AN UNSUPERVISED FEATURE SELECTION BASED ON CROW SEARCH ALGORITHM AND GREY WOLF OPTIMIZATION ALGORITHM FOR DNA-MICROARRAY DATA

[1]Artee Abudayor

[1]Computer Engineering Department, Engineering Faculty, Erciyes University
Talas, Kayseri, Turkey
[1]jamendat@gmail.com

[2]Özkan Ufuk Nalbantoğlu

[2]Computer Engineering Department, Engineering Faculty, Erciyes University
Talas, Kayseri, Turkey
[2]nalbantoglu@erciyes.edu.tr

Abstract

High-dimensional data encounter the curse of dimensionality in classification tasks, particularly in DNA-microarray technology, due to a lack of statistical power, complicating biomarker discovery. DNA-microarray technology expresses gene profiles, aids in the identification of diseases, and provides drugs to treat patients. Feature selection successfully reduces execution time and increases classification accuracy in preprocessing data by discarding irrelevant and redundant features for which technology has large dimensions and few samples in classification tasks. Most of the research on feature selection problems is considered in supervised feature selection, which is known as the label class in the classification process, but some of them cannot access the label class. Therefore, this study proposed an unsupervised filter-based feature selection algorithm based on a proposed algorithm based on CSA and GWO with a hybrid technique called UHCSAGWO. In the details of our proposed algorithm, we proposed three crucial mechanisms to find the significant feature subsets without using any learning model in the process of evaluating them, such as a pool for finding the top candidate features, a local search, and a new fitness function. To remove the irrelevant and redundant features in the local search step based on a pool to store top features and obtain a significant subset of the selected features. Then, to evaluate the subset of the selected feature by utilizing a new fitness function that is examined by relevant, redundant information is discarded without any classifiers. The performance of the proposed algorithm was compared to that of state-of-the-art feature selection algorithms using different classifiers on the meta-heuristic algorithms. The experimental results prove that the proposed algorithms can significantly reduce computational time and achieve better performance than others in classification error rate and other well-known criteria, such as precision and recall.

***Keywords*: Filter-based feature selection, DNA-Microarray, Unsupervised-based, Crow search algorithm (CSA), Grey-wolf optimizer (GWO)**

1. Introduction

As global cancer statistics have shown, the number of cancer cases is increasing, making cancer the most significant public health challenge of the twenty-first century, particularly since lung cancers. The analysis or diagnosis of the patient's illness is intended to expedite treatment and reduce the risk of disease, which is challenging in high-dimensional data problems. Moreover, the problems in big data areas have been of great interest in many fields, such as optimization problems, bioinformatics, data mining, and image processing, as shown in [1], [2]. DNA microarray-based profiling of gene expressions has been a mainstream technology in the disease/drug research as well as clinical grade monitoring in bioinformatics areas. The ability of the technology is to provide the expression profile of thousands of genes simultaneously at a high-throughput fashion [3]. In addition, the nature of the technology has large dimensions and few samples, which means it confronts the curse

DOI : 10.21817/indjcse/2023/v14i1/231402049

of dimensionality in classification tasks since lack of statistical power complicates the discovery of biomarkers. To improve the classification of the diseases as well as discovering disease-associated biomarkers, careful gene selection procedures should be employed. In addition, the feature selection [4] process, is the gene selection determining significant feature subsets without harming the classification performance attained by the entire data.

Feature selection (FS) is a dimensionality reduction technique for prediction or classification by removing redundant and irrelevant features. Additionally, it also encourages reduced execution time and increased classification accuracy. Feature selection techniques have been successfully applied to various expert system fields, including text mining, bioinformatics, and industrial applications. We can divide the categories of the feature selection in terms of the accessibility of class label data, the search strategy, and the relationship with the classifier algorithm. The accessibility of class labels leads to supervised, unsupervised, and semisupervised feature selections. In terms of search strategies, they can utilize random, heuristic, and complete searches to find the significant feature subsets. Lastly, the categories of feature selection relative to the classifier algorithm are divided into four main categories based on different evaluations: the filter, wrapper, embedded, and hybrid approaches.

To address the hard problem of subset selection, many researchers employ meta-heuristic algorithms to solve feature selection problems. The meta-heuristic algorithms are successful at solving combinatorial optimization problems such as binary programming, knapsack problem, etc. The algorithms can find sub-optimal solutions to complex problems in feasible since they utilize heuristic methods. Such practical algorithms in feature selection problems can be divided into three categories: filter, wrapper, and hybrid methods [5]. Firstly, filter methods are independently learning algorithms and have only one iteration. Filter methods are usually given a score for each feature or group of features. Therefore, it is easy to rank the features and select the best features among them or remove some features below a threshold, such as ant colony optimization (ACO) [6]. Secondly, wrapper methods utilize significant subsets of selected features to be evaluated by dependent learning algorithms by iteratively producing different candidate feature subsets in some strategies. Then, it uses a classifier algorithm to calculate the corresponding classification accuracy, such as GWO [4], ALO [7], and WOA [8]. Lastly, the combination of various methods is called the hybrid method, such as MFO with mRMR [9].

In this study, we focus on crow search algorithm (CSA) as a relatively new population-based meta-heuristic optimization method. CSA was proposed by Askazadeh et al. in 2016 [10]. The algorithm imitates the crows' behavior, in which a crow individual endeavors to hide the place for storing their food from other crows, who could follow them to steal their food. In addition, the strength of CSA includes a few control parameters and easy implementation. CSA has been widely applied to scientific research and real-world optimization problems such as feature selection problems. Moreover, many researchers have employed the CSA algorithm to solve feature selection problems for classification tasks in different domains, such as diseases [11], documents [12], big-data [13], and UCI standard datasets [14], [15]. Based on the reported studies, most researchers considered the wrapper approach to applying the algorithm to feature selection problems, but in some of them, it is not possible to access the information labeled that is unsupervised. Therefore, this study aimed to fill the research gap by addressing the lack of an unsupervised gene selection method in classifications for the DNA microarray research area.

We propose an unsupervised filter feature selection algorithm referencing a previously defined hybrid method based on CSA and GWO called HCSAGWO [16]. We call our unsupervised version as UHCSAGWO. In the details of our proposed algorithm, we proposed three crucial mechanisms to find the significant feature subsets without using any learning model in the process of evaluating them, such as a pool for finding the top candidate features, a local search, and a novel fitness function. A pool for finding the top candidate features prepares for the proposed local search by removing irrelevant features from the original dataset and can decrease redundancy without the help of any classifiers. Then, local search is a step to transform the search space for feature selection that can reduce the complexity cost, especially on DNA-microarray datasets. In addition, the local search evaluation utilizes the correlation of the features by considering the redundancy between the selected features and the pool that is proposed more efficiently. Lastly, it evaluates the subset of the selected feature by utilizing the fitness function that achieves an optimal solution by reducing the complexity of execution. The main advantage of our approach is offering high performance and finding significant feature subsets in DNA microarray while maintaining decent computational complexity.

The organization of this article is as follows: Section 2 presents the background information on meta-heuristic algorithms and filter-based feature selection algorithms, focusing on their inspiration and mathematical model. The proposed supervised-filter-based feature selection algorithm is presented in Section 3, whereas the experimental settings and the results on five-DNA microarray datasets for feature selection problems are discussed in Section 4. Finally, conclusions and future work are indicated in Section 5.

2. Related Works

The unsupervised filter-based feature selection method is a technique that gives scores to each feature or feature subset to find the most significant features by using relevant and non-redundant features, which is an independent learning algorithm in only one iteration process. The univariate approaches usually rank the genes or features by scoring each variable. Another multivariate approach gives scores for a set group of features. Moreover, these approaches are intended to select the best gene between them or remove any variables below a threshold of the specified score. Furthermore, many researchers attempted to employ metaheuristic algorithms to solve feature selection problems because they successfully find optimal solutions in a reasonable amount of time, even when the problems are complex, especially unsupervised-based feature selection, such as differential evolution (DE) algorithm combined with fuzzy rough set theory, known as DEFRS [17], to evaluate the performance of DEFRS that had used their proposed fitness function on different datasets, namely ionosphere, wbcd, sonar, hill, colon datasets, and so on. The fitness function aims to maximize the feature subset for the target and minimize the number of the selected features. An unsupervised gene selection ant colony optimization method is called MGSACO [6]. The MGSACO aims to minimize the redundancy between genes and maximize the relevance of genes for validating microarray datasets. Dual Regularized Unsupervised Feature Selection Based on Matrix Factorization and Minimum Redundancy (DR-FS-MFMR) [18]. The steps of the algorithm represented features in the feature weight matrix and correlation information that dictated the selection or discarding of the features, and the experimentation was tested on nine gene expression datasets. To solve a diverse grouping problem for gene expression RNA-Seq data that unbalanced class problems in multi-classification cancer, a grouping genetic algorithm (GGA) were proposed [19], in which the algorithm was combined with an Extreme Learning Machine (ELM) algorithm into the fitness function for evaluating the selected feature subsets. Biased random-key genetic algorithms, UFSBRKGA [20], were proposed as algorithms that utilized k-means to cluster features with different methods, namely laplacian scores and variance thresholding for feature selection, unsupervised discriminative feature selection, to achieve selected feature subsets. The proposed algorithm is able to achieve findings the significant features without noisy and missing data. A particle swarm optimization algorithm with genetic operators, H-FSPSOTC, was proposed in [21] for grouping text documents by k-means clustering. The experiment results were validated on eight text datasets with variant characteristics, which were obtained from the Laboratory of Computational Intelligence. A binary bat algorithm for filter-based feature selection was proposed for an information retrieval system that employed the sum of squared errors as the fitness function to weight the feature subsets [22]. The experimental results reveal that the proposed algorithm outperformed the genetic algorithm, bat algorithm, and ant colony optimization.

3. Methods

3.1. Meta-Heuristic Algorithms

Two well-known algorithms were used in this study, namely crow search algorithm (CSA) and grey-wolf Optimizer Algorithm (GWO).

In 2016, Askazadeh et al. proposed CSA by imitating the crows' behavior such that a crow follows other crows to steal their food from a hiding place [10]. To protect their food, the motion of each individual crow can be divided into two: firstly, an individual crow can find the other hiding place by following them. Finally, if a crow is aware that it is being followed by other crows, the crow may deceive the other crows in order to protect its location.

Mirjalili proposed GWO [23] in 2014 by imitating grey-wolf behaviors such as social leadership and hunting; social leadership is divided into four classes: alpha (α), beta (β), delta (δ), and omega (ω). In addition, three-leader wolves, α, β, and δ, are important to consider in this algorithm, whereas the last group, ω represents the rest of the candidate solutions. Consequently, this algorithm can be divided into three steps: encircling, hunting, and attacking.

3.2 Filter-based Feature Selection

The filter method is a technique that gives scores to each feature or feature subset to find the most significant features by using relevant and non-redundant features. It is an independent learning approach executed in a single shot. Moreover, the filter methods can be divided into two approaches: the univariate approach usually ranks the genes or features by scoring each variable. Another multivariate approach is to give scores for a set of features by selecting the best gene among them or by removing any variable below the threshold of the specified score.

- **ReliefF** is an extension on Relief algorithm which is inherently limited to two-class problems in continuous and discrete variables. The algorithm considers features based on the nearest neighbors: one to find the same class, known as the nearest hit H, and the other to find a different class, known as the nearest miss M [24]. X_i, M, and H values are used to update the quality estimate of all features. The advantage of ReliefF is that it is more robust than Relief and can manipulate incomplete and noisy data, as can be calculated in Eq. (1).

$$SC(f_i) = \frac{1}{p}\sum_{t=1}^{p}\left(-\frac{1}{m_{x_t}}\sum_{x_j\in NH(x_t)} d(f_{t,i}-f_{j,i}) + \sum_{x_j\in NM(x_t)} \frac{1}{m_{x_t}}\frac{p(y)}{1-p(y_{x_t})}\sum_{x_j\in NM(x_t)} d(f_{t,i}-f_{j,i})\right) \quad (1)$$

Where y_{x_t} and P(y) represent the class label of the instance x_t and the probability of an instance being from the y (the label class), respectively. NH(x), NM (x,y) present a set of nearest points to x with the same class as x and a different class (y), respectively. The sizes of the sets NH (x) and NM (x,y) are denoted by m_{x_t}, m_{x_t}; y, respectively.

- **Mutual Information (MI):** estimates the shared knowledge between a random variable Y (feature or class label) and another variable X, and information of Y may reduce the uncertainty of X [25], as shown in Eq. (2).

$$I(X;Y) = H(X) - H(X|Y) \quad (2)$$

Where H(X) is the Shannon entropy of variable X, which represents the uncertainty of X. H(X|Y) is the conditional entropy of X given Y, and it measures how much uncertainty is left in X when Y is introduced.

- **Pearson correlation coefficient (PCC):** PCC measures the linear correlation between two variables, feature X and class Y, by attaining a value between +1 and -1. If the value is 1,0,-1, it means there is a total positive correlation, no correlation, and a total negative correlation. Moreover, the algorithm is widely used as a measure of the degree of linear dependence between two variables. The equation of PCC is shown in Eq. (3)

$$r = r_{xy} = \frac{n\sum x_i y_i - \sum x_i y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (3)$$

4. The Proposed Algorithm

In this section, we propose an unsupervised method that utilizes the advantages of CSA and GWO called HCSAGWO [16], for global optimization problems to manipulate high-dimensional data. We utilized HCSAGWO for unsupervised filter-based feature selection methods, which are called UHCSAGWO.In addition, we can divide UHCSAGWO into 2 stages, including the initialization stage and a gene selection stage, as described in Figure 1. In the initialization stage, we prepare a pool to store the top features by employing 2 well-known unsupervised filter-based feature selections, namely reliefF and PCC, for finding the group of features that are significantly relevant features and removing irrelevant ones, as shown in Algorithm 1.

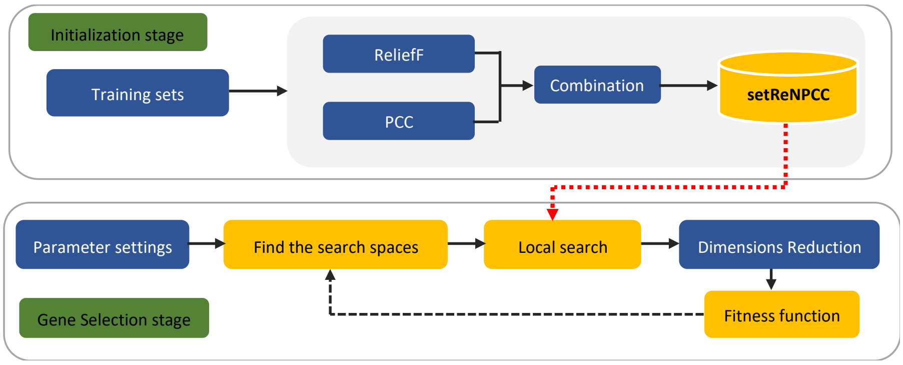
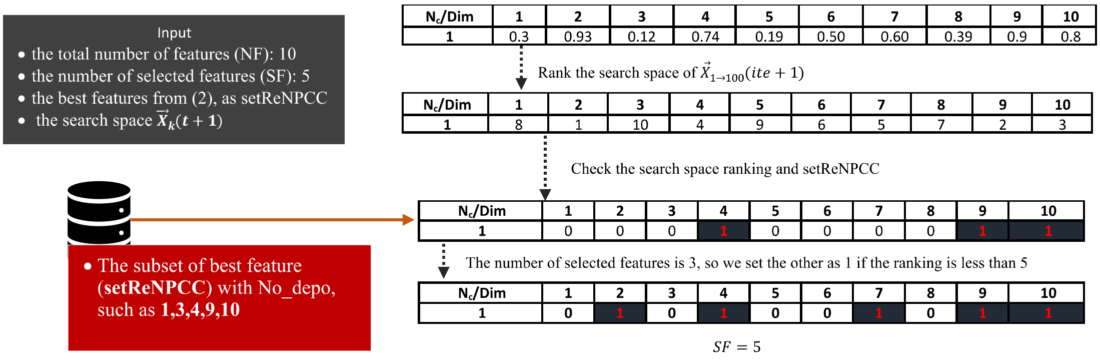


Figure 1. The overview of UHCSAGWO algorithm

As shown in Algorithm 1, we store a subset of top features with the size of depo (No_{depo}) as follows: if the total number of features in the dataset is greater than 10,000, No_{depo} is 400; otherwise, it is 200. Then, it evaluates all features in the dataset by utilizing PCC and ReliefF, giving scores and ranking each feature. Finally, subsetReNPCC is the group of top feature subsets that combine features by ranking them with a size of No$_{depo}$ as is the range between 1 and No$_{depo.}$

```
Algorithm 1: The pool for storing the top feature algorithm
Input: Training sets
Output: the subsetReNPCC
Evaluate features by utilizing PCC algorithm and ranking each feature, as $Rank_{PCC}$
Evaluate features by utilizing ReliefF algorithm and ranking each feature, as $Rank_{ReliefF}$
NF = the size of the training sets
$No_{depo} = 0$
// the size of the pool for storing the top features
if NF > 10000
    $No_{depo} = 400$
else
    $No_{depo} = 200$
end If
$Depo_{PCC} = Rank_{PCC}(1: No_{depo})$
$Depo_{ReliefF} = Rank_{ReliefF}(1: No_{depo})$
subsetReNPCC = Merge ($Depo_{PCC}$ $Depo_{ReliefF}$)
Return subsetReNPCC
```

The gene selection stage applies our proposed algorithm UHCSAGWO, an iterative process in a random search as proposed. At each iteration, our proposed algorithm is defined to random AP of each iteration by selection exploration and exploitation phases. In the exploration phase, we applied CSA algorithm that improved by employing inertia weight to aid the original one. In the exploration phase, GWO successfully finds a new position improving the proposed solutions further. Moreover, the algorithm selects feature subsets by utilizing a proposed local search to convert a search space into a binary vector, as 0 and 1 are meaning that a feature or a gene was unselected and selected, respectively. This step continues until each flock of crows selects gene subsets.



Input
- the total number of features (NF): 10
- the number of selected features (SF): 5
- the best features from (2), as setReNPCC
- the search space $\vec{X}_k(t+1)$

N$_c$/Dim	1	2	3	4	5	6	7	8	9	10
1	0.3	0.93	0.12	0.74	0.19	0.50	0.60	0.39	0.9	0.8

Rank the search space of $\vec{X}_{1\to100}(ite+1)$

N$_c$/Dim	1	2	3	4	5	6	7	8	9	10
1	8	1	10	4	9	6	5	7	2	3

Check the search space ranking and setReNPCC

N$_c$/Dim	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	0	0	0	0	1	1

- The subset of best feature (**setReNPCC**) with No_depo, such as **1,3,4,9,10**

The number of selected features is 3, so we set the other as 1 if the ranking is less than 5

N$_c$/Dim	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	0	0	1	0	1	1

$SF = 5$

Figure 2. The example of finding the significant feature subsets by the local search

In the step of the local search algorithm, it is essential to convert the continuous search space of UHCSAGWO to a binary version as 0 or 1 for feature selection problems. In the feature selection, the value of the binary vector is equal to 1, which means the feature is selected, whereas the value is 0, which means the corresponding feature is unselected. In the step of selecting each feature, we proposed a way to achieve selecting a subset of significant features by evaluating a proposed local search algorithm based on the subsetReNPCC, as represented in Algorithm 2.

```
Algorithm 2: The Local search algorithm
Input: Training sets, The number of selected features (SF), The number of selected features (SF), and subsetReNPCC
Output: the binary versions of the crow position (search space) as $\vec{X}$
NF= the size of subsetReNPCC; $cal_{log} = 0$; the size of Outputdata = NF, p =1
locMaxk = maxk( $\vec{X}$,NF) // to find max value of each search space by calling maxk function
// find the same position each dimension between setReNPCC and the max search space of the crows' position
for (i = 1: i≤ SF) do
    for (j = 1: j ≤ SF) do
     if setReNPCC(i) == locMaxk(j)  &&  P ≤ NF
         Outputdata(p)= setReNPCC(j);  // store the same position between setReNPCC and locMaxk
         p = p+1
     end If
   end for
end for
logOutput = locMaxk (1: $cal_{log}$) // set logOutputdata with the size of calLog
X(Outputdata(logOutput) )= 1 //set the digits or dimensions in logOutputdata by utilizing Outputdata with the size of SF
 if (X == 1) < SF
    for  j =1to $cal_{log}$ do
     if logOutputdata(i) ≠ logOutputdata(j)
```

```
            X(logOutputdata(logOutput)) = 1   // set the other digit are equal 1
            p = p+1
        end If
      end for
  end If
Return  $\vec{X}$ //the search space in binary vector, as 0 and 1.
```

For example, if x_h = [0.3, 0.93, 0.12, 0.74, 0.19, 0.50, 0.6, 0.39, 0.9, 0.8], the number of selected features (SF) is at 5, and the subsetReNPCC = [1,3, 4,9,10], as expressed in the example of finding the significant feature subsets by the local search in Figure 2. Then, ordering the search space is as Rank(x_h)= [8, 1, 10, 4, 9, 6, 5, 7, 2, 3]. Following that, we convert the positions in the search space to 0 or 1 by considering both the position of subsetReNPCC and the search space ranking. If both of them are in the same position, it sets them at 1. So, the new search space is [0, 0, 0, 1, 0, 0, 0, 0, 1, 1]. However, the SF value is set at 5, and the new search area that is to be set at 1 is only 3. Therefore, it requires converting the remaining search space to 5 by taking advantage of the ranking as follows: [0, 1, 0, 1, 0, 0, 1, 0, 1, 1].

In the feature selection phase, we proposed a new fitness function that is formulated by combining two objectives by setting a weight factor for discarding the redundant information in relevant features by utilizing two well-known filter-based algorithms to rank the selected dimension to estimate the weight of the subset of the selected features, as shown in Eq. (4)-(6).

$$Fit_{PCC} = \frac{Rank_{PCC}}{base_{Fit}} \; ; \; Fit_{ReliefF} = \frac{Rank_{ReliefF}}{base_{Fit}} ; \; base_{Fit} = \frac{Dim^2 * \alpha}{\gamma} \qquad (4)$$

$$Fit_{Redundancy} = Fit_{ReliefF} + Fit_{PCC}, Fit_{relevant} = Score_{NMI} \qquad (5)$$

$$Fit\_value = \sum(Fit\_Relevant) - (\beta \times \sum(Fit\_Redundancy) \qquad (6)$$

Where Dim dedicates the dimension of search space. α, β and μ are three parameters corresponding to the importance of evaluate the fitness value quality and selected feature subset size, α = [0.1, 0.3, 0.5, 0.7, 0.9], β = [0.7, 0.9], and γ = 5.

Beyond that, the candidate subsets of dimension or feature are evaluated using a new proposed fitness function. Then, the subset of genes with the better fitness value is kept as the best result in the current iteration. This process continues until the maximum number of iterations ($iter_{max}$) is reached. Finally, we can conclude our proposed algorithm for unsupervised-filter-based feature selection in pseudocode, as expressed in Algorithm 3, the proposed algorithm UHCSAGWO.

Algorithm 3: Unsupervised-filter-based feature selection algorithm-UHCSAGWO

```
Input: Training sets, The number of selected features (SF)
Output: The classification error rates, the computational time, the fitness values, the subset of selected features
Set the initial values of $N_C, AP, fl_h, ite_{max}, \vec{a}_{max}, \vec{a}_{min}, in_{\omega_{max}}$ and $in_{\omega_{min}}$
Calculate the candidate feature from datasets by PCC and ReliefF as subsetReNPCC in Algorithm 1
Initialize the crow position h randomly, as $\vec{X}_h$
Find the subset of selected features by Local Search that utilized subsetReNPCC and $\vec{X}_h$, as shown in Algorithm 2
Evaluate the fitness function of each crow $Fn(\vec{X})$.
Initialize the memory of search crow $\vec{M}$
Set t: = 1 //counter initialization.
While ($ite < ite_{max}$ number of iterations)
   Update $\vec{a}, in_\omega, in_{\omega1}, in_{\omega2}, in_{\omega3}, \vec{X}_1, \vec{X}_2, \vec{X}_3$ A, C and $\vec{X}^*(ite)$
     for (h = 1: h ≤ $N_C$) do
       Randomly choose one of crows to follow q
       if $rand() \geq AP_r(ite)$ then // exploitation phase
             $\vec{x}_h(ite+1) = (\vec{x}_h(ite) \times in_\omega) + fl_h(ite) \times rand() \times [\vec{M}_q(ite) - \vec{x}_h(ite)]$
       else
             $\vec{X}_h(ite+1) = \frac{(in_{\omega1} \times \vec{X}_1) + (in_{\omega2} \times \vec{X}_2) + (in_{\omega3} \times \vec{X}_3)}{(in_{\omega1} + in_{\omega2} + in_{\omega3})}$
       end If
     end for
     Check the feasibility of $\vec{X}(ite+1)$
     Find the subset of selected feature by Local Search that utilized subsetReNPCC and $\vec{X}$
     Evaluated the new position of crow $Fn(\vec{X}(ite+1))$
     Update the crow's memory $\vec{M}(ite+1)$
   Until ($ite < iteMax$) //Termination criteria satisfied
Produce the best solution $\vec{M}$
```

5. Experimental Settings

To evaluate our proposed algorithm, we demonstrate the process of experimentation that divides the dataset into training sets and testing sets, as shown in Figure 3. In Table 1, we have shown the parameter settings of the experimental report and a detailed list of algorithms that are used in this report, such as the proposed algorithms UHCSAGWO and other hybrid-based feature selection algorithms. We utilize five datasets from DNA-microarray datasets to evaluate the performance of our proposed method. The detailed distributions of names, the number of features, the number of samples, and the number of classes for each dataset are outlined in Table 2; the datasets are obtained from http://csse.szu.edu.cn/staff/zhuzx/Datasets.html [26].

This study was coded in MATLAB R2018a with Intel HD Graphics 6000, 1536 MB, 8 GB of memory, 1600 MHz DDR3, 1,6 GHz Dual-Core Intel Core i5, macOS Big Sur, and 128 GB HDD.

To verify the performance of our proposed algorithms, the experiments were conducted under the following three aspects: (1) The proposed UHCSAGWO algorithm for filter-based feature selection is compared with different tuning parameters of α and β to find the optimal solution for the proposed algorithm that uses SVM for classifier algorithm. The optimal-proposed algorithm is tested with a variety of classifiers, including SVM (linear kernel), DT, and NB. (2) On different classifier algorithms, the proposed UHCSAGWO algorithm with appropriate parameters α and β is compared with different numbers of selected features that range from 10 to 100. (3) The proposed UHCSAGWO algorithm is compared to other filter-based feature selection algorithms such as UFSACO, MGSACO, GSBACO, Term Variance (TV), Laplacian score (LS), MC, RRFS, and RSM. The results were obtained referring to [27]. Note: BCSA had proposed wrapper-based feature selection to be fair in comparison, so BCSA will utilize our proposed fitness function to evaluate their selected feature subsets for filter-based feature selection.

Table 1. Parameter settings of the algorithm used for comparison in this study

Algorithm	No. of Population	No. of Iteration	Parameters
UHCSAGWO	100	50	AP=0.8,fl=2,$in_{\omega_{max}}$=0.9, $in_{\omega_{min}}$= 0.4, and γ=7
BCSA	100	50	AP=0.1 and fl=2
UFSACO[27]	100	50	q_0=0.7, τ=0.2, β =0.1,ρ=0.2
MGSACO[6]	100	50	q_0=0.7, τ=0.2, β =0.1, ρ=0.2
GSBACO[28]	100	50	α=2, β =1,τ_0=0,q=3, ρ=0.2,ω=1.2,n_m=3
LS [29]	-	-	-
TV	-	-	-
RRFS [30]	-	-	The maximum allowed similarity between pairs of features is set in the range of [0.5,1)
RSM [31]	100	50	The size of the subspace in each iteration is set to 200

In this paper, we divided the dataset into 70% training sets and 30% testing sets that were used to assess the performance of the classifiers, such as SVM, DT, and NB. The population size is fixed at 100, whereas the number of maximum iterations is set at 50, and the results are averaged over 20 and 30 independent runs to achieve statistically average results. By following the experimental results, we evaluated the performance of the proposed algorithm in different aspects, namely the classification error rate (CERR) and the computational time to compare well-known unsupervised filter-based feature selection algorithms. Finally, we conducted an extensive comparison, focusing on meta-heuristic algorithms.

Table 2. The planning and control components.

Name	No. of Features	No. of Samples	No. of Classes
Colon	2000	62	2
Leukemia	7129	72	2
SRBCT	2308	83	4
Prostate	5966	102	2
Lung	12600	203	5

5.1 Experimental Results: Classification error rate

In this study, the performance of the proposed algorithm UHCSAGWO is evaluated employing different classifiers, such as SVM, DT, and NB. The best result in the table is shown in a bold label.

By tuning our proposed algorithm UHCSAGWO, we determine the appropriateness of parameters α and β in the proposed fitness function to achieve the lowest classification error rate on five DNA-microarray datasets (in percentage), as shown in Table 3. The table displays the experimental results of the tuning parameters α and β in terms of classification error rate by SVM classifier algorithm, with parameter α set between 0.1 and 0.9 and parameter β set between 0.7 and 0.9, and by evaluating the performances over 20 independently run tests and concluding the results in average (avg), best, worst, and standard deviation (std).

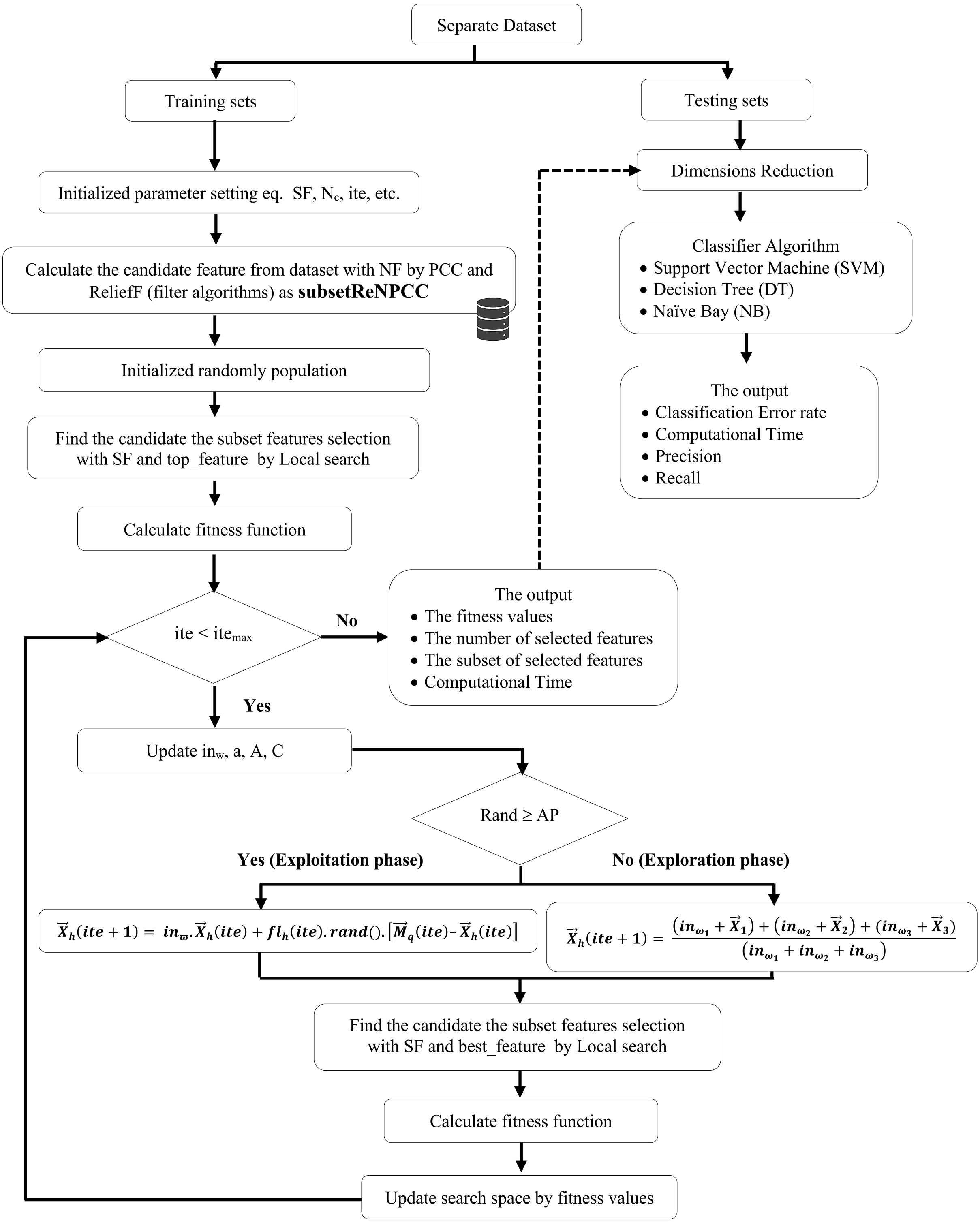

Separate Dataset
Training sets
Testing sets
Initialized parameter setting eq. SF, N_c, ite, etc.
Dimensions Reduction
Calculate the candidate feature from dataset with NF by PCC and ReliefF (filter algorithms) as **subsetReNPCC**
Classifier Algorithm
- Support Vector Machine (SVM)
- Decision Tree (DT)
- Naïve Bay (NB)
Initialized randomly population
The output
- Classification Error rate
- Computational Time
- Precision
- Recall
Find the candidate the subset features selection with SF and top_feature by Local search
Calculate fitness function
ite < ite_{max}
No
The output
- The fitness values
- The number of selected features
- The subset of selected features
- Computational Time
Yes
Update in_w, a, A, C
Rand ≥ AP
Yes (Exploitation phase)
No (Exploration phase)
$\vec{X}_h(ite+1) = in_{\varpi}.\vec{X}_h(ite) + fl_h(ite).rand().[\vec{M}_q(ite)-\vec{X}_h(ite)]$
$\vec{X}_h(ite+1) = \frac{(in_{\omega_1}+\vec{X}_1)+(in_{\omega_2}+\vec{X}_2)+(in_{\omega_3}+\vec{X}_3)}{(in_{\omega_1}+in_{\omega_2}+in_{\omega_3})}$
Find the candidate the subset features selection with SF and best_feature by Local search
Calculate fitness function
Update search space by fitness values

Figure 3. The process of experimentation of our proposed algorithm in this study

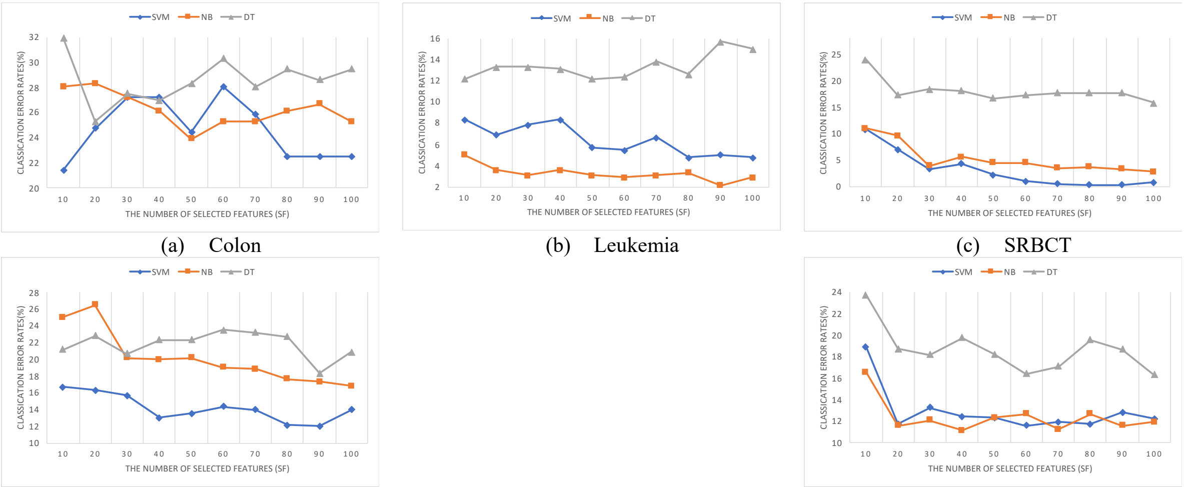
Therefore, the lowest CERR on SVM classifier is achieved by the proposed algorithm UHCSAGWO compared to the results of tuning parameters α and β over all the datasets. For example, for colon dataset, the average CERR obtained by α and β is 0.1 and 0.9, which is 24.72, while for the other results of tuning parameters α and β, this value is reported to be 26.67, 25.83, 26.39, 26.11, 27.50, respectively. Moreover, by obtaining an 6.90 average CERR on leukemia dataset, all the results of tuning parameters are the same value.

Table 3. The average, best, worst, and standard deviation classification error rates of tuning parameter performances by SVM classifier algorithm

Parameter Setting			**The results**					**The summaries**	
α	β		**Colon (20)**	**Leukemia (20)**	**SRBCT (20)**	**Prostate (20)**	**Lung (20)**	**Average**	**Ranking**
0.1	0.7	**Avg**	26.67	**6.90**	5.21	17.00	15.92	14.34	6
		Best	11.11	**0.00**	**0.00**	3.33	6.67	4.22	**3**
		Worst	50.00	**14.29**	25.00	36.67	25.00	30.19	8
		SD	11.20	**5.00**	6.32	8.71	4.88	7.22	5
0.1	0.9	**Avg**	**24.72**	**6.90**	7.08	**16.33**	**11.75**	**13.36**	**1**
		Best	**11.11**	**0.00**	**0.00**	**3.33**	**5.00**	3.89	2
		Worst	**44.44**	**14.29**	25.00	**36.67**	**16.67**	27.41	2
		SD	**8.73**	**5.00**	7.04	**8.98**	**3.65**	**6.68**	**1**
0.3	0.7	**Avg**	25.83	**6.90**	5.42	17.50	15.00	14.13	5
		Best	11.11	**0.00**	**0.00**	3.33	6.67	4.22	3
		Worst	44.44	**14.29**	25.00	36.67	25.00	29.08	4
		SD	11.72	**5.00**	5.91	9.17	4.62	7.28	8
0.3	0.9	**Avg**	25.83	**6.90**	5.63	17.50	14.42	14.06	3
		Best	11.11	**0.00**	**0.00**	3.33	8.33	4.56	4
		Worst	50.00	**14.29**	25.00	36.67	21.67	29.52	6
		SD	12.52	**5.00**	5.62	9.17	3.95	7.25	6
0.5	0.7	**Avg**	26.39	**6.90**	5.42	17.50	15.50	14.34	6
		Best	11.11	**0.00**	**0.00**	3.33	6.67	4.22	3
		Worst	55.56	**14.29**	29.17	36.67	21.67	31.47	9
		SD	10.64	**5.00**	6.77	9.17	4.23	7.16	3
0.5	0.9	**Avg**	26.11	**6.90**	**4.79**	17.50	14.42	13.94	2
		Best	11.11	**0.00**	**0.00**	3.33	6.67	4.22	3
		Worst	50.00	**14.29**	25.00	36.67	23.33	29.86	7
		SD	11.83	**5.00**	5.94	9.17	4.60	7.31	9
0.7	0.7	**Avg**	26.67	**6.90**	5.63	17.50	15.33	14.41	8
		Best	5.56	**0.00**	**0.00**	3.33	6.67	**3.11**	**1**
		Worst	44.44	**14.29**	**20.83**	36.67	21.67	27.58	3
		SD	11.90	**5.00**	**5.45**	9.17	4.38	7.18	4
0.7	0.9	**Avg**	27.50	**6.90**	5.21	17.50	14.83	14.39	7
		Best	11.11	**0.00**	**0.00**	3.33	6.67	4.22	3
		Worst	44.44	**14.29**	29.17	36.67	21.67	29.25	5
		SD	10.89	**5.00**	6.88	9.17	4.39	7.26	7
0.9	0.7	**Avg**	26.11	**6.90**	5.21	17.50	14.83	14.11	4
		Best	5.56	**0.00**	**0.00**	3.33	6.67	**3.11**	**1**
		Worst	38.89	**14.29**	**20.83**	36.67	25.00	**27.14**	**1**
		SD	10.98	**5.00**	5.72	9.17	4.80	7.13	2
0.9	0.9	**Avg**	26.67	**6.90**	5.63	17.50	15.33	14.41	8
		Best	5.56	**0.00**	**0.00**	3.33	6.67	**3.11**	**1**
		Worst	44.44	**14.29**	**20.83**	36.67	21.67	27.58	3
		SD	11.90	**5.00**	**5.45**	9.17	4.38	7.18	4

In addition, prostate and lung datasets reveal that the parameters α and β are set at 0.1 and 0.9, respectively, and achieved against others by 16.33 and 11.75, respectively. According to all the datasets, the tuning parameters α and β are set at 0.1 and 0.9, which outperforms others by 13.36, whereas the other average CERR are such as α = 0.1 and β = 0.7 as 14.34, α = 0.5 and β = 0.7 as 14.34, α = 0.5 and β = 0.9 as 13.94, α = 0.9 and β = 0.9 as 14.41, and so on. The parameters α and β are set to 0.9 and 0.7, respectively, for the worst and best CERR, which are excellent to execute the selected feature subsets, but the average CERR is not. Furthermore, the parameters α and β are set at 0.1 and 0.9, respectively, with some being more successful than others in terms of the standard deviation of the CERR. As all results, CERR results are α and β are 0.1 and 0.9, which quite achieve the minimum classification error rate over four datasets, except for SRBCT datasets in SVM classifier algorithm.

To assess the robustness of the tuning parameters α and β are at 0.1 and 0.9, respectively. We demonstrate to evaluate our proposed algorithm in different classifier algorithms such as SVM, DT, and NB classifiers with a different number of selected features that range between 10 to 100 features, as represented in Figure 4. As shown in the figure, for leukemia and lung datasets on DT classifier, the CERR of the proposed method algorithm outperforms others, whereas SRBCT and prostate datasets by SVM classifier algorithm obtain excellent achievement while the number of selected features increases, which was excellent as compared to DT and NB. However, in colon dataset, there was a fluctuation between CERR and the number of selected features; SVM classifier had the lowest CERR at 10. Therefore, the figure below confirms that our UHCSAGWO-SVM and UHCSAGWO-DT successfully manipulate the different numbers of selected features that are not affected the performance in terms of CERR.



(a) Colon (b) Leukemia (c) SRBCT

(d) prostate (e) Lung

Figure 4. The comparison of classification error rates with the number of selected features range between 10 to 100 on colon, leukemia, srbce, prostate, lung datasets, respectively.

Tables 4-6 represent the results of BCSA, UFSACO, MGSACO, GSBACO, TV, LS, MC, RRFS, RSM, and the proposed algorithm UHCSAGWO in terms of classification error rate for five high-dimensional datasets with 20 the number of selected features on SVM, DT, and NB classifier algorithms, respectively. In the evaluation results, UHCSAGWO performs superior to others in terms of classification error rate on 3 out of 5 datasets. Overall, the performance of our proposed is the best way to find the optima at 13.36, 15.91, and 19.50 for SVM, DT, and NB, respectively.

Table 4 shows the classification error rate values of different well-known feature selection algorithms on the SVM classifier. As can be seen, the proposed algorithm UHCSAGWO obtained the least CERR on leukemia and prostate datasets, with 6.90 and 16.33, respectively, in comparison to the others. On the other hand, the GSBACO algorithm was successful with the least CERR on colon and lung datasets at 20.36 and 11.429, respectively.

Table 4. The average classification error rates of different methods by SVM classifier algorithm

Datasets	UHCSAGWO	BCSA	UFSACO	MGSACO	GSBACO	TV	LS	MC	RRFS	RSM
Colon	24.72	26.11	21.81	21.81	**20.36**	21.81	33.63	38.18	24.54	24.54
Leukemia	**6.90**	9.05	41.02	17.94	23.7	20.58	35.29	38.23	23.52	37.64
SRBCT	7.08	5.00	28.27	25.51	22.069	39.31	36.55	45.51	31.72	37.93
Prostate	**16.33**	18.00	40.57	26.85	19.143	28	48	34.28	30.85	22.85
Lung	11.75	13.92	17.14	14.28	**11.429**	27.71	18	28.57	19.14	35.71
Average	**13.36**	14.42	29.76	21.28	19.34	27.48	34.29	36.95	25.95	31.73
Rank	**1**	2	7	4	3	7	8	9	5	7

According to Table 5, the lowest CERR on DT classifier is achieved by the proposed UHCSAGWO method compared to the others over all the datasets, such as leukemia, srbct, and lung datasets. For example, the CERR obtained by the proposed method UHCSAGWO for leukemia datasets is 3.57, whereas the values reported for BCSA, UFSACO, MGSACO, GSBACO, TV, LS, MC, RRFS, and RSM are 5.24, 30.76, 23.07, 23.529, 20.58, 29.41, 32.35, 20.58, and 38.82, respectively. Moreover, by obtaining a 15.91 average CERR on all the datasets, the proposed algorithm is better than UFSACO by 29.03, MGSACO by 23.83, GSBACO by 21.97, TV by 27.67, LS by 35.89, MC by 35.51, RRFS by 28.41, and RSM by 38.01 except BCSA by 14.71.

Table 5. The average classification error rates of different methods by DT classifier algorithm.

Datasets	UHCSAGWO	BCSA	UFSACO	MGSACO	GSBACO	TV	LS	MC	RRFS	RSM
Colon	28.33	26.39	**24.54**	23.63	20.909	31.81	39.09	33.63	34.54	28.18
Leukemia	**3.57**	5.24	30.76	23.07	23.529	20.58	29.41	32.35	20.58	38.82
SRBCT	**9.58**	10.00	27.58	22.75	21.149	22.75	45.51	44.13	28.96	58.62
Prostate	26.50	**18.67**	33.71	29.71	25.714	38.85	43.99	36	37.71	33.71
Lung	**11.58**	13.25	28.57	20	18.571	24.28	21.43	31.42	20.28	30.71
Average	15.91	14.71	29.03	23.83	21.97	27.65	35.89	35.51	28.41	38.01
Rank	2	1	6	3	2	5	8	9	4	7

As can be seen in Table 6, the average CERR of the proposed algorithm UHCSAGWO on the NB classifier algorithm is among those of the nine filter-based feature selection algorithms. The proposed algorithm provided the best average results of the CERR over five datasets, with a value of 19.50. The table demonstrates that MGSACO algorithms outperform those on colon, leukemia, and srbct datasets, whereas our proposed algorithm was successful on those others.

Table 6. The average classification error rates of different methods by NB classifier algorithm.

Datasets	UHCSAGWO	BCSA	UFSACO	MGSACO	GSBACO	TV	LS	MC	RRFS	RSM
Colon	25.28	28.89	28.18	**20**	18.182	41.81	47.27	31.81	32.72	26.36
Leukemia	13.33	15.71	41.02	**7.69**	26.77	32.35	8.82	29.41	35.29	42.35
SRBCT	17.29	20.21	20	**15.86**	11.034	38.62	32.41	37.93	28.27	37.92
Prostate	**22.83**	27.00	39.42	37.14	34.286	33.14	32.57	33.71	31.42	30.28
Lung	**18.75**	18.92	35.71	20	17.143	31.99	29.99	59.04	21.71	23.57
Average	**19.50**	22.15	32.87	20.14	21.48	35.58	30.21	38.38	29.88	32.1
Rank	**1**	4	7	2	3	6	9	10	5	8

5.2 Experimental Results: An extensive comparison with state-of-the-art methods on meta-heuristic algorithms

The performance of the proposed algorithm is compared with that of the other meta-heuristic algorithms based on ant colony optimization algorithms: The results of 30 independent runs of the proposed method, such as the UFSACO, MGSACO, and GSBACO methods, and the average and standard deviation of the classification performance on the different datasets for selecting 30 genes are presented in Tables 7-9 on colon, SRBCT, and leukemia datasets, respectively. However, the CERR values do not provide adequate information to assess the robustness of the results. In order to achieve the goal, the proposed method should be evaluated with other criteria such as precision and recall. These are the criteria that were used in this study. Therefore, precision and recall for all the algorithms on the SVM, DT, and NB classifiers in different datasets are reported in Tables 7-9.

For the colon dataset in Table 7, the proposed algorithm on the SVM classifier does not have good performance when compared with other algorithms, with values of 71.79 and 72.86 for precision and recall, respectively. The GSBACO algorithm performs well in terms of precision in both average and standard deviation among SVM, NB, and DT classifiers. For recall criteria, the UFSACO algorithm obtained the best performance when compared to others, such as 95.87 and 5.68 for average and standard deviation on the SVM classifier.

Table 7. The average of all metrics performance of different meta-heuristic algorithms on colon dataset

Algorithms		**UHCSAGWO**		**BCSA**		**UFSACO**		**MGSACO**		**GSBACO**	
		Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
SVM	AVG	71.79	72.86	74.07	71.95	75.86	**95.87**	80.39	95.1	**81.16**	94.51
	STD	10.46	10.90	11.56	10.63	9.75	**5.68**	11.46	5.82	**7.91**	5.84
DT	AVG	69.46	69.60	70.38	71.62	76.73	88.81	80.46	**90.66**	**82.65**	85.69
	STD	10.16	11.91	9.90	11.72	8.33	9.95	8.86	**9.17**	**8.01**	12.13
NB	AVG	72.25	73.96	73.89	76.01	75.86	**95.87**	82.91	81.22	**85.42**	89.02
	STD	10.87	11.83	9.57	9.65	9.75	**5.68**	10.57	17.44	**8.3**	6.8

According to Table 8-9, the experimental results on SVM, DT, and NB classifiers are achieved in precision and recall by the proposed UHCSAGWO method when compared against the other filter-based methods on srbct and leukemia datasets. For example, for the srbct and leukemia datasets, the precision obtained by the proposed method successfully finds the significant feature subsets of 96.78 and 92.97, respectively. However, for BCSA, UFSACO, MGSACO, and GSBACO, these values on precision by srbct dataset are reported to be 95.97, 74.18, 81.06, and 85.67 as represented by Table 8. Moreover, by obtaining a 96.50 average recall on srbct dataset, the proposed method is better than UFSACO by 83.93, MGSACO by 87.8, and GSBACO by 89.21except for BCSA by 96.55.

Table 8. The average of all metrics performance of different meta-heuristic algorithms on SRBCT dataset

Algorithms		**UHCSAGWO**		**BCSA**		**UFSACO**		**MGSACO**		**GSBACO**	
		Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
SVM	AVG	**96.78**	96.50	95.97	**96.55**	74.18	83.93	81.06	87.8	85.67	89.21
	STD	**3.30**	3.53	3.68	**3.04**	22.3	16.48	20.28	15.88	16.58	12.99
DT	AVG	81.50	81.74	**82.01**	**82.13**	80.87	74.46	83.6	78.73	84.27	77.95
	STD	10.12	**9.48**	**8.76**	9.67	14.01	15.81	15.51	13.06	16.61	15.32
NB	AVG	**96.16**	**95.79**	94.99	94.28	89.49	83.9	95.06	92.72	92.2	94.82
	STD	**3.37**	**4.15**	4.04	4.95	12.27	13.09	7.45	9.53	9.44	7.23

Table 9. The average of all metrics performance of different meta-heuristic algorithms on Leukemia dataset

Algorithms		UHCSAGWO		BCSA		UFSACO		MGSACO		GSBACO	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
SVM	AVG	**92.97**	**91.68**	90.05	91.58	75.46	90.59	77.9	92.26	75.33	93.32
	STD	**5.29**	**5.51**	6.45	5.75	12.61	10.62	10.57	8.81	8.48	7.33
DT	AVG	**86.50**	**84.33**	82.40	82.28	75.59	79.41	80.26	81.68	76.03	83.83
	STD	**6.46**	**6.59**	11.27	10.50	10.47	9.87	9.39	11.86	8.26	8.83
NB	AVG	**96.94**	**96.29**	93.05	93.09	83.56	78.28	83.43	80.75	72.01	92.16
	STD	**3.43**	**3.79**	6.44	6.44	11.44	14.29	9.26	14.23	9.21	5.52

As can be seen from Table 9, the standard deviation (STD) of the proposed method in recall criteria is the least of those of the other algorithms, particularly ACO-based algorithms. For example, for the leukemia dataset, STD value is 5.51, while BCSA, UFSACO, MGSACO, and GSBACO exhibit values of 15.75, 10.62, 8.81, and 7.33, correspondingly. In general, the results show that the proposed method accomplished the best value, as shown in Tables 8–9. The average and standard deviation of all the performances of the proposed algorithms on srbct and leukemia datasets are obtained with the highest precision and recall, which confirms our proposed algorithm's excellent performance on the SVM, DT, and NB classifiers. Therefore, we display an example of genes that are selected by the proposed algorithm, as shown in table 10. The numbers are shown in the table that refers to the sequence of the genes in the dataset.

Table 10. Examples of our proposed algorithms' significant features subset selection

Datasets	The subset of selected features
Colon	49 138 286 304 377 391 395 415 427 451 485 493 627 787 870 979 992 1060 1073 1113 1293 1356 1365 1416 1423 1580 1648 1836 1843 1887
Leukemia	173 200 293 450 654 695 758 804 874 900 1081 1159 1381 1450 1685 1745 1779 1834 1882 1909 1926 1953 2288 2348 2354 2642 3778 4107 4377 5231
SRBCT	166 169 171 174 255 433 445 509 672 719 796 847 850 1069 1093 1208 1494 1606 1662 1768 1795 1799 1834 1884 1886 1924 2000 2046 2162 2198

5.3 Experimental Results: Computational time

This experiment yields a comparison of the computational time of our proposed algorithm with other ACO-based algorithms over all the datasets. A comparison of the computational time (in milliseconds) of the UHCSAGWO algorithm with those of the other algorithms is represented in Figure 5.

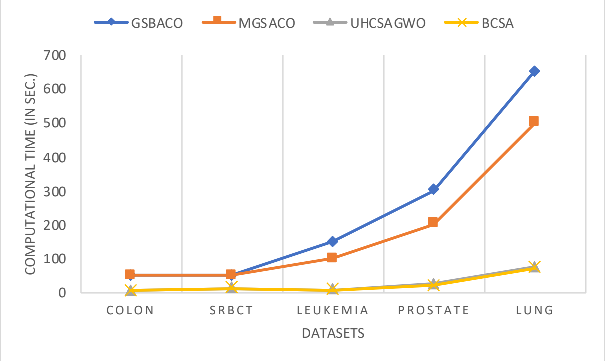


Figure 5. The comparison of computational time of different methods

It is clearly illustrated in the graph that the proposed method exhibits shorter CPU wall-clock times. This is because our proposed algorithm employs a proposed local search that could manipulate large features and significantly reduce execution time in high-dimensional datasets such as DNA-microarray data. Interestingly, the fitness function that we proposed could be examined to find the relevant features that help in classification tasks without any classifier algorithm. On the other hand, ACO-based algorithms employ a full-graph method to find significant feature subsets. As the problem dimensionality (i.e., number of features) increases, and the running time of such algorithms increases polynomially, rendering them infeasible for very large problems.

5.4 Experimental Results: Statistic Analysis

In this study, we utilized a statistical method based on nonparametric tests known as the Wilcoxon rank-sum test by employing the fitness value of our proposed algorithm, UHCSAGWO. The test compares two algorithms, or repeat measurements, from a pair of samples using our proposed algorithm with 5% accuracy. According to the test, we used a confidence level of 0.95 for statistical analysis, and p-values greater than or equal to 0.05 are shown in bold, as expressed in Table 11.

Table 11. The performance comparison of the Wilcoxon rank-sum test on five DNA microarray datasets

Datasets	**SF=20**										**SF=30**
	Our1	**Our3**	**Our4**	**Our5**	**Our6**	**Our7**	**Our8**	**Our9**	**Our10**	**BCSA**	**BCSA**
Colon	7.44E-02	1.01E-07	1.18E-07	1.01E-07	1.01E-07	1.01E-07	1.01E-07	1.01E-07	1.01E-07	**1.79E-01**	**1.37E-01**
Leukemia	7.25E-01	6.84E-01	6.94E-01	6.84E-01	6.84E-01	6.84E-01	6.84E-01	6.84E-01	6.84E-01	1.48E-07	6.51E-11
SRBCT	4.07E-01	1.97E-04	2.46E-04	3.41E-05	3.01E-05	4.35E-05	3.41E-05	6.49E-06	4.35E-05	3.55E-02	4.31E-05
Prostate	6.03E-01	3.19E-01	3.61E-01	2.67E-01	2.92E-01	2.67E-01	2.67E-01	2.55E-01	2.67E-01	2.79E-06	3.77E-09
Lung	4.74E-01	8.70E-04	1.29E-03	8.36E-05	1.76E-04	5.53E-05	9.96E-05	3.01E-05	5.53E-05	**2.67E-01**	4.32E-03

***Not:** Our1: UHCSAGWO$_{\alpha=0.1,\ \beta=0.7}$, Our3: UHCSAGWO$_{\alpha=0.3,\ \beta=0.7}$, Our4: UHCSAGWO$_{\alpha=0.3,\ \beta=0.9}$, Our5: UHCSAGWO$_{\alpha=0.5,\ \beta=0.7}$, Our6: UHCSAGWO$_{\alpha=0.5,\ \beta=0.9}$ Our7: UHCSAGWO$_{\alpha=0.7,\ \beta=0.7}$, Our8: UHCSAGWO$_{\alpha=0.7,\ \beta=0.9}$, Our9: UHCSAGWO$_{\alpha=0.9,\ \beta=0.7}$, Our10: UHCSAGWO$_{\alpha=0.9,\ \beta=0.9}$

Wilcoxon rank-sum test p-value scores were obtained with 5% accuracy from a pair of samples for two algorithms with 20 independent runs to test the null hypothesis for five DNA-microarray datasets summarized for different numbers of selected numbers such as 20 and 30. As for the results, it claimed p-values show that there are significant differences between the results obtained by the CSA, other proposed methods, and the proposed UHCSAGWO for all datasets. However, there is no significant difference between UHCSAGWO that at α=0.1, β= 0.9 and CSA for only 2 datasets in the number of selected features at 20, whereas there is only one colon dataset in the number of selected features at 30.

6. Discussion

The statistical performance reveals the potential of our proposed algorithm, UHCSAGWO, which can achieve significant feature subset selection by unsupervised-based feature selection on high-dimensional data, especially DNA microarray data. This study introduced three crucial mechanisms for proposed algorithms based on CSA and GWO, such as a pool for finding the top candidate features, a local search, and a novel fitness function. In the majority of situations, the performance of the UHCSAGWO is superior to the standard CSA because it has secured state-of-the-art classification error rates (CERR), precision, and recall compared to different datasets, such as colon, leukemia, srbct, prostate, and lung. For most of the cases, even the number of features selected by the proposed UHCSAGWO is below 99%, as seen from Tables 4–9. The reasons why the UHCSAGWO performs excellently and efficiently in execution to choose the crucial feature subset that is stable and robust are explained next. Begin with using the pool for storing setReNPCC, which helps the local search evaluate the significant feature by removing irrelevant features from the original dataset. Furthermore, we proposed a high-performance local search based on the search space and setReNPCC to select or unselect the feature. In addition, the strategy to select the feature subsets can increase the diversity of the search spaces and jump out to the global optimum to make the algorithm more effective. The main reason that UHCSAGWO can perform well in this type of problem is that it has other meta-heuristic algorithms that perform well in feature selection problems with different sizes of features, such as 2000 up to 12600 features.

The results of this problem showed that the UHCSAGWO method excellently chooses the feature subsets and outperforms the other methods in different metrics, such as classification error rates, as shown in Tables 4-6. It can be clearly seen that the proposed UHCSAGWO makes it quite difficult to judge the competitiveness of the algorithms based on only the number of features selected and CERR. Therefore, precision and recall for all the algorithms on the SVM, DT, and NB classifiers in different datasets are reported in Tables 7–9. As the comparison among BCSA, UFSACO, MGSACO, and GSBACO confirmed confidently, the advantages of the UHCSAGWO include performing much more efficiently, having greater robustness, and having only a few parameters to employ in the optimization problems. According to Figure 5, it is evident that the proposed algorithm, UHCSAGWO, reveals shorter CPU wall-clock times when compared with other meta-heuristic algorithms, especially ACO-based algorithms. As shown in Table 11, all statistical results support our proposed algorithm's claim that there are significant differences for CSA on the Wilcoxon rank-sum test.

Therefore, we can safely claim that UHCSAGWO has better capability in unsupervised-based feature selection because it cannot access the label for prediction as well as in enhancing CERR, precision, and recall. However, some evaluated datasets may fail to avoid local minima while finding global optima, as referred to by the NFL theorem [32]. According to the work's limitations, one possible shortcoming of the proposed UHCSAGWO algorithm is that certain parameters may take different values for other high-dimensional optimization problems, and trial and error are needed.

7. Conclusions and Future Direction

In DNA-microarray technology, feature selection methods play an important role in bringing out the significant feature subsets by discarding irrelevant and redundant information and providing valuable information to reveal disease biomarkers, as well as proposing clinical diagnostic and therapeutical hypotheses. In this study, an unsupervised filter-based feature selection algorithm inspired by the combination of CSA and GWO was presented. Three mechanisms have been presented: a pool for finding the top candidate features; a local search; and a novel fitness function. The pool plays an important role in finding the top candidate and preparing for the local search step to remove any irrelevant and redundant features. Then, an iterative process of the best-selected features was obtained by the local search within randomly explored or exploited capacities. In addition, our proposed algorithm can manipulate large features or high-dimensional data and can evaluate the subset of the selected features by utilizing a fitness function that is as close as possible to an optimal solution by reducing the performance complexity to estimate them. The experimental results reveal that UHCSAGWO achieves significant improvements in microarray data analysis when compared with other in state-of-the-art feature selection algorithms using different classifiers. Moreover, the results confirmed that UHCSAGWO could achieve better performance than the others under different criteria and reduce computational complexity. Thanks to its lightweight computational aspects, the proposed algorithm is plausible to be applied in ultra-high dimensional problems such as next-generation DNA sequencing data. Furthermore, due to its unsupervised feature selection nature UHCSAGWO can evaluate vast amounts of unlabelled microarray data which is the main mode of data regime in molecular databases (e.g. Gene Expression Omnibus (GEO) [33]). The proposed algorithm can be improved in future directions to manipulate large amounts of data by identifying significant features for global optimization, namely decision-making problems, etc., especially big data.

Funding

No funding is provided for the preparation of manuscript.

Conflict Of Interest Statement

The authors have no conflicts of interest to declare.

References

[1] Ray, P.; Reddy, S. S.; Banerjee, T. (2021). Various dimension reduction techniques for high dimensional data analysis: A review. Artificial Intelligence Review, 54(5), pp.3473–3515. https://doi.org/10.1007/s10462-020-09928-0

[2] Li, Y.; Huang, C.; Ding, L.; Li, Z.; Pan, Y.; Gao, X. (2019). Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. Methods, 166, pp.4–21.

[3] Ehrenreich, A. (2006). DNA microarray technology for the microbiologist: An overview. Applied Microbiology and Biotechnology, 73(2), pp.255–273.

[4] Al-Tashi, Q.; Rais, H. M.; Abdulkadir, S. J.; Mirjalili, S.; Alhussian, H. (2020). A review of grey wolf optimizer-based feature selection methods for classification. Evolutionary Machine Learning Techniques, pp.273–286.

[5] Chandrashekar, G.; Sahin, F. (2014). A survey on feature selection methods. Computers Electrical Engineering, 40(1), pp.16–28.

[6] Tabakhi, S.; Najafi, A.; Ranjbar, R.; Moradi, P. (2015). Gene selection for microarray data classification using a novel ant colony optimization. Neurocomputing, 168, pp.1024–1036.

[7] Mafarja, M.; Eleyan, D.; Abdullah, S.; Mirjalili, S. (2017). S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. Proceedings of the International Conference on Future Networks and Distributed Systems, pp.1–7.

[8] Tawhid, M. A.; Ibrahim, A. M. (2020). Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. International Journal of Machine Learning and Cybernetics, 11(3), pp.573–602.

[9] Dabba, A.; Tari, A.; Meftali, S. (2020). Hybridization of Moth flame optimization algorithm and quantum computing for gene selection in microarray data. Journal of Ambient Intelligence and Humanized Computing, pp.1–20.

[10] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Computers Structures, 169, pp.1–12.

[11] Masud, M.; Singh, P.; Gaba, G. S.; Kaur, A.; Alroobaea, R.; Alrashoud, M.; Alqahtani, S. A. (2021). CROWD: crow search and deep learning based feature extractor for classification of Parkinson's disease. ACM Transactions on Internet Technology (TOIT), 21(3), pp.1–18.

[12] Allahverdipour, A.; Soleimanian Gharehchopogh, F. (2018). An improved k-nearest neighbor with crow search algorithm for feature selection in text documents classification. Journal of Advances in Computer Research, 9(2), pp.37–48.

[13] Al-Thanoon, N. A.; Algamal, Z. Y.; Qasim, O. S. (2021). Feature selection based on a crow search algorithm for big data classification. Chemometrics and Intelligent Laboratory Systems, pp.104288.

[14] Samieiyan, B.; MohammadiNasab, P.; Mollaei, M. A.; Hajizadeh, F.; Kangavari, M. (2022). Novel optimized crow search algorithm for feature selection. Expert Systems with Applications, pp.117486.

[15] Adamu, A.; Abdullahi, M.; Junaidu, S. B.; Hassan, I. H. (2021). An hybrid particle swarm optimization with crow search algorithm for feature selection. Machine Learning with Applications, 6, pp.100108.

[16] Abudayor, A.; Nalbantoğlu, Ö. U. (2022). An Improved Crow Search Algorithm with Grey Wolf Optimizer for High-Dimensional Optimization Problems. International Conference on Soft Computing and Its Engineering Applications, pp.51–64.

[17] Hancer, E. (2020). New filter approaches for feature selection using differential evolution and fuzzy rough set theory. Neural Computing and Applications, pp.1–16.

[18] Saberi-Movahed, F.; Rostami, M.; Berahmand, K.; Karami, S.; Tiwari, P.; Oussalah, M.; Band, S. S. (2022). Dual regularized unsupervised feature selection based on matrix factorization and minimum redundancy with application in gene selection. Knowledge-Based Systems, 256, pp.109884.

[19] García-Díaz, P.; Sánchez-Berriel, I.; Martínez-Rojas, J. A.; Diez-Pascual, A. M. (2020). Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-Seq data. Genomics, 112(2), pp.1916–1925.

[20] Martarelli, N. J.; Nagano, M. S. (2020). Unsupervised feature selection based on bio-inspired approaches. Swarm and Evolutionary Computation, 52, pp.100618.

[21] Abualigah, L. M.; Khader, A. T. (2017). Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. The Journal of Supercomputing, 73, pp.4773–4795.

[22] Deshpande, M. A.; Deshpande, M. S.; Doke, M. M.; Chaudhari, M. A. (2016). Unsupervised Feature Selection Using Evolutionary Algorithms. World Journal of Research and Review, 3(1), pp.262925.

[23] Mirjalili, S.; Mirjalili, S. M.; Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69, pp.46–61.

[24] Peng, H.; Long, F.; Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), pp.1226–1238.

[25] Gao, W.; Hu, L.; Zhang, P. (2018). Class-specific mutual information variation for feature selection. Pattern Recognition, 79, pp.328–339. https://doi.org/10.1016/j.patcog.2018.02.020

[26] Microarray Datasets. (n.d.). Retrieved February 8, 2023, from https://csse.szu.edu.cn/staff/zhuzx/Datasets.html.

[27] Tabakhi, S.; Moradi, P.; Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. Engineering Applications of Artificial Intelligence, 32, pp.112–123.

[28] Naseri, A.; Hasheminejad, S. M. H. (2019). An unsupervised gene selection method based on multiobjective ant colony optimization. International Journal of Artificial Intelligence, 17(2), pp.1–22.

[29] He, X.; Cai, D.; Niyogi, P. (2006). Laplacian score for feature selection. Advances in Neural Information Processing Systems, pp.507–514.

[30] Ferreira, A. J.; Figueiredo, M. A. (2012). Efficient feature selection filters for high-dimensional data. Pattern Recognition Letters, 33(13), pp.1794–1804.

[31] Lai, C.; Reinders, M. J.; Wessels, L. (2006). Random subspace method for multivariate feature selection. Pattern Recognition Letters, 27(10), pp.1067–1076.

[32] Wolpert, D. H.; Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), pp.67–82.

[33] Barrett, T.; Edgar, R. (2006). Gene Expression Omnibus: Microarray data storage, submission, retrieval, and analysis. Methods in Enzymology, 411, pp.352–369.

Authors Profile

A. Abudayor received a BE degree in computer engineering from Naresuan University, Phisanulok, Thailand, in 2011 and an ME degree in computer engineering from Prince of Songkhla University, Songkhla, Thailand, in 2016. She is currently pursuing a Ph.D. degree in computer engineering at Erciyes University, Kayseri, Turkey. She has a total of five years of teaching experience. Her interests are optimization, heuristic algorithms, classification, data mining, soft computing, and machine learning.

Dr. Ö. U. Nalbantoğlu received his B.S. in Electrical and Electronics Engineering from Bogazici University, Istanbul, Turkey. He was awarded Ph.D. Electrical Engineering from University of Nebraska at Lincoln, Lincoln, Nebraska, United States. Currently, he is working as an Associate Professor in the Department of Computer Engineering at Erciyes University, Kayseri, Turkey. His research interests include machine learning, Artificial intelligence, bioinformatics, information theory, computational metagenomics, and meta-omics.