we utilize the  $\{0, 1\}$  version of RTEA, the obtained results will be in  $\{0, 1\}$ . With the help of R-GEO, RTEA performs the global exploration, and with R-LDO, it performs the exploitation. So, there is a balance between exploration phase and exploitation phase. The flowchart of our proposed method is given in Figure 1.

The HS algorithm depends on the value of HMCR and PAR in order to find out the global best solution. HMCR [76] mainly helps to find the ‘promising’ areas where optimal solution (global best) may lie i.e., it ensures exploration. On the other hand, PAR [76] helps to properly search the areas already discovered i.e., it ensures exploitation. Therefore, chances of obtaining the best solution depend on this HMCR and PAR, but their values need to be defined at the beginning of the HS algorithm. Appropriate setting of these values is needed for convergence of this algorithm. Again the same set of values may not to produce optimum results for all problems. In [76], the authors have tried to address this issue with an iterative approach, but then precision becomes another problem to deal with. With increase in precision by 10% margin, the time requirement increases exponentially. Here, we have tried to solve this problem using a completely different approach. We have enhanced both the exploration and exploitation operations of HS algorithm by a recently proposed meta-heuristic RTEA. In each iteration, we not only ‘enhance’ a particular harmony by finding its fitter neighbor using the R-LDO operator but also we excel the ‘improvise a new harmony’ step used in the HS algorithm with the help of R-GEO operator. This implies that we are able to reduce the dependency of the HS algorithm on initial values of HMCR and PAR to obtain the global optima. R-GEO, aids in the exploration process by considering randomly 4 harmonies present in current HM (population) and checking whether is it possible to form a better harmony or not. R-LDO tries to improve the exploitation by finding any better neighbor for a particular harmony. The results shown in subsection V-C and section VI validate this claim.

## V. RESULTS AND DISCUSSION

We have evaluated the proposed RTHS algorithm using three popular classifiers: KNN [81], Random Forest [82], Naive Bayes [83] for assessing the effectiveness of the same.

For each dataset, 80% of the instances are used to train the model and the rest 20% are used for testing. We have applied the FS methods on the trained data, and determined the features which are useful. These features form the optimal feature subset. From test data, only those features are selected

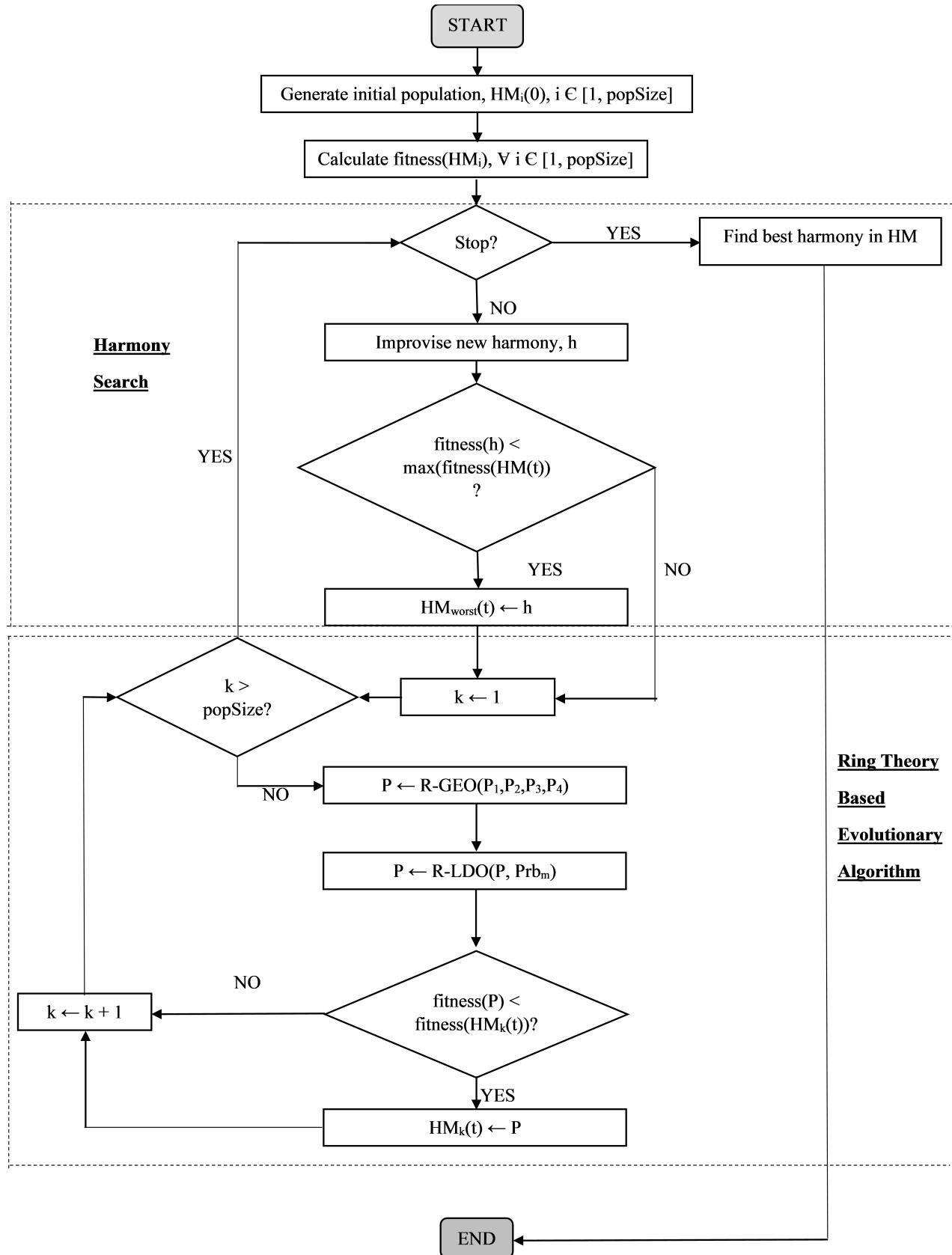


FIGURE 1: Pictorial representation of the proposed FS model named as RTHS: hybrid of Harmony Search (HS) algorithm and Ring Theory based Evolutionary Algorithm (RTEA)

and the test classification accuracy is measured based on these using the above mentioned classifiers. The proposed FS method is implemented using Python3 [84] whereas the graphs are plotted using Matplotlib [85].

### A. DATASET DESCRIPTION

In order to examine the performance of HS, RTEA and RTHS algorithms, 18 standard UCI datasets [17] have been considered. These datasets are selected from various backgrounds. The description of these datasets is presented in Table 1, which shows that there are 14 bi-class and 4 multi-class datasets. The datasets are diverse in terms of number of features and instances. These mixture helps us in establishing the robustness of the proposed FS method.

### B. PARAMETER TUNING

When we talk about multi-agent evolutionary algorithm, both population size and maximum number of iterations play a significant role for characterizing the behavior of one agent's learning ability from others' experiences and the step-by-step evolution of the agents respectively. For finding the appropriate values of these two parameters, we have performed experiments by varying one parameter w.r.t. the other.

Figure 2 shows the effect of the size of the population on achieved classification accuracy using the proposed FS method. Considering that the time requirement increases with increase in population size and the effect of the same on classification accuracy, we have fixed the values of population size to 20 as the standard population size, the maximum number of iterations to 30, HMCR to 0.8, PAR to 0.2 and  $Prb_m$  to 0.005 (as suggested in [15]) for all further experiments. Figure 3 shows the best value of the fitness function in each iteration.

### C. DISCUSSION

This section reports the results of the proposed FS method called RTHS for the datasets mentioned in Section V-A.

Table 2, Table 3 and Table 4 describe the results obtained by the proposed RTHS algorithm, as compared to RT and HS algorithms using KNN, Random Forest and Naive Bayes classifiers respectively. It can be concluded that the proposed RTHS algorithm performs the best over UCI datasets with KNN classifier. Besides, KNN is also widely used in the literature for FS purpose on UCI datasets [86, 87, 10]. Hence, for further experiments and analysis, we have used only KNN classifier with  $K = 5$ .

From Table 2, it is quite evident that the proposed FS method has performed significantly well. The RTHS algorithm has produced accuracy  $> 90\%$  (83.33%) for 15 datasets. Whereas it has achieved 100% accuracy for 9 (50%) datasets: CongressEW, Exactly, Ionosphere, M-of-n, PenglunEW, Sonar, Vote, WineEW, and Zoo, which is quite impressive. Out of 18 datasets, it achieves the highest accuracy in almost 17 cases (94.44%). Comparing these results

with HS algorithm, it can be observed that the RTHS algorithm performs better than HS algorithm in exactly 15 cases and in 3 cases they produce equivalent result. Comparing RTHS algorithm with RTEA, it is found that in 12 cases, they achieve the same result, and but in 5 cases, the RTHS algorithm outperforms RTEA. However, in the case of Tic-tac-toe dataset, the RTHS algorithm could not outperform RTEA in terms of classification accuracy.

Now, if we focus on the number of features selected, then it is quite clear that the RTHS algorithm selects the least number of features in exactly 15 cases (83.33%). Careful observation of Table 2 reveals that the RTHS algorithm outperforms both HS algorithm and RTEA with significant margin in most of the cases. It outperforms HS algorithm in 11 cases and gives equivalent result in 4 cases (CongressEW, Exactly2, HeartEW and M-of-n datasets). In case of Exactly, Ionosphere and Tic-tac-toe datasets, HS algorithm is able to produce better results than RTHS algorithm. But RTEA is unable to show better performance than RTHS algorithm for none of the cases, though it provides equivalent result in 4 cases (Breastcancer, BreastEW, Exactly2 and M-of-n datasets).

To visualize these results, bar charts showing the comparison of both average accuracies achieved and number of features utilized by the three algorithms, namely, RTHS algorithm, HS algorithm and RTEA, using KNN classifier have been plotted in Figure 4 and Figure 5 respectively. These bar charts show that all the three FS methods perform almost equivalently in terms of average accuracy, but in terms of average number of features selected, the RTHS algorithm has the upper hand.

From Table 3, it is quite clear that the proposed FS method performs much better than both RTEA and HS algorithm. For 14 datasets (77.78%), it has achieved  $> 90\%$  accuracy. Whereas for 7 datasets (38.89%), it has achieved an 100% accuracy. These datasets are: BreastEW, IonosphereEW, M-of-n, PenglunEW, Vote, WineEW and Zoo. For all the 18 datasets, it achieves the highest accuracy along with ties in case of 10 datasets with RTEA and 3 datasets with HS algorithm. This proves that using Random Forest classifier, the RTHS algorithm still shows superior performance. Figure 6 illustrates the comparison of average accuracies achieved by the three algorithms using Random Forest classifier.

Now, if we consider the number of selected features, we can see that RTHS algorithm is slightly behind the HS algorithm and RTEA. For 8 datasets (44.44%), it selects the least number of features along with ties in 8 datasets with RTEA and in 4 datasets with HS algorithm. From this perspective, the RTHS algorithm may seem to be inefficient, but looking at the achieved classification accuracy, it can be said that the RTHS algorithm produces far better results than others. Figure 7 shows the comparison of number of features chosen by the three algorithms using Random Forest classifier. By looking at Figure 7, it is obvious that in terms of average number of features selected, the RTHS algorithm selects least number of features than RTEA.

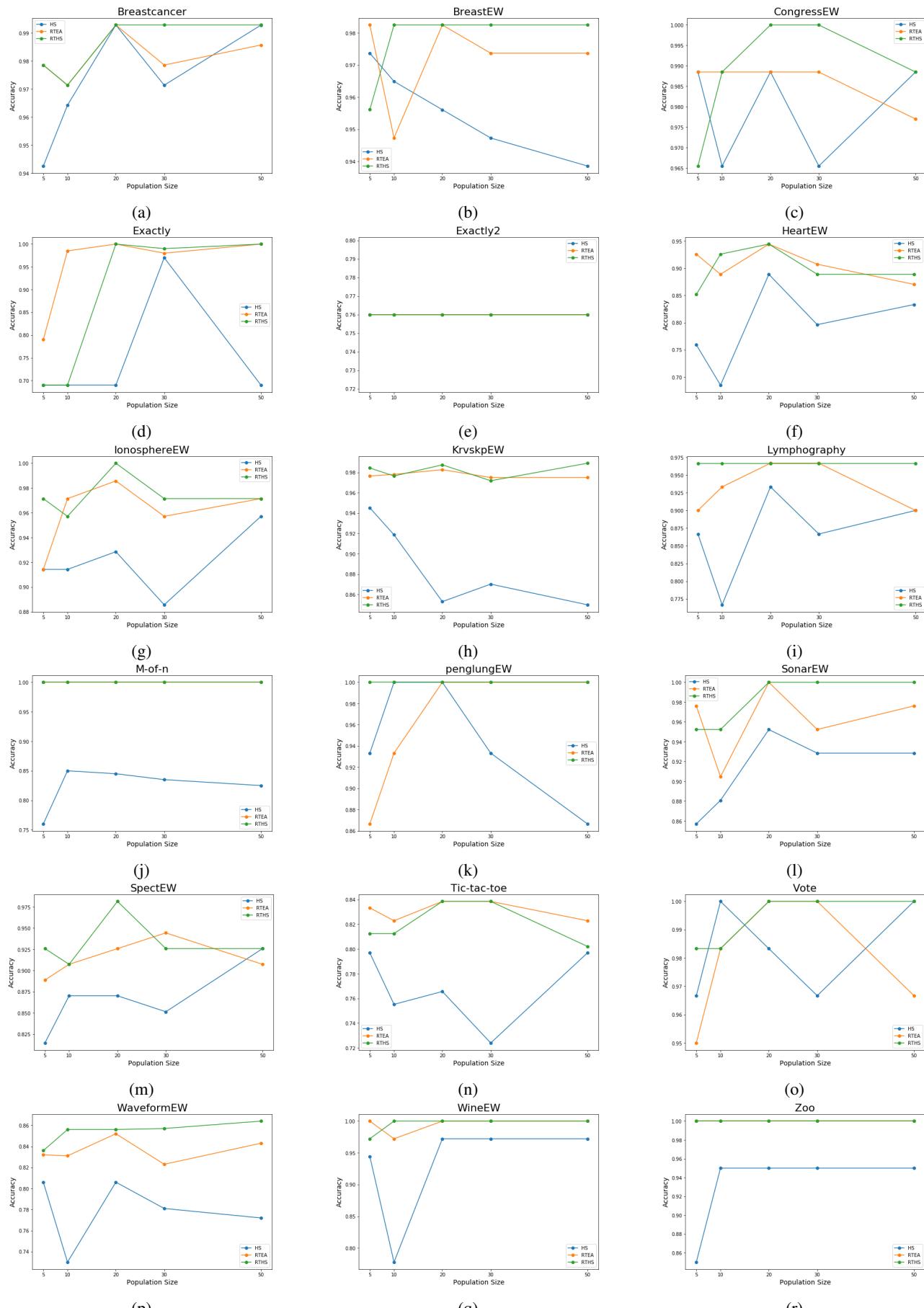


FIGURE 2: Effect of population size on classification accuracy for 18 UCI datasets using HS algorithm, RTEA and RTHS algorithm

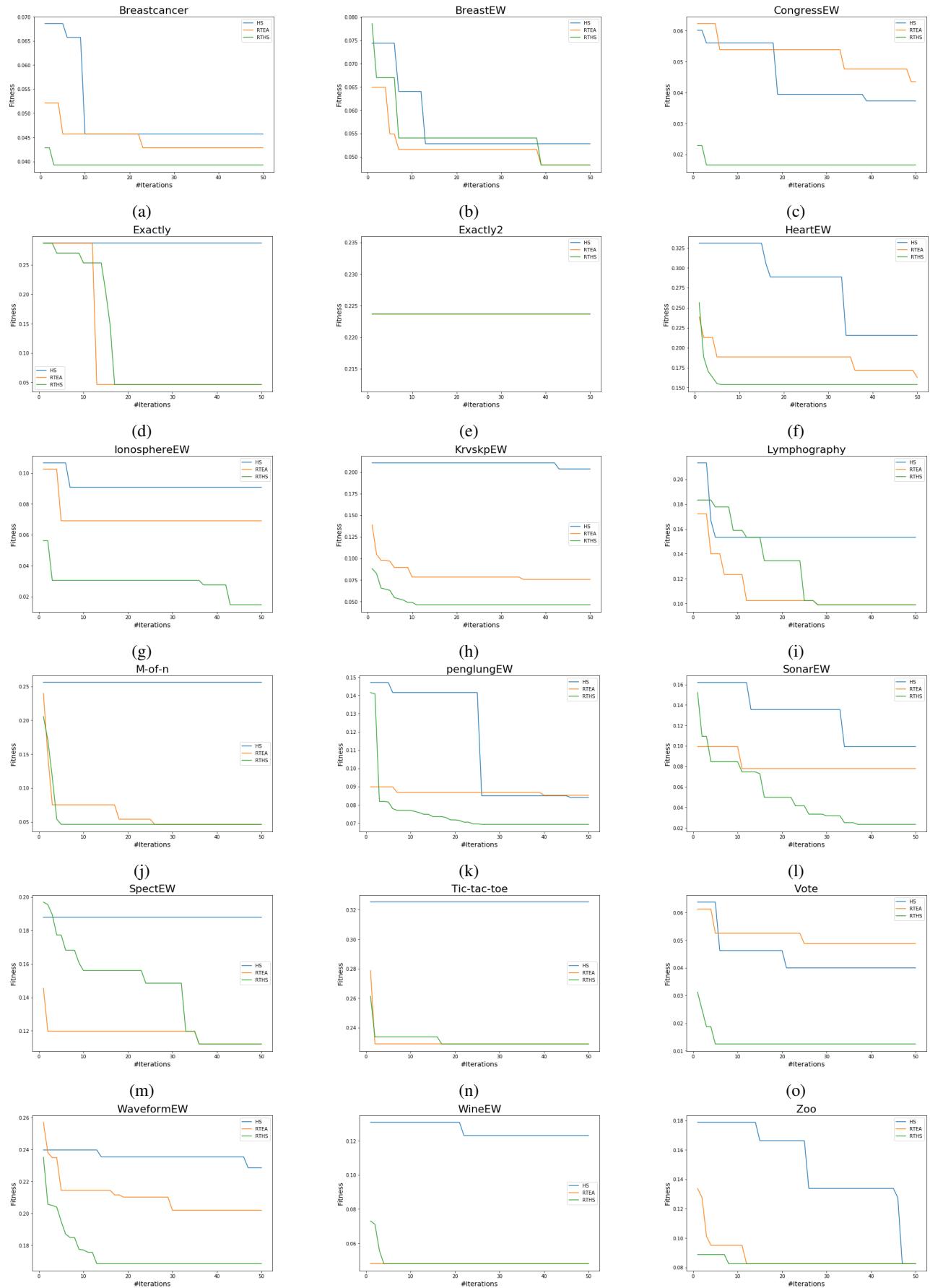


FIGURE 3: Best fitness value in each iteration for 18 UCI datasets using HS algorithm, RTEA and RTHS algorithm

TABLE 1: Description of the datasets used to evaluate the present FS method

| 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              |                 |
| 9       | Lymphography | 18                | 148            | 4              | Biology         |
| 10      | M-of-n       | 13                | 1000           | 2              | Biology         |
| 11      | PenglunEW    | 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              |                 |

Going through Table 4, it can be easily said that the RTHS algorithm has achieved the highest accuracy for all the 18 UCI datasets along with ties for 4 datasets with HS algorithm and 9 datasets with RTEA. In case of 14 datasets (77.78%), the proposed FS method has achieved accuracy  $> 90\%$ . On the other hand, for 7 datasets (38.89%), which are: BreastEW, IonosphereEW, Lymphography, PenglunEW, Vote, WineEW and Zoo, the proposed RTHS algorithm has produced an 100% accuracy. These prove that the RTHS algorithm produces the best results in terms of classification accuracy. Figure 8 illustrates the comparison of average accuracies achieved by the three algorithms using Naive Bayes classifier. Now, considering the number of selected features, Figure 9 tells us a lot. For 13 datasets (72.22%), the RTHS algorithm selects the least number of features along with ties in 5 datasets with HS algorithm and ties in 4 datasets with RTEA.

Based on the above discussion as well as observing Table 2, we can conclude that the RTHS algorithm significantly helps both HS algorithm and RTEA to explore different parts of the search space and to achieve better solution in terms of both achieved classification accuracy and selected number of features.

During this comparison, if the classification accuracy is found to be higher for any particular classifier, then that classifier is considered as better and if the classification accuracies remain the same for any two classifiers, then the number of selected features is used as the deciding factor to break the tie. While observing Table 2, Table 3 and Table 4, we can also conclude that the KNN classifier produces better results than both Random Forest and Naive Bayes classifiers.

Comparing the results obtained from both KNN and Random Forest classifiers, it can be observed that the KNN classifier performs better than Random Forest for exactly 13 datasets (72.22%) considering both the achieved classification accuracy and number of selected features. In case of BreastEW, Tic-tac-toe and WaveformEW datasets, Random

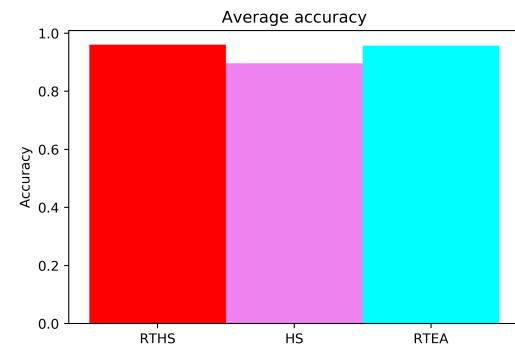


FIGURE 4: Graphical comparison of average accuracies achieved by RTHS algorithm, HS algorithm and RTEA using KNN classifier

Forest has upper hand over KNN classifier. Again, in case of Exactly2 and M-of-n datasets, two classifiers produce the same classification accuracy and choose the same number of features.

Furthermore, comparing the results obtained by KNN and Naive Bayes classifiers, it can be inferred that the KNN classifier outperforms the Naive Bayes classifier for exactly 12 datasets (66.67%) while considering the above mentioned factors. For 4 datasets, namely, BreastEW, Lymphography, WaveformEW and Zoo, the Naive Bayes classifier produces the best results. For Exactly2 and Vote datasets, the two classifiers result in a tie.

We can finally conclude from the above discussion that the KNN classifier produces better results than Random Forest and Naive Bayes classifier. So, for performing comparison of the proposed RTHS algorithm with state-of-the-art FS methods, we consider KNN classifier only.

TABLE 2: Performance comparison of HS algorithm, RTEA and RTHS algorithm in terms of accuracy and number of selected features using KNN classifier

| Sl. No. | Dataset      | Original |           | HS           |           | RTEA         |           | RTHS         |           |
|---------|--------------|----------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
|         |              | Accuracy | #Features | Accuracy     | #Features | Accuracy     | #Features | Accuracy     | #Features |
| 1       | Breastcancer | 96       | 9         | <b>99.28</b> | 5         | <b>99.28</b> | <b>3</b>  | <b>99.28</b> | 3         |
| 2       | BreastEW     | 92.63    | 30        | 95.61        | 6         | 98.24        | <b>2</b>  | <b>98.25</b> | 2         |
| 3       | CongressEW   | 92.18    | 16        | 98.85        | <b>4</b>  | 98.85        | 7         | <b>100</b>   | 4         |
| 4       | Exatly       | 72.3     | 13        | 69           | <b>1</b>  | <b>100</b>   | 6         | <b>100</b>   | 6         |
| 5       | Exactly2     | 73.3     | 13        | <b>76</b>    | <b>1</b>  | <b>76</b>    | <b>1</b>  | 76           | 1         |
| 6       | HeartEW      | 68.15    | 13        | 88.89        | <b>5</b>  | <b>94.4</b>  | 6         | <b>94.4</b>  | 5         |
| 7       | IonosphereEW | 83.43    | 34        | 92.85        | <b>3</b>  | 97.14        | 8         | <b>100</b>   | 8         |
| 8       | KrvskpEW     | 96.1     | 36        | 85.29        | 10        | 97.96        | 17        | <b>98.75</b> | 9         |
| 9       | Lymphography | 81.33    | 18        | 93.33        | 10        | <b>96.67</b> | 7         | <b>96.67</b> | 4         |
| 10      | M-of-n       | 87.4     | 13        | 84.5         | <b>6</b>  | <b>100</b>   | <b>6</b>  | <b>100</b>   | 6         |
| 11      | PenglunEW    | 81.33    | 325       | <b>100</b>   | 77        | <b>100</b>   | 80        | <b>100</b>   | 13        |
| 12      | SonarEW      | 80.95    | 60        | 95.23        | 16        | <b>100</b>   | 16        | <b>100</b>   | 8         |
| 13      | SpectEW      | 82.22    | 22        | 87.03        | 8         | 96.29        | 10        | <b>98.15</b> | 4         |
| 14      | Tic-tac-toe  | 81.1     | 9         | 76.56        | <b>3</b>  | <b>85.41</b> | 6         | 83.8         | 9         |
| 15      | Vote         | 92.33    | 16        | 98.33        | 3         | <b>100</b>   | 4         | <b>100</b>   | 2         |
| 16      | WaveformEW   | 81.44    | 40        | 80.6         | 16        | <b>84.1</b>  | 20        | <b>84.1</b>  | 15        |
| 17      | WineEW       | 66.67    | 13        | 97.22        | 3         | <b>100</b>   | 3         | <b>100</b>   | 2         |
| 18      | Zoo          | 87       | 16        | 95           | 6         | <b>100</b>   | 5         | <b>100</b>   | 4         |

TABLE 3: Performance comparison of HS algorithm, RTEA and RTHS algorithm in terms of accuracy and number of selected features using Random Forest classifier

| Sl no. | Dataset      | Original |           | HS         |           | RTEA         |           | RTHS         |           |
|--------|--------------|----------|-----------|------------|-----------|--------------|-----------|--------------|-----------|
|        |              | Accuracy | #Features | Accuracy   | #Features | Accuracy     | #Features | Accuracy     | #Features |
| 1      | Breastcancer | 97.14    | 9         | 97.86      | <b>2</b>  | 98.57        | <b>2</b>  | <b>98.86</b> | 2         |
| 2      | BreastEW     | 92.3     | 30        | 97.37      | <b>4</b>  | 99.12        | 6         | <b>100</b>   | 7         |
| 3      | CongressEW   | 91.4     | 16        | 96.55      | 4         | 96.55        | <b>2</b>  | <b>97</b>    | 2         |
| 4      | Exactly      | 65       | 13        | <b>69</b>  | <b>1</b>  | <b>69</b>    | <b>1</b>  | 69           | 1         |
| 5      | Exactly2     | 72       | 13        | <b>76</b>  | <b>1</b>  | <b>76</b>    | <b>1</b>  | 76           | 1         |
| 6      | HeartEW      | 67.77    | 13        | 85.19      | 5         | 85.19        | <b>4</b>  | <b>87.04</b> | 5         |
| 7      | IonosphereEW | 81.84    | 34        | 91.43      | 12        | <b>100</b>   | <b>9</b>  | <b>100</b>   | 9         |
| 8      | KrvskpEW     | 94.27    | 36        | 95.46      | 13        | <b>98.59</b> | <b>14</b> | <b>98.59</b> | 14        |
| 9      | Lymphography | 83.33    | 18        | 90         | <b>5</b>  | 90           | 9         | <b>93.33</b> | 9         |
| 10     | M-of-n       | 74.64    | 13        | 80         | <b>4</b>  | <b>100</b>   | 8         | <b>100</b>   | 6         |
| 11     | PenglunEW    | 80.43    | 325       | 93.33      | <b>95</b> | 86.67        | 122       | <b>100</b>   | 121       |
| 12     | SonarEW      | 76.26    | 60        | 85.71      | 20        | <b>90.48</b> | <b>17</b> | <b>90.48</b> | 20        |
| 13     | SpectEW      | 81.37    | 22        | 87.04      | <b>6</b>  | <b>96.3</b>  | 10        | <b>96.3</b>  | 7         |
| 14     | Tic-tac-toe  | 81.58    | 9         | 68.75      | <b>1</b>  | <b>93.75</b> | 9         | <b>93.75</b> | 8         |
| 15     | Vote         | 88.34    | 16        | 95         | <b>1</b>  | 98.33        | 6         | <b>100</b>   | 5         |
| 16     | WaveformEW   | 78.5     | 40        | 79         | <b>14</b> | 84.6         | 23        | <b>86.2</b>  | 16        |
| 17     | WineEW       | 67.7     | 13        | 97.22      | 4         | <b>100</b>   | <b>3</b>  | <b>100</b>   | 3         |
| 18     | Zoo          | 90       | 16        | <b>100</b> | <b>6</b>  | <b>100</b>   | 6         | <b>100</b>   | 6         |

## VI. COMPARISON

To check the effectiveness of the proposed FS method, we have compared it with 10 state-of-the-art FS methods that include four popular meta-heuristic FS methods namely, GA, PSO, ALO, and GSA, and six hybrid meta-heuristic FS methods namely, Serial grey-whale optimizer (HSGW), random switching grey-whale optimizer (RSGW), adaptive switching grey-whale optimizer (ASGW), WOASAT-2, BG-WOPSO and WOA-CM. HSGW, RSGW, and ASGW are three different FS strategies formed by hybridizing both GWO and WOA methods [86]. WOASAT-2 method [10] is hybrid of WOA and SA methods. BGWOPSO method [70] is developed by hybridizing GWO and PSO methods. In WOA-CM method [88], the performance of WOA is enhanced by using both crossover and mutation. The values of the control parameters of these FS methods are described in Table 5.

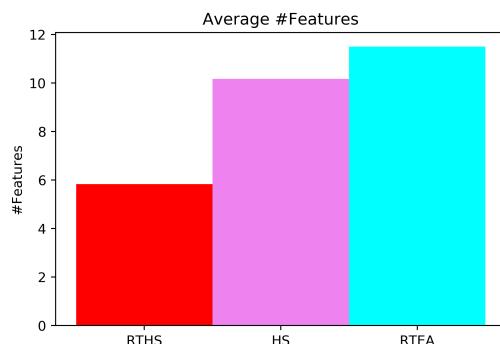


FIGURE 5: Graphical comparison of average #features selected by RTHS algorithm, HS algorithm and RTEA using KNN classifier

TABLE 4: Performance comparison of HS algorithm, RTEA and RTHS algorithm in terms of accuracy and number of selected features using Naive Bayes classifier

| Sl no. | Dataset      | Original   |           | HS         |           | RTEA         |           | RTHS         |           |
|--------|--------------|------------|-----------|------------|-----------|--------------|-----------|--------------|-----------|
|        |              | Accuracy   | #Features | Accuracy   | #Features | Accuracy     | #Features | Accuracy     | #Features |
| 1      | Breastcancer | 91.43      | 9         | 97.85      | 2         | 98.57        | 2         | <b>98.86</b> | 2         |
| 2      | BreastEW     | 98.245     | 30        | 99.28      | 6         | 98.24        | 10        | <b>100</b>   | 6         |
| 3      | CongressEW   | 97.7       | 16        | 97.7       | 4         | <b>98.85</b> | 2         | <b>98.85</b> | 1         |
| 4      | Exactly      | <b>69</b>  | 13        | <b>69</b>  | 1         | <b>69</b>    | 1         | <b>69</b>    | 1         |
| 5      | Exactly2     | <b>76</b>  | 13        | <b>76</b>  | 1         | <b>76</b>    | 1         | <b>76</b>    | 1         |
| 6      | HeartEW      | 92.59      | 13        | 92.59      | 4         | <b>94.44</b> | 7         | <b>94.44</b> | 7         |
| 7      | IonosphereEW | 94.29      | 34        | 94.28      | 3         | 97.14        | 5         | <b>100</b>   | 9         |
| 8      | KrvskpEW     | 66.2       | 36        | 80.75      | 17        | 93.58        | 14        | <b>96.4</b>  | <b>11</b> |
| 9      | Lymphography | 93.33      | 18        | 93.33      | 9         | 96.67        | 5         | <b>100</b>   | 6         |
| 10     | M-of-n       | 96.5       | 13        | 83.5       | 5         | <b>98</b>    | 9         | <b>98</b>    | 6         |
| 11     | PenglungEW   | 66.7       | 325       | 86.67      | 28        | 86.67        | 66        | <b>100</b>   | <b>29</b> |
| 12     | SonarEW      | 88         | 60        | 88         | 9         | 90.48        | 25        | <b>97.61</b> | 6         |
| 13     | SpectEW      | 66.67      | 22        | 88.89      | 9         | 92.59        | 6         | <b>94.44</b> | <b>4</b>  |
| 14     | Tic-tac-toe  | 75         | 9         | 69.79      | 3         | <b>72.91</b> | 5         | <b>72.91</b> | 5         |
| 15     | Vote         | <b>100</b> | 16        | 98.33      | 2         | <b>100</b>   | 3         | <b>100</b>   | 2         |
| 16     | WaveformEW   | 81.5       | 40        | 82         | 16        | 83.6         | 18        | <b>84.3</b>  | <b>10</b> |
| 17     | WineEW       | <b>100</b> | 13        | <b>100</b> | 5         | <b>100</b>   | 3         | <b>100</b>   | <b>3</b>  |
| 18     | Zoo          | <b>100</b> | 16        | <b>100</b> | 5         | <b>100</b>   | 4         | <b>100</b>   | <b>3</b>  |

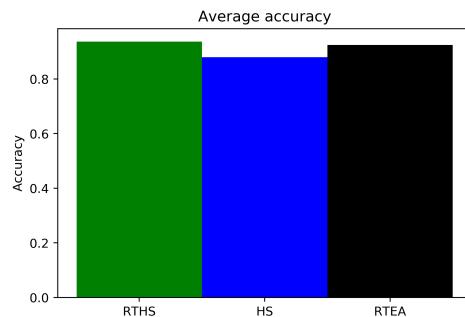


FIGURE 6: Graphical comparison of average accuracies attained by RTHS algorithm, HS algorithm and RTEA using Random Forest classifier

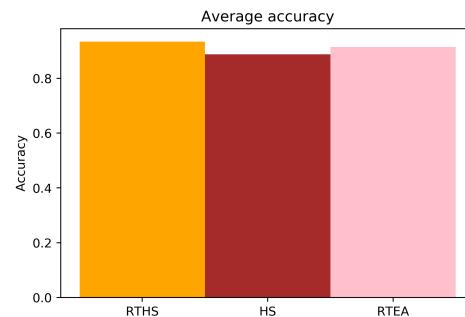


FIGURE 8: Graphical comparison of average accuracies attained by RTHS algorithm, HS algorithm and RTEA using Naive Bayes classifier

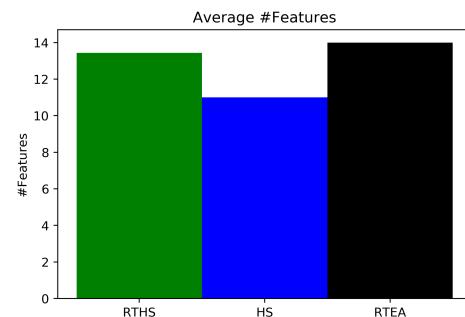


FIGURE 7: Graphical comparison of average #features selected by RTHS, HS and RTEA algorithms using Random Forest classifier

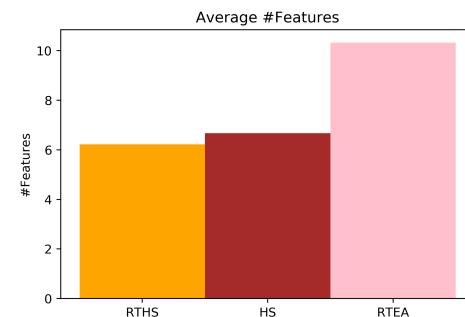


FIGURE 9: Graphical comparison of average #features selected by RTHS algorithm, HS algorithm and RTEA using Naive Bayes classifier

TABLE 5: Parameters setting for state-of-the-art FS methods used here for comparison

| Algorithm        | Parameters  |
|------------------|---|
| BGA              | popSize = 8<br>maxIter = 70<br>Mutation and Crossover rate = 0.8  |
| BPSO             | popSize = 8<br>maxIter = 70<br>inertia factor = 0.1<br>individual-best acceleration factor = 0.1            |
| BALO             | popSize = 8<br>maxIter = 200  |
| BGSA             | popSize = 8<br>maxIter = 20<br>G=1<br>$\alpha = 20$   |
| HSGW, RSGW, ASGW | popSize = 20<br>maxIter = 100<br>a in GWO and WOA = [2 0]<br>$\alpha$ in GSA = 20<br>G0 in GSA = 100        |
| WOASAT-2         | popSize = 10<br>maxIter = 100<br>a in WOA = [2 0]<br>selection pressure = 0.5                               |
| BGWOPSO          | popSize = 10<br>maxIter = 100<br>$c_1 = c_2 = c_3 = 0.5$<br>$w = 0.5 + \text{rand}() / 2$<br>$l \in [1, 0]$ |
| WOA-CM           | popSize = 10<br>maxIter = 100<br>a in WOA = [2 0]   |

Table 6 shows the performance of the RTHS algorithm in terms of classification accuracy. From Table 6, it can be observed that the RTHS algorithm performs the best in 16 cases (88.9%), which is quite impressive. In case of Tic-tac-toe dataset, it stands *third* position following ASGW and RSGW methods.

It is worth mentioning that the RTHS algorithm outperforms BGA method completely in 15 cases, and ties in 2 cases. For Exactly2 dataset, BGA performs better than RTHS algorithm. On the other hand, the RTHS algorithm outperforms BPSO in 15 cases and ties in 2 cases. For Exactly2 dataset, BPSO performs slightly better than RTHS algorithm. However, the proposed algorithm completely outperforms both BALO and BGSA methods for exactly 17 datasets except Exactly2.

Again, the HSGW method outperforms RTHS algorithm in case of Exactly2 dataset and ties in 4 cases. The RTHS algorithm outperforms HSGW in 13 cases. With respect to RSGW method, the RTHS algorithm has 4 ties, 12 wins and 2 losses. ASGW ties with RTHS method in 4 cases whereas the former method outperforms the latter in 3 cases. Furthermore, it loses to RTHS algorithm for the rest 11 cases. In comparison with WOASAT2 method, the RTHS algorithm has 2 ties and 16 wins. Again, the BGWOPSO method has 3 ties and 15 loses when comparing with RTHS algorithm. Both WOA-CM and RTHS tie in case of Exactly2 dataset whereas for the rest 17 datasets, the proposed RTHS algorithm wins.

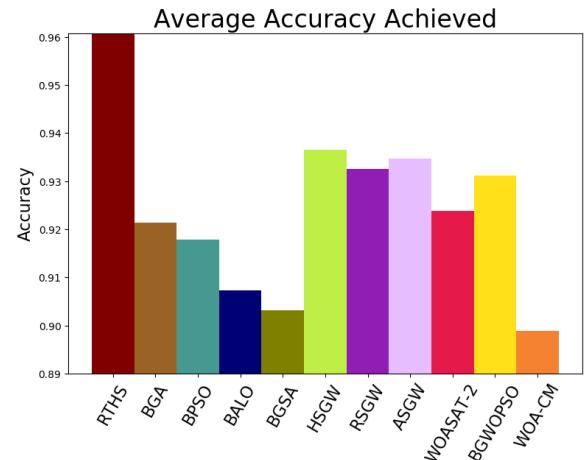


FIGURE 10: Graphical comparison of average accuracies achieved by the proposed method and 10 state-of-the-art FS methods over the 18 UCI datasets

Table 7 shows the performance of the RTHS algorithm w.r.t. number of selected features. The RTHS algorithm selects the lowest number of features in case of 8 datasets. In case of Exactly2 dataset, the RTHS algorithm fails to achieve the highest accuracy but in terms of number of selected features, it performs the best followed by BGA, BPSO and BGSA methods. So, considering both Table 6 and Table 7, it can be concluded that the RTHS algorithm performs the best w.r.t. the 10 state-of-the-art FS methods considered here for comparison. Figure 10 shows the graphical comparison of the average classification accuracies achieved by RTHS algorithm and 10 state-of-the-art FS methods. It clearly shows that the RTHS algorithm attains the highest average classification accuracy as compared to other state-of-the-art FS methods. Figure 11 depicts the graphical comparison of the average number of features selected by RTHS algorithm as well as 10 state-of-the-art FS methods. Figure 11 implies that the proposed RTHS algorithm selects the lowest number of features as compared to other state-of-the-art FS methods.

To determine the statistical significance of the RTHS algorithm, the Wilcoxon rank-sum test [89] has been performed. It is a non-parametric statistical test where pairwise comparison of the proposed FS method is done w.r.t. 10 other state-of-the-art FS methods. Here, the null hypothesis states that the two sets of results follow the same distribution. If the distribution of two results are statistically different, then the generated  $p$ -value obtained from the test statistics will be  $< 0.05$ , when the test is performed at 0.05% significance level. If this condition is satisfied, then the null hypothesis is rejected. From the test results provided in Table 8, it can be concluded that the proposed RTHS algorithm is statistically significant w.r.t. 10 other state-of-the-art FS methods.

Observing both Table 6 and Table 7, we can conclude that the top – 3 FS methods which perform better as compared

TABLE 6: Comparison of classification accuracies obtained by proposed FS method with 10 state-of-the-art FS methods over 18 UCI datasets

| Dataset              | RTHS          | BGA      | BPSO     | BALO  | BGSA   | HSGW         | RSGW     | ASGW         | WOASAT-2 | BGWOPSO  | WOA-CM   |
|----------------------|---------------|----------|----------|-------|--------|--------------|----------|--------------|----------|----------|----------|
| Breastcancer         | <b>0.9928</b> | 0.9743   | 0.9629   | 0.974 | 0.9686 | 0.986        | 0.971    | 0.985        | 0.97     | 0.98     | 0.968    |
| BreastEW             | <b>0.9825</b> | 0.9754   | 0.9719   | 0.974 | 0.9544 | 0.981        | 0.982    | <b>1</b>     | 0.98     | 0.97     | 0.971    |
| CongressEW           | <b>1</b>      | 0.9679   | 0.9633   | 0.981 | 0.9633 | 0.975        | 0.961    | 0.994        | 0.98     | 0.98     | 0.956    |
| Exactly              | <b>1</b>      | <b>1</b> | <b>1</b> | 0.965 | 0.994  | <b>1</b>     | 0.997    | 0.999        | <b>1</b> | <b>1</b> | <b>1</b> |
| Exactly2             | 0.76          | 0.77     | 0.768    | 0.762 | 0.77   | <b>0.815</b> | 0.779    | 0.777        | 0.75     | 0.76     | 0.742    |
| HeartEW              | <b>0.9444</b> | 0.8741   | 0.837    | 0.838 | 0.8296 | 0.923        | 0.848    | 0.831        | 0.85     | 0.85     | 0.807    |
| IonosphereEW         | <b>1</b>      | 0.9489   | 0.9489   | 0.904 | 0.9432 | 0.944        | 0.978    | 0.972        | 0.96     | 0.95     | 0.926    |
| KrvskpEW             | <b>0.9875</b> | 0.985    | 0.9731   | 0.973 | 0.9549 | 0.973        | 0.972    | 0.971        | 0.98     | 0.97     | 0.972    |
| Lymphography         | <b>0.9667</b> | 0.8378   | 0.8919   | 0.917 | 0.8649 | 0.934        | 0.893    | 0.884        | 0.89     | 0.92     | 0.852    |
| M-of-n               | <b>1</b>      | <b>1</b> | <b>1</b> | 0.967 | 0.994  | <b>1</b>     | <b>1</b> | <b>1</b>     | <b>1</b> | <b>1</b> | 0.991    |
| PenglungEW           | <b>1</b>      | 0.9189   | 0.9189   | 0.827 | 0.8333 | 0.942        | <b>1</b> | <b>1</b>     | 0.94     | 0.96     | 0.792    |
| SonarEW              | <b>1</b>      | 0.9904   | 0.9423   | 0.845 | 0.9135 | 0.964        | 0.979    | 0.948        | 0.97     | 0.96     | 0.919    |
| SpectEW              | <b>0.9815</b> | 0.8955   | 0.8881   | 0.899 | 0.8433 | 0.862        | 0.815    | 0.87         | 0.88     | 0.88     | 0.866    |
| Tic-tac-toe          | 0.838         | 0.7996   | 0.7996   | 0.783 | 0.7766 | 0.828        | 0.859    | <b>0.865</b> | 0.79     | 0.81     | 0.785    |
| Vote                 | <b>1</b>      | 0.9733   | 0.96     | 0.972 | 0.96   | 0.983        | 0.996    | 0.984        | 0.97     | 0.97     | 0.939    |
| WaveformEW           | <b>0.841</b>  | 0.7836   | 0.756    | 0.797 | 0.7344 | 0.748        | 0.757    | 0.746        | 0.76     | 0.8      | 0.753    |
| WineEW               | <b>1</b>      | 0.9888   | 0.9775   | 0.972 | 0.9775 | <b>1</b>     | <b>1</b> | <b>1</b>     | 0.99     | <b>1</b> | 0.959    |
| Zoo                  | <b>1</b>      | 0.902    | 0.9608   | 0.98  | 0.9804 | <b>1</b>     | <b>1</b> | <b>1</b>     | 0.97     | <b>1</b> | 0.98     |
| <b>Average Rank</b>  | 1.5           | 4.38     | 5.72     | 5.67  | 7.33   | 3.61         | 3.77     | 3.77         | 4.61     | 4.22     | 7.44     |
| <b>Assigned Rank</b> | <b>1</b>      | 5        | 8        | 7     | 9      | 2            | 3        | 3            | 6        | 4        | 10       |

TABLE 7: Comparison of number of selected features by the proposed FS method with 10 state-of-the-art FS methods over 18 UCI datasets

| Dataset              | RTHS      | BGA      | BPSO     | BALO  | BGSA      | HSGW    | RSGW   | ASGW   | WOASAT-2 | BGWOPSO | WOA-CM |
|----------------------|-----------|----------|----------|-------|-----------|---------|--------|--------|----------|---------|--------|
| Breastcancer         | <b>3</b>  | 4        | 4        | 4.7   | 4         | 5       | 5.933  | 4.867  | 4.2      | 4.4     | 4.302  |
| BreastEW             | <b>2</b>  | 8        | 9        | 13.85 | 10        | 16.667  | 17.5   | 15.833 | 11.6     | 13.6    | 15.81  |
| CongressEW           | 4         | <b>2</b> | 3        | 6.65  | 4         | 8.867   | 9.7    | 8.833  | 6.4      | 4.4     | 6.448  |
| Exactly              | 6         | 6        | 6        | 5.75  | <b>4</b>  | 6.7     | 7.1    | 6.867  | 6        | 6       | 6.045  |
| Exactly2             | <b>1</b>  | <b>1</b> | <b>1</b> | 1.5   | <b>1</b>  | 9.033   | 9.2    | 7.933  | 2.8      | 1.6     | 5.252  |
| HeartEW              | 5         | 5        | <b>3</b> | 8.6   | <b>3</b>  | 8.767   | 6.133  | 6.367  | 5.4      | 5.8     | 6.995  |
| IonosphereEW         | 8         | 7        | 7        | 11.75 | 9         | 18.167  | 20.5   | 17.3   | 12.8     | 13      | 14.416 |
| KrvskpEW             | 9         | 11       | 12       | 16.15 | 14        | 24.8    | 24.8   | 24.5   | 18.4     | 15.8    | 18.54  |
| Lymphography         | <b>4</b>  | 5        | 5        | 7.35  | 6         | 10.567  | 10.567 | 11.2   | 7.2      | 9.2     | 8.208  |
| M-of-n               | 6         | 6        | 6        | 6     | <b>5</b>  | 6.8     | 7.1    | 6.867  | 6        | 6       | 6.006  |
| PenglungEW           | <b>13</b> | 84       | 130      | 133.1 | 140       | 165.333 | 181.2  | 170.3  | 127.4    | 130.8   | 128.05 |
| SonarEW              | <b>8</b>  | 19       | 22       | 26.6  | 24        | 34.3    | 36.433 | 35.3   | 26.4     | 31.2    | 35.64  |
| SpectEW              | 4         | 5        | 6        | 7.65  | 5         | 10.233  | 13.3   | 10.167 | 9.4      | 8.4     | 8.052  |
| Tic-tac-toe          | 9         | 5        | 6        | 5     | <b>4</b>  | 7       | 7      | 7      | 6        | 5.2     | 6.903  |
| Vote                 | <b>2</b>  | 5        | <b>3</b> | 6.6   | 4         | 7.567   | 8.8    | 8.967  | 5.2      | 3.4     | 7.408  |
| WaveformEW           | 15        | 15       | 15       | 20.5  | <b>14</b> | 26.933  | 27.533 | 25.833 | 20.6     | 14.2    | 25.4   |
| WineEW               | <b>2</b>  | 4        | 5        | 5.4   | 4         | 4.533   | 5.867  | 5.933  | 6.4      | 6       | 6.799  |
| Zoo                  | <b>4</b>  | <b>4</b> | 5        | 5.7   | 6         | 5.533   | 5.3    | 7.6    | 5.6      | 6.8     | 6      |
| <b>Average Rank</b>  | 1.8       | 2.05     | 2.55     | 5     | 2.94      | 7.4     | 8.33   | 7.7    | 4.72     | 4.88    | 6.27   |
| <b>Assigned Rank</b> | <b>1</b>  | 2        | 3        | 7     | 4         | 9       | 11     | 10     | 5        | 6       | 8      |

TABLE 8: *p*-values produced by the Wilcoxon rank-sum test for the classification accuracies achieved by the RTHS algorithm, HS algorithm, and RTEA as compared to themselves as well as 10 other state-of-the-art FS methods considered here

|      | RTHS  | HS    | RTEA  | BGA   | BPSO  | BALO  | BGSA  | HSGW  | RSGW  | ASGW  | WOASAT-2 | BGWOPSO | WOA-CM |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|---------|--------|
| RTHS | -     | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 | 0.000 | 0.008 | 0.011 | 0.048 | 0.000    | 0.001   | 0.000  |
| HS   | 0.000 | -     | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000    | 0.000   | 0.000  |
| RTEA | 0.000 | 0.001 | -     | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000    | 0.000   | 0.000  |

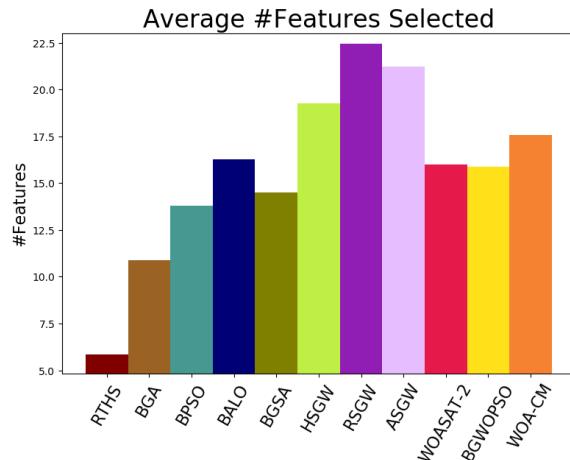


FIGURE 11: Graphical comparison of average number of features selected by the proposed method and 10 state-of-the-art FS methods over the 18 UCI datasets

to the proposed RTHS algorithm are: HSGW, RSGW, and ASGW. Table 9 provides the detailed performance results attained by the proposed FS method as well as these *top – 3* methods in terms of four statistical popular measures such as precision, recall, f-score, and roc\_auc\_score for all the 18 UCI datasets considered in the present work.

## VII. CONCLUSION

In this paper, we have proposed a hybrid meta-heuristic method for FS, named as RTHS, based on a well-known meta-heuristic called HS algorithm and a recently proposed meta-heuristic called RTEA. The proposed RTHS has been applied on 18 standard UCI datasets and compared with 10 state-of-art meta-heuristic and hybrid meta-heuristic FS approaches. The obtained results prove the superiority of RTHS over other methods. Hence, we can say RTHS can be considered as a competent method for solving FS problems. Observing the results meticulously, we can understand that RTEA helps HS algorithm to overcome its limitations in terms of exploration and exploitation (as discussed in section IV). But, there may be some cases, where it may fail to find global optima as per the requirement of the problem, which is in accordance with *No Free Lunch* theorem [74]. At the same time, we need to perform a bit exhaustive experiments to find the ideal value of the parameters used in this algorithm for different problems, which is another shortcoming of the proposed work. As future scope of the work, we can apply the proposed RTHS on other popular and interesting research problems, like facial emotion recognition, musical symbol recognition, handwritten or printed script recognition, etc. RTHS can be applied on high dimensional datasets like gene expression data. It would be interesting to hybridize this with other recently proposed or classical meta-heuristic algorithms.

## References

- [1] Lu Xiong et al. “Multi-feature fusion and selection method for an improved particle swarm optimization”. In: *Journal of Ambient Intelligence and Humanized Computing* (Dec. 2019). DOI: [10.1007/s12652-019-01624-4](https://doi.org/10.1007/s12652-019-01624-4).
- [2] Hossam Faris et al. “Time-varying hierarchical chains of salps with random weight networks for feature selection”. In: *Expert Systems with Applications* 140 (Feb. 2020), p. 112898. DOI: [10.1016/j.eswa.2019.112898](https://doi.org/10.1016/j.eswa.2019.112898).
- [3] Heming Jia et al. “Spotted Hyena Optimization Algorithm With Simulated Annealing for Feature Selection”. In: *IEEE Access* 7 (2019), pp. 71943–71962. DOI: [10.1109/access.2019.2919991](https://doi.org/10.1109/access.2019.2919991).
- [4] Girish Chandrashekhar and Ferat Sahin. “A survey on feature selection methods”. In: *Computers & Electrical Engineering* 40.1 (Jan. 2014), pp. 16–28. DOI: [10.1016/j.compeleceng.2013.11.024](https://doi.org/10.1016/j.compeleceng.2013.11.024).
- [5] Kenji Kira and Larry A. Rendell. “A Practical Approach to Feature Selection”. In: *Machine Learning Proceedings 1992*. 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).
- [6] C. E. Shannon. “A Mathematical Theory of Communication”. In: *Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. DOI: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
- [7] Xiaofei He, Deng Cai, and Partha Niyogi. “Laplacian Score for Feature Selection”. In: *Proceedings of the 18th International Conference on Neural Information Processing Systems*. NIPS’05. Vancouver, British Columbia, Canada: MIT Press, 2005, pp. 507–514.
- [8] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. “Feature selection for text categorization on imbalanced data”. In: *ACM SIGKDD Explorations Newsletter* 6.1 (June 2004), p. 80. DOI: [10.1145/1007730.1007741](https://doi.org/10.1145/1007730.1007741).
- [9] Quanquan Gu, Zhenhui Li, and Jiawei Han. “Generalized Fisher Score for Feature Selection”. In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. UAI’11. Barcelona, Spain: AUAI Press, 2011, pp. 266–273. ISBN: 9780974903972.
- [10] Majdi M. Mafarja and Seyedali Mirjalili. “Hybrid Whale Optimization Algorithm with simulated annealing for feature selection”. In: *Neurocomputing* 260 (Oct. 2017), pp. 302–312. DOI: [10.1016/j.neucom.2017.04.053](https://doi.org/10.1016/j.neucom.2017.04.053).
- [11] Manosij Ghosh et al. “Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods”. In: *Medical & Biological Engineering & Computing* 57.1 (Aug. 2018), pp. 159–176. DOI: [10.1007/s11517-018-1874-4](https://doi.org/10.1007/s11517-018-1874-4).
- [12] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. “Grey Wolf Optimizer”. In: *Advances*

TABLE 9: Comparison of the proposed method with the *top – 3* FS methods in terms of precision, recall, f-score and roc\_auc\_score for 18 UCI datasets

| Dataset      | HSGW         |               |               |          | RSGW          |               |               |               | ASGW          |               |               |               | RTHS          |               |               |               |
|--------------|--------------|---------------|---------------|----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|              | precision    | recall        | fscore        | roc_auc  | precision     | recall        | fscore        | roc_auc       | precision     | recall        | fscore        | roc_auc       | precision     | recall        | fscore        | roc_auc       |
| Breastcancer | <b>1</b>     | <b>1</b>      | <b>1</b>      | <b>1</b> | 0.9552        | 0.9678        | 0.9609        | 0.9925        | 0.98          | 0.9891        | 0.9843        | 0.9981        | <b>1</b>      | <b>1</b>      | <b>1</b>      | <b>1</b>      |
| BreastEW     | 0.9539       | 0.9335        | 0.9422        | 0.9726   | 0.9672        | 0.9573        | 0.9619        | 0.9904        | 0.936         | 0.9315        | 0.9337        | 0.9487        | <b>0.9741</b> | <b>0.9692</b> | <b>0.9716</b> | <b>0.9959</b> |
| CongressEW   | 0.9512       | 0.9512        | 0.9512        | 0.977    | 0.9662        | 0.9604        | 0.9632        | 0.9655        | 0.9714        | <b>0.9815</b> | <b>0.9759</b> | 0.9854        | <b>0.9612</b> | 0.9663        | 0.9636        | <b>0.992</b>  |
| Exactly      | 0.8143       | 0.8086        | 0.8113        | 0.8624   | 0.7779        | 0.7779        | 0.7779        | 0.8463        | <b>0.9006</b> | <b>0.8734</b> | <b>0.8852</b> | <b>0.9704</b> | <b>0.9006</b> | 0.8734        | 0.8852        | <b>0.9704</b> |
| Exactly2     | 0.5505       | 0.5192        | 0.5007        | 0.5914   | <b>0.6241</b> | <b>0.5718</b> | <b>0.5766</b> | <b>0.698</b>  | 0.5609        | 0.534         | 0.5291        | 0.6407        | <b>0.6241</b> | <b>0.5718</b> | <b>0.5766</b> | <b>0.698</b>  |
| HeartEW      | 0.5907       | 0.5917        | 0.5903        | 0.6785   | 0.739         | 0.7417        | 0.7393        | 0.7792        | <b>0.85</b>   | <b>0.85</b>   | <b>0.85</b>   | <b>0.9</b>    | <b>0.85</b>   | <b>0.85</b>   | 0.82          | <b>0.9</b>    |
| IonosphereEW | 0.9018       | 0.78          | 0.8045        | 0.9293   | <b>0.9327</b> | 0.86          | 0.8825        | <b>0.9942</b> | 0.9245        | 0.84          | 0.8639        | 0.9267        | 0.9252        | <b>0.8889</b> | <b>0.9029</b> | 0.948         |
| KrvskpEW     | 0.9646       | 0.9634        | 0.9639        | 0.9925   | 0.9765        | 0.9764        | 0.9765        | <b>0.9965</b> | 0.9704        | 0.97          | 0.9702        | 0.9843        | <b>0.9825</b> | <b>0.9834</b> | <b>0.9828</b> | 0.9895        |
| Lymphography | 0.4234       | 0.4409        | 0.432         | 0.8923   | 0.3946        | 0.4006        | 0.3944        | 0.874         | 0.4039        | 0.4069        | 0.401         | 0.7794        | <b>0.6132</b> | <b>0.668</b>  | <b>0.6403</b> | <b>0.9279</b> |
| M-of-n       | 0.9379       | 0.9329        | 0.9353        | 0.9774   | 0.8527        | 0.8443        | 0.8481        | 0.935         | 0.9922        | 0.9865        | 0.9892        | 0.9996        | <b>1</b>      | <b>1</b>      | <b>1</b>      | <b>1</b>      |
| PenglungEW   | <b>0.881</b> | <b>0.9048</b> | <b>0.8762</b> | 0.9512   | 0.7929        | 0.7857        | 0.7732        | <b>0.998</b>  | 0.7571        | 0.8095        | 0.7651        | 0.964         | <b>0.881</b>  | <b>0.9048</b> | <b>0.8762</b> | 0.9243        |
| SonarEW      | 0.8413       | 0.8249        | 0.8286        | 0.9073   | 0.8612        | 0.8513        | 0.8542        | 0.881         | 0.8333        | 0.8295        | 0.8309        | 0.8936        | <b>0.9273</b> | <b>0.9302</b> | <b>0.9282</b> | <b>0.9611</b> |
| SpectEW      | 0.7778       | 0.7378        | 0.7545        | 0.8076   | 0.7473        | 0.6924        | 0.7126        | <b>0.8763</b> | 0.7976        | <b>0.8171</b> | <b>0.8066</b> | 0.8594        | <b>0.7999</b> | 0.704         | 0.7333        | 0.7378        |
| Tic-tac-toe  | 0.6864       | 0.645         | 0.6515        | 0.747    | 0.8995        | <b>0.8204</b> | <b>0.8433</b> | <b>0.8703</b> | <b>0.9139</b> | 0.806         | 0.8325        | 0.8057        | 0.812         | 0.7223        | 0.7394        | 0.7833        |
| Vote         | 0.951        | 0.943         | 0.9467        | 0.97     | 0.9259        | 0.9459        | 0.9314        | 0.9636        | <b>0.9647</b> | <b>0.9647</b> | <b>0.9647</b> | 0.9959        | <b>0.9647</b> | <b>0.9647</b> | <b>0.9647</b> | <b>0.998</b>  |
| WaveformEW   | 0.811        | 0.8111        | 0.811         | 0.9365   | 0.8202        | 0.8195        | 0.8187        | 0.9508        | <b>0.8367</b> | 0.8361        | 0.836         | 0.951         | 0.8354        | <b>0.8365</b> | <b>0.838</b>  | <b>0.965</b>  |
| WineEW       | 0.9697       | 0.9762        | 0.9718        | 0.9978   | 0.9744        | 0.9762        | 0.9743        | <b>1</b>      | 0.9697        | 0.9762        | 0.9718        | 0.9794        | <b>0.976</b>  | <b>0.976</b>  | <b>0.976</b>  | 0.99          |
| Zoo          | <b>1</b>     | <b>1</b>      | <b>1</b>      | <b>1</b> | 0.9841        | 0.9286        | 0.944         | 0.9605        | <b>1</b>      |

- in *Engineering Software* 69 (Mar. 2014), pp. 46–61. DOI: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [13] B. Chatterjee et al. “Late Acceptance Hill Climbing Based Social Ski Driver Algorithm for Feature Selection”. In: *IEEE Access* 8 (2020), pp. 75393–75408. DOI: [10.1109/ACCESS.2020.2988157](https://doi.org/10.1109/ACCESS.2020.2988157).
- [14] Zong Woo Geem, Joong Hoon Kim, and G.V. Loganathan. “A New Heuristic Optimization Algorithm: Harmony Search”. In: *SIMULATION* 76.2 (Feb. 2001), pp. 60–68. DOI: [10.1177/003754970107600201](https://doi.org/10.1177/003754970107600201).
- [15] He Yichao and Xi-Zhao Wang. “Ring Theory-Based Evolutionary Algorithm and Its Application to D0-1KP”. In: *Applied Soft Computing* 77 (Feb. 2019), pp. 714–722. DOI: [10.1016/j.asoc.2019.01.049](https://doi.org/10.1016/j.asoc.2019.01.049).
- [16] Kushal Kanti Ghosh et al. “Improved Binary Sailfish Optimizer based on Adaptive  $\beta$ -Hill Climbing for Feature Selection”. In: *IEEE Access* (2020), pp. 1–1. DOI: [10.1109/access.2020.2991543](https://doi.org/10.1109/access.2020.2991543).
- [17] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: <http://archive.ics.uci.edu/ml>.
- [18] Michel Gendreau and Jean-Yves Potvin. “Metaheuristics in Combinatorial Optimization”. In: *Annals of Operations Research* 140.1 (Nov. 2005), pp. 189–213. DOI: [10.1007/s10479-005-3971-7](https://doi.org/10.1007/s10479-005-3971-7).
- [19] Iztok Fister Jr. et al. *A Brief Review of Nature-Inspired Algorithms for Optimization*. 2013. arXiv: [1307.4186 \[cs.NE\]](https://arxiv.org/abs/1307.4186).
- [20] Mohamed Abdel-Basset, Laila Abdel-Fatah, and Arun Kumar Sangaiah. “Metaheuristic Algorithms: A Comprehensive Review”. In: *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*. 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).
- [21] Amin Foroughi Nematollahi, Abolfazl Rahiminejad, and Behrooz Vahidi. “A novel meta-heuristic optimization method based on golden ratio in nature”. In: *Soft Computing* 24.2 (Apr. 2019), pp. 1117–1151. DOI: [10.1007/s00500-019-03949-w](https://doi.org/10.1007/s00500-019-03949-w).
- [22] Riccardo Leardi. “Application of genetic algorithm-PLS for feature selection in spectral data sets”. In: *Journal of Chemometrics* 14.5-6 (2000), pp. 643–655. 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).
- [23] Pedram Ghamisi and Jon Atli Benediktsson. “Feature Selection Based on Hybridization of Genetic Algorithm and Particle Swarm Optimization”. In: *IEEE Geoscience and Remote Sensing Letters* 12.2 (Feb. 2015), pp. 309–313. DOI: [10.1109/lgrs.2014.2337320](https://doi.org/10.1109/lgrs.2014.2337320).
- [24] Jinjie Huang, Yunze Cai, and Xiaoming Xu. “A hybrid genetic algorithm for feature selection wrapper based on mutual information”. In: *Pattern Recognition Letters* 28.13 (Oct. 2007), pp. 1825–1844. DOI: [10.1016/j.patrec.2007.05.011](https://doi.org/10.1016/j.patrec.2007.05.011).
- [25] Stjepan Oreski and Goran Oreski. “Genetic algorithm-based heuristic for feature selection in credit risk assessment”. In: *Expert Systems with Applications* 41.4 (Mar. 2014), pp. 2052–2064. DOI: [10.1016/j.eswa.2013.09.004](https://doi.org/10.1016/j.eswa.2013.09.004).
- [26] Basabi Chakraborty. “Feature subset selection by particle swarm optimization with fuzzy fitness function”. In: *2008 3rd International Conference on Intelligent System and Knowledge Engineering*. IEEE, Nov. 2008. DOI: [10.1109/iske.2008.4731082](https://doi.org/10.1109/iske.2008.4731082).
- [27] Sangwook Lee et al. “Modified binary particle swarm optimization”. In: *Progress in Natural Science* 18.9 (Sept. 2008), pp. 1161–1166. DOI: [10.1016/j.pnsc.2008.03.018](https://doi.org/10.1016/j.pnsc.2008.03.018).
- [28] Xiangyang Wang et al. “Feature selection based on rough sets and particle swarm optimization”. In: *Pattern Recognition Letters* 28.4 (Mar. 2007), pp. 459–471. DOI: [10.1016/j.patrec.2006.09.003](https://doi.org/10.1016/j.patrec.2006.09.003).
- [29] Liangjun Ke, Zuren Feng, and Zhigang Ren. “An efficient ant colony optimization approach to attribute reduction in rough set theory”. In: *Pattern Recognition*

- Letters* 29.9 (July 2008), pp. 1351–1357. DOI: [10.1016/j.patrec.2008.02.006](https://doi.org/10.1016/j.patrec.2008.02.006).
- [30] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi. “GSA: A Gravitational Search Algorithm”. In: *Information Sciences* 179.13 (June 2009), pp. 2232–2248. DOI: [10.1016/j.ins.2009.03.004](https://doi.org/10.1016/j.ins.2009.03.004).
- [31] Lawrence Davis. “Handbook of genetic algorithms”. In: (1991).
- [32] W.Daniel Hillis. “Co-evolving parasites improve simulated evolution as an optimization procedure”. In: *Physica D: Nonlinear Phenomena* 42.1-3 (June 1990), pp. 228–234. DOI: [10.1016/0167-2789\(90\)90076-2](https://doi.org/10.1016/0167-2789(90)90076-2).
- [33] Rainer Storn and Kenneth Price. In: *Journal of Global Optimization* 11.4 (1997), pp. 341–359. DOI: [10.1023/a:1008202821328](https://doi.org/10.1023/a:1008202821328).
- [34] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992. ISBN: 0262111705.
- [35] David Fogel. “Artificial Intelligence through Simulated Evolution”. In: *Evolutionary Computation: The Fossil Record* (Nov. 2009), pp. 227–296. DOI: [10.1109/9780470544600.ch7](https://doi.org/10.1109/9780470544600.ch7).
- [36] D. Simon. “Biogeography-Based Optimization”. In: *IEEE Transactions on Evolutionary Computation* 12.6 (Dec. 2008), pp. 702–713. DOI: [10.1109/tevc.2008.919004](https://doi.org/10.1109/tevc.2008.919004).
- [37] Hamid Salimi. “Stochastic Fractal Search: A powerful metaheuristic algorithm”. In: *Knowledge-Based Systems* 75 (Feb. 2015), pp. 1–18. DOI: [10.1016/j.knosys.2014.07.025](https://doi.org/10.1016/j.knosys.2014.07.025).
- [38] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN’95 - International Conference on Neural Networks*. Vol. 4. Nov. 1995, 1942–1948 vol.4. DOI: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [39] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. “Ant colony optimization”. In: *IEEE Computational Intelligence Magazine* 1.4 (Nov. 2006), pp. 28–39. DOI: [10.1109/mci.2006.329691](https://doi.org/10.1109/mci.2006.329691).
- [40] Seyedali Mirjalili. “The Ant Lion Optimizer”. In: *Advances in Engineering Software* 83 (May 2015), pp. 80–98. DOI: [10.1016/j.advengsoft.2015.01.010](https://doi.org/10.1016/j.advengsoft.2015.01.010).
- [41] Muzaffar Eusuff, Kevin Lansey, and Fayzul Pasha. “Shuffled frog-leaping algorithm: a memetic metaheuristic for discrete optimization”. In: *Engineering Optimization* 38.2 (Mar. 2006), pp. 129–154. DOI: [10.1080/03052150500384759](https://doi.org/10.1080/03052150500384759).
- [42] K. M. Passino. “Biomimicry of bacterial foraging for distributed optimization and control”. In: *IEEE Control Systems Magazine* 22.3 (June 2002), pp. 52–67. ISSN: 1941-000X. DOI: [10.1109/MCS.2002.1004010](https://doi.org/10.1109/MCS.2002.1004010).
- [43] Dervis Karaboga and Bahriye Basturk. “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”. In: *Journal of Global Optimization* 39.3 (Apr. 2007), pp. 459–471. DOI: [10.1007/s10898-007-9149-x](https://doi.org/10.1007/s10898-007-9149-x).
- [44] Xin-She Yang. “Firefly Algorithms for Multimodal Optimization”. In: *Stochastic Algorithms: Foundations and Applications*. Springer Berlin Heidelberg, 2009, pp. 169–178. DOI: [10.1007/978-3-642-04944-6\\_14](https://doi.org/10.1007/978-3-642-04944-6_14).
- [45] A.S. Joshi et al. “Cuckoo Search Optimization- A Review”. In: *Materials Today: Proceedings* 4.8 (2017), pp. 7262–7269. DOI: [10.1016/j.matpr.2017.07.055](https://doi.org/10.1016/j.matpr.2017.07.055).
- [46] Seyedali Mirjalili and Andrew Lewis. “The Whale Optimization Algorithm”. In: *Advances in Engineering Software* 95 (May 2016), pp. 51–67. DOI: [10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008).
- [47] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. “Grasshopper Optimisation Algorithm: Theory and application”. In: *Advances in Engineering Software* 105 (Mar. 2017), pp. 30–47. DOI: [10.1016/j.advengsoft.2017.01.004](https://doi.org/10.1016/j.advengsoft.2017.01.004).
- [48] A. Kaveh and Neda Farhoudi. “A new optimization method: Dolphin echolocation”. In: *Advances in Engineering Software* 59 (May 2013), pp. 53–70. DOI: [10.1016/j.advengsoft.2013.03.004](https://doi.org/10.1016/j.advengsoft.2013.03.004).
- [49] Mohit Jain, Vijander Singh, and Asha Rani. “A novel nature-inspired algorithm for optimization: Squirrel search algorithm”. In: *Swarm and Evolutionary Computation* 44 (Feb. 2019), pp. 148–175. DOI: [10.1016/j.swevo.2018.02.013](https://doi.org/10.1016/j.swevo.2018.02.013).
- [50] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (May 1983), pp. 671–680. DOI: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671).
- [51] Peter J. M. van Laarhoven and Emile H. L. Aarts. “Simulated annealing”. In: *Simulated Annealing: Theory and Applications*. Springer Netherlands, 1987, pp. 7–15. DOI: [10.1007/978-94-015-7744-1\\_2](https://doi.org/10.1007/978-94-015-7744-1_2).
- [52] Tamás Vicsek et al. “Novel Type of Phase Transition in a System of Self-Driven Particles”. In: *Physical Review Letters* 75.6 (Aug. 1995), pp. 1226–1229. DOI: [10.1103/physrevlett.75.1226](https://doi.org/10.1103/physrevlett.75.1226).
- [53] Abdolreza Hatamlou. “Black hole: A new heuristic optimization approach for data clustering”. In: *Information Sciences* 222 (Feb. 2013), pp. 175–184. DOI: [10.1016/j.ins.2012.08.023](https://doi.org/10.1016/j.ins.2012.08.023).
- [54] A. Kaveh and S. Talatahari. “A novel heuristic optimization method: charged system search”. In: *Acta Mechanica* 213.3-4 (Jan. 2010), pp. 267–289. DOI: [10.1007/s00707-009-0270-4](https://doi.org/10.1007/s00707-009-0270-4).
- [55] Seyedali Mirjalili. “SCA: A Sine Cosine Algorithm for solving optimization problems”. In: *Knowledge-Based Systems* 96 (Mar. 2016), pp. 120–133. DOI: [10.1016/j.knosys.2015.12.022](https://doi.org/10.1016/j.knosys.2015.12.022).
- [56] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Abdolreza Hatamlou. “Multi-Verse Optimizer: a nature-inspired algorithm for global optimization”. In: *Neural Computing and Applications* 27.2 (Mar. 2015), pp. 495–513. DOI: [10.1007/s00521-015-1870-7](https://doi.org/10.1007/s00521-015-1870-7).

- [57] R.V. Rao, V.J. Savsani, and D.P. Vakharia. “Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems”. In: *Computer-Aided Design* 43.3 (Mar. 2011), pp. 303–315. DOI: [10.1016/j.cad.2010.12.015](https://doi.org/10.1016/j.cad.2010.12.015).
- [58] Esmaeil Atashpaz-Gargari and Caro Lucas. “Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition”. In: *2007 IEEE Congress on Evolutionary Computation*. IEEE, Sept. 2007. DOI: [10.1109/cec.2007.4425083](https://doi.org/10.1109/cec.2007.4425083).
- [59] T. Ray and K.M. Liew. “Society and civilization: an optimization algorithm based on the simulation of social behavior”. In: *IEEE Transactions on Evolutionary Computation* 7.4 (Aug. 2003), pp. 386–396. DOI: [10.1109/tevc.2003.814902](https://doi.org/10.1109/tevc.2003.814902).
- [60] Ali Husseinzadeh Kashan. “League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships”. In: *Applied Soft Computing* 16 (Mar. 2014), pp. 171–200. DOI: [10.1016/j.asoc.2013.12.005](https://doi.org/10.1016/j.asoc.2013.12.005).
- [61] A. Kaveh. “Tug of War Optimization”. In: *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Springer International Publishing, 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).
- [62] Reza Moghdani and Khodakaram Salimifard. “Volleyball Premier League Algorithm”. In: *Applied Soft Computing* 64 (Mar. 2018), pp. 161–185. DOI: [10.1016/j.asoc.2017.11.043](https://doi.org/10.1016/j.asoc.2017.11.043).
- [63] E.-G. Talbi. “Taxonomy of Hybrid Metaheuristics”. In: *Journal of Heuristics* 8.5 (2002), pp. 541–564. DOI: [10.1023/a:1016540724870](https://doi.org/10.1023/a:1016540724870).
- [64] Il-Seok Oh, Jin-Seon Lee, and Byung-Ro Moon. “Hybrid genetic algorithms for feature selection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.11 (Nov. 2004), pp. 1424–1437. DOI: [10.1109/tpami.2004.105](https://doi.org/10.1109/tpami.2004.105).
- [65] Olivier C. Martin and Steve W. Otto. “Combining simulated annealing with local search heuristics”. In: *Annals of Operations Research* 63.1 (Feb. 1996), pp. 57–75. DOI: [10.1007/bf02601639](https://doi.org/10.1007/bf02601639).
- [66] Manosij Ghosh et al. “Feature selection for facial emotion recognition using late hill-climbing based memetic algorithm”. In: *Multimedia Tools and Applications* 78.18 (June 2019), pp. 25753–25779. DOI: [10.1007/s11042-019-07811-x](https://doi.org/10.1007/s11042-019-07811-x).
- [67] Majdi Mafarja and Salwani Abdullah. “Investigating memetic algorithm in solving rough set attribute reduction”. In: *International Journal of Computer Applications in Technology* 48.3 (2013), p. 195. DOI: [10.1504/ijcat.2013.056915](https://doi.org/10.1504/ijcat.2013.056915).
- [68] Mohammad Tubishat et al. “Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection”. In: *Expert Systems with Applications* 145 (May 2020), p. 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”. In: *WSEAS Trans. Inf. Sci. Appl* 9.11 (2012), pp. 340–349.
- [70] Qasem Al-Tashi et al. “Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection”. In: *IEEE Access* 7 (2019), pp. 39496–39508. DOI: [10.1109/access.2019.2906757](https://doi.org/10.1109/access.2019.2906757).
- [71] Xinsheng Lai and Mingyi Zhang. “An efficient ensemble of GA and PSO for real function optimization”. In: *2009 2nd IEEE International Conference on Computer Science and Information Technology*. IEEE, 2009. DOI: [10.1109/iccsit.2009.5234780](https://doi.org/10.1109/iccsit.2009.5234780).
- [72] Mohammad Ehsan Basiri and Shahla Nemati. “A novel hybrid ACO-GA algorithm for text feature selection”. In: *2009 IEEE Congress on Evolutionary Computation*. IEEE, May 2009. DOI: [10.1109/cec.2009.4983263](https://doi.org/10.1109/cec.2009.4983263).
- [73] Ezgi Zorarpacı and Selma Ayşe Özal. “A hybrid approach of differential evolution and artificial bee colony for feature selection”. In: *Expert Systems with Applications* 62 (Nov. 2016), pp. 91–103. DOI: [10.1016/j.eswa.2016.06.004](https://doi.org/10.1016/j.eswa.2016.06.004).
- [74] D.H. Wolpert and W.G. Macready. “No free lunch theorems for optimization”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (Apr. 1997), pp. 67–82. DOI: [10.1109/4235.585893](https://doi.org/10.1109/4235.585893).
- [75] Xin-She Yang. “Harmony Search as a Metaheuristic Algorithm”. In: *Music-Inspired Harmony Search Algorithm*. Springer Berlin Heidelberg, pp. 1–14. DOI: [10.1007/978-3-642-00185-7\\_1](https://doi.org/10.1007/978-3-642-00185-7_1).
- [76] M. Mahdavi, M. Fesanghary, and E. Damangir. “An improved harmony search algorithm for solving optimization problems”. In: *Applied Mathematics and Computation* 188.2 (May 2007), pp. 1567–1579. DOI: [10.1016/j.amc.2006.11.033](https://doi.org/10.1016/j.amc.2006.11.033).
- [77] Kang Seok Lee and Zong Woo Geem. “A new metaheuristic algorithm for continuous engineering optimization: harmony search theory and practice”. In: *Computer Methods in Applied Mechanics and Engineering* 194.36–38 (Sept. 2005), pp. 3902–3933. DOI: [10.1016/j.cma.2004.09.007](https://doi.org/10.1016/j.cma.2004.09.007).
- [78] Mahamed G.H. Omran and Mehrdad Mahdavi. “Global-best harmony search”. In: *Applied Mathematics and Computation* 198.2 (May 2008), pp. 643–656. DOI: [10.1016/j.amc.2007.09.004](https://doi.org/10.1016/j.amc.2007.09.004).
- [79] Zong Woo Geem. “Optimal cost design of water distribution networks using harmony search”. In: *Engineering Optimization* 38.3 (Apr. 2006), pp. 259–277. DOI: [10.1080/03052150500467430](https://doi.org/10.1080/03052150500467430).
- [80] John B Fraleigh. *A first course in abstract algebra*. Pearson Education India, 2003.
- [81] N. S. Altman. “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression”. In: *The Amer-*

- ican Statistician* 46.3 (Aug. 1992), p. 175. DOI: [10.2307/2685209](https://doi.org/10.2307/2685209).
- [82] Andy Liaw, Matthew Wiener, et al. “Classification and regression by randomForest”. In: *R news* 2.3 (2002), pp. 18–22.
- [83] Irina Rish et al. “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.
- [84] Guido van Rossum and Fred L. Drake. *The Python Language Reference Manual*. Network Theory Ltd., 2011. ISBN: 1906966141.
- [85] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [86] Majdi Mafarja et al. “Efficient Hybrid Nature-Inspired Binary Optimizers for Feature Selection”. In: *Cognitive Computation* 12.1 (July 2019), pp. 150–175. DOI: [10.1007/s12559-019-09668-6](https://doi.org/10.1007/s12559-019-09668-6).
- [87] E. Emamy, Hossam M. Zawbaa, and Aboul Ella Hassanien. “Binary grey wolf optimization approaches for feature selection”. In: *Neurocomputing* 172 (Jan. 2016), pp. 371–381. DOI: [10.1016/j.neucom.2015.06.083](https://doi.org/10.1016/j.neucom.2015.06.083).
- [88] Majdi Mafarja and Seyedali Mirjalili. “Whale optimization approaches for wrapper feature selection”. In: *Applied Soft Computing* 62 (Jan. 2018), pp. 441–453. DOI: [10.1016/j.asoc.2017.11.006](https://doi.org/10.1016/j.asoc.2017.11.006).
- [89] Frank Wilcoxon. “Individual Comparisons by Ranking Methods”. In: *Springer Series in Statistics*. Springer New York, 1992, pp. 196–202. DOI: [10.1007/978-1-4612-4380-9\\_16](https://doi.org/10.1007/978-1-4612-4380-9_16).



PAWAN KUMAR SINGH (email: [pksingh.it@jadavpuruniversity.in](mailto:pksingh.it@jadavpuruniversity.in)) received his B.Tech degree in Information Technology from West Bengal University of Technology in 2010. He received his M.Tech in Computer Science and Engineering and Ph.D. (Engineering) degrees from Jadavpur University (J.U.) in 2013 and 2018 respectively. He also received RUSA 2.0 fellowship for pursuing his post-doctoral research in J.U. in 2019. He is currently working as an Assistant Professor in the Department of Information Technology in J.U. He has published more than 50 research papers in peer-reviewed journals and international conferences. His areas of current research interest are Computer Vision, Pattern Recognition, Handwritten Document Analysis, Image Processing, Machine Learning and Artificial Intelligence. He is a member of the IEEE (U.S.A.), 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 (email: [geem@gachon.ac.kr](mailto:geem@gachon.ac.kr)) is an Associate Professor in Department of Energy IT at Gachon University, South Korea. He has obtained B.Eng. in Chung-Ang University, Ph.D. in Korea University, and M.Sc. in Johns Hopkins University, and researched at Virginia Tech, University of Maryland - College Park, and Johns Hopkins University. He invented a music-inspired optimization algorithm, Harmony Search, which has been applied to various scientific and engineering problems. His research interest includes phenomenon-mimicking algorithms and their applications to energy, environment and water fields. He has served for various journals as an editor (Associate Editor for Engineering Optimization; Guest Editor for Swarm & Evolutionary Computation, Int. Journal of Bio-Inspired Computation, Journal of Applied Mathematics, Applied Sciences, and Sustainability).



SHAMEEM AHMED (email: [shameemahmed20-apr2000@gmail.com](mailto:shameemahmed20-apr2000@gmail.com)) is currently a sophomore undergraduate student of Computer Science and Engineering at Jadavpur University, Kolkata, India. His areas of interest include Machine Learning, Optimization, and Image Processing.



RAM SARKAR (email: [rsarkar@ieee.org](mailto:rsarkar@ieee.org)) received his B.Tech degree in Computer Science and Engineering from University of Calcutta in 2003. He received his M.E. degree in Computer Science and Engineering and PhD (Engineering) degree from Jadavpur University in 2005 and 2012 respectively. He joined in the department of Computer Science and Engineering of Jadavpur University as an Assistant Professor in 2008, where he is now working as an Associate Professor. He received Fulbright-Nehru Fellowship (USIEF) for post-doctoral research in University of Maryland, College Park, USA in 2014-15. His areas of current research interest are Image Processing, Pattern Recognition, Machine Learning, and Bioinformatics. He is a senior member of the IEEE, U.S.A.



KUSHAL KANTI GHOSH (email: [kushalkanti1999@gmail.com](mailto:kushalkanti1999@gmail.com)) is currently a senior year undergraduate student of Computer Science and Engineering at Jadavpur University, Kolkata, India. His areas of interest include Machine Learning, Optimization, Game Theory, and Image Processing.