# Differential Evolution With Duplication Analysis for Feature Selection in Classification

Sara Kamran, MohammadMahdi Ebadzadeh, Peng Wang, Graduate Student Member, Bing Xue , Senior Member, IEEE,Jing Liang , Senior Member, IEEE, and Mengjie Zhang

**Abstract—By selecting a small subset of relevant features, feature selection can reduce the dimensionality of the problem while maintaining or increasing the discriminating ability of the data. However, many existing feature selection approaches ignore the fact that there are multiple optimal solutions to a feature selection problem. Multiple feature subsets with different features selected can achieve very similar or the same classification accuracy. To search for multiple optimal feature subsets, a niching-based differential evolution (DE) method with duplication analysis is proposed. Unlike traditional approaches that rely on distance-based classifiers, the proposed method integrates a Support Vector Machine (SVM) to evaluate fitness. SVM's ability to construct an optimal hyperplane in high-dimensional spaces provides a more robust and stable fitness landscape, which is crucial for identifying diverse yet equally effective feature subsets. Specifically, the duplicated feature subsets in the population are modified by a proposed subset repairing scheme designed to produce unique feature subsets and increase population diversity. Furthermore, the mutation operator in DE is improved to leverage both niche and global information to produce promising feature subsets, while a new selection method considering diversity is adopted to form the subsequent population for the next generation. In the experiments, the proposed method is compared with seven evolutionary feature selection algorithms and two typical feature selection methods on 18 datasets. The results show that the proposed algorithm achieves higher classification accuracy than the compared methods on most of the used datasets. Furthermore, the proposed method can find different feature subsets with very similar or the same classification accuracy.**

***Index Terms*—Classification, differential evolution (DE), feature selection, multiple optimal feature subsets.**

## I. INTRODUCTION

The advancement of data collection technologies leads to the number of features collected in many classification tasks becoming increasingly large, that is, high-dimensional data. The existence of redundant, irrelevant, and/or noisy features degrades the classification accuracy and increases the training time of a learning algorithm. By selecting a small subset of features with high discriminating ability, feature selection [1] is a crucial dimension reduction technique. Feature selection not only increases the classification accuracy but also simplifies the learned model [2].

Redundant and interactive features cause different feature subsets (solutions) to have very similar or the same classification performance, that is, multiple solutions with different features selected can reach the same classification accuracy [3], [4], [5], [6]. In some scenarios, such as biomarker detection, different feature subsets representing varying conceptual designs may lead to similar performance with the learned model, which can help identify hidden biological information. Liu et al. [4] experimentally founded that two different solutions, {*X*56494_*at*, *M*57710_*at*, *U*19495_*s*_*at*} and {*X*62078_*at*, *L*33842_*rna*1_*at*, *J*02645_*at*} in DLBCL dataset [7], can achieve the same accuracy of 91.7%. The feature *M*57710_*at* in the first solution is up-regulated in the lymphoma samples, while the feature *L*33842_*rna*1_*at* in the second solution is overexpressed in the lymphoma patients [4]. Both solutions with the same classification accuracy may indicate that two different functional modules can differentiate the lymphoma patients from the normal controls.

Furthermore, when providing different feature subsets with very similar or the same classification accuracy, users can pick one from the multiple feature subsets based on their preferences, for example, a smaller number of features included, a lower cost of feature collection, and/or a lower redundant rate. For example, Yue et al. [5] founded that employing *K*-nearest neighbor (KNN) with two different solutions, *S*1: {Clump Thickness, Uniformity of Cell Size, Bland Chromatin} and *S*2: {Uniformity of Cell Size, Uniformity of Cell Shape, Bland Chromatin}, on breast cancer Wisconsin dataset [7] obtain the same classification accuracy of 97.81%. Clump Thickness in *S*1 is easier to collect than Uniformity of Cell Shape in *S*2. Most users are likely to prefer *S*1 since *S*1 has a lower feature collection cost. However, without searching for multiple optimal feature subsets, the algorithm may miss the first feature subset.

While most existing niching-based methods for feature selection rely on K-Nearest Neighbor (KNN) due to its computational simplicity, it often struggles with the curse of dimensionality and sensitivity to noisy features. To overcome these limitations, this study adopts a Support Vector Machine (SVM) as the core classifier within the NDEDA framework. By leveraging SVM's ability to define an optimal hyperplane, the proposed method achieves more robust and stable fitness evaluations in high-dimensional spaces compared to traditional distance-based classifiers. This integration not only enhances classification accuracy but also ensures that the identified multiple optimal subsets possess superior generalization capabilities.

To find multiple optimal feature subsets, niching-based evolutionary computation (EC) methods have been developed for feature selection [5], [6], [8]. Compared with classical search methods, such as sequential forward or backward floating selection method [9], EC methods do not require domain knowledge or assumption of the search/solution space. More importantly, the population-based search and the potential global search ability of EC methods are good for handling feature selection tasks. One such EC method, differential evolution (DE) [10], has been extensively applied to address many real-world problems including feature selection [11], [12]. Furthermore, niching techniques, including crowding-based [13], sharing-based [14], and speciation-based [15] methods, have the potential to find multiple optimal solutions. Niching is a kind of method for finding and retaining multiple stable niches or the areas in the solution space around multiple optimal solutions to prevent convergence to a local optimum.

Although there are some applications of EC methods with niching techniques for feature selection, for example, [5] and [8], they are often used on the data with tens or hundreds of features. Their performance on high-dimensional data with thousands of features or more is still limited. DE incorporates niching techniques, such as [16], [17], [18], and [19] can reach the convergence of population members around different basins of attraction in parallel. However, such algorithms still have some deficiencies, such as confronting difficulties in tackling problems with high dimensionality or many optima [20]. Those motivate us to propose a new niching-based DE method for feature selection.

Some mutation mechanisms in DE including DE/current-to-best/1 showed better performance than some classic mutation operators, for example, DE/rand/1 [21]. However, if only the local information from the niche guides the individuals without simultaneously considering the global information from the whole population, it might increase the risk of an algorithm getting trapped in a local optimum [22]. Another advantage of considering both the niche and the whole population is that it can help an algorithm find multiple optimal solutions [20]. It is expected that different areas of the search space can be covered so that multiple optimal solutions can be found. Therefore, a new mutation operator considering these points will be developed in this work to update the population.

Another obstacle of applying EC methods to feature selection is the existence of duplicated and reappearing feature subsets during training. Although different encoding schemes, including real-valued encoding [23], binary encoding [24], [25], and hybrid encoding [26], have been used, the duplicated and reappearing feature subsets can show in different forms since a feature selection task is inherently a combinatorial optimization problem. For real-valued encoding, some solutions are different in the search space but they may select the same features. In the current generation, duplicated feature subsets can appear in two situations: 1) *parent offspring* duplication (e.g., feature subsets $S3$ and $S7$ in Fig. 1) and 2) *offspring-offspring* duplication (e.g., feature subsets $S5$ and $S6$ in Fig. 1). In addition, some solutions eliminated from the previous generations may reappear despite not being duplicated with any individual in the current generation (e.g., feature subset $S8$ in Fig. 1). If all feature subsets produced during the evolutionary process are preserved into a set, for example, a unique set, feature subset $S8$ in Fig. 1 can be identified as a reappearing solution since $S8$ selects the same features as the solution in the previous generations. Likely, the classification error rate of the features selected by $S8$ is quite high, thus, this solution has been eliminated from the previous generations. However, the newly generated individuals from the current generation may still include this solution.

The duplicated feature subsets seriously reduce the population diversity. Without any strategy to deal with the duplication, a significant amount of memories and evaluations will be wasted. Meanwhile, the duplication limits the performance of the EC-based feature selection methods since the duplicated feature subsets in the population hinder the emergence of new and good feature subsets. More importantly, the existence of the duplicated feature subsets prevents the one method from finding multiple optimal feature subsets. A simple way is to remove all the duplicated feature subsets, but it may lead to a very small number of individuals to the next-generation. Furthermore, some useful information contained in the duplicated solutions will be lost. To overcome these limitations, a subset repairing scheme will be proposed. By modifying the duplicated feature subsets, the proposed scheme can increase

the population diversity. For the reappearing feature subsets, although they do not affect the diversity of the current population, the influence of these solutions on the feature selection performance should be explored. Therefore, experiments will be designed to investigate the impact of the reappearing feature subsets on an algorithm's performance.

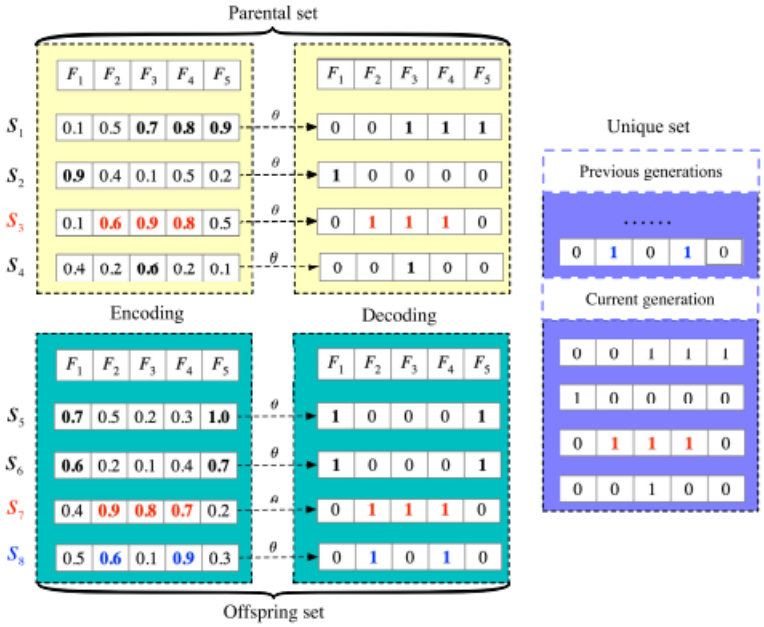


Fig. 1. Duplicated and reappearing feature subsets. Suppose that an EC method is to handle a feature selection task with five features: *F*1–*F*5. In the current generation, four new individuals *S*5–*S*8 are generated using *S*1–*S*4. Unique set (called ***U****)* stores all unique feature subsets during evolution, and its size is called UN, that is, |***U***| = *UN*. More descriptions of UN can be seen in Section IV-C. The feature subsets kept in ***U*** before the current generation are marked as Previous generations.

In this article, a niching-based DE method with duplication analysis is proposed (called NDEDA), which is expected to find different feature subsets with very similar or the same classification accuracy. Specifically, the following contributions are included.

1) An improved mutation operator in DE is proposed. In the developed mutation operator, both the current best individual from the niche ($l_{best}$) and the best individual from the whole population ($g_{best}$) are used to produce new individuals. The results show that the developed mutation operator can generate better feature subsets and speed up the convergence.
2) A subset repairing mechanism is proposed to modify the duplicated feature subsets. In a duplicated feature subset, at least half of the original selected features is retained. The reason is that multiple duplicated feature subsets may select some common useful features. The results show that the proposed subsets repairing scheme can increase the population diversity and improve the feature selection performance.
3) A selection operator is proposed for removing the possible duplicated feature subsets if the population still has the duplicated solutions after performing the subset repairing mechanism. Furthermore, the feature subset with a lower size during selection is preferred. The results show that the proposed selection operator can help the proposed method achieve a fitter fitness value and better feature selection performance.

The remainder of this article is organized as follows. Section II presents the related work. Section III discusses the proposed NDEDA algorithm. Next, the experiments are conducted in Section IV. The effectiveness of NDEDA is demonstrated in Section V. This article is concluded in Section VI.

## II. RELATED WORK

### *A. Typical Feature Selection Methods*

Feature selection methods can be separated into wrapper-based, filter-based, and embedded-based methods according to the evaluation criteria [27]. A wrapper method employs a specific learning algorithm such as KNN to evaluate feature subsets, while a filter method, for example, the correlated feature selection method [28] or the fast correlation method (FCBF) [29], typically utilizes certain statistical measures instead of a learning algorithm to evaluate the goodness of a feature subset. In the embedded methods, for example, the robust feature selection method [30] and genetic programming [31], feature selection is embedded into the learning process of a classifier. Once the learning process is completed, the features used in the classifier are selected. Generally, a wrapper or an embedded method can usually achieve better classification accuracy than a filter method but with the price of longer computation time and less generality [2].
ReliefF [32] and FOCUS [33] are two popular filter-based feature selection methods. A representative of the wrapper-based methods is the minimum redundancy maximum relevance method [34] which selects features with the highest relevance to the class while the lowest redundancy among the features. In addition, sparsity regularization in dimensionality reduction, for example, [30] or [35], belongs to the embedded-based methods. However, most classical feature selection methods require to define the number of selected features in advance. Moreover, those methods can only output one feature subset ignoring the existence of multiple optimal feature subsets. Due to the global search ability and without using any prior knowledge, EC methods, for example, DE, are popular in designing feature selection methods.

*B. EC-Based Feature Selection Methods*

Niching-based EC methods are efficient in addressing multimodal numerical/benchmark optimization problems, where there are multiple optimal solutions. In order to find multiple optimal feature subsets, some niching techniques are also used for feature selection. We attempt to describe them in the following aspects.

*1) Typical Niching Techniques:* Some well-known niching techniques, including crowding [13], speciation [15], and sharing [14] methods, have been incorporated with EC algorithms to search for multiple optima. In the crowding methods, the environmental constraint is constructed by sampling a small number of individuals from the population. Offspring will compete with the nearest parent in the sample size, and the better one will survive. In the speciation methods, the fittest members in the whole population are taken as seeds, bounded by a radius to envelop the members of their species. Sharing methods typically divide the whole population based on the distance or other standards between individuals. An individual is compelled to share its fitness with other members in the same niche except for the fittest one. Many crowding-based and speciation-based DE algorithms have been proposed, such as neighborhood-based crowding DE (NCDE) [16], neighborhood-based speciation DE (NSDE) [16], and local binary pattern-based adaptive DE (LBPADE) [22]. However, these methods cannot handle well high-dimensional problems or problems with many local optima [18], [20]. To improve the performance, Wang et al. [18] designed an automatic niching DE method to form the niches via affinity propagation clustering. Chen et al. [19] proposed a distributed individuals DE algorithm where each individual is assigned to a virtual subpopulation to find the optimal solution in a small region. Jiang et al. [20] applied entropy measurement to determine the niche centers. When applying these methods to feature selection, they might not be able to achieve the promising performance due to the existence of duplicated feature subsets. In addition, some clustering-based niching methods may have high computational costs, especially for high-dimensional datasets.
To find multiple optimal feature subsets to a classification task, Kamyab and Eftekhari [8] proposed a sharing-based particle swarm optimization (PSO) [36] method. The proposed method can find different feature subsets with the same classification accuracy, but the number of features in most of the used datasets is smaller than 100. Yue et al. [5] proposed a ring topology-based PSO method for solving feature selection problems. However, the method requires a large population of individuals that are uniformly distributed in the search space for the purpose of forming stable species. Wang et al. [37] proposed a niching-based DE method for feature selection, but the method suffers from high computation costs because of the complex environmental selection process.

*2) DE for Feature Selection:* Real-valued or float number encoding is commonly used when applying EC methods to feature selection. In [38], an individual in DE is encoded with float numbers which are rounded to the nearest integers (the feature index). To avoid a certain feature being used more than once in a solution, Khushaba et al. [38] proposed a repair mechanism based on the selected frequency of features. The results on 12 gene datasets showed that the proposed method obtained better performance than other EC methods, including PSO and genetic algorithms (GAs) [39]. Zhang et al. [40] proposed a binary mutation mechanism and one-bit purifying search in DE for feature selection. The proposed mutation approach randomly selects three individuals from the whole population, and one is taken as a base vector while the remaining two are used to determine the mutation probability. The

proposed method obtains promising results on 20 datasets, but it has high computational complexity. To predict heart disease, a modified DE-based feature selection algorithm was proposed in [41]. Results showed that the variant algorithm can achieve better classification performance than standard DE on the cardiovascular disease dataset.

*3) Other EC Methods for Feature Selection:* A sticky PSO method was developed in [42] where the update of the position of particles is determined by the so-called *stickiness* property and previous location without the item of velocity to address knapsack and feature selection problems. Tran et al. [43] proposed a wrapper-based variable-length PSO representation to reduce computing consumption. After dividing the features in a dataset into different clusters/groups, Song et al. [44] employed integer encoding to address feature selection tasks. Xue et al. [45] introduced a GA-based ensemble feature selection method where ten extreme learning machines with multiple near-optimal feature subsets from the final-generation are used. The ensemble method uses majority voting to further improve the prediction accuracy.

Recently, researchers transformed a feature selection task into a multi objective optimization problem and/or a sparse optimization problem. To simultaneously minimize the classification error rate and minimize the number of selected features, Xue et al. [46] developed the first study using a multi objective DE method addressing feature selection problems. Experiments on nine datasets showed that the proposed method outperformed some classical feature selection methods. Tian et al. [26] proposed a sparse feature selection algorithm which can obtain a set of feature subsets with high sparsity.

In summary, many current feature selection methods ignored the existence of multiple optimal feature subsets. Although many niching-based EC methods have been proposed and obtained good performance in addressing multimodal continuous function optimization problems, they are seldom used for addressing feature selection tasks. Furthermore, several niching-based feature selection methods, for example, [5] and [8], need to be improved because they ignore the negative impact of duplicated feature subsets on the algorithm. Therefore, in the following section, we proposed a new method called NDEDA, which employs a niching-based DE technique and analyzes duplications of solutions in the population to search for multiple optimal feature subsets.

**Algorithm 1:** NDEDA

**Input**: *MaxFe*: Maximal number of fitness evaluations,
*N*: Population size
**Output**: A set of feature subsets

```
1  begin
2      Initialize population with size N,
3      Set the current number of fitness evaluations fe as 0,
4      Get the fitness values of individuals via Eq. (1), fe = fe + N,
5      while fe < MaxFe do
6          for i = 1 to N do
7              Find the |N| neighbors of x_i to form a niche,
8              Generate a mutant vector via Eq. (2),
9              Generate a trial vector via Eq. (3),
10         end
11         Perform subset repairing scheme via Alg. 2,
12         Calculate the fitness values of new individuals via Eq. (1),
13         Get fe = fe + N,
14         Select N individuals to enter the next generation,
15     end
16     Output optimal and near-optimal feature subsets.
17 end
```

## III. PROPOSED METHOD

### *A. Overall Algorithm*

Algorithm 1 shows the overall NDEDA algorithm. In the beginning, the population is randomly initialized (line 2). Then, the population undergoes the evolution process (lines 5–15). To produce promising candidate feature subsets, the mutation operator of NDEDA considers the fitness landscape of individuals (lines 7 and 8). Furthermore, to improve the population diversity, the proposed subset repairing scheme is adopted (line 11). The evolutionary learning process continues until the stop condition is met (line 5).

### *B. Representation and Fitness Function*

To deal with a feature selection task, each individual is represented as a vector with real-valued encoding (limited in [0, 1]). The number of original features in a dataset is the dimensionality of a vector, called D. A preset threshold $\theta$ is utilized to determine the feature is chosen or not. For example, in an individual $\vec{x} = (x_1, x_2, \ldots, x_D)$, if $x_d \geq \theta$ ($d \in [1, D]$), the corresponding feature will be selected; otherwise, not selected.

The fitness function is shown in (1), and λ is a small constant which is to ensure the size of a feature subset is considered when the classification error rate of multiple feature subsets is almost the same

$$f = ER_{\text{SVM}} + \lambda * \text{Size} \quad (1)$$

where $ER_{SVM}$ represents the classification error rate obtained by an SVM classifier using 5-fold cross-validation. This ensures that the discriminating ability of the subset is evaluated through a robust margin-based learning algorithm, providing more stable fitness values in high-dimensional spaces compared to distance-based methods. In addition, Size is the number of selected features by a solution. Followed by the setting in [47] and [48], λ is set to 1e - 6.

*C. Niching-Based Mutation*

To find multiple optimal feature subsets, the developed mutation strategy combines the local information from the niche and the global information from the whole population. Furthermore, the niching-based mutation is used as a stable and an effective way to exchange evolutionary information between solutions [49].

This work employs *k*-nearest neighbors of an individual $\vec{x}_i$ to form a niche. Hamming distance is used to simulate the surrounding information of the solution space. More specifically, the KNNs of an individual $\vec{x}_i$ is kept in a set $N_i$, $|N_i| = k$, and $k$ can be set to 8 which is similar to the regular settings in [22]. The fitness values of the individuals in the same niche are compared. The individuals in $\boldsymbol{Ni}$ with fitter fitness value than the current individual $\vec{x}_i$ will be selected and stored in another set $S_i$. The developed mutation strategy is as follows:

$$\vec{v}_i = \begin{cases} \vec{x}_i + F * (\vec{x}_{lbest} - \vec{x}_i) + F * (\vec{x}_{g_{r1}} - \vec{x}_{g_{r2}}) & if\ |S_i| \geq |N_i|/2 \\ \vec{x}_i + F * (\vec{x}_{gbest} - \vec{x}_i) + F * (\vec{x}_{l_{r1}} - \vec{x}_{l_{r2}}) & othrwise \end{cases} \quad (2)$$

where $\vec{x}_{\text{lbest}}$ and $\vec{x}_{\text{gbest}}$ mean the individual with the best fitness value in the niche $N_i$ of $\vec{x}_i$ and in the whole population P, respectively. Meanwhile, $g_{r1}$ and $g_{r2}$ are randomly selected from P ($g_{r1} \neq g_{r2} \neq \vec{x}_{\text{lbest}} \neq i$), while $l_{r1}$ and $l_{r2}$ are randomly selected from $N_i$ ($l_{r1} \neq l_{r2} \neq \vec{x}_{\text{gbest}} \neq i$). Scaling factor F, $F \in [0, 1]$, controls the learning degree of $\vec{x}_i$ from the others.

Two situations are considered in the proposed mutation mechanism. One situation is $|S_i| \geq |N_i|/2$ in (2), which means half or more solutions in the niche of $\vec{x}_i$ have better fitness value than $\vec{x}_i$. In this situation, $\vec{x}_{\text{lbest}}$ from the current niche can be used to guide $\vec{x}_i$, which can enhance the exploitation ability of $\vec{x}_i$. Meanwhile, $g_{r1}$ and $g_{r2}$ as the global disturbance are added in order to prevent being trapped into a local optimum. The second situation is $|S_i| < |N_i|/2$, which means not so many solutions have better fitness value than $\vec{x}_i$. In this situation, $\vec{x}_{\text{gbest}}$ from the whole population can be used to guide $\vec{x}_i$, which can enhance the exploration ability of $\vec{x}_i$ and locate the potential optima quickly. Meanwhile, $l_{r1}$ and $l_{r2}$ as the local disturbance are added in order to retain the local exploitation information. Both the proposed mutation operator in this work and the mutation operator in [22] consider the local information from the niche and the global information from the whole population. To better maintain the global search ability, $\vec{x}_{\text{gbest}}$ is adopted to guide the individuals in the proposed mutation operator. Furthermore, more stringent condition (i.e., $|S_i| \geq |N_i|/2$) is set when using $\vec{x}_{\text{lbest}}$ to update $\vec{x}_i$, which is to avoid falling into the local optima. The comparisons between the method [22] and the proposed NDEDA method will be shown in Section V-A. Furthermore, the analysis using only the information from $\vec{x}_{\text{gbest}}$ or $\vec{x}_{\text{lbest}}$ for the mutation will be shown in Section V-D.

*D. Crossover*

After performing the mutation operator, the crossover operator is performed on the mutant vector $\vec{v}_i$ and the individual $\vec{x}_i$ to produce a trial vector $\vec{u}_i$, as shown in

$$\vec{u}_{i,j} = \begin{cases} \vec{v}_{i,j} & if\ (rand(0,1) \leq CR\ or\ (j = j_{rand}) \\ \vec{x}_{i,j} & othrwise \end{cases} \quad (3)$$

where $j_{rand}$ is a random integer within 1 and $D$ (the dimensionality of the problem), and $rand(0,1)$ is a random number in the range of 0 and 1. Crossover rate CR, $CR \in [0, 1]$, controls the information learned from the mutant vectors.

*E. Subset Repairing Scheme*

After performing mutation and crossover operators, there may be some duplicated feature subsets. To address this issue, a subset repairing mechanism is proposed to generate new feature subsets that have never appeared so far. The proposed mechanism randomly replaces the selected and unselected positions to modify a duplicated feature subset.

The proposed mechanism not only increases the diversity of the population but also enhances the search ability of NDEDA.

**Algorithm 2:** Subset Repairing Scheme

**Input**: Parental set $\boldsymbol{P}$, offspring set $\boldsymbol{O}$, unique set $\boldsymbol{U}$
**Output**: Repaired offspring set $\boldsymbol{O}$

```
1  begin
2      U = U ∪ P, and remove duplicated solutions from U,
3      for i = 1, ..., |O| do
4          if O(i) is a duplicated feature subset in U then
5              Get the inverted number N_iv via Eq. (4),
6              Randomly select N_iv positions from the selected features
               and the unselected features, termed R_s and R_u,
               respectively,
7              Produce a feature subset x_new via Eq. (5),
8              if x_new ∉ U then
9                  Put x_new to U, and set O(i) = x_new
10             else
11                 Return Line 5
12             end
13         else
14             Put O(i) to U
15         end
16     end
17 end
```

Algorithm 2 presents the details of the proposed subset repairing mechanism in one-generation. Let set $\boldsymbol{U}$ store all the unique feature subsets in the solution space, and its size increases with evolution. Noted that if an offspring is duplicated with the solution in $\boldsymbol{U}$, the duplication can include two situations, that is, the *parent-offspring* duplication and the *offspring-offspring* duplication (line 4 of Algorithm 2). For both situations, $\vec{x}_{dup}$ will be modified by using (4) and (5). The modification process will not stop until a new unique feature subset is produced (lines 8–12 of Algorithm 2). Here, a threshold $\tau$ for stopping the while-loop iteration can be additionally set so as not to get lost in infinite looping. The sensitivity analysis of $\tau$ will be shown in Section IV-B.

To maintain part of the original selected features from $\vec{x}_{dup}$, the number of positions to be inverted, that is, $\vec{N}_{iv}$, is calculated by

$$\vec{N}_{iv} = \begin{cases} 1 & if\ |S| \leq 2 \\ rand[1, \min(|S|/2, D-S)] & othrwise \end{cases} \quad (4)$$

where $S$ means the number of selected features in $\vec{x}_{dup}$, and $D$ is the original number of features in the dataset. The symbol $\lfloor\ \rfloor$ means rounding down. Noted that if $S$ is not greater than 2 *Niv* will be set to 1

$$\vec{x}_{new,j} = \begin{cases} rand\ [0, \theta) & for\ j \in R_s \\ rand[\theta, 1] & for\ j \in R_u \end{cases} \quad (5)$$

where $\vec{x}_{new,j}$ means the $j$th dimension of $\vec{x}_{new}$. Meanwhile, $R_s$ and $R_u$ represent $\vec{N}_{iv}$ numbers randomly selected from $I_s$ and $I_u$, respectively. Here, $I_s$ stores the indexes whose position value in $\vec{x}_{dup}$ is no less than $\theta$, and $I_u$ stores the indexes whose position value in $\vec{x}_{dup}$ is smaller than $\theta$.

In (4), $rand[1, \min(|S|/2, D-S)]$ is lower or equal to $S/2$ which means that at least half of the selected features from $\vec{x}_{dup}$ will be retained to $\vec{x}_{new}$ when the number of selected features is more than 1. This is to maintain part of the useful features in the duplicated solutions. The minimal value of *Niv* in (4) is set to 1, which means that if only one feature is selected from $\vec{x}_{dup}$, the newly produced solution $\vec{x}_{new}$ will randomly select another different feature. In this way, the proposed subset repairing mechanism can modify the duplicated solutions while maintaining their original sizes.

An example in Fig. 2 is given to show the process of modifying the two duplicated feature subsets, that is, feature subsets *S*6 and *S*7 in Fig. 1. For *S*6, $I_s = [1,5]$ and $I_u = [2,3,4]$, since features *F*1 and *F*5 are selected while *F*2–*F*4 are not selected. Based on (4), *Niv* of *S*1 is 1. Therefore, one position from $I_u$ and *Is* will be randomly chosen, respectively. Suppose that the selected position from $I_u$ is 2 and that from $I_s$ is 5, $R_u = [2]$ and $R_s = [5]$. The value of the second dimension in *S*6 will be replaced by a randomly generated number between $\theta$ and 1. Meanwhile, the value of the fifth dimension in *S*6 will be replaced by a randomly generated number between 0 and $\theta$. Then, a new feature subset is formed to replace *S*6. A similar way is used to modify *S*7 in Fig. 2.

The proposed subset repairing scheme does not depend on the encoding method. For other methods using different encoding mechanisms, the subset repairing scheme can also be used. Take binary encoding as an example. When modifying the duplicated feature subsets in a method using binary encoding, the only minor change is in (5). In other

words, $rand\ [0, \theta)$ will be changed as 0, and $rand[\theta, 1]$ will be set to 1. The remaining parts can be directly applied.

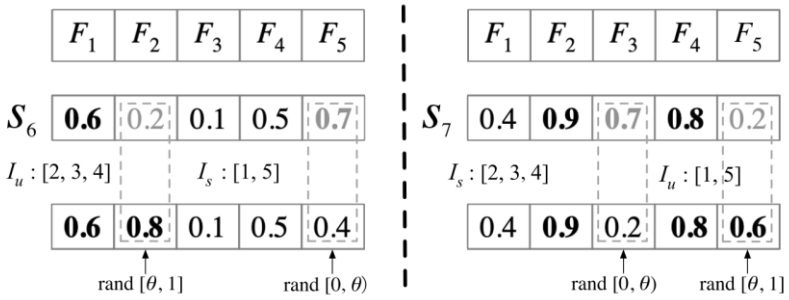


Fig. 2. Example of modifying the two duplicated feature subsets from Fig. 1.

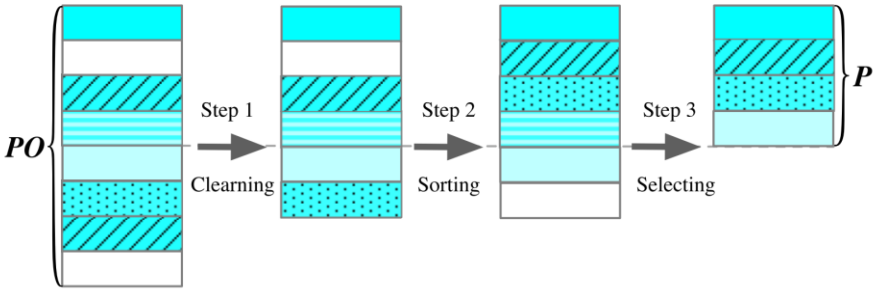


Fig. 3. Selection process to form a new population. In each column, the squares with the same color and stripe mean the duplicated feature subsets.

*F. New Selection Strategy*

To select a certain number of individuals to enter the next generation, the current population P and offspring set O are combined as a new set PO. As shown in Fig. 3, the developed selection strategy includes three steps. The first step is to clear PO. There might be some duplicated feature subsets still in PO after performing the subset repairing scheme. Therefore, those possible duplicated solutions will be cleared. Followed by the work in [27], among the multiple duplicated feature subsets, the solution with the highest confidence score ($C_f$) will be picked and the remaining solutions will be deleted. The $C_f$ value of a solution $\overrightarrow{x_i}$ is calculated as follows:

$$C_f(\overrightarrow{x_i}) = \sum_{j=1}^{D} Conf(j) \qquad (6)$$

$$Conf(j) = \begin{cases} \dfrac{x_{i,j} - \theta}{1 - \theta}, & x_{i,j} > \theta \\ \dfrac{\theta - x_{i,j}}{\theta}, & x_{i,j} \le \theta \end{cases} \qquad (7)$$

where Conf(j) means the confidence degree in the decision of feature selection for the jth feature. Meanwhile, $x_{i,j}$ is the jth dimension of $\overrightarrow{x_{i,j}} \in [1, D]$, and D means the number of features in a dataset. In (6), Cf $(\overrightarrow{x_i})$ means the sum of the confidence degree of $\overrightarrow{x_i}$. The larger the Cf value, the higher the confidence, and the less likely the decision is changed.

After finding and choosing the most confident solution from each duplicated group, the second step is to sort all individuals in PO in an ascending order based on their fitness values.

Finally, the top N individuals will be selected. Noted that the (N + 1)th individual in PO might have the same fitness value as the Nth individual. For this situation, NDEDA will choose the one with a smaller size, that is, a smaller number of features is selected. Suppose that after Step 2 in Fig. 3, the last three feature subsets have the same fitness value. The solution with a smaller size will be selected, and the top three individuals are added to form a new population.

## IV. EXPERIMENT DESIGN

### *A. Benchmark Techniques*

To validate the effectiveness of the proposed algorithm, NDEDA is compared with five multimodal algorithms. They are *r*3pso [50], NCDE [16], NSDE [16], neighborhood-based sharing DE (NShDE) [16], and LBPADE [22]. Furthermore, a sinusoidal-based binary dragonfly (SBDA) method [51] and a variable-size cooperative coevolutionary PSO method (VS-CCPSO) [52] are compared with NDEDA. NDEDA is also compared with two traditional feature selection methods. They are a correlation-based filter method using symmetrical uncertainty (FCBF) [29] and a sparse multinomial logistic regression method using a Bayesian *l*1 regularization (SBMLR) [35]. The two traditional methods can automatically determine the number of features to select.

Table I. INFORMATION OF THE USED DATASETS

| Number | Dataset | # Features | # Classes | # Instances |
|---|---|---|---|---|
| 1 | Zoo | 16 | 7 | 101 |
| 2 | WBCD | 30 | 2 | 569 |
| 3 | Ionosphere | 34 | 2 | 351 |
| 4 | Movement | 90 | 15 | 360 |
| 5 | Hillvally | 100 | 2 | 1,212 |
| 6 | Musk1 | 166 | 2 | 2,031 |
| 7 | Multiple (pix) | 240 | 10 | 2,000 |
| 8 | Arrhythmia | 279 | 16 | 452 |
| 9 | CNAE | 856 | 9 | 1,080 |
| 10 | QSAR | 1,024 | 2 | 3,000 |
| 11 | AD | 1,558 | 2 | 3,279 |
| 12 | SRBCT | 2,308 | 4 | 83 |
| 13 | Leukemia | 5,147 | 2 | 72 |
| 14 | Leukemia1 | 5,327 | 3 | 72 |
| 15 | 9Tumor | 5,726 | 9 | 60 |
| 16 | DLBCL | 7,050 | 2 | 77 |
| 17 | Leukemia2 | 11,225 | 3 | 72 |
| 18 | 11Tumor | 12,533 | 11 | 174 |

### *B. Datasets and Parameter Settings*

Table I shows the 18 datasets used in the experiments, which are selected to have various numbers of features (from 16 to 12 533), classes (from 2 to 16), and instances (from 60 to 3279). Each dataset is randomly divided into a training set and a test set with the proportions of 70% and 30%, respectively, [53]. The calculation of the classification error rate is based on a Support Vector Machine (SVM) with a linear kernel, utilizing five-fold cross-validation on the training set to ensure robust fitness evaluation. The SVM classifier is chosen to balance classification accuracy and computational efficiency while providing more stable performance in high-dimensional spaces compared to distance-based methods.

Table II PARAMETER SETTINGS

| Algorithms | Parameter Values |
|---|---|
| *r*3pso [50] | The acceleration coefficient $c_1 = c_2 = 2.05$, inertia weight $w = 0.7298$ |
| NCDE [16] | $F = 0.5$, $CR = 0.5$, crowding factor $CF = 2$ |
| NSDE [16] | $F = 0.5$, $CR = 0.5$, radius radius parameter $r_s = 2$ |
| NShDE [16] | $F = 0.5$, $CR = 0.5$, sharing radius $\sigma_{\text{share}} = 0.5$ |
| LBPADE [22] | Neighborhood window = $3 \times 3$ |
| SBDA [51] | The inertia weight $w = 0.9 - (0.5 * g)/g_{\max}$, trigonometric offset angle $\beta = \pi/2$ |
| VS-CCPSO [52] | The number of feature subspaces $M = 8$. |

For each feature selection algorithm, the number of runs is 30 on each training set. The specific parameters of the benchmark algorithms are shown in Table II. The maximal number of the fitness evaluations MaxFe is set to 100 (generations) times of N (population size) for all the datasets. Meanwhile, N is set to the number of original features in a dataset but limited to 300 as suggested in [43]. For DE-based feature selection methods including NDEDA, F in (2) and CR in (3) are set to 0.5 [37], [54]. The threshold θ for selecting features is set to 0.6 [55].

For the constant (τ) in NDEDA, the average training classification accuracy and the average number of the selected features of τ at 1, 2, 3, 4, and 5 are reported in Tables III and IV, respectively. In Table III, NDEDA with τ = 2 can achieve the best classification performance on 7 out of the 18 datasets. When τ is larger, for example, 3, 4, or 5, the performance slightly decreases on some datasets. This is because multiple modifications may be too disruptive by

potentially removing some informative features. Furthermore, $\tau = 1$ means a duplicated solution is modified only once. A too small $\tau$ value may not be enough, thus, the classification results of $\tau = 2$ are better than that of $\tau = 1$. In Table IV, NDEDA with $\tau = 3$ or with $\tau = 5$ produces smaller feature subsets than NDEDA with $\tau = 2$. However, according to the results in Table III, NDEDA with $\tau = 2$ can achieve the best overall classification performance. For feature selection tasks, the classification accuracy is more of a concern than the subset size. Therefore, this work sets $\tau$ in NDEDA to 2.

Table III TRAINING CLASSIFICATION RESULTS OF NDEDA WITH DIFFERENT $\tau$ VALUES

| Dataset | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ | $\tau = 5$ |
|---|---|---|---|---|---|
| Zoo | 95.17±0.09 | **95.33**±0.07 | 95.19±0.09 | 95.22±0.07 | 95.20±0.11 |
| WBCD | **94.89**±0.05 | **94.89**±0.05 | 94.88±0.05 | 94.87±0.06 | 94.84±0.31 |
| Ionosphere | **93.05**±0.40 | 93.02±0.38 | 92.94±0.42 | 93.01±0.40 | 92.98±0.38 |
| Movement | 75.56±1.00 | **75.88**±1.07 | 75.87±1.10 | 75.49±1.15 | 75.54±1.23 |
| Hillvally | 63.31±0.68 | 63.17±0.60 | 63.25±0.64 | **63.42**±0.70 | 63.21±0.66 |
| Musk1 | 99.36±0.14 | **99.39**±0.15 | **99.39**±0.14 | 99.35±0.17 | 99.33±0.16 |
| Multiple | 98.85±0.08 | 98.84±0.10 | **98.88**±0.10 | 98.86±0.09 | 98.85±0.08 |
| Arrhythmia | 70.97±0.94 | 70.89±0.95 | 71.11±0.86 | 71.15±0.97 | **71.25**±0.80 |
| CNAE | **91.39**±0.76 | 91.22±0.87 | 91.15±1.25 | 91.09±0.95 | 90.93±0.96 |
| QSAR | 94.84±0.11 | **94.85**±0.10 | 94.83±0.11 | **94.85**±0.13 | **94.85**±0.10 |
| AD | 97.10±0.12 | 97.13±0.11 | **97.16**±0.12 | 97.11±0.13 | 97.11±0.18 |
| SRBCT | 99.52±0.40 | 99.57±0.29 | 99.57±0.40 | **99.64**±0.24 | 99.51±0.28 |
| Leukemia | **97.72**±0.94 | 97.51±0.80 | 97.58±1.15 | 97.58±0.90 | 97.68±0.60 |
| Leukemia1 | 98.02±0.49 | 98.10±0.59 | 98.10±0.60 | **98.23**±0.74 | 98.10±0.79 |
| 9Tumor | 69.93±3.01 | 70.31±3.45 | 70.05±3.24 | **71.41**±3.33 | 70.25±2.73 |
| DLBCL | 99.04±0.9 | 98.89±0.85 | 99.01±0.86 | **99.06**±0.86 | 98.97±0.87 |
| Leukemia2 | 99.03±0.95 | **99.52**±0.68 | 98.99±0.97 | 99.05±0.95 | 98.86±0.99 |
| 11Tumor | 90.01±1.14 | **90.72**±1.42 | 90.55±1.62 | 90.52±1.34 | 90.65±1.34 |

*C. Performance Indicators*

To compare different methods, four performance indicators are utilized. They are: 1) *Ac*: the test classification accuracy; 2) *Size*: the number of selected features; 3) *Num*: the number of different feature subsets with very similar or the same training accuracy; and 4) UN: the number of unique feature subsets.

Among them, UN is an important indicator to indicate the duplication degree of the feature subsets produced by one method. The value of UN increases with generations, and it will reach the top (called MaxUN) when an algorithm stops. If the value of UN is equal to fe (the current number of fitness evaluations), there are no duplicated feature subsets produced at this generation. Generally, MaxUN is smaller than MaxFe due to the existence of the duplicated and/or reappearing feature subsets. MaxUN = MaxFe indicates no duplicated or reappearing solution is generated during training.

## V. RESULTS

The six algorithms, r3pso, NCDE, NSDE, NShDE, LBPADE, and NDEDA, aim to find multiple optimal feature subsets. However, some feature subsets could achieve very similar but different classification accuracy due to feature interaction and cross-validation. If the absolute difference in classification accuracy among two or more feature subsets is less than or equal to a small constant $\varepsilon$, these feature subsets will be considered to have the same classification performance. For each of the eight feature selection methods, in the final population, the algorithm will first find the feature subset with the lowest fitness value, called $S_{min}$. The feature subsets whose absolute difference in classification accuracy to $S_{min}$ is less than or equal to $\varepsilon$ will be output. To allow flexibility in using the classification accuracy, $\varepsilon$ is set to 1/T where T means the number of training instances on one dataset.

To ensure a fair and consistent comparison with the original study, the baseline results using KNN (Tables I-VI) are preserved as a benchmark. Meanwhile, the newly proposed SVM-based evaluations, including comparative performance and systematic ablation studies, are presented in Tables X-XIII to highlight the improvements achieved by the proposed NDEDA framework.

Tables V and VI show the average Ac and the average Size results of the obtained feature subsets of the eight algorithms on the test sets, respectively. The *F*1 results of the eight methods are shown in Online Supplementary Materials1, which reveal similar patterns as the classification accuracy results. In Tables V and VI, the highest *Ac* value or the lowest *Size* value obtained on each dataset are in bold. The signs "↑," "↓," and "*o*" indicate that the

corresponding benchmark algorithm is significantly better than, worse than or has no significant difference from NDEDA, respectively. In other words, the more ↓, the better the proposed NDEDA method. The Wilcoxon test with a significance level of 0.05 is used. Moreover, the Friedman test is employed to give the relative performance ranking among the eight algorithms.

Table IV SUBSET SIZES OF NDEDA WITH DIFFERENT $\tau$ VALUES

| Dataset | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ | $\tau = 5$ |
|---|---|---|---|---|---|
| Zoo | **5.7**±0.3 | 5.8±0.2 | 5.8±0.3 | 5.8±0.2 | 5.8±0.3 |
| WBCD | **4.0**±0.2 | 4.1±0.2 | 4.1±0.3 | 4.1±0.2 | 4.1±0.3 |
| Ionosphere | **4.1**±0.4 | 4.2±0.3 | **4.1**±0.4 | **4.1**±0.3 | 4.2±0.4 |
| Movement | 26.2±4.6 | **23.7**±3.8 | 24.4±4.3 | 25.6±5.3 | 24.3±4.4 |
| Hillvally | 25.2±4.1 | **24.8**±3.9 | 25.2±4.5 | 25.4±4.4 | 25.6±5.0 |
| Musk1 | 42.9±4.4 | 43.5±5.6 | 42.9±5.4 | **41.8**±6.0 | 44.4±6.2 |
| Multiple | 82.6±6.2 | **81.4**±6.2 | 82.2±6.0 | 81.7±6.4 | 81.6±7.6 |
| Arrhythmia | 41.4±10.3 | 42.3±8.8 | **39.8**±6.3 | 42.9±9.0 | 43.4±7.8 |
| CNAE | 292.1±22.2 | 290.2±22.8 | **253.2**±21.3 | 253.4±17.8 | 256.6±16.7 |
| QSAR | 276.6±23.3 | 273.9±17.9 | **268.2**±21.1 | 276.0±24.4 | 274.9±23.5 |
| AD | 417.2±28.5 | 419.9±22.0 | 371.6±25.3 | **367.8**±32.7 | 383.5±23.0 |
| SRBCT | 389.4±26.8 | 391.4±37.9 | 393.4±33.3 | 391.7±32.3 | **389.1**±42.9 |
| Leukemia | 836.8±60.4 | 833.4±66.3 | 849.7±59.9 | 836.0±58.1 | **832.5**±63.5 |
| Leukemia1 | 991.3±60.6 | 979.7±52.0 | 972.0±50.8 | 976.6±58.3 | 979.1±62.2 |
| 9Tumor | 1,437.2±104.9 | 1,431.3±97.2 | 1,446.3±94.7 | 1,455.5±121.8 | **1,384.3**±65.3 |
| DLBCL | 1,322.2±81.1 | 1,320.1±62.8 | **1,302.5**±65.3 | 1,323.5±90.5 | 1,322.1±87.9 |
| Leukemia2 | 2,031.3±132.6 | 2,083.2±129.0 | 2,031.6±128.9 | 2,043.7±143.9 | **1,997.7**±158.9 |
| 11Tumor | 3,188.8±170.6 | 3,223.1±233.6 | **3,142.1**±201.2 | 3,195.5±154.6 | 3,148.6±139.8 |

### *A. Results of NDEDA*

*1) NDEDA Versus r3pso and SBDA:* According to Table V, NDEDA generates feature subsets giving significantly higher classification accuracy than both *r*3pso and SBDA on 13 datasets. Furthermore, in Table VI, NDEDA selects significantly fewer features on the 18 datasets than both *r*3pso and SBDA. For example, NDEDA selects two times fewer features than *r*3pso and SBDA on the WBCD, Ionosphere, Arrhythmia, SRBCT, Leukemia, Leukemia1, DLBCL, and Leukemia2 dataset. According to the Friedman test results from Tables V and VI, *r*3pso and SBDA show the worst performance among the eight algorithms.

Table V AVERAGE RESULTS OF *Ac (%)* ON THE TEST SETS

| Dataset | *r*3pso | NCDE | NSDE | NShDE | LBPADE | SBDA | VS-CCPSO | NDEDA |
|---|---|---|---|---|---|---|---|---|
| Zoo | 89.52±4.16o | 90.25±1.38o | 90.25±0.21o | 90.21±0.31o | 89.68±1.04↓ | 90.22±2.70o | **90.43**±1.31o | 90.17±0.59 |
| WBCD | 92.74±1.26↓ | 93.73±0.92↓ | 93.68±1.14↓ | 93.54±1.29↓ | 94.10±0.15o | 94.31±0.50o | **94.76**±1.11↑ | 94.15±0.02 |
| Ionosphere | 80.91±2.29↓ | 85.56±2.97↓ | 87.51±2.84o | 87.49±3.19o | 85.25±2.67↓ | 80.13±2.24↓ | 86.95±3.48o | **88.15**±1.87 |
| Movement | 75.27±2.02↓ | 76.22±1.96↓ | 76.13±1.54↓ | 76.68±1.43o | 77.07±1.16o | 76.48±1.60↓ | 73.98±2.25↓ | **77.39**±1.55 |
| Hillvally | 52.28±1.78↓ | 54.15±1.45↓ | 54.09±1.29↓ | 54.38±1.35↓ | 54.53±1.15↓ | 53.41±1.95↓ | 54.93±1.84o | **55.38**±1.29 |
| Musk1 | 96.25±0.56↓ | 97.49±0.47↓ | 97.68±0.42↓ | 97.57±0.38↓ | 97.61±0.53↓ | 96.46±0.54↓ | 97.95±0.56o | **98.08**±0.58 |
| Multiple | 96.65±0.37o | 96.61±0.32o | 96.64±0.37o | **96.74**±0.34o | 96.58±0.39o | 96.73±0.38o | 96.49±0.38o | 96.40±0.35 |
| Arrhythmia | 57.95±2.06↓ | 61.76±2.20o | **63.48**±2.29↑ | 62.53±2.61o | 62.32±1.73o | 59.58±1.83↓ | 59.85±2.77↓ | 62.14±1.92 |
| CNAE | 76.34±5.82↓ | 84.69±2.52↓ | 86.22±1.82↓ | 86.23±1.29↓ | 84.27±1.54↓ | 81.52±2.20↓ | 74.83±1.15↓ | **87.28**±1.53 |
| QSAR | 93.52±0.35↓ | 93.60±0.25↓ | 93.72±0.25↓ | 93.65±0.21↓ | 93.53±0.30↓ | 93.56±0.32↓ | 93.87±0.35o | **93.93**±0.26 |
| AD | 96.92±0.33↓ | 97.22±0.38↓ | 97.19±0.47↓ | 97.10±0.41↓ | 97.30±0.34↓ | 96.91±0.47↓ | 97.36±0.50↓ | **97.69**±0.28 |
| SRBCT | 92.45±2.30↓ | 95.31±1.69↓ | 96.29±1.60↓ | 95.69±1.54↓ | 96.10±1.72↓ | 94.00±2.88↓ | 89.60±7.05↓ | **98.95**±1.59 |
| Leukemia | 81.89±5.11↓ | 86.61±2.29↓ | 87.71±2.25o | 87.49±2.35↓ | 87.07±3.25↓ | 85.61±4.40↓ | **91.67**±5.26o | 89.31±3.96 |
| Leukemia1 | 73.11±7.02↓ | 79.86±4.22o | **81.72**±5.09↑ | 79.00±3.58o | 77.95±4.17o | 75.91±7.06o | 67.27±4.61↓ | 78.70±6.42 |
| 9Tumor | 41.17±4.96↓ | 44.85±4.21↓ | 44.45±5.26↓ | 44.86±6.10↓ | 45.34±6.23↓ | 45.93±7.30↓ | 37.41±7.98↓ | **49.52**±6.72 |
| DLBCL | 97.09±1.23↓ | 97.59±1.27o | 97.27±1.63o | 97.74±1.33o | 97.26±1.04↓ | **98.47**±2.01o | 94.58±3.90↓ | 98.30±2.70 |
| Leukemia2 | 89.67±1.19↓ | 90.06±0.99↓ | 90.17±0.92↓ | 90.21±1.23↓ | 90.64±0.91↓ | 90.15±1.69↓ | 92.27±4.41o | **94.18**±2.78 |
| 11Tumor | 73.97±2.97↓ | 75.23±1.80↓ | 75.55±2.83↓ | 75.20±2.71↓ | 76.41±2.79↓ | 76.23±3.18↓ | 77.99±3.62↓ | **79.46**±3.36 |
| Rank | 7.00 (0/16/2) | 4.86 (0/13/5) | 3.92 (2/11/5) | 3.97 (0/11/7) | 4.39 (0/13/5) | 5.11 (0/13/5) | 4.50 (1/9/8) | 2.25 |

*2) NDEDA Versus NCDE, NSDE, NShDE, and LBPADE:* As shown in Table V, NDEDA achieves significantly better Ac performance on at least 9 out of the used 18 datasets than the four DE-based methods. On the Multiple and DLBCL datasets, although the Ac results from the three methods (NCDE, NSDE, and NShDE) and NDEDA are similar in the Wilcoxon tests, NDEDA selects fewer features. On five datasets, LBPADE presents a similar result with NDEDA on Ac, but NDEDA selects fewer features on average apart from the Zoo dataset. The highest dimensionality reduction can be seen on the 11Tumor dataset. NDEDA selects 1100 fewer features than LBPADE and still has a significantly larger Ac value on average. The main reason is that although both LBPADE and NDEDA employ the current best individual

from the niche ($l_{best}$) to guide the search in mutation, NDEDA also considers the best individual from the whole population ($g_{best}$) to produce new individuals. Based on the results in Tables V and VI, NDEDA is superior to the four DE methods.

Table VI AVERAGE NUMBER OF SELECTED FEATURES *(Size)*

| Dataset | *r*3pso | NCDE | NSDE | NShDE | LBPADE | SBDA | VS-CCPSO | NDEDA |
|---|---|---|---|---|---|---|---|---|
| Zoo | 6.9±1.1↓ | 5.5±0.5↑ | **5.2**±0.4↑ | **5.2**±0.3↑ | 5.7±0.6o | 7.0±1.5↓ | 5.6±1.2o | 5.8±0.3 |
| WBCD | 12.2±2.3↓ | 11.8±1.7↓ | 11.4±1.7↓ | 11.6±1.7↓ | 4.9±0.7↓ | 14.2±2.7↓ | 8.6±2.1↓ | **4.1**±0.3 |
| Ionosphere | 12.2±3.3↓ | 5.5±1.8↓ | 4.8±1.7o | 5.3±1.7↓ | 5.3±1.5↓ | 13.8±3.6↓ | 4.5±1.1o | **4.2**±0.4 |
| Movement | 36.0±3.6↓ | 34.5±3.8↓ | 34.9±4.0↓ | 33.7±4.0↓ | 32.2±3.5↓ | 44.3±5.2↓ | **20.7**±4.7↑ | 23.7±3.8 |
| Hillvally | 40.2±4.6↓ | 35.9±4.8↓ | 32.9±3.6↓ | 34.0±3.3↓ | 31.8±3.9↓ | 44.5±4.0↓ | **20.0**±5.9↑ | 24.8±3.9 |
| Musk1 | 69.6±5.0↓ | 63.1±4.8↓ | 60.9±4.4↓ | 61.0±4.6↓ | 60.0±3.6↓ | 82.8±6.8↓ | **40.9**±5.3o | 43.5±5.6 |
| Multiple | 101.9±6.8↓ | 100.1±6.5↓ | 99.9±5.8↓ | 105.2±5.9↓ | 95.6±4.5↓ | 123.2±7.2↓ | **78.0**±6.9↑ | 81.4±6.2 |
| Arrhythmia | 117.2±7.3↓ | 98.5±5.7↓ | 97.3±7.1↓ | 100.1±6.7↓ | 89.5±6.6↓ | 134.1±7.8↓ | **25.9**±5.8↑ | 42.3±8.8 |
| CNAE | 364.2±14.8↓ | 347.4±11.7↓ | 347.8±12.1↓ | 353.9±13.2↓ | 321.9±22.0↓ | 435.9±10.8↓ | **146.4**±14.9↑ | 290.2±22.8 |
| QSAR | 416.0±16.7↓ | 388.5±12.5↓ | 383.1±13.7↓ | 391.5±11.2↓ | 352.8±18.7↓ | 511.4±14.9↓ | **226.6**±38.2↑ | 273.9±17.9 |
| AD | 639.0±17.5↓ | 589.4±17.7↓ | 572.8±18.1↓ | 592.5±23.1↓ | 532.9±32.1↓ | 779.0±20.9↓ | **325.8**±35.0↑ | 419.9±22.0 |
| SRBCT | 954.3±21.0↓ | 878.6±13.5↓ | 854.3±17.3↓ | 868.5±11.5↓ | 759.8±43.4↓ | 1,147.9±22.0↓ | **40.6**±7.1↑ | 391.4±37.9 |
| Leukemia | 2,060.4±36.7↓ | 1,879.4±28.7↓ | 1,750.8±26.1↓ | 1,856.7±21.1↓ | 1,424.2±144.9↓ | 2,577.3±33.1↓ | **159.2**±56.9↑ | 833.4±66.3 |
| Leukemia1 | 2,148.5±43.7↓ | 1,984.0±22.7↓ | 1,868.1±35.1↓ | 1,977.2±37.3↓ | 1,595.2±137.6↓ | 2,652.7±33.1↓ | **264.0**±100.5↑ | 979.7±52.0 |
| 9Tumor | 2,347.4±43.0↓ | 2,201.2±46.1↓ | 2,158.7±47.6↓ | 2,204.2±33.5↓ | 1,985.3±133.1↓ | 2,858.9±47.0↓ | **671.8**±163.8↑ | 1,431.3±97.2 |
| DLBCL | 2,909.6±39.2↓ | 2,681.2±27.0↓ | 2,586.5±32.8↓ | 2,635.9±19.4↓ | 2,186.6±119.4↓ | 3,530.3±36.3↓ | **555.9**±270.5↑ | 1,320.1±62.8 |
| Leukemia2 | 4,560.1±54.2↓ | 4,205.1±45.8↓ | 4,030.1±78.4↓ | 4,190.7±46.4↓ | 3,069.6±252.6↓ | 5,612.7±41.8↓ | **589.3**±174.7↑ | 2,083.2±129.0 |
| 11Tumor | 5,177.2±135.6↓ | 4,843.2±67.9↓ | 4,777.5±123.5↓ | 4,835.1±58.6↓ | 4,382.1±305.5↓ | 6,266.0±49.3↓ | **1,511.6**±465.1↑ | 3,223.1±233.6 |
| Rank | 6.89 (0/18/0) | 5.64 (1/17/0) | 3.89 (1/16/1) | 5.03 (1/17/0) | 3.11 (0/17/1) | 8.00 (0/18/0) | 1.33 (14/1/3) | 2.11 |

*3) NDEDA Versus VS-CCPSO:* VS-CCPSO significantly outperforms NDEDA only on the WBCD dataset in terms of *Ac* results. However, NDEDA selects more features than VSCCPSO on 14 datasets. On the CNAE, SRBCT, and 9Tumor datasets, although NDEDA selects more features, it improves the accuracy by more than 9% than VS-CCPSO. There are two main strategies in VS-CCPSO resulting in smaller feature subsets. One is the removal operator. Before training, part of the original features in a dataset has already been removed in VS-CCPSO. Another one is the cooperative coevolutionary operator. VS-CCPSO groups feature into different clusters, and candidate feature subsets are formed by picking up features from each group.

To have a further analysis, the removal operator of VSCCPSO is added to NDEDA, and the newly formed method is called r-NDEDA. Specifically, both r-NDEDA and VSCCPSO evaluate the correlation between each feature and the class labels using symmetric uncertainty. Features whose correlations to the class labels are lower than $0.1 \times cd_{max}$, where $cd_{max}$ is the maximal feature importance of the current dataset, will be removed. The performance of VS-CCPSO, NDEDA, and r-NDEDA is shown in Online Supplementary Materials. As shown in Table S.I, based on the Friedman test, both NDEDA and r-NDEDA show better rankings than VS-CCPSO in terms of the test classification accuracy. Furthermore, NDEDA and r-NDEDA, respectively, achieve the highest Ac values on eight and seven datasets among the three methods. In Table S.II, r-NDEDA selects significantly fewer features than NDEDA on seven datasets. On the CNAE, QSAR, and AD datasets, r-NDEDA, respectively, selects three times fewer features than NDEDA and VS-CCPSO, and the classification accuracies among the three methods are close. The results indicate that the removal operator can further reduce the number of selected features. However, some informative features might be deleted by the removal operator. In addition, VS-CCPSO with both the removal and the cooperative coevolutionary operators have even smaller feature subsets, but the accuracy is scarified.

In summary, NDEDA wins 190, draws 42, and losses 20 out of the 252 comparisons. The results show that NDEDA not only achieves higher *Ac* results than the other seven algorithms but also selects fewer features except for VS-CCPSO.

*B. Comparisons With Traditional Methods*

Table VII shows the returned feature subset size, the best and mean *Ac* of each method. As a reference, the performance of using all features on each dataset is also given in Table VII. The last column (Column *W)* shows the results of the Wilcoxon test compared the corresponding method over NDEDA using the same symbols and meanings as in Tables V and VI. The smallest size, the highest average, and the best *Ac* obtained on each dataset are in bold.

*1) NDEDA Versus Full:* In Table VII, the number of features selected by NDEDA on all datasets is one order of magnitude smaller than the original size. The features selected by NDEDA obtain significantly higher classification accuracy than using all features on 13 out of the 18 datasets. On the SRBCT, Leukemia1, and 11Tumor datasets, the feature subsets from NDEDA obtain 10% higher accuracy than *Full* on average and 16% higher in the best case.

*2) NDEDA Versus FCBF and SBMLR:* In Table VII, both FCBF and SBMLR select less than 306 features on all datasets, obtaining significantly smaller subset sizes than NDEDA on most datasets. However,

NDEDA achieves significantly higher classification accuracy than both FCBF and SBMLR on 13 datasets with about 10% or even more improvements on four datasets (Zoo, CNAE, 9Tumor, and 11Tumor). On the 9Tumor dataset, although FCBF and SBMLR use only less than 50 features, NDEDA achieves more than 16% higher accuracy than both methods on average and 18% higher in the best case by using over one thousand features. On three small datasets (Zoo, WBCD, and Ionosphere), NDEDA has the smallest size and the highest accuracy. On DLBCL, although both NDEDA and FCBF achieve the top accuracy, FCBF selects significantly fewer features. On the Hillvally, QSAR, and Leukemia1 datasets, the average classification accuracy between SBMLR and NDEDA is similar, but the average size of feature subsets from SBMLR is smaller.

TABLE VII AVERAGE TEST RESULTS OF THE THREE METHODS

| Dataset | Method | Size | Best | Mean±Std | $W$ |
|---|---|---|---|---|---|
| Zoo | Full | 16.0 | 87.10 | | ↓ |
| | FCBF | 6.0 | 80.65 | | ↓ |
| | SBMLR | 11.0 | 80.65 | | ↓ |
| | NDEDA | **5.8** | **91.40** | **90.17**±0.59 | |
| WBCD | Full | 30.0 | 91.81 | | ↓ |
| | FCBF | 5.0 | 93.57 | | ↓ |
| | SBMLR | 5.0 | 93.57 | | ↓ |
| | NDEDA | **4.1** | **94.18** | **94.15**±0.02 | |
| Ionosphere | Full | 34.0 | 79.25 | | ↓ |
| | FCBF | 9.0 | 79.25 | | ↓ |
| | SBMLR | 5.0 | 85.85 | | ↓ |
| | NDEDA | **4.2** | **90.57** | **88.15**±1.87 | |
| Movement | Full | 90.0 | 73.15 | | ↓ |
| | FCBF | **9.0** | 75.00 | | ↓ |
| | SBMLR | 66.0 | 72.22 | | ↓ |
| | NDEDA | 23.7 | **81.48** | **77.39**±1.55 | |
| Hillvally | Full | 100.0 | 51.10 | | ↓ |
| | FCBF | **1.0** | 47.53 | | ↓ |
| | SBMLR | 2.0 | 54.95 | | o |
| | NDEDA | 24.8 | **57.97** | **55.38**±1.29 | |
| Musk1 | Full | 166.0 | 95.41 | | ↓ |
| | FCBF | **3.0** | 91.15 | | ↓ |
| | SBMLR | 10.0 | 90.49 | | ↓ |
| | NDEDA | 43.5 | **99.14** | **98.08**±0.58 | |
| Multiple | Full | 240.0 | **97.33** | | o |
| | FCBF | **24.0** | 94.00 | | ↓ |
| | SBMLR | 173.0 | 97.17 | | o |
| | NDEDA | 81.4 | 97.23 | 96.40±0.35 | |
| Arrhythmia | Full | 279.0 | 54.41 | | ↓ |
| | FCBF | 8.0 | 59.56 | | ↓ |
| | SBMLR | **4.0** | 58.82 | | ↓ |
| | NDEDA | 42.3 | **65.48** | **62.14**±1.92 | |
| CNAE | Full | 856.0 | 83.02 | | ↓ |
| | FCBF | **20.0** | 69.44 | | ↓ |
| | SBMLR | 42.0 | 75.93 | | ↓ |
| | NDEDA | 290.2 | **91.39** | **87.28**±1.53 | |
| QSAR | Full | 1,024.0 | 93.33 | | o |
| | FCBF | **18.0** | 92.89 | | ↓ |
| | SBMLR | 144.0 | 93.44 | | o |
| | NDEDA | 273.9 | **94.58** | **93.93**±0.26 | |
| AD | Full | 1,558.0 | 96.04 | | ↓ |
| | FCBF | 41.0 | 95.63 | | ↓ |
| | SBMLR | **4.0** | 92.07 | | ↓ |
| | NDEDA | 419.9 | **98.28** | **97.69**±0.28 | |
| SRBCT | Full | 2,308.0 | 84.00 | | ↓ |
| | FCBF | 32.0 | 96.00 | | ↓ |
| | SBMLR | **19.0** | 96.00 | | ↓ |
| | NDEDA | 391.4 | **100** | **98.95**±1.59 | |
| Leukemia | Full | 5,147.0 | 81.82 | | ↓ |
| | FCBF | 41.0 | 90.91 | | o |
| | SBMLR | **5.0** | 72.73 | | ↓ |
| | NDEDA | 833.4 | **96.83** | 89.31±3.96 | |
| Leukemia1 | Full | 5,327.0 | 68.18 | | ↓ |
| | FCBF | 36.0 | 68.18 | | ↓ |
| | SBMLR | **14.0** | 77.27 | | o |
| | NDEDA | 979.7 | **93.43** | **78.70**±6.42 | |
| 9Tumor | Full | 5,726.0 | 50.00 | | o |
| | FCBF | 49.0 | 33.33 | | ↓ |
| | SBMLR | **15.0** | 22.22 | | ↓ |
| | NDEDAE | 1,431.3 | **61.13** | **49.52**±6.72 | |
| DLBCL | Full | 7,050.0 | 95.83 | | ↓ |
| | FCBF | 56.0 | **100** | | ↑ |
| | SBMLR | **4.0** | 95.83 | | ↓ |
| | NDEDA | 1,320.1 | **100** | 98.30±2.70 | |
| Leukemia2 | Full | 11,225.0 | 90.91 | | ↓ |
| | FCBF | 57.0 | 90.91 | | ↓ |
| | SBMLR | **14.0** | 90.91 | | ↓ |
| | NDEDA | 2,083.2 | **100** | **94.18**±2.78 | |
| 11Tumor | Full | 12,533.0 | 69.81 | | ↓ |
| | FCBF | 305.0 | 67.92 | | ↓ |
| | SBMLR | **55.0** | 69.81 | | ↓ |
| | NDEDA | 3,223.1 | **86.56** | **79.46**±3.36 | |

The results indicate that both FCBF and SBMLR scale very well in most datasets. However, the inferior results of FCBF and SBMLR show that the heuristic search might get stuck in local optima while the global search help NDEDA overcome this problem to obtain better classification performance.

*C. Further Analysis on the Obtained Feature Subsets*

This section reports the average and the standard deviation of the number (called *Num)* of feature subsets, including different features with very similar or the same classification accuracy. Noted that both SBDA and VS-CCPSO

are not niching-based feature selection methods, thus, their *Num* results are omitted. The *Num* results of FCBF and SBMLR are also omitted since both methods are deterministic methods. Each of them can only output one feature subset in a single run. The results of the remaining methods from their respective 30 runs are shown in Table VIII. Then, the feature subsets obtained from NDEDA are analyzed.

*1) Num Results:* In Table VIII, on the 18 datasets except for Zoo, NDEDA obtains the largest *Num* value compared with the five methods. On the Zoo dataset, NCDE, NSDE, and NShDE achieve the largest number (over 6) of different feature subsets with very similar or the same accuracy. On the remaining datasets, NDEDA can find significantly more feature subsets with the same classification accuracy than *r*3pso, NCDE, NSDE, NShDE, and LBPADE. The superiority of NDEDA is especially obvious on the Arrhythmia and the last seven datasets where NDEDA can find over 200 different feature subsets with the same classification accuracy, but the average *Num* value of the other compared methods is lower than 106. The results show that NDEDA can find more and better different feature subsets with similar or the same classification accuracy than the compared methods.

*2) Different Solutions With the Same Accuracy:* The results of NDEDA on the Zoo dataset are shown in Fig. 4 since it only has 16 features that are easy to visualize. The result of NDEDA on the WBCD dataset can be seen in the Online Supplementary Materials. In Fig. 4, each column (Feature) represents one of the 16 original features in the Zoo dataset, whereas each row (Subset) means one of the obtained feature subsets from one method. The square *(i, j)* will be colored if the *j*th feature is included in the *i*th feature subset. The figure also gives the training accuracy (on the right side) of the learning algorithm using each feature subset and the frequency (in the upper side) with which a feature has been selected across the obtained subsets.

TABLE VIII AVERAGE NUMBER OF DIFFERENT FEATURE SUBSETS WITH THE SAME TRAINING ACCURACY (Num)

| Dataset | *r*3pso | NCDE | NSDE | NShDE | LBPADE | NDEDA |
|---|---|---|---|---|---|---|
| Zoo | 1.5±0.8↓ | **7.2**±2.0↑ | 6.1±2.0o | 6.9±2.4↑ | 2.2±0.8↓ | 5.7±0.8 |
| WBCD | 2.5±1.2↓ | 6.0±4.0↓ | 5.1±3.3↓ | 6.2±3.7↓ | 10.2±3.1↓ | **24.3**±4.3 |
| Ionosphere | 2.2±1.3↓ | 2.3±1.5↓ | 1.8±0.9↓ | 2.3±1.2↓ | 1.9±1.1↓ | **3.2**±1.8 |
| Movement | 2.1±1.4↓ | 2.0±1.3↓ | 2.1±1.2↓ | 2.0±1.6↓ | 2.8±1.8↓ | **6.3**±5.6 |
| Hillvally | 1.2±0.5↓ | 1.5±0.8↓ | 1.5±0.7↓ | 1.4±0.7↓ | 1.3±0.5↓ | **3.9**±2.9 |
| Musk1 | 1.6±0.9↓ | 2.9±2.3↓ | 2.2±1.3↓ | 3.0±1.6↓ | 2.4±1.9↓ | **80.8**±42.4 |
| Multiple | 2.8±1.7↓ | 3.9±2.6↓ | 2.9±1.9↓ | 3.1±1.7↓ | 2.1±1.1↓ | **91.7**±52.9 |
| Arrhythmia | 2.1±1.3↓ | 3.0±2.0↓ | 2.8±1.7↓ | 2.7±2.0↓ | 2.6±1.6↓ | **246.8**±78.8 |
| CNAE | 1.1±0.3↓ | 1.5±0.7↓ | 1.4±0.7↓ | 1.3±0.5↓ | 1.6±1.0↓ | **34.4**±78.0 |
| QSAR | 1.9±1.1↓ | 2.7±2.3↓ | 1.8±1.0↓ | 2.3±1.3↓ | 2.2±1.4↓ | **15.8**±24.6 |
| AD | 1.5±0.7↓ | 1.9±1.3↓ | 1.7±0.9↓ | 1.9±1.1↓ | 2.9±2.2↓ | **31.5**±49.7 |
| SRBCT | 10.8±6.1↓ | 27.7±19.1↓ | 22.3±9.1↓ | 27.3±19.0↓ | 20.5±13.7↓ | **298.6**±7.4 |
| Leukemia | 7.4±5.3↓ | 43.7±19.9↓ | 41.4±17.6↓ | 46.0±18.0↓ | 66.2±43.9↓ | **273.7**±33.3 |
| Leukemia1 | 4.2±3.0↓ | 10.2±5.9↓ | 15.9±12.2↓ | 14.5±11.0↓ | 21.9±12.4↓ | **246.9**±51.0 |
| 9Tumor | 3.7±3.0↓ | 3.7±3.2↓ | 4.3±3.2↓ | 4.9±3.0↓ | 11.6±10.9↓ | **295.7**±17.0 |
| DLBCL | 10.5±5.7↓ | 105.5±17.7↓ | 75.4±10.0↓ | 101.9±16.1↓ | 96.7±48.8↓ | **297.9**±10.8 |
| Leukemia2 | 28.6±20.4↓ | 32.9±27.2↓ | 48.2±35.4↓ | 42.2±32.0↓ | 105.9±52.9↓ | **272.0**±48.1 |
| 11Tumor | 4.4±3.6↓ | 7.0±5.6↓ | 4.6±3.2↓ | 3.8±2.8↓ | 13.3±13.5↓ | **299.8**±0.9 |

In Fig. 4, four different feature subsets on the Zoo dataset can achieve the same classification accuracy of 94.28568%. One note is that there are two common features, $F_4$ and $F_{12}$, selected in the four solutions. This shows that the two features $F_4$ and $F_{12}$ are more likely to be strongly relevant features. According to the description from UCI Machine Learning Repository [7], features $F_4$ and $F_{12}$ are the *milk* and the *fins* of different animals, respectively. This shows that *milk* and *fins* are two essential factors in identifying different animals. Furthermore, the irrelevant features, for example, $F_6$ (i.e., *aquatic*) and $F_7$ (i.e., *predator*), are easy to identify by their low frequency, which helps to categorize the remaining variables as weakly relevant features.

Although two feature subsets, $S_3$: {$F_4$, $F_5$, $F_8$, $F_9$, $F_{12}$, $F_{15}$} and $S_4$: {$F_2$, $F_4$, $F_5$, $F_8$, $F_{12}$, $F_{15}$} in Fig. 4, achieve the same training classification accuracy, they may have different feature collection costs. More specifically, $F_9$ in $S_3$ is the *backbone* information of different animals, while $F_2$ in $S_4$ is the *feathers* information. Generally, $F_2$ is easier to collect than $F_9$. Therefore, if both $S_3$ and $S_4$ are provided, a decisionmaker is more likely to choose $S_4$, since $S_4$ has a lower feature collection cost.

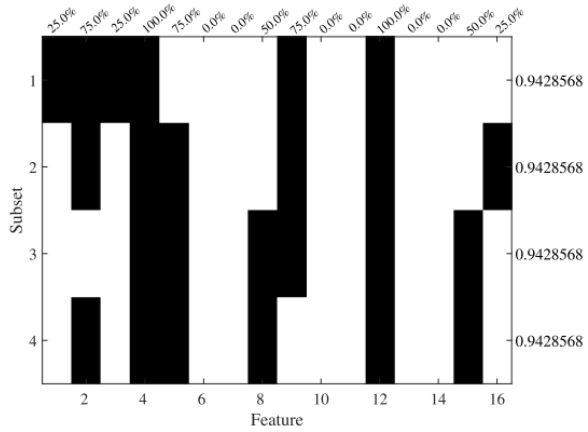


Fig. 4. Frequency matrix on the Zoo dataset. "Feature" means one of the original features in the Zoo dataset, and "Subset" means one of the obtained feature subsets from NDEDA. The training accuracy using each feature subset is shown on the right side, and the selected frequency of each feature is shown on the upper side

*D. Analysis on the Contribution of Each Major Component*

NDEDA has three major components: 1) the developed mutation operator; 2) the selection mechanism; and 3) the subset repairing scheme. To investigate the performance of the three components, four algorithms are formed. Three of them are variant algorithms of NDEDA using different mutation operators. In terms of the selection mechanism, comparing the fitness value of the current individual with the nearest individual in its parental population is another popular way. Considering both points, the newly formed algorithms are shown as follows:

1) NDEDA-l uses $\vec{x}_{lbest}$, $\vec{x}_{gr1}$, and $\vec{x}_{gr2}$ in (2) to generate mutant vectors. That means $\vec{x}_{gbest}$, $\vec{x}_{lr1}$, and $\vec{x}_{lr2}$ do not participate in mutation. Moreover, a new offspring is compared with the nearest individual from its parental population.
2) NDEDA-g uses $\vec{x}_{gbest}$, $\vec{x}_{lr1}$ and $\vec{x}_{lr2}$in (2) to generate mutant vectors, then, NDEDA-g degenerates to use DE/best/1 as its mutation operator. The used selection operator is the same as NDEDA-l.
3) NDEDA-lg uses (2) to generate mutant vectors. The used selection operator is the same as NDEDA-g.
4) NDEDA-lgs uses (2) to generate mutant vectors. The used selection operator is shown in Section III-F.
5) NDEDA-sr uses (2) as its mutation operator. Section III-E shows its selection operator. The subset repairing scheme used in NDEDA-sr is to modify both the duplicated solutions in the current population and possible reappearing feature subsets.

The only difference between NDEDA-*lgs* and NDEDA is that NDEDA employs the proposed subset repairing scheme, while NDEDA-lgs does not. By employing the subset repair scheme, NDEDA-sr modifies both duplicated solutions and possible reappearing feature subsets, whereas NDEDA modifies only duplicated solutions in the current population.

The detailed Ac and Size results of the six algorithms are shown in Tables S.IV and S.V in the Online Supplementary Materials, respectively. For the readers' convenience, the average Friedman's rankings are shown in Table IX. Furthermore, the average plots of the lowest fitness value (fmin) with generation are shown in Fig. 5. Four datasets (Ionosphere, Hillvally, Arrhythmia, and Leukemia) are chosen as representatives. Also, Fig. 6 shows the UN results from NDEDA-lgs, NDEDA-sr, and NDEDA on the four datasets.

1) Effect of the Proposed Mutation Operator: In Fig. 5, NDEDA-lg shows better and faster convergence performance than NDEDA-l and NDEDA-g on the four datasets. This shows that a DE-based feature selection method using only the local information of the niche or only the global information of the whole population in mutation may lead to a local optimum. Furthermore, the fitness values from both NDEDA lg and NDEDA-lgs are lower than those from NDEDA-l and NDEDA-g. The results in Tables S.IV and S.V in the Online Supplementary Materials and Table IX indicate that NDEDA-lg can significantly reduce the number of selected features and maintain or even improve the classification accuracy over NDEDA-l and NDEDA-g. The results show that the proposed mutation operator can help NDEDA obtain feature subsets with a lower training error rate by selecting fewer features.

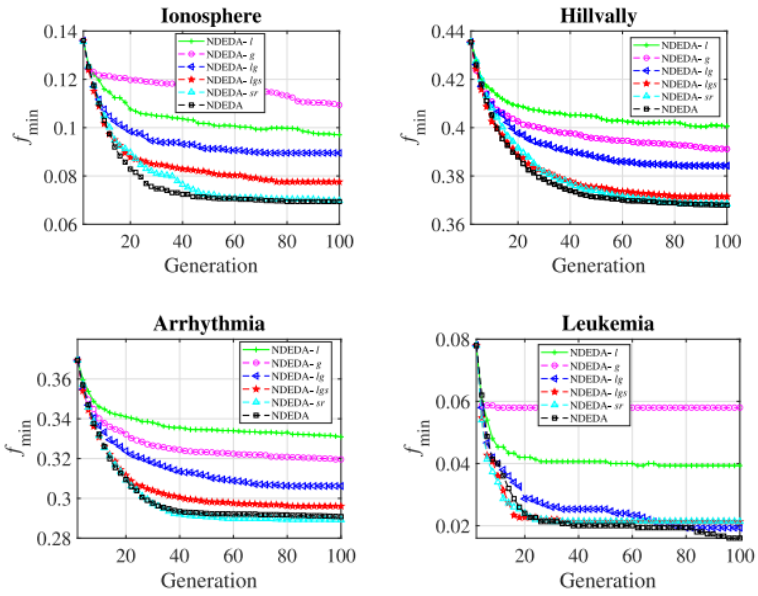


Fig. 5. Convergence curve of six algorithms on four training sets.

2) Effect of the Proposed Selection Operator: Comparison between NDEDA-lg and NDEDA-lgs shows that a lower fitness value can be obtained by NDEDA-lgs in Fig. 5. In Table IX, NDEDA-lgs achieves better rankings than NDEDAl, NDEDA-g, and NDEDA-lg on the Ac and Size results. The results indicate that the developed selection operator can further improve the feature selection performance.

Table IX. AVERAGE RANKINGS OF VARIANT METHODS OF NDEDA BASED ON FRIEDMAN TEST

| Indicator | NDEDA | NDEDA-$l$ | NDEDA-$g$ | NDEDA-$lg$ | NDEDA-$lgs$ | NDEDA-$sr$ |
|---|---|---|---|---|---|---|
| $Ac$ | 0.66 | 1.39 | 1.48 | 1.04 | 0.76 | 0.66 |
| $Size$ | 0.38 | 1.50 | 1.68 | 1.10 | 0.82 | 0.75 |

*3) Effect of the Proposed Subset Repairing Scheme:* In Fig. 5, NDEDA shows lower fitness values than NDEDA-l, NDEDA-g, NDEDA-lg, and NDEDA-lgs. Fig. 6 reveals the reason. In Fig. 6, the average number of unique feature subsets from NDEDA, that is, UN, increases along with the evolutionary process. At the end of the evolution, that is, the 100$^{th}$ generation, the UN value from NDEDA is close or equal to MaxFe. That means almost all fitness evaluations of NDEDA are consumed on evaluating unique feature subsets. The results show that the proposed subset repairing scheme modifies the duplicated feature subsets in the population and produces new feature subsets. The results in Tables S.IV and S.V in the Online Supplementary Materials and Table IX reveal that the produced new feature subsets can help NDEDA find feature subsets with a lower classification error rate.
Although NDEDA-sr has the best UN performance, that is, the UN value of NDEDA-sr linearly increases from 0 to MaxFe in Fig. 6, NDEDA-sr shows almost the same classification performance as NDEDA based on the results in Fig. 5, Table S.IV in the Online Supplementary Materials, and Table IX. Noted that NDEDA-sr and NDEDA differ only in that NDEDA-sr modifies reappearing feature subsets by using the proposed subset repairing scheme. On small datasets such as Ionosphere, the reappearing feature subsets might have poor fitness values. It is hard to produce new feature subsets with fitter fitness by modifying them. On large datasets such as Leukemia, there are few reappearing feature subsets. Therefore, the UN values between NDEDA-sr and NDEDA are almost the same in Fig. 6.

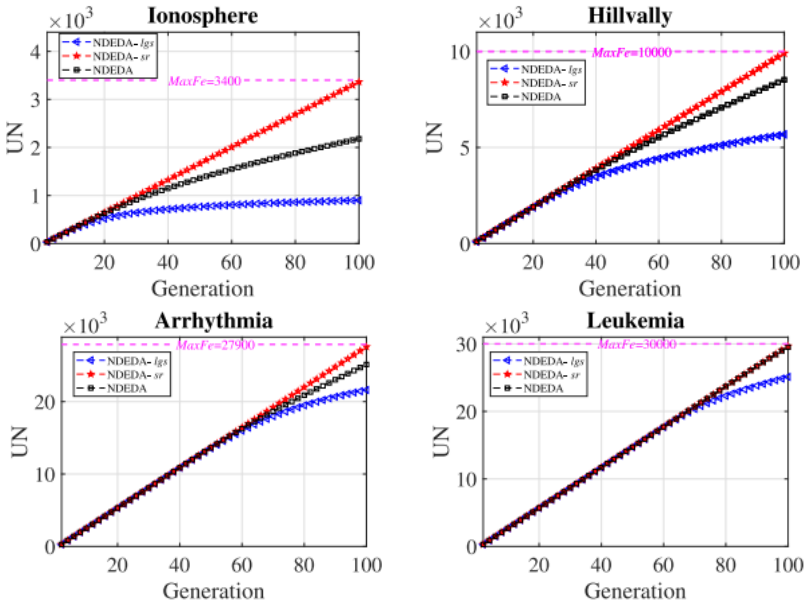


Fig. 6. Number of unique feature subsets (UN) produced with generations from NDEDA-lgs and NDEDA on four datasets. In each subfigure, MaxFe means the maximal time of fitness evaluation.

*E. Performance Comparison of SVM and KNN*

Table X presents a comparative analysis of SVM and KNN on five benchmark datasets, evaluating their performance in terms of accuracy and number of selected features. Across most datasets, SVM demonstrates higher classification accuracy and fewer selected features compared to KNN. For instance, SVM achieves an accuracy of 94.24% on the WDBC dataset with only 3.4 features, while KNN attains 94.15% with 4.1 features. Similarly, on the Ionosphere dataset, SVM outperforms KNN by 2.35% in accuracy while selecting fewer features. These results highlight SVM's ability to balance model performance and feature reduction, making it a more efficient choice for high-dimensional data. Additionally, the stability of SVM's fitness values (ranging between 0.056 and 0.061) underscores its robustness in maintaining consistent performance across different datasets.

Table X Performance Comparison of SVM and KNN Classifiers on Benchmark Datasets

| dataset | metric | KNN | SVM |
|---|---|---|---|
| **WDBC** | Accuracy | 94.15 | 94.24 |
| | Size | 4.1 | 3.4 |
| **Zoo** | Accuracy | 90.17 | 91.40 |
| | Size | 5.8 | 5.0 |
| **Ionosphere** | Accuracy | 88.15 | 90.50 |
| | Size | 4.2 | 4.0 |
| **Hillvally** | Accuracy | 55.38 | 57.90 |
| | Size | 24.8 | 22.5 |
| **CNAE-9** | Accuracy | 87.28 | 89.10 |
| | Size | 290.2 | 275.0 |

*F. Ablation Study of NDEDA Components: Impact on Accuracy and Feature Selection*

Building upon the structural analysis in Section V-D, a more detailed ablation study is provided here to quantify the exact impact of each component on accuracy and feature reduction.

A systematic ablation study was conducted to evaluate the individual contributions of the Repair Scheme, Inversion mechanism, Disturbance terms, and Guidance vectors to the overall performance of NDEDA (Tables XI and XII). Removing the Repair Scheme results in a noticeable drop in accuracy (e.g., from 90.17% to 89.45% on Zoo) and an increase in feature count. This indicates that the repair mechanism is vital for eliminating redundant individuals and preserving population diversity. Similarly, removing the Inversion mechanism transforms the process into a random

mutation, diminishing the preservation of informative features and causing performance degradation. The disturbance terms are critical for preventing premature convergence by introducing global and local perturbations. Their removal significantly impacts complex datasets; for example, accuracy on CNAE-9 drops from 87.28% to 81.20%. The most significant loss, however, occurs when removing the Guidance Terms ($\vec{x}_{lbest}$ and $\vec{x}_{\mathrm{gbest}}$). In this scenario, the accuracy for the Ionosphere dataset falls sharply from 88.15% to 80.20%, and the feature count nearly triples to 12.4. These results confirm that the guidance vectors are essential for providing directional exploration and avoiding a purely random search. In summary, the synergy between these four components allows NDEDA to effectively balance exploration and exploitation, ensuring robust feature selection in high-dimensional scenarios.

Table XI Ablation Study of NDEDA Components: Impact on Accuracy

| Dataset | Remove Guidance ($l_{best}$ / $g_{best}$) | Remove Disturbance | Remove Inversion | Remove Repair | NDEDA (all) |
|---|---|---|---|---|---|
| Zoo | 85.30% | 88.20% | 89.60% | 89.45% | 90.17% |
| Ionosphere | 80.20% | 84.50% | 87.10% | 86.90% | 88.15% |
| WDBC | 89.50% | 92.10% | 93.95% | 93.80% | 94.15% |
| Hillvalley | 48.10% | 52.30% | 54.50% | 54.10% | 55.38% |
| CNAE-9 | 74.50% | 81.20% | 85.90% | 85.50% | 87.28% |

Table XII Ablation Study of NDEDA Components: Impact on Number of Selected Features

| Dataset | Remove Guidance ($l_{best}$ / $g_{best}$) | Remove Disturbance | Remove Inversion | Remove Repair | NDEDA (all) |
|---|---|---|---|---|---|
| Zoo | 8.5 | 7.1 | 6.2 | 6.5 | 5.8 |
| Ionosphere | 12.4 | 6.8 | 4.5 | 5.3 | 4.2 |
| WDBC | 11.2 | 9.5 | 5.1 | 5.8 | 4.1 |
| Hillvalley | 45.6 | 38.4 | 28.5 | 31.2 | 24.8 |
| CNAE-9 | 520.8 | 410.5 | 320.0 | 345.0 | 290.2 |

*G. Comparative Analysis of Mutation Strategies*

To further validate the effectiveness of the proposed mutation operator in NDEDA, a comparative experiment was conducted against two conventional DE mutation strategies: DE/best/1 and DE/rand/1 (Table XIII). The results demonstrate that NDEDA consistently achieves superior classification accuracy across all tested datasets. For instance, on the high-dimensional CNAE-9 dataset, NDEDA reached an accuracy of 87.28%, significantly outperforming DE/best/1 (84.27%) and DE/rand/1 (76.34%). Moreover, NDEDA exhibited a remarkable capability in dimensionality reduction; on the WDBC dataset, it selected only 4.1 features on average, whereas DE/best/1 and DE/rand/1 required 11.4 and 11.8 features, respectively. This enhanced performance is attributed to the synergy between local niche information and global guidance ($x_{lbest}$ and $x_{gbest}$ ), which effectively balances exploration and exploitation. This mechanism prevents the algorithm from stagnating in local optima while ensuring the selection of a parsimonious feature subset.

Table XIII Comparative Results of NDEDA versus Standard DE Mutation Strategies

| Dataset | **DE/best/1** | | **DE/rand/1** | | **NDEDA** | |
|---|---|---|---|---|---|---|
| | Acc | size | Acc | size | Acc | size |
| **Zoo** | 89.20 | 6.5 | 88.50 | 7.0 | 90.17 | 5.8 |
| **Ionosphere** | 85.25 | 5.3 | 80.91 | 13.8 | 88.15 | 4.2 |
| **WDBC** | 93.68 | 11.4 | 92.74 | 11.8 | 94.15 | 4.1 |
| **Hillvally** | 54.38 | 32.9 | 52.28 | 44.5 | 55.38 | 24.8 |
| **CNAE-9** | 84.27 | 353.9 | 76.34 | 435.9 | 87.28 | 290.2 |

*H. Efficiency and Complexity of NDEDA*

The average running time of different methods can be seen in Table S.III in the Online Supplementary Materials. In Table S.III, all the nine methods use a relatively short time, less than 30 min, on the small datasets, such as Zoo, WBCD, Ionosphere, Movement, and Hillvally. On the Musk1, Multiple, QSAR, and AD datasets, r3pso spends the

longest training time. The slowest algorithms on the seven high-dimensional datasets (SRBCT, Leukemia, Leukemia1, DLBCL, 9Tumor, Leukemia2, and 11Tumor) are the proposed NDEDA and r-NDEDA algorithms. This is because both NDEDA and rNDEDA need to identify the duplicated feature subsets from the population one by one and then modify them. The time consumed by the subset repairing scheme increases as the number of features increases.

NDEDA mainly includes six parts: 1) initialization; 2) niching-based mutation; 3) crossover; 4) evaluation; 5) subset repairing scheme; and 6) selection strategy. Assume using a population with its size of N to solve a problem with D decision variables, the overall complexity of NDEDA is analyzed as follows.

The initialization, crossover, selection, and evaluation operators in NDEDA execute O(N) basic operations, which can be finished in linear time scale. Therefore, their computational complexities are O(N). Since the calculating of the Hamming distance has the complexity of O(D), and that of the niching-based mutation in NDEDA is O(ND). The computational complexity of the proposed subset repairing scheme consists of two parts: 1) the detection steps of the duplicated solutions and 2) the modification process. After obtaining the initialized N unique feature subsets, if all new individuals produced from one-generation are unique, the detection steps execute $O(3N^2/2 - N/2)$ basic operations while the modification process will not be activated. If only one unique feature subset is produced in one-generation, the detection steps will execute O(N) operations. Under this situation, the (N - 1) duplicated solutions will be improved by the modification operator. Therefore, the overall computational complexity of NDEDA is between $O(ND + N^2 - N)$ and $O(ND + 3N^2/2 - N/2)$.

## VI. CONCLUSION AND FUTURE WORK

The goal of this article was to design a new feature selection method to search for multiple optimal feature subsets to increase classification performance and reduce the number of the selected features. The goal has been successfully achieved by introducing a new mutation operator, a subset repairing scheme, and a new selection operator. A significant contribution of this work is the integration of a Support Vector Machine (SVM) as the fitness evaluator, which provides a more robust and stable fitness landscape in high-dimensional spaces compared to traditional distance-based classifiers. The proposed NDEDA algorithm was examined and compared with seven EC-based feature selection algorithms and two classical feature selection methods on 18 datasets. The results showed that NDEDA achieved higher classification accuracy with a much smaller number of selected features than *r*3pso, NCDE, NSDE, NShDE, LBPADE, and SBDA on most of the used datasets. Furthermore, the comparative analysis between SVM and KNN evaluators confirmed that the margin-based learning of SVM leads to superior generalization and more effective feature reduction. Comprehensive ablation studies further indicated that each component, including the repair scheme, inversion mechanism, and the guidance-based mutation operator, plays a synergistic role in balancing exploration and exploitation. Although VS-CCPSO, FCBF, and SBMLR scaled better than NDEDA in most datasets, NDEDA achieved higher classification accuracy. More importantly, NDEDA successfully found different feature subsets with very similar or the same classification performance. Further analyses indicated that the proposed mutation operator can speed up the convergence and produce promising feature subsets with fewer features. The proposed subset repairing scheme can increase the population diversity, and the selection operator can further improve the feature selection performance.

Further research in this direction could include investigating a more efficient strategy that can obtain multiple optimal feature subsets with less common features selected, applying ensemble learning techniques to integrate the obtained multiple feature subsets to further improve the classification accuracy, and using feature clustering methods to better explore the search space. Some of these directions are currently pursued by the authors. In addition, solving feature selection tasks under privacy protection and optimizing the computational overhead of the SVM-based evaluation remain challenging topics for future investigation.

## REFERENCES


[1] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[2] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[3] G. Karakaya, S. Galelli, S. D. Ahipa¸saoglu, and R. Taormina, ˘ "Identifying (quasi) equally informative subsets in feature selection problems for classification: A max-relevance min-redundancy approach," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1424–1437, Jun. 2016.

[4] J. Liu, C. Xu, W. Yang, Y. Shu, W. Zheng, and F. Zhou, "Multiple similarly effective solutions exist for biomedical feature selection and classification problems," *Sci. Rep.*, vol. 7, no. 1, pp. 1–10, 2017.

[5] C. Yue, J. Liang, B. Qu, K. Yu, and H. Song, "Multimodal multiobjective optimization in feature selection," in *Proc. IEEE Congr. Evol. Comput.*, 2019, pp. 302–309.

[6] P. Wang, B. Xue, J. Liang, and M. Zhang, "Multiobjective differential evolution for feature selection in classification," *IEEE Trans. Cybern.*, early access, Dec. 7, 2021, doi: 10.1109/TCYB.2021.3128540.

[7] D. Dua and C. Graff. "UCI machine learning repository." 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[8] S. Kamyab and M. Eftekhari, "Feature selection using multimodal optimization techniques," *Neurocomputing*, vol. 171, pp. 586–597, Jan. 2016.

[9] P. Pudil, J. Novovicová, and J. Kittler, "Floating search methods in fea- ˇ ture selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[10] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[11] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, 2010.

[12] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[13] K. A. De Jong, "An analysis of the bevavior of a class of genetic adaptive systems," Ph.D. thesis, Univ. Michigan, Ann Arbor, 1975.

[14] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. Int. Conf. Genet. Algorithms*, Hillsdale, NJ, USA, 1987, pp. 41–49.

[15] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, 2002.

[16] B. Qu, P. N. Suganthan, and J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[17] Z. Wang *et al.*, "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, Dec. 2018.

[18] Z. Wang *et al.*, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, Feb. 2020.

[19] Z. Chen, Z. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.

[20] Y. Jiang, Z. Zhan, K. C. Tan, and J. Zhang, "Optimizing niche center for multimodal optimization problems," *IEEE Trans. Cybern.*, early access, Dec. 17, 2021, doi: 10.1109/TCYB.2021.3125362.

[21] L. Tang, Y. Zhao, and J. Liu, "An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 209–225, Apr. 2014.

[22] H. Zhao *et al.*, "Local binary pattern-based adaptive differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3343–3357, Jul. 2020.

[23] A. Datta, S. Ghosh, and A. Ghosh, "Wrapper based feature selection in hyperspectral image data using self-adaptive differential evolution," in *Proc. IEEE Int. Conf. Image Inform. Process.*, 2011, pp. 1–6.

[24] A. A. Bidgoli, S. Rahnamayan, and H. Ebrahimpour-Komleh, "Opposition-based multi-objective binary differential evolution for multi-label feature selection," in *Proc. Int. Conf. Evol. Multi-Criter. Optim.*, 2019, pp. 553–564.

[25] X. Zhao, L. Bao, Q. Ning, J. Ji, and X. Zhao, "An improved binary differential evolution algorithm for feature selection in molecular signatures," *Mol. Informat.*, vol. 37, no. 4, 2018, Art. no. 1700081.

[26] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, Apr. 2020.

[27] P. Wang, B. Xue, M. Zhang, and J. Liang, "A grid-dominance based multi-objective algorithm for feature selection in classification," in *Proc. IEEE Congr. Evol. Comput.*, 2021, pp. 2053–2060.

[28] M. A. Hall and L. A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper," in *Proc. FLAIRS Conf.*, vol. 1999, 1999, pp. 235–239.

[29] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. Intern. Conf. Mach. Learn.*, 2003, pp. 856–863.

[30] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint *l*2,1-norms minimization," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 23, 2010, pp. 1813–1821.

[31] D. P. Muni, N. R. Pal, and J. Das, "Genetic programming for simultaneous feature selection and classifier design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 106–117, Feb. 2006.

[32] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, no. 1, pp. 23 69, 2003.

[33] H. Almuallim and T. G. Dietterich, "Learning boolean concepts in the presence of many irrelevant features," *Artif. Intell.*, vol. 69, nos. 1–2, pp. 279–305, 1994.

[34] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and minredundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[35] G. C. Cawley, N. L. Talbot, and M. Girolami, "Sparse multinomial logistic regression via Bayesian *l*1 regularisation," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 19, 2007, pp. 209–216.

[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.
[37] P. Wang, B. Xue, J. Liang, and M. Zhang, "Differential evolution based feature selection: A Niching-based multiobjective approach," *IEEE Trans. Evol. Comput.*, early access, Apr. 20, 2022, doi: 10.1109/TEVC.2022.3168052.
[38] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Feature subset selection using differential evolution and a statistical repair mechanism," *Exp. Syst. Appl.*, vol. 38, no. 9, pp. 11515–11526, 2011.
[39] D. Whitley, "A genetic algorithm tutorial," *Stat. Comp.*, vol. 4, no. 2, pp. 65–85, 1994.
[40] Y. Zhang, D. Gong, X. Gao, T. Tian, and X. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
[41] T. Vivekanandan and N. C. S. N. Iyengar, "Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," *Comput. Biol. Med.*, vol. 90, pp. 125–136, Nov. 2017.
[42] B. H. Nguyen, B. Xue, P. Andreae, and M. Zhang, "A new binary particle swarm optimization approach: Momentum and dynamic balance between exploration and exploitation," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 589–603, Feb. 2021.
[43] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, Jun. 2019.
[44] X. Song, Y. Zhang, D. Gong, and X. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9573–9586, Sep. 2022, doi: 10.1109/TCYB.2021.3061152.
[45] X. Xue, M. Yao, and Z. Wu, "A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm," *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 389–412, 2018.
[46] B. Xue, W. Fu, and M. Zhang, "Differential evolution (DE) for multiobjective feature selection in classification," in *Proc. Annu. Conf. Genet. Evol. Comput.*, 2014, pp. 83–84.
[47] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comp.*, vol. 18, pp. 261–276, May 2014.
[48] B. Zhang, A. K. Qin, and T. Sellis, "Evolutionary feature subspaces generation for ensemble classification," in *Proc. Genet. Evol. Comput.Conf.*, 2018, pp. 577–584.
[49] H. Xu, B. Xue, and M. Zhang, "A duplication analysis-based evolutionary algorithm for biobjective feature selection," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 205–218, Apr. 2021, doi: 10.1109/TEVC.2020.3016049.
[50] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.
[51] A. I. Hammouri, M. Mafarja, M. A. Al-Betar, M. A. Awadallah, and I. Abu-Doush, "An improved dragonfly algorithm for feature selection," *Knowl. Based Syst.*, vol. 203, Sep. 2020, Art. no. 106131.
[52] X. Song, Y. Zhang, Y. Guo, X. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 882–895, Oct. 2020.
[53] K. Chen, B. Xue, M. Zhang, and F. Zhou, "Correlation-guided updating strategy for feature selection in classification with surrogate-assisted particle swarm Optimisation," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1015–1029, Oct. 2022, doi: 10.1109/TEVC.2021.3134804.
[54] C. Yue *et al.*, "Differential evolution using improved crowding distance for multimodal multiobjective optimization," *Swarm Evol. Comput.*, vol. 62, Apr. 2021, Art. no. 100849.
[55] P. Wang, B. Xue, J. Liang, and M. Zhang, "Improved crowding distance in multi-objective optimization for feature selection in classification," in *Proc. Int. Conf. Appl. Evol. Comput.*, 2021, pp. 489–505.