

OPEN

# An Efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets

Jamshid Pirgazi<sup>1\*</sup>, Mohsen Alimoradi<sup>2</sup>, Tahereh Esmaeili Abharian<sup>2</sup> & Mohammad Hossein Olyaei<sup>1</sup>

Feature selection problem is one of the most significant issues in data classification. The purpose of feature selection is selection of the least number of features in order to increase accuracy and decrease the cost of data classification. In recent years, due to appearance of high-dimensional datasets with low number of samples, classification models have encountered over-fitting problem. Therefore, the need for feature selection methods that are used to remove the extensions and irrelevant features is felt. Recently, although, various methods have been proposed for selecting the optimal subset of features with high precision, these methods have encountered some problems such as instability, high convergence time, selection of a semi-optimal solution as the final result. In other words, they have not been able to fully extract the effective features. In this paper, a hybrid method based on the IWSSr method and Shuffled Frog Leaping Algorithm (SFLA) is proposed to select effective features in a large-scale gene dataset. The proposed algorithm is implemented in two phases: filtering and wrapping. In the filter phase, the Relief method is used for weighting features. Then, in the wrapping phase, by using the SFLA and the IWSSr algorithms, the search for effective features in a feature-rich area is performed. The proposed method is evaluated by using some standard gene expression datasets. The experimental results approve that the proposed approach in comparison to similar methods, has been achieved a more compact set of features along with high accuracy. The source code and testing datasets are available at [https://github.com/jimmy2020/SFLA\\_IWSSr-Feature-Selection](https://github.com/jimmy2020/SFLA_IWSSr-Feature-Selection).

The basic issue about big data is a large number of features. Among the features available, only a few of them will be useful to distinguish samples that belong to different classes and many of the features are irrelevant, noise, or redundant. Irrelevant features do not necessarily lead to noise generation in big data analysis; they result in increasing the dimensions of the dataset and computational complexity in clustering and classification operations, and consequently they decrease the rate of classification accuracy. Therefore, it is necessary to select the appropriate features. In feature selection, the redundant features are usually removed from dataset because there is a subset of other features that can provide the information that is provided by these redundant features. On the other hand, noise features that do not provide any information about labels should also be removed because they will reduce the efficiency of the algorithm. Therefore, only relevant features which consist of significant information about given dataset will remain<sup>1</sup>. Consequently, a method for identifying diverse features, calculating relationships between features and selecting relevant features is needed through a huge amount of data.

For a dataset containing  $N$  number of features, there are  $2^N$  number of candidate subsets. The purpose of designing different feature selection methods has always been to find the most compressed subset with the highest precision among the candidate subsets. Considering the wide scope of possible solutions and increasing the size of this set of responses due to increment of the number of features exponentially, finding the best subset of  $N$  (medium or large) features is extremely costly. Computational complexity of selecting features is another major challenge for researchers<sup>2</sup>.

<sup>1</sup>Faculty of Engineering, Department of Computer Engineering, University of Gonabad, Gonabad, Iran. <sup>2</sup>Faculty of Electronic, Computer & IT Department of Computer, Qazvin Islamic Azad University, Qazvin, Iran. \*email: [j.pirgazi@znu.ac.ir](mailto:j.pirgazi@znu.ac.ir)

Different methods proposed for selecting a subset of features, have encountered some problems such as instability, high convergence time and falling in local optima as a final result, etc. Despite the success they have gained, they have not been able to extract the most effective features.

The feature selection methods are generally divided into four categories: filter methods, wrapper methods, hybrid methods and embedded methods. Each of these methods is described in detail<sup>3</sup>.

The filter methods for selection a subset of proper features use intrinsic and statistical characteristics of the features and they are independent of any learning algorithm. In these methods, weight is assigned to each feature based on the degree of relevance of features to class labels; correlation criteria and information theory-based criteria are used for weighting features usually. Due to the need for less computations, filter methods are effective for high-dimensional datasets, but they do not have the proper accuracy<sup>4</sup>. The filter methods are divided into two groups of univariate and multivariate. In univariate methods, relevance of only one feature is measured according to the evaluation criterion. In these methods, dependencies between features do not play a role in the process of feature selection. The methods are such as: *t*-statistics (TS)<sup>5</sup>, Signal-to-Noise Ratio (SNR)<sup>6</sup>, and Pearson Correlation coefficient (PC)<sup>7</sup> and F-Test (FT)<sup>8</sup>. In multivariate methods, the relationship between the features is considered. This makes these methods slower than univariate methods. The methods are such as: minimum Redundancy Maximum Relevance (mRMR)<sup>9</sup>, Correlation based Feature Selection (CFS)<sup>10</sup>, Fast Correlation Based Filter (FCBF)<sup>11</sup> and Mutual Information Feature Selection (MIFS)<sup>12</sup>, Max-Relevance-Max-Distance (MRMD)<sup>13</sup>, Analysis of variance (ANOVA)<sup>14</sup> and F-Score<sup>15</sup>. In these methods, the features are sorted based on their weights, and the features that have higher weights are selected as relevant features.

MRMD method considers distance between two kinds of features. This method is based on distance function to measure the independence of every feature. The higher the distance, the more the independence. Therefore, Pearson's correlation coefficient is used to measure the relevance between features. Distance functions such Euclidean distance, Cosine distance and Tanimoto distance are exploited to calculate the redundancy.

In ANOVA, a method is proposed to improve the prediction accuracy of mitochondrial proteins of the malaria parasite. In this method, firstly, the protein samples are formulated using the *g*-gap dipeptide composition. Then, Analysis of variance is proposed to select the best subset of features. Finally, the support vector machine (SVM) is used to perform the prediction.

The most important defect of filter algorithms is the lack of utilization of the classification accuracy in selection of a subset of features. To solve this problem, new methods called wrapper methods are proposed. Wrapper methods use learning algorithms and a classifier to find a subset of features. In these methods, the learning model has the tasks of searching in the space of primary features and selecting the subset of the candidate features.

Also, the classifier is used to estimate the performance of the subset of the selected candidate features. Compared to the filter methods, the wrapper methods have higher computational costs and they are not suitable for high-dimensional datasets; however, they are more successful in finding the subset of effective features and the high accuracy of selecting a subset of features using these methods is noticeable<sup>16</sup>. Many of wrapper methods have used heuristic search algorithms to find a subset of features. These methods start with a randomly generated solution, and in each iteration they are one step closer to the best subset of the solution. The evolutionary algorithms used in wrapper methods include Genetic Algorithm<sup>17–19</sup>, Simulated Annealing algorithm<sup>20,21</sup>, Ant Colony Optimization algorithm<sup>22–24</sup>, Shuffled Frog Leaping Algorithm<sup>25,26</sup>, Particle Swarm Optimization algorithm<sup>27,28</sup>, Binary Wolf Search Algorithm<sup>29,30</sup> and so on.

Some methods use exhaustive searches. In<sup>31</sup>, in order to select a subset of features, it first starts with a complete set of features, and then some of the features are removed by the first depth method. In<sup>32</sup>, the features are selected using the beam search. This method arranges the features in a queue based on importance, and then all possible states are evaluated using beam searches. The main drawback of these methods is their computational complexity. Heuristic methods were proposed to solve this problem. Sequential feature selection methods such as Sequential Backward Selection (SBS)<sup>33</sup> and Sequential Forward Selection (SFS)<sup>34</sup>, greedy methods such as hill-climbing<sup>35</sup>, Bayesian search methods such as Bayesian features selection<sup>36</sup>, meta-heuristic methods such as Ant Colony Optimization algorithm<sup>22</sup>, and Genetic Algorithm<sup>17</sup>, etc. are some methods that use heuristic search.

Another category of feature selection methods is Hybrid methods that combine filter and wrapper methods. So in the first step, based on a filter method, some features are selected based on importance. Then, in the selected features space, a wrapper method is applied to select the effective features<sup>37–39</sup>. In<sup>40</sup>, Incremental Wrapper Subset Selection (IWSS) is presented. In this method, after the weight of the features in the filter phase is calculated, the incremental algorithm is used to select the subset of features. First, the feature subset is empty. In the first iteration, the features are added with greater weights to the subset of features and the classifier is created based on the features and the dataset. The accuracy rate of classifier is stored as the best result. In next iterations, each time a feature with more weight is added to the subset, again the classifier is trained. If the recognition rate of the classifier is better than the one stored, the added feature is considered as a relevant feature and it is retained in the subset, otherwise the feature is removed from the subset. In<sup>41</sup>, the hybrid local search strategy embedded in the particle swarm optimization algorithm has been used to select relevant features. The purpose of the local search in this method is to optimize the particle swarm to select distinctive features based on their correlation information.

In<sup>37</sup>, a hybrid approach based on the Greedy Randomized Adaptive Search Procedure is proposed. In the first step, by using a filter method, the process of ranking the features is done, and features that have high degrees of relation with class labels are more weighty and less important features are less weighty. In the second stage, in the wrapper method, GRASP method is used to find the best subset. In the GRASP method, a subset of features is randomly selected based on their weights. Then in the next step by using the IWSS, SFS, IWSS, Hill-Climbing, and Best Agglomerative Ranked Subset (BARS)<sup>42</sup> methods; redundant and irrelevant features are removed. Use of an improvement phase is also considered in FICA<sup>43</sup>. In this paper, at the filter step, features weighting is performed. Then, in the wrapper phase, using the Fuzzy Imperialist Competitive Algorithm (FICA) and the IWSSr algorithm, searching for effective features in the weighted feature space is done. In the other work,

by using mutual information and adaptive genetic algorithm, gene expression data are classified<sup>44</sup>. In this method, the features are ranked base on maximizing the mutual information and then, by using the adaptive genetic algorithm, the optimal subset of features is selected. In<sup>45</sup>, an effective hybrid gene selection method based on ReliefF and Ant colony optimization (ACO) algorithm for tumor classification is proposed. At first, ReliefF is used to estimate the weights of features according to how well their values distinguish between close instances. Then a new pruning rule based on ACO is designed to reduce dimensionality and obtain a new candidate subset with the smaller number of genes.

A two-step feature selection is proposed to exclude redundant and noise information for identifying origin of replication in *Saccharomyces cerevisiae*. In this method, at first, the weight of the features is calculated based on the F-score technique. Then, the MRMR technique is used to maximize the correlation between features and class labels while minimize the correlation between features and features<sup>46</sup>.

In the embedded methods, selecting the features subset is considered as a part of the model construction. This kind of methods can be considered as a search in the feature and model space; such as Adaboost<sup>47</sup>, random forest, and decision tree<sup>48</sup>. SVM-RFE is also one of the embedded methods<sup>49</sup>. In this method, the algorithm starts with a set containing all features. In each iteration, the weight vector coefficients  $w$  is used to evaluate the features. Each element of this vector corresponds to a feature. In this case, the feature with the lowest score, ie,  $c_i = (w_i)^2$ , is removed. These weights indicate the relation of each feature with class label. Another algorithm proposed in this field is the KP-SVM algorithm<sup>50,51</sup>. The algorithm tries to find the appropriate features by updating the parameter  $\sigma$  in the RBF kernel.

In this paper, a hybrid method is proposed for selecting features in high dimensional datasets. In the proposed method, in the filter phase, the Relief method is used for weighting the features. Then, in the wrapper step, by using the SFLA and the IWSSr algorithm<sup>52</sup>, the search is performed to find the best subset of the features. The proposed method is evaluated with ten standard gene expression datasets. The results of the experiments confirm the effectiveness of the proposed approach in comparison with similar methods, in terms of Accuracy, Specificity, Sensitivity, Balance Rate and accessing to a subset of more compact features. The rest of the paper is organized as follows. Section 2 and 3 present an overview of the SFLA and IWSSr approaches and Section 4 describes the phases of the proposed method in detail. Section 5 provides the results of the method in the *gene* datasets. Finally, Section 6 summarizes the results.

## An Overview of the SFLA

SFLA is a new population-based metaheuristic optimization method that imitates the memetic evolution of a group of frogs when looking for a place with the maximum amount of available food. The SFLA has both definite and random strategies in finding the optimal response. The definite strategy allows the algorithm to use surface-level information efficiently in order to guide heuristic search. Random elements control the flexibility and power of the search pattern in the proposed method.

In this method, each frog is considered as a solution to the problem and a bunch of frogs forms a population that moves in order to reach a specific target. During the process of reaching the optimal answer, the population is divided into a number of subsets. The effects of the frogs in each subgroup modify the decision variables. After a certain number of evolutions, information is transmitted between the frogs during the process of combining subsets and forming a new population and a targeted search is carried out to determine the optimal answer. This trend continues until certain convergence conditions are established<sup>53,54</sup>.

In the SFLA, a primitive population of *sfla\_p* frogs is randomly generated from possible answers. The position or situation of a frog is a possible solution to the problem. These frogs are implemented by vectors and structures to indicate the variables or problem solutions. In the algorithm, the entire initial population is first divided into *sfla\_m* groups called memplex. Different memplexes that have *sfla\_n* frogs are bunch of frogs that are individually searching for a solution in the search space. In each memplex, a submemplex is created to avoid falling in local optima<sup>23</sup>. Each submemplex consists of *sfla\_q* frogs and the frogs are selected randomly based on the following probability function:

$$P_j = \frac{2(sfla\_n + 1 - j)}{sfla\_n (sfla\_n + 1)}, \quad j = 1, 2, \dots, sfla\_n \quad (1)$$

Where  $P_j$  is the probability of choosing  $j$ th frog for selection and *sfla\_n* is the number of frogs in the memplex. Since in each memplex the frogs are sorted according to a descending order of fitness, by decreasing the fitness value, the probability of selecting frogs is lowered. Therefore, a better-positioned frog in the search space will have a greater chance of choosing as a member of the submemplex. In each submemplex, the worst frog ( $P_w$ ), performs leaping based on its own experiences and the position of best frog in memplex ( $P_b$ ). Therefore, the worst frog is first selected from the submemplex. The leaping step size for frog  $P_w$  is as follows:

$$S_B = \begin{cases} \min\{\text{int}(\text{rand} \cdot [P_b - P_w]), S_{\max}\} & \text{for a positive step} \\ \max\{\text{int}(\text{rand} \cdot [P_b - P_w]), -S_{\max}\} & \text{for a negative step} \end{cases} \quad (2)$$

Where *rand* is a random number in the range [0,1] and  $S_{\max}$  is the maximum leap length. In the next step, the worst frog position is edited by the following equation:

$$P'_w = P_w + S_B \quad (3)$$

If the new frog ( $P'_w$ ) is better than the original frog, this frog is replaced with the original frog, otherwise the  $P_w$  frog is edited according to the best frog of the total population ( $P_G$ ) according to the following:

In D: training Data, ITmax: Total Iteration number, sfla\_p: Population size  
 sfla\_m: Number of memplexes, sfla\_n: Population size of each memplex, sfla\_q: Population  
 size of submemplexes, S<sub>max</sub>: The maximum leap length allowed to change

Out Frog : best subset of feature

- 1 Create an initial population of SFLA\_P frogs generated randomly.
- 2 Divide the frogs into afa\_m memplexes each holding sfla\_n frogs.
- 3 i= 0
- 4 while I < ITmem
- 5 create a submemplex for each memplex
- 6 the position of the worst frog P<sub>w</sub>' for the memplex is adjusted such as (3)
- 7 if (fitness(P<sub>w</sub>') < fitness(P<sub>w</sub>))
- 8 the position of the worst frog P<sub>w</sub>' for the memplex is adjusted such as (5)
- 9 if (fitness(P<sub>w</sub>'') < fitness(P<sub>w</sub>))
- 10 a random frog is generated which replaces the worst frog.
- 11 otherwise
- 12 P<sub>w</sub> = P<sub>w</sub>'
- 13 otherwise
- 14 P<sub>w</sub> = P<sub>w</sub>'
- 15 i = i + 1
- 16 frog shuffling together
- 17 Check the convergence. If the convergence criteria are satisfied stop, otherwise return to the step 2.
- 18 finish

**Figure 1.** Pseudo code of SFLA.

$$S_G = \begin{cases} \min\{\text{int}(\text{rand.}[P_G - P_w]) \cdot S_{\max}\} & \text{for a positive step} \\ \max\{\text{int}(\text{rand.}[P_G - P_w]) \cdot -S_{\max}\} & \text{for a negative step} \end{cases} \quad (4)$$

$$P''_w = P_w + S_G \quad (5)$$

Similar to the previous one, if the  $P''_w$  frog is better than the original frog ( $P_w$ ), this frog is replaced with the  $P''_w$  frog and if neither of these is satisfied, a new random frog is replaced with the worst frog of submemplex. After the  $IT_{mem}$  steps of dividing memplex into submemplexes, again all the frogs are combined and re-divided into  $sfla_m$  memplexes. This operation continues to meet the end conditions of the program. The pseudo code of SFLA is shown in Fig. 1. Based on this algorithm, the worst frog can leap toward the best frog. By repeating this process, gradually the average fitness of the frog population increases during the evolutionary stages and converges to a certain degree. With respect to this process,  $P_G$  and  $P_w$  are changed in each iteration and the value of fitness increases to converge to the desired response<sup>55</sup>.

## An Overview of IWSSr Algorithm

IWSSr algorithm<sup>52</sup> that is an extension of IWSS algorithm, is one of the wrapper-based features subset selection algorithms. In this method, first, in the filter phase, the relevance of each feature with the class labels is calculated and a weight is assigned to each feature. In IWSSr, the SU criterion is used for weighting features. SU is a nonlinear information theory based criterion. This criterion evaluates each feature independently and it assigns to each feature a number in the range [0,1] indicating the weight of each feature based on its relevance to class labels. A large number indicates the high importance of the feature. This criterion is calculated as follows:

$$SU_{i,c}(F_i, C) = 2 \frac{H(F_i) - H(F_i|C)}{H(F_i) - H(C)} \quad (6)$$

Where  $C$  is the class label,  $F_i$  represents  $i$ th feature and  $H$  indicates entropy. In the following, at wrapper phase, the features are arranged in descending order by weights. Then an incremental mechanism is used to select a subset of features. Figure 2 shows the pseudo code of IWSSr algorithm. In this algorithm,  $S$  is the subset of selected features. At first, the candidate subset is empty and in first iteration, the feature that has the highest score is added to the candidate subset.

Then a classifier is trained based on the candidate subset and the existing training data. The classification accuracy is maintained as the best result. The next step is done in two phases. In the first phase, a feature with a high score that has not been evaluated yet, is replaced with each feature in the candidate subset. After each replacement, a new classifier is trained by using the obtained subset. then the classification accuracy is calculated. If the addition of a new feature causes increase in classification accuracy compared to the previous subset, the result is maintained as the best. In this way, the dependence of this feature with all previous selected features is measured and if it does not depend on any of the selected features, it will be added to the candidate subset.

In the second phase, the feature that is under review (the feature that was replaced with the features in the selected subset in the first phase) is added to the selected subset  $S$  (which was obtained in the previous stage) and a new classifier is trained based on the new subset and the classification accuracy is calculated. If the accuracy of the subset is higher than the accuracy of the candidate subset of the first phase, it is maintained as the best result. After the first and second phases, if we have achieved a better subset in each of these phases, the optimal subset is selected as the subset of this iteration and the feature is applied to the selected subset.

```

In : D: training Data, C: classifier, F: feature set
Out: S feature subset
1  S = {F[1]}
2  Bestfeature = evaluate(C, S,D)
3  for i=2 to F.size()
4      best = null;
5      for j=1 to S.size()
6          S_sel= update(S, swap(S[j],F[i])
7          Data = evaluate (C, S_sel,D)
8          if (Data > Bestfeature )
9              best =(S[j],F[i])
10             Bestfeature = Data
11         S_aux= S ∪ {F[i]}
12         Data = evaluate(C, S_sel,D)
13         if (Data> Bestfeature)
14             best= add(F[i])
15             Bestfeature = Data
16         if (best != null)
17             update(S, best)

```

**Figure 2.** Pseudo code of IWSSr.

```

In : D training dataset, Repetition of the whole process n
Out: weight of features
1  Set all Weight W[A]=0
2  For i=1:n do
3      Randomly select an instance R
4      Find nearest hit H and nearest miss M
5      For A =1 to all_features do
6          W[A] = W[A] - diff(A,R,H)/n + diff(A,R,M)/m

```

**Figure 3.** The general scheme of the Relief algorithm.

## Materials and Methods

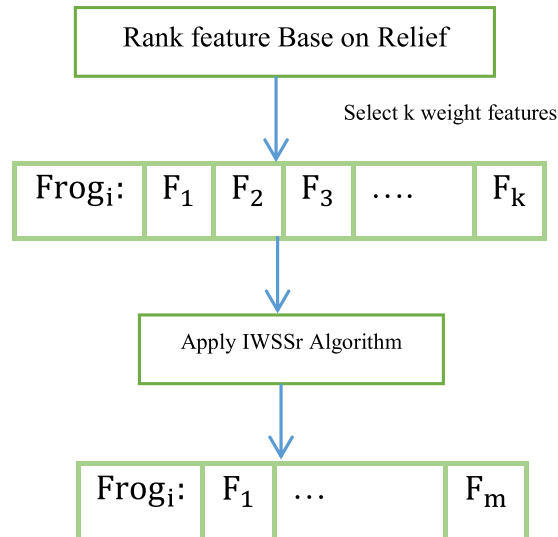
The proposed algorithm is a feature selection system called IWSSr and Shuffled Frog Leaping Algorithm (IWSSr-SFLA). In this paper, a hybrid method is proposed for selecting features in high dimensional datasets. In the proposed method in the filter phase, the Relief method is used for weighting the features. Then, in the wrapping phase, by using the combination of Shuffled Frog Leaping Algorithm and the IWSSr algorithm, the search is performed to find the best subset of features.

In the first phase, the Relief method, estimates the quality of features according to how well their values distinguish between instances that are near to each other. The Relief method calculates the correlation between features found by nearest-neighbor algorithm. Its output is a set containing weights of features<sup>56</sup>. It arranges the set in descending order. Figure 3 shows the general scheme of the Relief algorithm<sup>56</sup>.

As we can see in Fig. 3, at first, one sample is randomly selected, then its two neighbors are searched. One neighbor along with selected sample are in a same class and the other neighbor is in a different class. Function  $\text{Diff}(A,R,H)$  calculates the difference between the values of the feature  $A$  and the first neighbor, and  $\text{Diff}(A,R,M)$  calculates the difference between the values of the feature  $A$  and the second neighbor. then the weight of each feature is updated. For discrete features the difference is 1 (when the values are different) and 0 (when the values are the same). For continuous features, the difference is the normalized value of the real difference of two values of feature, in the range of [0,1]. The Relief algorithm works well for noisy or correlated features. It depends on the number of features and the number of samples in the dataset. It is noticeable to point that the time complexity of the algorithm is linear.

In the wrapping phase, a primary population of frogs is initially created, each containing a subset of the features. In order to find the best subset for a more efficient classification, the primary population should be trained. After some learning phases, the best frog (which is closest to the target) is selected as a solution. At each training phase, the entire population is first divided into a number of memplexes.

In each memplex, a submemplex is selected and in this category the worst frog is initially trained or leaped towards the best frog of the memplex. If the better frog is created, this frog is replaced with the worst frog. Otherwise, the worst frog will be leaped according to the best frog of the entire set. This time, As the previous stage, if the frog is improved, it is replaced, and if not, a new frog is created. After creating the new frog randomly, the replacement of the new frog is done if its fitness is better than the original frog, otherwise the original frog is



**Figure 4.** Create frogs in the proposed algorithm.

remaining unchanged. The division of the memplexes into submemplexes is repeated  $IT_{mem}$  times. After completing the learning phases, the whole set and the best frog get closer to the goal.

**Initial population creation.** In the proposed algorithm, an Initial population with the number of  $sfla\_p$  frogs is initially created randomly. Each frog has a subset of features for classifying data. Therefore any of the frogs will be a solution to the problem. In the initial population, a random percentage of the features are selected based on the weights assigned to them in the filtering phase. Due to random weighted selection, high weight features are more likely to be selected. Figure 4 shows how to create the frogs in the proposed algorithm.

**Evaluation of the initial population.** After selecting the features for each frog, the redundant features of each frog are removed by using the IWSSr algorithm and after applying this algorithm, the cost of each frog is calculated. The initial population is evaluated using a quality check function. The frog, which includes more relevant features, earns a higher value of fitness.

$$F = \left( \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) / 2 \quad (7)$$

Where  $TN$  is the number of negative samples which are correctly classified.  $FN$  is the number of positive samples identified as negative samples.  $TP$  is the number of positive samples which are correctly classified.  $FP$  is the number of negative samples identified as positive.

**Termination conditions of the program.** The termination conditions refer to the user-defined conditions. The conditions can be a user-defined constant number of iterations for training, reaching the maximum percentage of diagnosis or not changing the entire population. In the experiments, after  $IT_{max}$  iterations, the learning process is terminated.

**Division of memplexes into submemplexes.** In each memplex which has  $sfla\_n$  frogs, a submemplex is created that contains  $sfla\_q$  frogs. To do this, frogs of memplex are sorted by descending value of fitness. The probability of choosing each frog in submemplex is calculated based on Eq. (1). Therefore, the submemplex is created based on the fitness of each frog.

**Leap or improve the frog.** After each submemplex creation, the worst frog position ( $P_w$ ) is edited based on the position of the best frog of the memplex ( $P_b$ ) (or the best frog of the total population ( $P_G$ )). This edition is called leaping. Therefore, the leaping in the SLEA is an operation in which, the frog with a lower fitness can be improved according to a frog which has better fitness. The leaping action can vary depending on different issues. The improvement phase of the worst frog which is indicated by the IWF as shown in Fig. 5, is illustrated as a flowchart in Fig. 6.

To improve the worst frog ( $P_w$ ) according to better frog in the memplex ( $P_b$ ), at first, the number of features that are removed from or added to the frog is calculated using the following equation:

$$S_b = \begin{cases} \min\{\text{int}(\text{rand}[SP_b - SP_w]), S_{max}\} & \text{if } SP_b > SP_w \\ \max\{\text{int}(\text{rand}[SP_b - SP_w]), -S_{max}\} & \text{else} \end{cases} \quad (8)$$

Where  $SP_w$  and  $SP_b$  are the number of features in the worst and better frogs respectively.  $rand$  is a random number in the range of  $[0,1]$  and  $S_{max}$  is the maximum number of feature changes allowed. In order to make changes in the worst frog, at first, according to the SU criterion, the features of the worst and better frogs are arranged. Then,

```

Input: Training Data
Output: S: The selected subset
Parameters: N: number of features /Par: Parameters of SFLA /
/m: Repetition of the whole process of Relief method
////////// Filtering Section //////////
1   Set W[J] =0
2   For i=1:m
3     Randomly select an instance R
4     Find nearest hit H and nearest miss M
5     For J =1 to N do
6       W[J] = W[J] - diff(J,R,H)/m + diff(J,R,M)/m
7   Probsel[i]= W [i]/Σj=1N W[j]
////////// IWSSr_SFLA Section //////////
8   Generate the initial population by using Probsel
9   Apply IWSSr Algorithm
10  Evaluate the initial population using Fitness function
11  while (Itr < ITmax)
12    Partition the population into sfla_m memplex
13    k=0
14    while( k < sfla_m)
15      k=k+1
16      select k-th memplex;
17      i=1
18      while (i < ITmem)
19        Generate a submemplex base on  $P_j = \frac{2(sfla\_n+1-j)}{sfla\_n(sfla\_n+1)}$  .j = 1.2....sfla_n
20
21        Select the best frog Fb from memplex
22        Select the worst frog Fw from submemplex
23        P'w =IWF(FG, Fw)
24        fit =Evaluate(P'w)
25        If (fit(P'w) > fit(Fw))
26          Replace Fw with P'w
27          i=i+1
28        else
29          Select the best frog FG from whole population
30          P''w =IWF(FG, Fw)
31          fit =Evaluate(P''w)
32          If (fit(P''w) > fit(Fw))
33            Replace Fw with P''w
34            i=i+1
35          else
36            Randomly generate a new frog (P'''w)
37            fit =Evaluate(P'''w)
38            If (fit(P'''w) > fit(Fw))
39              Replace Fw with P'''w
40              i=i+1
41          else
42            Fw = Fw
43        Shuffle all the frog
44        Itr= Itr +1
45
46  S= FG

```

Leaping

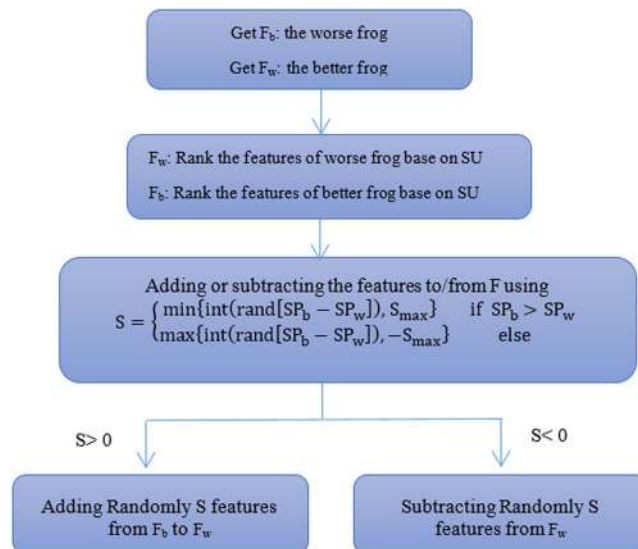
**Figure 5.** Pseudo code of the proposed hybrid algorithm.

if  $S_b$  is a positive number, then  $S_b$  features are randomly added to the worst frog from the better frog. In this case, the features that have high weights are more likely to be selected. Similarly, if  $S_b$  is negative, then  $S_b$  features are randomly deleted from the worst frog. In this case, features that are less weighted are more likely to be selected. In the next step, by using the IWSSr algorithm, the redundant features of the worst frogs are removed.

## Results and Discussions

**Datasets.** In order to evaluate the proposed method, the experiments are performed by MATLAB software on ten gene expression datasets. Summary of the datasets are given in Table 1. Each dataset is described as follows:

Prostate dataset: This dataset contains 12600 genes for 136 samples. 77 samples include prostate tumor and 59 samples are normal<sup>57</sup>. Colon dataset: This dataset contains 2000 genes and 62 samples. 40 samples contain colon cancer and 22 samples are normal<sup>58</sup>. Central Nervous System dataset (CNS): This dataset contains 7129 genes and 60 samples. The dataset includes 21 benign samples and 39 malignant samples<sup>59</sup>. Diffuse Large b-cell lymphoma dataset (DLBCL): This dataset contains 11226 genes for 77 samples. 58 samples including lymphoma tissue, are large cell B, and 19 samples of lymphoma tissue are Follicular lymphoma<sup>60</sup>. Dorothea dataset: This dataset



**Figure 6.** Leap algorithm for worst frog Improvement ( $F_w$ ) by the help of better frog ( $F_b$ ) (IWF).

Data set	Original Data	Training Data	Independent Data	#Gene	#Classes	#class1	#class2
Colon	62	50	12	2000	2	40	22
Arcene	100	80	20	10000	2	44	56
Prostate1	88	71	17	12625	2	38	50
DLBCL	77	61	16	11226	2	58	19
Lung	181	145	36	12533	2	150	31
Dorothea	800	640	160	100000	2	610	190
Prostate	136	109	27	12600	2	77	59
CNS	60	48	12	7129	2	21	39
Leukemia	72	58	14	7129	2	47	25
Breast	97	78	19	24481	2	51	46

**Table 1.** Microarray data sets used in the experiments.

contains 100,000 features and 800 samples. 190 samples are positive and 610 are negative<sup>57</sup>. Leukemia dataset: This dataset contains 7129 genes and 72 samples. Diseases of the leukemia collection are divided into two categories of Acute Lymphoblastic Leukemia (ALL) and Acute Myelogenous Leukemia (AML). The dataset consists of 47 ALL samples and 25 AML samples<sup>61</sup>. Arcene dataset: This dataset contains 10,000 genes and 100 samples. This dataset consists of 56 cancer samples and 44 normal samples<sup>57</sup>. Lung cancer: Gene expression dataset for lung cancer classification between two classes: adenocarcinoma (ADCA); malignant pleural mesothelioma (MPM). The lung dataset contains 181 tissue samples (150 ADCA and 31 MPM). Each sample is described by 12533 genes<sup>62</sup>. Breast cancer: Patients outcome prediction for breast cancer. The training data contains 97 patient samples, 46 of which are from patients who had developed distance metastases within 5 years (labelled as “relapse”), the rest 51 samples are from patients who remained healthy from the disease after their initial diagnosis for interval of at least 5 years (labelled as “non-relapse”). In this data, the number of genes are 24481<sup>57</sup>. Prostate1 dataset: This dataset contains expression levels of 12625 genes taken over 88 samples. (38 normal samples and 50 abnormal)<sup>63</sup>.

**Performance metrics.** To compare the results of the proposed method, seven hybrid methods LFS, IWSS, IWSSr, BARS, GRASP, SVM-RFE and FICA and three filter methods FCBF<sup>24</sup>, F-Score and PCA<sup>51</sup> have been used. The PCA method has been proposed for high-dimensional datasets in recent years. To demonstrate the performance of the proposed method some metrics such as, the number of features obtained, the number of evaluations performed to reach the final subset, accuracy, specificity, sensitivity, and balance rate according to the following formula are measured<sup>64,65</sup>. The number of evaluations indicates the number of subsets tested to reach the final subset.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$



Parameter	Value	Comments
$sfla\_p$	100	Population size
$sfla\_m$	10	Number of memplexes
$sfla\_n$	10	Population size of each memplex
$sfla\_q$	4	Population size of submemplexes
$IT_{max}$	40	Total Iteration number
$IT_{mem}$	10	The number of replications of the division of memplexes into submemplexes
$S_{max}$	5	The maximum leap length allowed to change

**Table 2.** SFLA parameters used in the problem.

DataSet	IWSS		IWSSr		LFS		BARS		FCBF		PCA	
	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts
Colon	80.65	3.8	83.87	2.8	80.80	4.1	85.70	3.0	77.40	14.6	72.50	28.9
Arcene	70.00	13.4	72.00	6.2	73.00	4.5	74.00	4.9	70.00	34.2	—	—
Prostate1	76.23	12.8	77.42	8.3	73.12	3.6	85.34	4.1	63.12	32.4	59.12	37.1
DLBCL	83.11	3.2	81.23	2.7	88.67	4.1	75.21	2.8	96.45	56.2	68.11	42.7
Lung	97.20	2.7	97.20	2.4	93.60	2.5	98.30	3.0	99.40	115.2	85.61	125.2
Dorothea	93.50	7.4	92.90	6.3	90.30	5.5	93.80	7.3	92.60	92.8	—	—
Prostate	77.90	11.1	78.70	7.0	75.40	4.5	86.80	3.7	61.30	35.8	57.35	36.6
CNS	85.21	3.2	86.10	3.1	83.23	3.4	89.12	2.8	93.24	42.2	77.32	44.1
Leukemia	87.50	2.5	87.50	3.0	93.00	3.2	90.50	2.3	95.80	45.8	79.10	53.8
Breast	69.21	11.1	70.21	9.2	70.43	10.1	72.81	9.34	69.43	107.3	63.10	96.3
<b>Mean</b>	<b>82.05</b>	<b>7.122</b>	<b>82.71</b>	<b>5.1</b>	<b>82.15</b>	<b>4.55</b>	<b>85.15</b>	<b>4.32</b>	<b>81.87</b>	<b>57.65</b>	<b>70.27</b>	<b>58.09</b>

**Table 3.** Result of feature selection algorithm.

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (10)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (11)$$

$$BR = \frac{\text{Specificity} + \text{Sensitivity}}{2} \quad (12)$$

The classifier used in the proposed method is support vector machine and in the methods to be compared, Bayesian classifier is used.

When using feature selection methods, it is important to make sure that there is no overlap between the training and test data. Cross validation is an approach that puts data into categories effectively to evaluate feature selection and classification methods. In this approach, the efficiency of the proposed methods is evaluated on the basis of a number of categories derived from the original data. At first, the whole samples of a dataset are randomly divided into  $k$  categories for training and testing purposes. In  $k$  steps,  $(k-1)$  batches are used for model training and one batch is used for testing. At each step, the features and parameters used to test the model are obtained from the training stage and with the help of samples in the training categories. Finally, the efficiency of the proposed method is obtained based on the  $k$  outputs of the training and testing phases<sup>66,67</sup>.

In this paper, Cross Validation (CV) method is used to train and then test the support vector machine classifier based on selected features to determine the percentage of recognition of test data, where  $k=10$ . Since in the 10-fold CV method, the samples are randomly divided into 10 categories, the results depend on how the samples are grouped. To solve this problem, the samples are randomly divided into 10 groups 10 times.

The final number of features is equal to the average of selected features and other criteria are equal to the average of the criteria in selected subset after 10 times execution of proposed method. The performance criteria of the proposed method is also obtained based on the average of 10-fold CV repetitions.

The initial value of hyper parameters of the proposed method is given in Table 2. All hyper parameters are selected based on multiple tests and they are identical in all datasets. To determine the value of hyper parameters, the Random search method is used. For this purpose, a set of hyper parameters is chosen and the model is built based on training data and then it is evaluated based on evaluation data. This process is repeated with other hyper parameters. The hyper parameters that report the best accuracy are selected. In this paper, Population size is set from 80 to 120, Number of memplexes is set from 8 to 12, Population size of submemplexes is set from 3 to 6. The maximum leap length allowed to change ( $S_{max}$ ) is set from 3 to 8.

	Grasp + HC		Grasp + IWSS		Grasp + IWSSr		Grasp + BARS		Grasp + SFS		FICA + IWSSr		F-Score		SVM-RFE		Proposed method	
	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts	Acc	Atts
Colon	81.10	3.0	79.60	3.4	82.20	3.1	80.00	2.9	80.00	3.5	93.60	4.5	83.74	55	93.70	9.8	94.72	5.3
Arcene	80.00	5.7	79.30	6.0	78.50	5.7	79.00	5.2	79.30	6.3	93.40	7.1	73.25	110	89.11	13.5	95.16	8.5
Prostate1	80.45	4.3	79.12	4.1	78.49	3.7	81.12	4.7	78.43	6.3	—	—	68.74	105	82.71	17.2	88.52	8.2
DLBCL	85.65	2.1	84.60	2.2	85.61	2.1	89.11	2.2	85.70	2.4	99.10	4.5	93.11	100	95.23	15.7	99.21	6.8
Lung	95.60	2.2	95.08	2.2	95.70	2.4	96.02	2.3	96.20	2.4	98.90	3	82.16	105	98.73	9.4	99.16	5.6
Dorothea	93.30	3.7	93.30	4.2	92.90	3.8	93.50	5.0	93.20	4.4	75.80	3	76.24	310	84.32	21.7	91.43	7.2
Prostate	77.80	5.0	78.60	5.7	77.50	4.6	78.60	5.1	78.10	5.6	92.40	4.4	54.33	250	92.20	14.4	94.18	7.8
CNS	91.46	2.6	93.12	2.8	87.32	2.8	92.14	3.1	91.12	3.1	—	—	66.53	90	76.96	16.3	95.64	6.7
Leukemia	92.60	2.7	93.70	2.7	91.60	2.8	93.30	2.8	93.60	3.3	99.60	1.8	75.57	70	100.00	8.6	99.62	5.2
Breast	79.63	4.3	80.11	3.1	78.38	3.5	81.24	2.7	80.91	3.6	—	—	73.82	120	86.09	17.3	88.17	10.2
Mean	85.75	3.56	85.65	3.64	84.82	3.45	86.40	3.60	85.65	4.09	—	—	74.74	131.5	89.90	14.39	93.34	7.12

**Table 4.** Comparison of proposed method with GRASP and FICA.

	Training data				Independent data			
	Accuracy	Sensitivity	Specificity	Balance rates	Accuracy	Sensitivity	Specificity	Balance rates
Colon	94.50	95.87	86.11	90.99	93.33	95.00	90.00	92.50
Arcene	94.75	92.57	96.44	94.50	94.00	92.21	95.45	93.83
Prostate1	88.87	86.77	90.50	88.63	88.23	87.77	91.00	89.38
DLBCL	99.50	98.89	94.81	96.85	98.12	98.33	95.00	96.66
Lung	99.13	99.58	96.80	98.19	99.16	99.66	96.66	98.16
Dorothea	91.25	93.70	90.26	91.98	90.37	91.31	89.47	90.39
Prostate	94.18	98.36	90.45	94.41	94.44	96.25	91.81	94.03
CNS	95.31	90.54	97.21	93.88	94.99	92.50	96.25	94.37
Leukemia	99.34	100.00	97.29	98.64	98.57	99.00	97.50	98.25
Breast	88.12	88.23	87.95	88.09	87.89	89.00	85.55	87.27

**Table 5.** Performance results of proposed method in training and independent data.

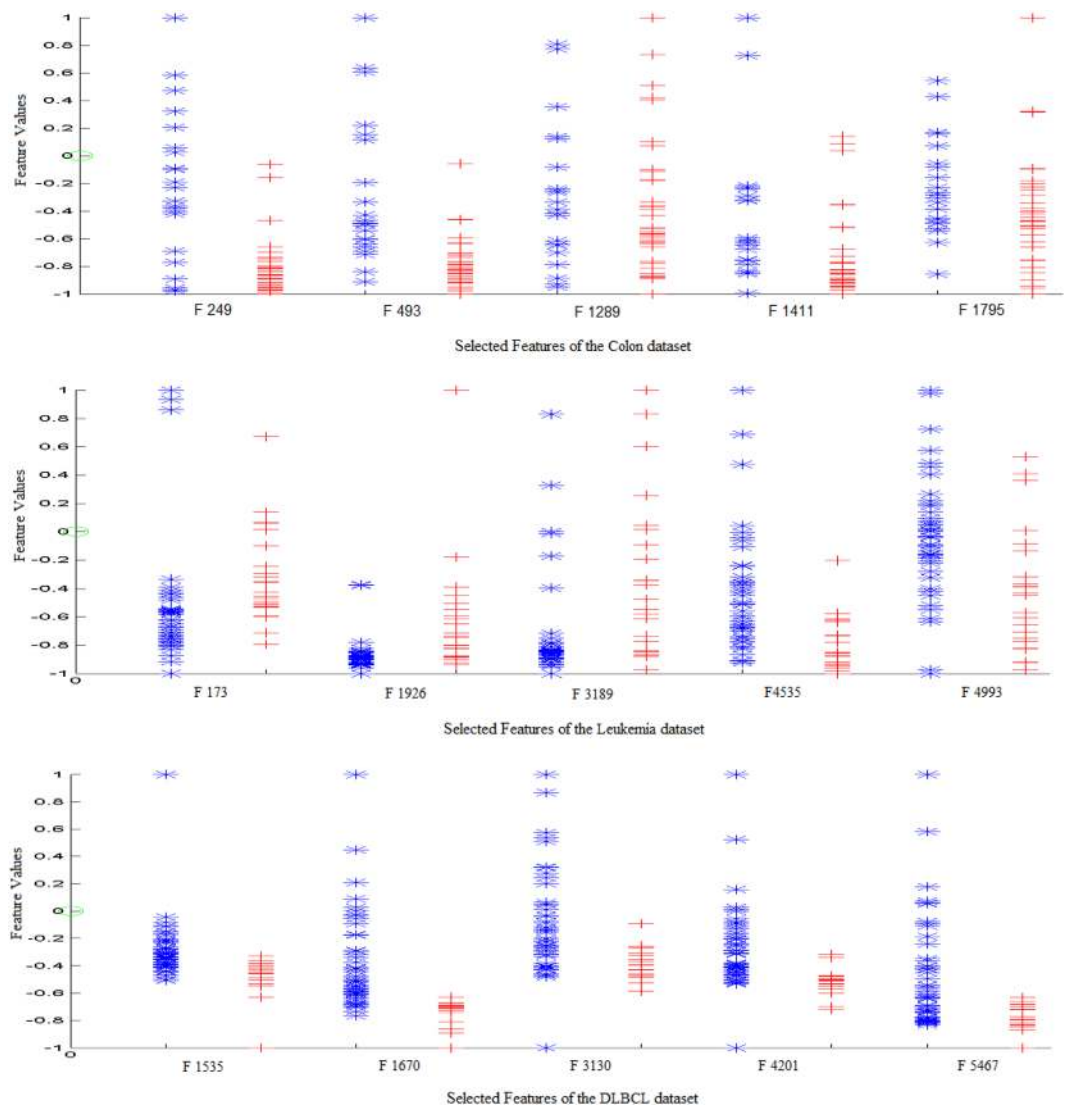
**Experimental results.** In Tables 3 and 4, the results of the implementation of the proposed method have been shown along with comparative methods. In this following tables, acc refers to the accuracy and atts refers to the attribute. According to Table 3, the results approve that the BARS method has fewer features and better accuracy than other methods.

The main idea behind this approach is based on relevancy and redundancy; so the features are added to the selected set that have better information for the classification of the data. The results show that the LFS<sup>5</sup> method has fewer features, but does not have good accuracy. Due to the use of only 100 filtered features to select the subset of features in the wrapper phase, the relationship between the features cannot be considered. The IWSS and IWSSr are wrapper methods. Although the IWSS method finds the subset fast because of relying on the univariate ranking of features, does not consider the relationship between the features. It often fails to find redundant features and the average number of features found by this method is high. In the IWSSr method, in each step of the implementation, the dependence of the assessed feature with all of the features in the selected subset is examined.

Therefore, in addition to the high accuracy, it finds a subset of more compact features in comparison with the IWSS method. However, this method requires a high evaluation time compared to similar methods and runs slow on high dimensional datasets. FCBF and PCA methods are filter-based. These methods only consider the linear relationship between features to find irrelevant features, so they cannot remove the redundant features, and the number of features found in these methods is high.

In Table 4 the proposed method is compared with Grasp, IFCA, F-score and SVM-RFE. In the Grasp method, after finding the candidate subsets, in the local search phase, the methods of IWSS, IWSSr, SFS, BARS, and Hill Climbing are used separately to select the best subset of features. The BARS method selects the best subset of features using a combination of candidate subsets of features and removing the redundant features. The GRASP method, using a two-step algorithm as well as the application of various techniques in the improving phase section, has made progresses in comparison with other methods. However, it is less efficient than the proposed method and FICA. FICA method, because of using the IWSSr method, considers the relationships between features. The Fuzzy Imperialist Competitive Algorithm has been able to remove redundant features properly.

Additionally, the fuzzy influence of imperialist in colonies and the distribution of relevant features of imperialists in the colonial subsets leads to select the subset of optimal features with high-performance. Although this method finds a subset of more compact features than the proposed method, the results show that the accuracy of this method is competitive with the proposed method.

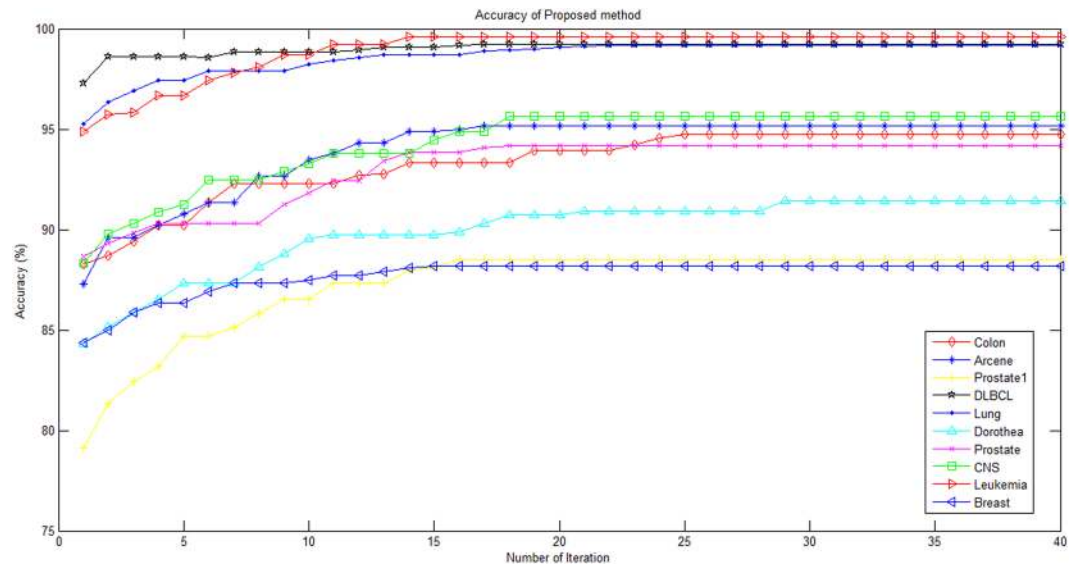


**Figure 7.** Distribution of selected feature values using the proposed method.

The F-score method is usually utilized to compute the degree of difference between two sets of real numbers. The larger the F value, the better the predictive ability of the feature<sup>68</sup>. In this study, F value for all features is calculated in the datasets. Then, the 55 high F values are selected for classification using SVM. Although this method is simple, its detection rate is lower than the proposed method. This method does not indicate mutual information of features. In other words, F-score reveals the discriminative power of each feature independently from other features. Also, the number of selected features in this method is much higher than the other methods.

The SVM-RFE method (Support Vector Machine based on Recursive Feature Elimination) ranks the genes by training a SVM model and selects important genes using recursive feature elimination strategy. In this method, RFE is applied for eliminating unimportant features<sup>69</sup>. Therefore, firstly, the SVM training using initial set of features is performed and the weight is assigned to each feature. Then, these absolute weights are sorted in descending order. Finally, the less weighted features are deleted. The results show that accuracy rate of this method is appropriate, but, the main problem of SVM-RFE is its time complexity, especially when the dimensionality of input data is extremely high. Furthermore, the number of selected genes in this method is higher than the other methods.

The results show that the accuracy of the proposed method in all datasets except Dorothea, is better than other methods. First, it is able to remove irrelevant features in the filter phase, then it removes the redundant features from the subset of features using the hybrid of the SFLA and IWSSr. In this method, due to the improvement of worst frogs, based on better frogs in the memplex and the best frog in the whole set, the redundant and irrelevant features of the frogs are removed and the relevant and useful features are added to the frogs. Removing and adding features are done based on their importance and their relationship with each other. Therefore, the selected feature set is more compact in the best frog and includes relevant features. The results show that in 8 datasets of 10 datasets used, an accuracy of 90% and in 3 datasets, a high accuracy of 98% is achieved. In 10 datasets used in the proposed method the average accuracy of 93.34% is obtained that is better than what obtained from other methods. Additionally, the average of selected features is 7.12, that can be compared to other methods.



**Figure 8.** Mean accuracy of frog's populations in the 40 iterations of training.

Dataset	Minimum number of iterations	average number of iterations	Maximum number of iterations	Average accuracy
Colon	12	13.9	18	94.72
Arcene	4	70.8	9	95.16
Prostate1	6	9.40	15	88.52
DLBCL	9	14.7	16	99.21
Lung	5	80.7	10	99.16
Dorothea	9	11.4	14	91.43
Prostate	17	22.2	30	94.18
CNS	8	9.80	21	95.64
Leukemia	11	16.4	19	99.62
Breast	12	9.50	22	88.17
Average	9.3	12.38	17.4	93.34

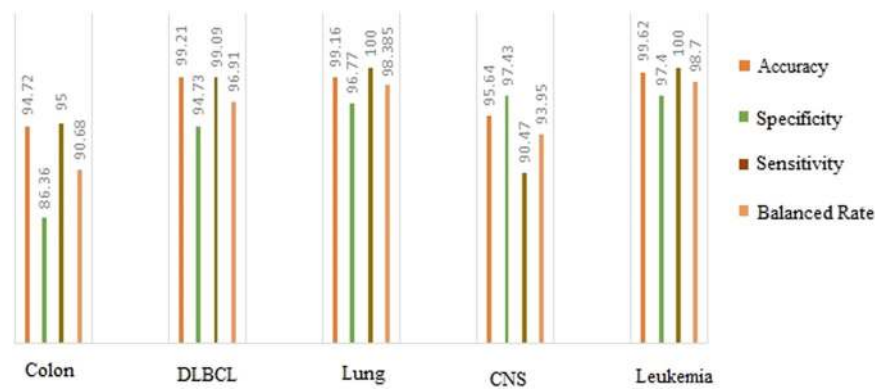
**Table 6.** minimum, maximum and average number of iterations performed by the proposed algorithm.

In order to better evaluation, in this study, each dataset is divided into two datasets; a training dataset and an independent dataset. 80% of the original data is chosen randomly for the training dataset and 20% for the independent dataset. For this purpose, the training dataset is used to train, evaluate and justify the proposed method, and the independent dataset is applied for final performance evaluation of the proposed method. The samples are randomly divided into 2 groups 10 times and the results are averaged over 10 times. The results of these experiments are shown in Table 5. The results approve that the proposed method is robust and it has high accuracy rate. Therefore, the method can be used to classify gene expression data with high accuracy.

In addition, a more detailed analysis of the proposed method, focusing on the features selected, shows some interesting aspects. Figure 7 shows value of the selected features for all samples in some datasets. The proposed method has selected features whose values are less overlapping in the two classes. So these features have distinguished the patterns of two classes even better. It shows that the proposed method has selected appropriate features properly based on the available information. Also, the features in the negative class, especially in the DLBCL and Colon Datasets, have less variance. This property may be important in this regard that in the test and not seen samples, the value of the features is also in the range shown in the Fig. 7. Therefore, the error rate in this class can be less in comparison with other class. However, the value of features in the positive class has more variance. This causes the test data to deviate more than the mean, and the error rate in this class increases. Therefore, feature selection methods should select features that have a high classification accuracy on the test and training data.

To study the process of convergence of the algorithm, the mean accuracy of the method on the datasets in 40 iterations is shown in Fig. 8. As you can see, the learning process is going fast at the beginning, on average in step 20, the algorithm has converged on most datasets, and the accuracy has not increased from this iteration.

Moreover, the minimum, maximum and average number of iterations to achieve convergence of the proposed algorithm using the datasets are shown in Table 6. The Arcene dataset with an average of 7.8 iterations has the lowest convergence time and Breast dataset with an average of 9.5 iterations has the highest convergence time. Overall, the average number of iterations required for all datasets is 12.38 reps.



**Figure 9.** Comparing the performance criterion (accuracy, Specificity, Sensitivity and Balanced Rate) of proposed method.

By checking the number of samples in two classes of data it is clear that the number of data for two classes in Colon, DLBCL, Lung, CNS and Leukemia datasets is not balanced. In this type of data, the method cannot be evaluated only based on the “precision” criterion. Because the method may be biased to the majority class. In order to better evaluate, the accuracy, specificity, sensitivity and balance rates of proposed method in the mentioned datasets are shown in Fig. 9. Obviously, the proposed method has classified the class with more samples properly. However, the class with fewer samples has been classified with fewer classification rate. Due to the low number of samples in the class for correct learning, the classification operation is justifiable. Generally, all the criteria except Specificity in Colon dataset is higher than 90%. The results of the Fig. 9 show that the performance of proposed method in the classification of unbalanced data is also acceptable.

## Conclusion

In this paper, a two-step hybrid algorithm based on Shuffled Frog Leaping Algorithm is proposed. This method uses the advantages of filter and wrapping methods for selecting efficient features. In the filter phase of the proposed method, the Relief method is used for weighting the features of the dataset. Then, in wrapping phase, in the weighted space, by using the Shuffled Frog Leaping Algorithm and the IWSSr algorithm, the search is performed to find the effective and relevant features. In the phase of modifying frogs, removing and adding features are based on their importance and weight. Therefore, the proposed method detects the relationship between the features properly and removes the redundant and irrelevant features from the selected feature set. The proposed method is evaluated using ten gene standard datasets. The experimental results of the proposed algorithm approve that it has the highest accuracy (an average of 93.34%) in comparison with similar methods. Also, the number of features found in each dataset with an average of 7.12 causes high efficiency and a subset of compressed features is achieved.

Received: 5 September 2019; Accepted: 22 November 2019;

Published online: 09 December 2019

## References

- Liu, H. & Motoda, H. *Feature selection for knowledge discovery and data mining*. 454 (Springer Science & Business Media, 2012).
- Liang, S., Ma, A., Yang, S., Wang, Y. & Ma, Q. A review of matched-pairs feature selection methods for gene expression data analysis. *Computational and structural biotechnology journal* **16**, 88–97 (2018).
- Hira, Z. M. & Gillies, D. F. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics* (2015).
- Sun, Y., Lu, C. & Li, X. The cross-entropy based multi-filter ensemble method for gene selection. *Genes* **9**, 258 (2018).
- Speed, T. *Statistical analysis of gene expression microarray data*. (Chapman and Hall/CRC, 2003).
- Golub, T. R. *et al.* Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science* **286**, 531–537 (1999).
- Leung, Y., Chang, C., Hung, Y. & Fung, P. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. 5846–5849 (IEEE).
- Ding, C. & Peng, H. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* **3**, 185–205 (2005).
- Peng, H., Long, F. & Ding, C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1226–1238 (2005).
- Hall, M. A. Correlation-based feature selection for machine learning. (1999).
- Yu, L. & Liu, H. In *Proceedings of the 20th international conference on machine learning (ICML-03)*. 856–863 (2003).
- Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks* **5**, 537–550 (1994).
- Zou, Q., Zeng, J., Cao, L. & Ji, R. A novel features ranking metric with application to scalable visual and bioinformatics data classification. *Neurocomputing* **173**, 346–354 (2016).
- Ding, H. & Li, D. Identification of mitochondrial proteins of malaria parasite using analysis of variance. *Amino acids* **47**, 329–333 (2015).
- Ou, Y.-Y. Identifying the molecular functions of electron transport proteins using radial basis function networks and biochemical properties. *Journal of Molecular Graphics and Modelling* **73**, 166–178 (2017).

16. Brankovic, A., Hosseini, M. & Piroddi, L. A distributed feature selection algorithm based on distance correlation with an application to microarrays. *IEEE/ACM transactions on computational biology and bioinformatics* (2018).
17. Wu, Y.-L., Tang, C.-Y., Hor, M.-K. & Wu, P.-F. Feature selection using genetic algorithm and cluster validation. *Expert Systems with Applications* **38**, 2727–2732 (2011).
18. Benitez, I. P., Sison, A. M. & Medina, R. P. In *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. 238–243 (IEEE).
19. Yang, J. & Honavar, V. In *Feature extraction, construction and selection* 117–136 (Springer, 1998).
20. Jeong, I.-S. *et al.* A Feature Selection Approach Based on Simulated Annealing for Detecting Various Denial of Service Attacks. *Software Networking* **2018**, 173–190 (2018).
21. Debuse, J. C. & Rayward-Smith, V. J. Feature subset selection within a simulated annealing data mining algorithm. *Journal of Intelligent Information Systems* **9**, 57–81 (1997).
22. Sivagaminathan, R. K. & Ramakrishnan, S. A hybrid approach for feature subset selection using neural networks and ant colony optimization. *Expert systems with applications* **33**, 49–60 (2007).
23. Kabir, M. M., Shahjahan, M. & Murase, K. A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications* **39**, 3747–3763 (2012).
24. Deriche, M. In *2009 6th International Multi-Conference on Systems, Signals and Devices*. 1–4 (IEEE).
25. Hu, B. *et al.* Feature Selection for Optimized High-Dimensional Biomedical Data Using an Improved Shuffled Frog Leaping Algorithm. *IEEE/ACM transactions on computational biology and bioinformatics* **15**, 1765–1773 (2016).
26. Pirgazi, J. & Khanteymooori, A. R. SFLA based gene selection approach for improving cancer classification accuracy. *AUT Journal of Modeling and Simulation* **47**, 1–8 (2015).
27. Xue, B., Zhang, M. & Browne, W. N. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics* **43**, 1656–1671 (2012).
28. Chakraborty, B. In *2008 3rd international conference on intelligent system and knowledge engineering*. 1038–1042 (IEEE).
29. Li, J., Fong, S., Wong, R. K., Millham, R. & Wong, K. K. Elitist binary wolf search algorithm for heuristic feature selection in high-dimensional bioinformatics datasets. *Scientific reports* **7**, 4354 (2017).
30. Too, J., Abdullah, A., Mohd Saad, N., Mohd Ali, N. & Tee, W. A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification. *Computers* **7**, 58 (2018).
31. Narendra, P. M. & Fukunaga, K. A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers*, 917–922 (1977).
32. Doak, J. E. *Intrusion detection: The application of feature selection, a comparison of algorithms, and the application of a wide area network analyzer*. (U. of Calif., Davis, 1992).
33. Cotter, S. F., Kreutz-Delgado, K. & Rao, B. D. Backward sequential elimination for sparse vector subset selection. *Signal Processing* **81**, 1849–1864 (2001).
34. Almuallim, H. & Dietterich, T. G. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence* **69**, 279–305 (1994).
35. Caruana, R. & Freitag, D. In *Machine Learning Proceedings 1994* 28–36 (Elsevier, 1994).
36. Mitchell, T. J. & Beauchamp, J. J. Bayesian variable selection in linear regression. *Journal of the American Statistical Association* **83**, 1023–1032 (1988).
37. Bermejo, P., Gámez, J. A. & Puerta, J. M. A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. *Pattern Recognition Letters* **32**, 701–711 (2011).
38. Brahim, A. B. & Limam, M. A hybrid feature selection method based on instance learning and cooperative subset search. *Pattern Recognition Letters* **69**, 28–34 (2016).
39. Shukla, A. K., Singh, P. & Vardhan, M. A hybrid framework for optimal feature subset selection. *Journal of Intelligent & Fuzzy Systems* **36**, 2247–2259 (2019).
40. Ruiz, R., Riquelme, J. C. & Aguilar-Ruiz, J. S. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition* **39**, 2383–2392 (2006).
41. Moradi, P. & Gholampour, M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing* **43**, 117–130 (2016).
42. Ruiz, R., Riquelme, J. C. & Aguilar-Ruiz, J. S. In *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*. 148–162 (2008).
43. Moradkhani, M., Amiri, A., Javaherian, M. & Safari, H. A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm. *Applied Soft Computing* **35**, 123–135 (2015).
44. Lu, H. *et al.* A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* **256**, 56–62 (2017).
45. Sun, L., Kong, X., Xu, J., Zhai, R. & Zhang, S. A Hybrid Gene Selection Method Based on ReliefF and Ant Colony Optimization Algorithm for Tumor Classification. *Scientific Reports* **9**, 8978 (2019).
46. Dao, F.-Y. *et al.* Identify origin of replication in *Saccharomyces cerevisiae* using two-step feature selection technique. *Bioinformatics* **35**, 2075–2083 (2018).
47. Wang, R. AdaBoost for feature selection, classification and its relation with SVM, a review. *Physics Procedia* **25**, 800–807 (2012).
48. Ram, M., Najafi, A. & Shakeri, M. T. Classification and biomarker genes selection for cancer gene expression data using random forest. *Iranian journal of pathology* **12**, 339 (2017).
49. Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine learning* **46**, 389–422 (2002).
50. S. Maldonado, S., Weber, R. & Basak, J. Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences* **181**, 115–128 (2011).
51. Hall, M. A. Correlation-based feature selection of discrete and numeric class machine learning. (2000).
52. Bermejo, P., Gámez, J. A. & Puerta, J. M. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*. 367–374 (IEEE).
53. Eusuff, M., Lansley, K. & Pasha, F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization* **38**, 129–154 (2006).
54. Bhattacharjee, K. K. & Sarmah, S. P. Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. *Applied soft computing* **19**, 252–263 (2014).
55. Pirgazi, J., Khanteymooori, A. R. & Amiri, A. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 1–5 (IEEE).
56. Robnik-Šikonja, M. & Kononenko, I. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning* **53**, 23–69 (2003).
57. Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M. & Herrera, F. A review of microarray datasets and applied feature selection methods. *Information Sciences* **282**, 111–135 (2014).
58. Alon, U. *et al.* Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* **96**, 6745–6750 (1999).
59. Pomeroy, S. L. *et al.* Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature* **415**, 436 (2002).
60. Shipp, M. A. *et al.* Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine* **8**, 68 (2002).

61. Stuart, R. O. *et al.* In silico dissection of cell-type-associated patterns of gene expression in prostate cancer. *Proceedings of the National Academy of Sciences* **101**, 615–620 (2004).
62. Gordon, G. J. *et al.* Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer research* **62**, 4963–4967 (2002).
63. Singh, D. *et al.* Gene expression correlates of clinical prostate cancer behavior. *Cancer cell* **1**, 203–209 (2002).
64. Ho, Q.-T. & Ou, Y.-Y. Classifying the molecular functions of Rab GTPases in membrane trafficking using deep convolutional neural networks. *Analytical biochemistry* **555**, 33–41 (2018).
65. Le, N. Q. K., Huynh, T.-T., Yapp, E. K. Y. & Yeh, H.-Y. Identification of clathrin proteins by incorporating hyperparameter optimization in deep learning and PSSM profiles. *Computer Methods and Programs in Biomedicine* **177**, 81–88 (2019).
66. Jung, Y. & Hu, J. AK-fold averaging cross-validation procedure. *Journal of nonparametric statistics* **27**, 167–179 (2015).
67. Arlot, S. & Celisse, A. A survey of cross-validation procedures for model selection. *Statistics surveys* **4**, 40–79 (2010).
68. Ou, Y.-Y. Prediction of FAD binding sites in electron transport proteins according to efficient radial basis function networks and significant amino acid pairs. *BMC bioinformatics* **17**, 298 (2016).
69. Li, Z., Xie, W. & Liu, T. Efficient feature selection and classification for microarray data. *PLoS one* **13**, e0202167 (2018).

### Author contributions

J.P., M.A. and T.E.A. designed the research, J.P. and M.A. collected data, J.P., T.E.A. and M.H.O. wrote and performed computer programs, J.P., M.A., T.E.A. and M.H.O. analyzed and interpreted the results, M.A. and T.E.A. wrote the first version of the manuscript, J.P., M.A. and M.H.O. revised and edited the manuscript.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to J.P.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019