# Modified Floating Search Feature Selection Based on Genetic Algorithm

*Kanyanut* Homsapaya*, and *Ohm* Sornil

Graduate School of Applied Statistics National Institute of Development Administration, 118 Serithai Rd., Bangkapi, Bangkok, 10240, Thailand

**Abstract.** Classification performance is adversely impacted by noisy data .Selecting features relevant to the problem is thus a critical step in classification and difficult to achieve accurate solution, especially when applied to a large data set. In this article, we propose a novel filter-based floating search technique for feature selection to select an optimal set of features for classification purposes. A genetic algorithm is utilized to increase the quality of features selected at each iteration. A criterion function is applied to choose relevant and high-quality features which can improve classification accuracy. The method is evaluated using 20 standard machine learning datasets of various sizes and complexities. Experimental results with the datasets show that the proposed method is effective and performs well in comparison with previously reported techniques.

**Key words:** Feature selection, floating search, genetic algorithm.

## 1 Introduction

Classification, a process for predicting a class of a given input data, is one of the most fundamental tasks in data mining. A number of methods are commonly used for data classification, such as decision trees; rule-based, probabilistic and instance-based methods; support vector machines (SVMs); and neural networks. Noisy and irrelevant data are major obstacles to data mining.

Selecting features relevant to the problem is a critical first step in classification, especially when applied to a large dataset. The aim is to select a representative subset of highly relevant dimensions while removing irrelevant and redundant ones [1]. Feature selection can significantly improve the running time of a machine learning algorithm as well as improve the quality of the model.

Consequently, Bins and Draper [2] proposed a technique to reduce a large set of features (1 000) to a much smaller subset without removing any highly important features or decreasing classification accuracy. There are three steps in the algorithm: first, irrelevant features are removed using a modified form of the relief algorithm [3]; second, redundant features are eliminated using K-means clustering [4]; and, lastly, a combinatorial feature selection algorithm is employed to the current feature subsets using the sequential floating backward selection (SFBS) algorithm. The basic concept is to filter feature subsets on each step until the smallest possible one is obtained. In this article, we propose a technique to improve the effectiveness of the floating search feature selection method which leads to a higher classification rate. Our method employs a genetic algorithm to enrich and improve the resultant features after each iteration of the sequential forward feature search (SFFS) process.

## 2 Backgrounds

### 2.1. Feature selection

Two important components of the feature selection process: subset generation and subset evaluation are shown in Figure 1. The subset generation engine identifies feature subset candidates and subset evaluation measures the quality of the subsets. Lastly, in order to terminate the process, a stopping criterion is tested at every iteration.

There are three main types of feature selection method: filter, wrapper and hybrid. Wrapper methods rely on a classification algorithm employed as the subset evaluation process for feature subsets [5]. Maroño et al. [6] proposed a wrapper method by applying ANOVA. In general, the wrapper approach gives a higher performance than the filter approach since the feature selection process is optimized for the specific classification algorithm. Nevertheless, when wrapper methods are applied to huge dimensional datasets, they will incur high computational cost and may become unfeasible.

Filter methods use an independent criterion which relies on general characteristics of the data to evaluate and select feature subsets without involving a classification algorithm. Common evaluation functions usually are measures such as distance, mutual information (MI), dependency or entropy, calculated directly from the training data. Karegowda et al. [7] developed a filter-based technique in a cascade fashion with a genetic algorithm (GA) using a correlation-based criterion. Dash and Liu [1] proposed Hybrid methods exploit the positive aspects of both wrapper and filter methods. It utilizes a filter-based technique to select highly representative features and applies a wrapper-based technique to add candidate features and evaluate the candidate subsets in order to select the best ones.

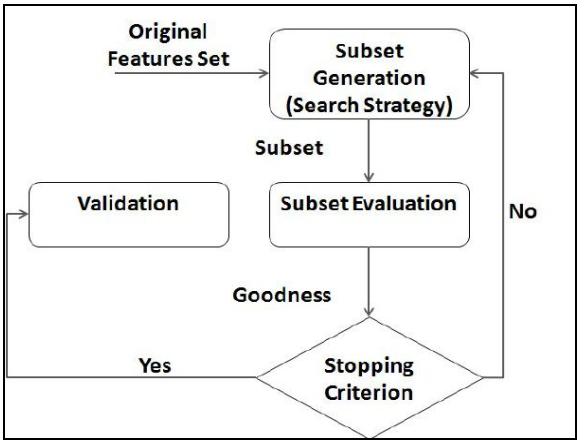


**Fig. 1.** The feature selection process.

The sequential forward search (SFS) method operates in a forward search manner starting with an empty set and adds one feature subset during each round until a new feature subset that maximizes the criterion function value is found, whereas the sequential backward search (SBS) method starts with a full feature subset and eliminates a feature on each iteration until a predetermined criterion is satisfied. A drawback of both methods is that they have a nesting effect problem, which means that the features discarded cannot be re-selected, and the features selected cannot be removed later. Since these algorithms do not examine all possible feature subsets, they are not guaranteed to produce an optimal result. Generalized forms GSFS and GSBS based on group collection feature testing are better solutions, but at the cost of increased computational time. The plus l take away r (PTA) method was proposed to take care of the nesting problem [8].

### 2.2 Floating search method

Pudil et al. [9] proposed “floating” search methods based on two main categories: the search process in a forward direction (SFFS) and the process in a backward direction (SBFS). These methods use a criterion function to select a feature and compare candidate subsets. SFFS and SBFS can be classified as a wrapper or a filter approach depending on the criterion function used. They perform well but the computational time is long, especially with large datasets. The floating search methods can be viewed as predictive text algorithms (PTAs) without the use of a fixed parameter. They have been shown to give very good performance (close to optimum results) and to overcome the nesting problem. SFFS, SBFS, and bidirectional selection as a combination of both are greedy search algorithms that add or discard features one at a time [9]. The floating search method consists of two phases: forward and backward. SFFS starts with

an empty set and sequentially adds one feature at a time. The structure of the floating search algorithm is shown in Figure 2.

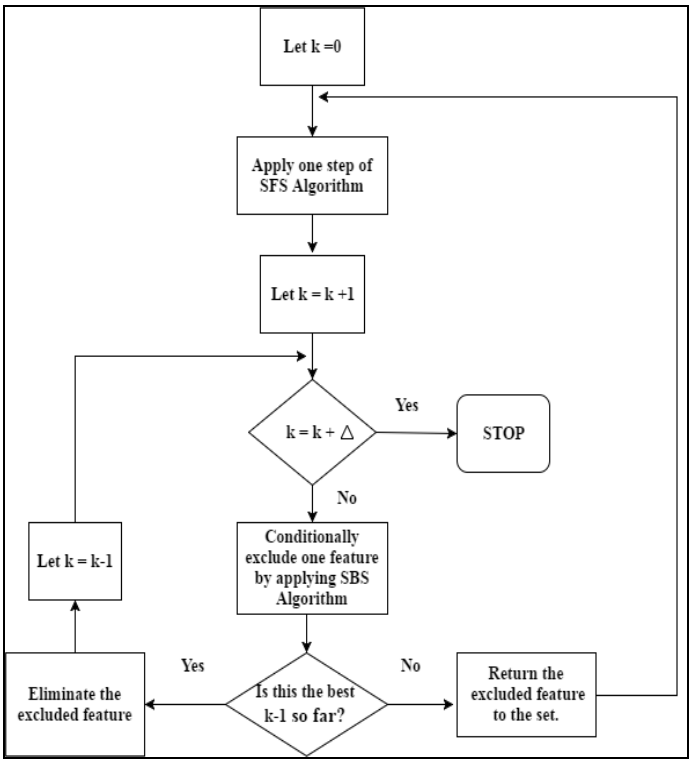


**Fig. 2.** The structure of a floating search algorithm.

SBFS, the counterpart of the forward search, is initialized with a full set and sequentially eliminates one feature at a time after execution of SFFS. An SFFS search selects the best unselected feature according to a criterion function to form a new feature subset, and an SBFS search iteratively determines which members of the selected subset are to be removed if the remaining set improves performance according to the same criterion function in the forward search. The algorithm loops back to a forward search until the stopping condition is reached. There are disadvantages when using either algorithm. With SFFS, it is not possible to succeed in eliminating redundant features generated in the search process, whereas SBFS cannot re-calculate evaluation feature usefulness together with other features at the same time.
Improved versions of SFFS have been proposed in many researches to obtain better performance. Somol et al. [10] presented the adaptive sequential forward floating selection (ASFFS) algorithm with a parameter "r" which specifies the number of features to be added in the inclusion phase calculated dynamically. Parameter "o" is used in the exclusion phase to remove the maximum number of features if it improves performance. Nakariyakul and Casasent [11] came up with an improved forward floating search algorithm, which has a new search step to check whether to replace a weak feature and remove it again until the replacement can no longer improve the criterion function. They found that this method obtained optimal solutions for many feature subsets and was less computationally intensive than exhaustive search optimal feature selection algorithms. Chaiyakarn and Sornil [12] proposed a filter-based method to return a small subset of features for classification by employing two different criterion functions in the forward and backward steps. The functions helped remove redundant features, maximize inter-class distances, and minimize intra-class distances.

### 2.3 Feature subset evaluation

In order to perform feature selection with the filter approach, a measure is needed to evaluate the relevance of the subset to the classification process.

#### *2.3.1 Mahalanobis distance*

The Mahalanobis distance is very helpful solution of determining the "similarity" of a set of values from an "unknown" sample to a set of values measured from a collection of "known" samples. Yongli [13] used the Mahalanobis of candidate feature subsets and selected the best quality subset to be used as input data. One of the main reasons the Mahalanobis distance method is used is that it is very sensitive to inter-variable changes in the training data. The Mahalanobis distance between two points $x = (x_1,...,x_p)^t$ and $y = (y_1,...,y_p)^t$ in the p-dimensional space $R_p$ is defined as:

$$d_S(x,y) = \sqrt{(x-y)^t S^{-1}(x-y)} \tag{1}$$

and $d_S(x,0) = \|x\|_S = \sqrt{x^t S^{-1} x}$ is the norm of x.

#### *2.3.2 Mutual information*

In order to perform feature selection with the filter approach, measures are needed to evaluate the relevance of the subset to the classification process. MI is a widely used measure to evaluate candidate feature subsets. Battiti [14] used MI on candidate feature subsets to select a quality subset to be used as input data for a neural network classifier. MI measures absolute dependencies between random variables and can be calculated as follows:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{2}$$

Where $H$ is an entropy function, $Y$ is a class attribute, and $X$ is the feature to select. Given a random variable $X$ such that:

$X = 0$ with probability $p$

$X = 1$ with probability $1 - p$

$$H(X) = -p \log p - (1-p)\log(1-p) = H(p) \tag{3}$$

A genetic algorithm (GA), introduced by John Holland in 1975 [15], is an adaptive optimization search algorithm to find an optimal solution inspired by natural selection in biological systems. The genes of an organism are gathered into structures called chromosomes, and a set of chromosomes is referred to as a population. In general, there are three operations employed in GAs. First, selection is an operator for selecting potentially useful solutions for recombination, and is achieved by either tournament or roulette wheel selection. Second, crossover refers to the process of producing an offspring chromosome from two matching parent chromosomes. Third, mutation causes genetic diversity of chromosomes by making random binary changes in a chromosome, thus adversely affecting their fitness value. These principles have led to new solutions in the pursuit of better search solutions.

### 2.4 Genetic algorithm

GAs have been successfully applied to feature selection [16] with the objective to save on computational time without processing in an exhaustive fashion, which is achieved by finding promising regions and selecting quality feature subsets. Furthermore, hybrid GAs [17] are involved in a new search method that includes local search operators to improve the fine-tuning quality of a simple GA search.

The fitness function, based on the principle of survival of the fittest, is the process whereby a GA evaluates each individual's fitness and obtains the optimal solution after applying the genetic operators. This process is repeated many times and over many generations until the stopping criterion is satisfied. For feature selection, the feature subsets are represented as a binary; a feature is either included or not included in the feature subset.

## 3 The proposed algorithm

We now discuss our algorithm to select the best subset of size d of the total of D features. The inclusion step using MI as the criterion function (J) is executed to create a set of candidates for inclusion. In the exclusion step, a candidate feature subset is used to generate smaller subsets from the result of the inclusion step by removing one feature and re-evaluating them. A selection subset of size k + 1 is generated and compared to the previously best subset of size k + 1

from the inclusion part. If evaluation of the new subset is more qualified than the formerly selected set, the exclusion step retains the better one and iterates to smaller subsets, or else the algorithm goes back to the inclusion step.
Our feature improvement step based on GA is included after the exclusion step at each iteration. The chromosome structure consists of binary genes, corresponding to individual features. The value of 1 at the *i*th gene means that the *i*th feature is selected; otherwise it is 0.

The initial population is generated from the resulted subsets of size k + 1 from the exclusion step by first removing the weakest features from the best subset resulting in a subset of size k. Each remaining feature is thus added to that subset generating the niched initial population for GA. The fitness function used in this study is MI. Then, a new population is created by selection, crossover and mutation operations. The process is terminated when the current feature set reaches the size of D-2 features. We now provide an illustrative example of how the proposed algorithm works and how it improves SFFS. Assume that the first five feature sets selected by the SFS method at each size are {f1}, {f1, f4}, {f1, f4, f5}, {f1, f4, f5, f7} with the corresponding J values of 4.1, 6.2, 9.1 and 10.2, respectively, and the next iteration is to determine subsets with five features.

### 3.1 Step 1: Inclusion

A feature is added to the feature subset. The SFS method adds a feature to the subset up to a total of five: J (f1, f4, f5, f7, f6) = 13. Assume that feature f6 is chosen using the SFS method and J for the 5th features is 14.

### 3.2 Step 2: Exclusion

A feature is removed from the feature subset. The SBS method is applied in this step by backtracking and conditionally removing one feature from the subset selected in Step 1 and returning an improved subset, e.g. (f1, f5, f6, f7) J value = 11; (f1, f4, f5, f7) J value = 9; (f1, f4, f7, f6) J value = 9.5; and (f4, f5, f7, f6) J value = 10. In this case, the best feature subset of size 4 is (f1, f5, f6, f7).

### 3.3 Step 3: Feature improvement

The weakest feature is removed from the subset from of size k the previous step which is (f1, f5, f6, f7) by iteratively evaluating the smaller subsets: (f1, f5, f7), (f1, f5, f6), (f5, f6, f7) and (f1, f7, f6). In this case, we assume that the best performance subset of size 3 is (f5, f7, f6). Then, each feature is added to each subset of (f5, f7, f6) in order to find the best four feature subset, either (f5, f7, f6, f1), (f5, f7, f6, f2), (f5, f7, f6, f3), (f5, f7, f6, f4), (f5, f7, f6, f8), or (f5, f7, f6, f9). Top n chromosomes is selected as the initial population for GA and passed through the crossover and mutation operations.

#### *3.3.1 Step 3.1: Crossover operation*

Once a pair of chromosomes has been selected, crossover can take place to produce child chromosomes. A crossover point is randomly chosen from two randomly selected individuals (parents). This point occurs between two bits and divides each individual into left and right sections. Crossover then swaps the left (or the right) section of the two individuals thus (Figure 3):

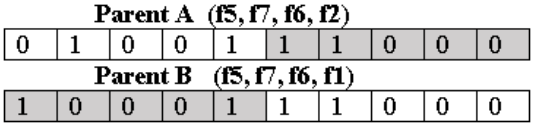


**Fig. 3.** Crossover operation.

Suppose the crossover point randomly occurs after the sixth bit, then each new child receives one half of each parent's bits (Figure 4):

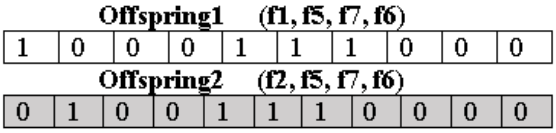


**Fig. 4.** Child chromosome.

This algorithm continues to select parental chromosomes to apply the crossover operation. Child chromosomes may have one bit more than the current size of k features subset. In this case, a random bit is automatically flipped to preserve the size of the chromosome (i.e. current feature set size).

### *3.3.2 Step 3.2: Mutation operation*

The mutation operation (Figure 5) is applied to all of the offspring chromosomes from the crossover step. Mutation operates at the bit level by randomly flipping bits in the new chromosome within the current population (turning a '0' into '1', and vice versa).

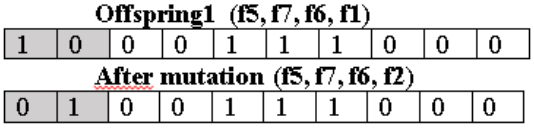


**Fig. 5.** Mutation operation.

After all child chromosomes have passed through the mutation operator, the resultant chromosomes are evaluated by the fitness function. After this, we can discover the best performing features subset, which is (f5, f7, f6, f2). We assume that J ({f5, f7, f6}) = 8.35, and that J ({f5, f7, f6, f2}) = 12, which is larger than the prior largest value for four features, J = 11 Thus, the best 4-feature subset becomes {f5, f7, f6, f2} with J = 12, whereas the best 3-feature subset remains {f1, f4, f5} since J ({f1, f4, f5}) = 9.1 > J ({f5, f7, f6}) = 8.35.

The improvement step helps discover subsets not discoverable by the greedy nature of SFFS. From the above example, the SFFS algorithm is not able to produce this best 4-feature four subset because it cannot backtrack to the set {f5, f7, f6}, thus could not add feature f2 to subset {f5, f7, f6}. The example above demonstrates the advantage of our proposed algorithm. The algorithm replaces the weak feature (feature f1 in our example) in the feature set {f1, f5, f7, f6} with feature 2, which results in a new set of four features {f5, f7, f6, f2} which has a larger J value. Therefore, the search strategy of our proposed algorithm is more thorough than the SFFS algorithm, so it is more effective.

### **3.4 Step 4: Terminating condition**

After each iteration, the selection/crossover per mutation cycle continues until all possible combinations of chromosomes in the population have been evaluated. The higher the fitness value, the higher the probability of that chromosome being selected for reproduction. This generational process is repeated until a pre-determined termination condition has been reached. We terminate the algorithm when the current feature set reaches d < D features, where D is the total number of features in the dataset). The pseudo-code is depicted in Figure 6.

A fitness function is commonly needed in GAs to evaluate a candidate chromosome of an individual to assess whether the latter should survive or not. At each iteration, calculation of the fitness function is processed repeatedly, which, because of its simplicity, is a fast process, although it still impacts performance. In our model, we use the Mahalanobis criterion as a fitness function.

Input: $Y_m$ is a feature set, $m$ is a predefined number of selected features, J is a criterion function. $P_c$ is probability of crossover, $P_m$ is probability of mutation, Population is set of individuals, max_generation is the maximum number of generations, and Fitness is a function which determines quality of individuals. Output: The best solution in all generation.

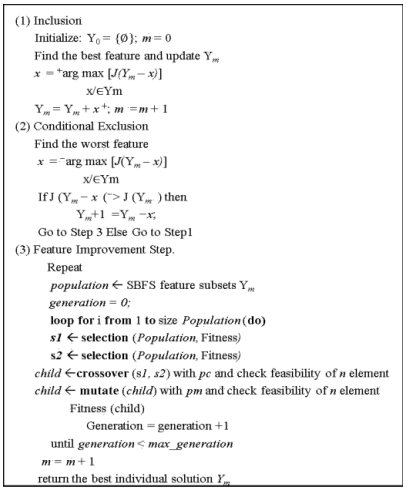


**Fig. 6**. Pseudo-code of the proposed algorithm.

## 4 Experimental evaluation

To evaluate the proposed feature selection algorithm, 20 standard datasets of various sizes and complexities from the UCI machine learning repository [18] are used in the experiments. These datasets have been frequently used as a benchmark to compare the performance of classification methods and consist of a mixture of numeric, real and categorical attributes. Details of the datasets are shown in Table 1.

Three classification modeling techniques are used in the experiments which consist of Classification and Regression Tree (CART), Support Vector Machine (SVM), and Naïve Bayes. Training and testing data is used as provided in the datasets. For those not providing separate testing data, a 5-fold cross validation is applied. To evaluate a feature subset, MI is applied as the criterion function.

### 4.1 The classifiers

CART is a well-known decision tree algorithm for supervised machine learning that is applied to both classification and regression problems. It was first introduced by Brieman et al. [19]. A decision tree represents a series of decisions. The key components of the tree are a set of rules for splitting each node in the tree, and assigning a class outcome to each terminal node.

The Naïve Bayes algorithm is a statistical classifier for supervised learning [19], and is based on the principle of conditional probability. It can predict class membership probabilities, such as the probability that a given sample belongs to a particular class, and its performance has been shown to be excellent in some domains but poor on specific domains, e.g. those with correlated features. The classification system is based on Bayes 'rule under the assumption that the effect of an attribute on a given class is independent from the other attributes. This assumption is called the class conditional independence which makes computation simple.

SVMs, originally proposed by Cortes and Vapnik [20], have become important in many classification problems for a variety of reasons, such as their flexibility, computational efficiency, and capacity to handle high dimensional data. They are a recent method to extract information from a dataset. Classification is achieved by a linear or nonlinear separating surface in the input space of the dataset. SVMs have been applied to a number of applications, such as bioinformatics, face recognition, text categorization, handwritten digit recognition, and so forth. SVM is a binary classifier assigning a new data to a class by minimizing the probability of error

### 4.2 Performance of the proposed techniques using classifiers

We studied the effectiveness of the proposed feature selection using three different classification methods: CART, SVM and Naïve Bayes on 20 standard UCI datasets. The results in Table 1 show that, in 97.7 % of the cases, the proposed technique improved classification effectiveness and greatly reduced the number of features selected, thus increasing classification efficiency, for all of the classification methods. We actually achieved 100 % selection accuracy from four

datasets with the proposed method. In a comparison of the classification methods, SVM yielded the highest classification accuracy in 65 % of the datasets while CART gave the highest accuracy in 35 % of the datasets.

**Table 1.** Classification effectiveness.

| Dataset | Original Datasets | No. of Attributes | PM* CART | PM* SVM | PM* Naïve Bayes |
|---|---|---|---|---|---|
| Wine | 88.87 | 13 | 100.00 (7) | 100.00 (7) | 96.43 (7) |
| Breast Cancer (Original) | 93.13 | 10 | 97.42 (5) | 97.55 (5) | 92.12 (5) |
| Breast Cancer (WDBC) | 92.23 | 32 | 95.5 (5) | 96.10 (8) | 91.00 (8) |
| Breast Cancer (WPBC) | 72.00 | 34 | 84.23 (6) | 86.36 (6) | 79.00 (6) |
| Iris | 94.00 | 4 | 98.95 (3) | 100.00 (3) | 94.60 (3) |
| Pima -Indian Diabetes | 72.51 | 8 | 74.38 (4) | 77.00 (4) | 71.89 (4) |
| Abalone | 49.07 | 8 | 55.00 (4) | 58.50 (4) | 50.00 (4) |
| Dermatology | 95.08 | 34 | 98.03 (26) | 97.5 (26) | 94.15 (26) |
| Heart | 76.67 | 13 | 82.00 (6) | 84.00 (6) | 79.00 (6) |
| German | 68.50 | 20 | 73.00 (6) | 72.5 (6) | 70.00 (6) |
| Lung cancer | 59.67 | 56 | 78.00 (21) | 85.33 (21) | 72.00 (21) |
| Soy bean | 85.00 | 35 | 100.00 (22) | 100.00 (22) | 98.28 (22) |
| Spambase | 93.26 | 57 | 96.00 (26) | 93.00 (26) | 91.76 (26) |
| Glass Identification | 62.00 | 10 | 63.13 (5) | 67.00 (5) | 65.00 (5) |
| Teaching Assistant | 54.92 | 5 | 60.03 (2) | 61.86 (2) | 62.00 (5) |
| Contact Lens | 76.00 | 4 | 80.00 (2) | 100.00 (2) | 85.00 (2) |
| Sonar | 69.50 | 60 | 84.00 (18) | 85.70 (18) | 75.00 (7) |
| Statlog (Australian) | 65.45 | 14 | 75.30 (7) | 80.00 (7) | 75.24 (7) |
| Ionosphere | 84.00 | 34 | 91.00 (6) | 92.62 (7) | 90.10 (5) |
| Image Segmentation | 85.00 | 19 | 92.00 (14) | 90.57 (14) | 84.10 (14) |

*Proposed Method
*Note.* Results expressed as percentages.

**Table 2.** Comparison with other previously reported methods on common datasets [21–27].

| Dataset | PM* CART | PM* SVM | [Liu, 2004] | [Ratanamahatana, 2003] | [Anwar, 2014] | [Yang, 2010] | [Gupta, 2015] | [Tsai, 2011] | [Lavanya, 2011] |
|---|---|---|---|---|---|---|---|---|---|
| Breast Cancer (Original) | 97.42 | 97.55 | - | 97.4 | 94.4 | 96.5 | - | - | 94.8 |
| Breast Cancer (WDBC) | 95.5 | 96.1 | 95.4 | - | - | - | - | - | 93.0 |
| Iris | 98.95 | 100 | 97.3 | | - | 97.3 | 96.7 | 96.6 | - |
| Pima Indian Diabetes | 74.38 | 77.0 | 73.8 | 79.9 | 76.0 | 73.2 | - | - | - |
| German | 73.0 | 72.5 | 72.6 | 76.2 | - | 74.5 | - | 69.9 | - |
| Soybean | 100 | 100 | - | 88.3 | - | 97.8 | - | - | - |
| Wine | 100 | 100 | - | - | 91.6 | 98.3 | - | - | - |
| Heart | 82.0 | 84.0 | - | - | 61.1 | 84.8 | 87.1 | - | - |
| Sonar | 84.0 | 84.7 | - | - | 83.7 | - | - | - | - |
| Abalone | 55.0 | 58.5 | 54.5 | - | - | - | 30.0 | 25.7 | - |
| Dermatology | 98.03 | 97.5 | - | - | - | 95.4 | - | - | - |
| Contact Lenses | 80.0 | 100 | - | - | - | - | 75.0 | - | - |

*Proposed Method
Note. Results expressed as percentages.

## 5 Conclusion

Feature selection is critical to the performance of classification. We propose a feature selection algorithm that improves the performance of SFFS by incorporating a feature improvement step based on a genetic algorithm. This step helps discover important subsets that are not possible using SFFS alone. The algorithm employs mutual information as the feature subset evaluation function. The proposed technique was evaluated using 20 standard datasets from the UCI

repository using three different classification methods. The results show that the proposed feature selection technique significantly improves classification accuracy and gives a much smaller feature set, thus improves efficiency. In addition, it performed very well in comparison with other previously reported studies.

## References

1. M. Dash, H. Liu. Intelligent Data Analysis, **1**(1-4):131–156 (1997). https://www.sciencedirect.com/science/article/pii/S1088467X97000085
2. J. Bins, B. Draper. *Feature selection from huge feature sets*. Proceedings on Eighth IEEE International Conference on Computer Vision (Vancouver, Canada, 2001). http://ieeexplore.ieee.org/document/937619/
3. K. Kira, L. Rendell. *The Feature Selection Problem: Traditional Methods and a New Algorithm*, AAAI-92 Proceedings, pp .129-134, (San Jose, California, 1992) https://www.aaai.org/Papers/AAAI/1992/AAAI92-020.pdf
4. MacQueen, J., *Some Methods for Classification and Analysis of Multivariate Observations,* Proceedings of the Fifth Berkley Symposium on Mathematics, Statistics and Probability, pp. 281-297, 1967.
5. I. Guyon, A. Elisseeff. J. Mach. Learn. Res. **3**:1157–1182 (2003). http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf
6. Maroño, N.S., Betanzos, A. & Castillo, E., *A new wrapper method for feature subset selection*. Proceedings European Symposium on Artificial Neural Networks, pp. 515–520. (Belgium, 2005). https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2005-106.pdf
7. A.G. Karegowda, M.A. Jayaram, A.S. Manjunath. IJCA, **23**(2):1–10 (2011). http://www.ijcaonline.org/archives/volume23/number2/2865-3711
8. H. Zhang, G. Sun. Pattern Recognit. **35**(3):701-711 (2002). https://www.sciencedirect.com/science/article/pii/S0031320301000462
9. P. Pudil, J. Novovicova, J. Kittler. Pattern Recognit. Lett. **15**(11):1119–1125 (1994). https://www.sciencedirect.com/science/article/pii/0167865594901279
10. P. Somol, P. Pudil, J. Novovicova. *Flexible hybrid sequential floating search in statistical feature selection*. Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science. A. Fred, T. Caelli, R.P.W. Duin, A. Campilho, D. Ridder (eds.), Springer, **4109**:632–639 (2006). https://link.springer.com/chapter/10.1007/11815921_69
11. S. Nakariyakul, D.P. Casasent. Pattern Recognit. **41**(9):1932–1940 (2009). https://www.sciencedirect.com/science/article/pii/S0031320308004937
12. O. Sornil. *Filter-based feature selection using two criterion functions and evolutionary fuzzification*. International Workshop on Multi-disciplinary Trends in Artificial Intelligence. Springer International Publishing. (Chiang Mai, Thailand, 2016). https://link.springer.com/chapter/10.1007/978-3-319-49397-8_15
13. Z. Yongli, Z. Yungui, T. Weiming, C. Hongzhi. *An improved feature selection algorithm based on MAHALANOBIS distance for network intrusion detection*. International Conference on Sensor Network Security Technology and Privacy Communication System (SNS & PCS), IEEE (Nangan, China, 2013). http://ieeexplore.ieee.org/document/6553837/
14. R. Battiti. IEEE Transactions on Neural Networks, **5**(4):537–550 (1994). http://ieeexplore.ieee.org/document/298224/
15. D. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Boston: Addison Wesley (1989). https://dl.acm.org/citation.cfm?id=534133
16. F. Brill, D. Brown, W. Martin. IEEE Transactions: Neural Networks, **3**(2):324–328 (1992). http://ieeexplore.ieee.org/document/125874/
17. I.S. Oh, J.S. Lee, B.R. Moon. IEEE Transactions: Pattern Analysis and Machine Intelligence, **26**(11):1424–1437 (2004). http://ieeexplore.ieee.org/document/1335448/
18. A. Asuncion, D.J. Newman. *UCI machine learning repository*. University of California, Department of Information and Computer Science. [Online] from http://archive.ics.uci.edu/ml/index.php
19. L. Brieman, J. Friedman, R. Olshen, C. Stone. *Classification of regression trees*. Routledge: Wadsworth Inc. (1984). https://books.google.co.id/books?id=gLs6DwAAQBAJ
20. C. Cortes, V. Vapnik. Machine Learning, 20(3):273–297 (1995). https://link.springer.com/article/10.1007/BF00994018
21. Ratanamahatana, C. & Gunopulos, D. Appl. Artif. Intell., 17(5-6):475–487 (2003). http://www.tandfonline.com/doi/abs/10.1080/713827175
22. H. Liu, H. Motoda, L. Yu. Artificial Intelligence, **159**(1–2):49–74 (2004). https://www.sciencedirect.com/science/article/pii/S0004370204000980
23. H. Anwar, U. Qamar, A.W.M. Qureshi. Sci. World J. **2014**(Article ID 313164):1–9 (2014). https://www.hindawi.com/journals/tswj/2014/313164/

24. T. Yang, L. Cao, C. Zhang. *A novel prototype reduction method for the K-nearest neighbor algorithm with K ≥ 1*. Pacific-Asia Conference on Knowledge Discovery and Data Mining **Part II**:89–100 (Hyderabad, India, 2010). https://link.springer.com/chapter/10.1007/978-3-642-13672-6_10
25. A. Gupta. IJSTR, **4**(5):85–94 (2015). http://www.ijstr.org/final-print/may2015/Classification-Of-Complex-Uci-Datasets-Using-Machine-Learning-And-Evolutionary-Algorithms.pdf
26. C.F. Tsai, W.Y. Lin, Z.F. Hong, C.Y. Hsieh. EURASIP Journal on Advances in Signal Processing, **2011**(62):2–11 (2011). https://link.springer.com/article/10.1186/1687-6180-2011-62
27. D. Lavanya, D. K.U. Rani. IJCSE **2**(5):756–763 (2011). http://www.ijcse.com/docs/INDJCSE11-02-05-167.pdf