Graph Based Feature Selection Using Symmetrical Uncertainty in Microarray Dataset

Soodeh Bakhshandeh
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
soodeh.bakhshandeh@gmail.com
Reza Azmi*
Department of Computer Engineering, Alzahra University, Tehran, Iran
azmi@alzahra.ac.ir
Mohammad Teshnehlab
Department of Control Engineering, K. N. Toosi University. Tehran, Iran
teshnehlab@kntu.ac.ir

Received: 30/Apr/2019 Revised: 25/Sep/2019 Accepted: 04/Nov/2019

Abstract

Microarray data using small samples and thousands of genes provides a difficult challenge for researchers. Utilizing gene selection helps to select the most relevant genes from original dataset with the purpose of dimensionality reduction of microarray data as well as increasing the prediction performance. In this paper, a new gene selection method based on community detection technique and ranking the best genes, is proposed. In order to select the best genes, Symmetric Uncertainty calculates the similarity between two genes, and between gene and its class label. In the first phase, this leads to representation of search space in form of graph. In the second phase, the proposed graph is divided into several clusters, using community detection algorithm. Finally, after ranking the genes, the ones with maximum ranks are selected as the best genes. This approach is a supervised/unsupervised filter-based gene selection method, which not only minimizes the redundancy between genes, but also maximizes the relevance of genes and their class labels. Performance of the proposed method is compared with twelve well-known unsupervised/supervised gene selection approaches over twelve microarray datasets using four classifiers including SVM, DT, NB and k-NN. The results illustrate the advantages of the proposed approach.

Keywords: Gene selection; Microarray data; Filter method; Graph-based clustering; Feature Selection.

1. Introduction

In recent years, a new research path has been opened in Bioinformatics and machine learning field. This field contains monitoring thousands of gene expressions for detecting or classifying the specific type of tumor in DNA microarray datasets. Applying machine learning techniques to microarray data results extracting valuable information from dataset and building a model for classifying data into different categories based on the data classes. In training and testing phase, the researchers deal with small samples consisting of thousands of genes which may lead to "curse of dimensionality" [1]. In high dimensional data, many features are irrelevant and redundant which requires large storage space and consequently, have impact on performance and increasing the cost of learning process [2]. For improving the performance of learning model, the dimensionality reduction methods have been introduced in several papers. These methods reduce the cost and risk of over-fitting, but on the other hand they increase the ability of learning model for classifying the high-dimensionality data in different classes.

For reducing the dimensionality of features (genes), data could be transformed from the original space, with data dimension, into a new space with lower dimension, using feature (gene) extraction techniques. Another technique for reducing the dimensionality of features (genes) is feature (gene) selection. If there are n features (genes), the whole search space size will be2^n. Therefore, the time complexity of feature selection process is 2^n, which is NP hard problem [3]. In past years, some methods, such as feature selection method, have been introduced to find a near-optimal feature (gene) subset. Feature selection is a preprocessing step that identifies and removes irrelevant and redundant genes from the training data. In classification phase, the gene selection method leads to increasing the influence of training data.

The gene selection methods can be classified into filter, wrapper, embedded, and hybrid categories. The goal of filter, which is independent of any learning algorithm, is to reduce the data dimensionality based on the statistical properties of data. Univariate and multivariate strategies are used for evaluating the relevance of each gene to others in filter method. While univariate methods rank the genes individually and independent of other genes, the multivariate methods consider the correlation between genes. Therefore, because the univariate methods ignore the relation

* Corresponding Author

between genes, they are rather fast but with low accuracy compared to multivariate methods [4]. While Laplacian score [5], term variance [6], mutual information [7], and information gain [8] are popular univariate methods, the well-known multivariate methods are mRMR [9], FCBF [10], RRFS [11], UFSACO [12], RSM [13].

The wrapper method optimizes a predictor for selection process. The predictor evaluates the quality of the selected genes iteratively. Greedy and stochastic search strategies are two categories of wrapper methods. Sequential forward selection and sequential backward selection are two classical greedy search methods and ant colony optimization (ACO) [14], particle swarm optimization (PSO) [15] are two stochastic search methods. Greedy strategy searches based on single track and stochastic method uses the randomness nature of data. The performance of wrapper method is better comparing to filter method because of using learning model, however the computational cost is higher.

The embedded methods work on training a learning model for classification and using it for building the optimal subset of features. Since the embedded approach uses all the genes for training the classifier, the training phase is so time consuming. Kernel penalized SVM (KP-SVM) [16], First Order Inductive Learner (FOIL) [17] and SVM-RFE [18] are three examples of the embedded methods.

The last category of gene selection methods is hybrid methods. This method combines the advantages of filter and wrapper methods for improving the performance of the selected genes in classifying phase. Combining of SVM-RFE and mRMR [19], combining correlation-based feature selection (CFS) and Taguchi-genetic algorithm [20] and combining Fisher score with a GA and PSO [21], are some hybrid methods that are introduced during recent years.

In this paper, a filter-based gene selection method is introduced that uses class label and the correlation between the genes for removing the redundant and irrelevant genes. The proposed method works in three steps; graph representation, gene clustering and selecting the best features from each cluster. In the first step, the gene search space is represented as a graph, corresponding to the similarity between features and also the similarity between feature and its class label. Afterwards, in the second step, a clustering algorithm is applied to the constructed graph to divide the genes into several clusters. Finally, the best genes from each cluster are selected as the final gene subset. The proposed method is compared with other feature/gene selection methods (Mutual correlation (MC), fast clustering-based feature selection algorithm (FAST), Graph Clustering with Node Centrality for unsupervised feature selection (GCNC), Unified-Feature Association Map (U-FAM), Feature Selection method with Joint Maximal Information Entropy between features and class (FS-JMIE), a Correlation based Memetic Algorithm (MA-C), Dense Subgraph Finding with Feature Clustering (DSFFC), Distributed dCor-based FS (D^2CORFS), a ReliefF and ACO-based gene selection (RFACO-GS), A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm (FICA-IWSSr), Greedy Randomized Adaptive Search Procedure (GRASP), and Support Vector Machine Recursive Feature Elimination (FCSVM-REF)). The comparison is done in terms of classification accuracy, number of selected genes, feature reduction, parameter effects and execution time. The novelties of the proposed method lie in the following aspects:

1. Using Symmetric Uncertainty (SU) for representing the search space as a graph: In this paper, a new method is introduced for calculating the weights of graph based on the similarity between two genes and between each gene and its class label by using SU. Changing the value of the parameters in the first step of the proposed method, leads to supervised or unsupervised gene selection method. If class of dataset samples is available, supervised method is used, otherwise unsupervised method is applied. By using this selection method, the results are considerably improved.
2. Using a new ranking method for determining the important nodes in a graph: In this paper a new ranking method is used for identifying the selection of the influential genes in the third step. After applying the ranking method, a meta-heuristic method is used for selecting the best genes in the third step, resulting minimum redundancy between selected genes in the proposed method.

The rest of the paper is organized as follows: Section 2 contains a brief review on other related works. In Section 3, the materials and concepts, which are used in the proposed method, are explained in more detail. In Section 4, the proposed method is presented. In Section 5, in order to demonstrate the advantages of the proposed method, experimental results on some datasets using different classifiers are presented. Additionally, the reasons backing up the efficiency of the proposed method is described. At the end, in Section 6, the conclusion part is provided.

2. Related Works

As the information of redundant genes presents in other genes, these genes do not help the prediction result. The irrelevant genes do not contribute in increasing the predictive accuracy either. Therefore, the goal of gene selection is removing the redundant and irrelevant genes from gene subset. In past years, some gene/feature selection methods have been introduced considering these goals. Some of them have been successful in removing irrelevant features like [22-24] and some others on eliminating both irrelevant and redundant features ([25, 26]).

As mentioned in section 1, filter gene selection methods are successful in case of dealing with large

number of features. There have been also some graph-based feature selection methods, which work on the relation between different features.

In [27], a fast clustering based feature selection algorithm (FAST) for high dimensional data is proposed, which works in two main steps. In the first step, by using graph-theoretic clustering methods, features are divided into different clusters. In the second step, the relationship between all features and the target class is calculated and based on storage of the relation, some features are selected. In this paper, after eliminating irrelevant features, a minimum spanning tree is constructed, which after partitioning it, the representative features are selected.

In [28], an unsupervised feature selection is presented that works based on graph theory. First, a graph is constructed based on the dissimilarity of features. Then the densest subgraph by maximum average weight is identified. As the reduced subgraph contains the features with less average correlation, these features are the selected features. As in constructing the graph, feature dissimilarity is taken into account, so the feature relevancy does not influence the graph.

In [29], for eliminating both redundant and irrelevant features, the densest subgraph and feature clustering are combined. For removing the redundant features, the densest subgraph like [28] is obtained, and for eliminating the irrelevant features, a specific feature clustering is applied to the feature set.

In [30], ant colony optimization (ACO) with a new fitness function is used for a new unsupervised gene selection method that is called MGSACO. In this method, first a graph represents the search space, using similarity between genes. After that, ACO with new fitness function is used for gene selection.

In [31], a graph is constructed using Pearson correlation coefficient measure for representation the features relationship. Then Louvain community detection algorithm is used for clustering the features. In the last phase, a novel ACO-based search strategy is proposed for selecting a feature subset from the cluster group.

In [32], a new feature selection method called Probabilistic Attribute-Value for Class Distinction (Pavicd), is introduced for removing irrelevant and redundant genes. It works on the space of feature values instead of the features' space. It only requires an evaluation function, for estimating the prediction of class label by one or more features, and a threshold for analysis of relevance.

In [33], by combining Genetic Algorithm (GA) and Local Search (LS), a correlation based memetic framework is introduced. For tuning the population of GA, symmetrical uncertainty measure is used.

In [34], a joint maximal information entropy between features and their classes is used for feature selection process. For measuring the feature subset, a joint maximal information entropy is defined, and a binary particle swarm optimization searches the feature space for finding the best feature subset.

3. Materials and Methods

The proposed approach is based on graph-theoretic principles, clustering and ranking concepts which have been covered in this section.

3.1 Symmetric Uncertainty (SU)

Entropy is the uncertainty measure in the distribution of variable and is defined as:

$$H(X) = -\sum_i P(x_i) log_2(P(x_i)) \quad (1)$$

Where X and $P(X)$ are discrete variable and probability mass function of X, respectively and $H(X)$ indicates entropy of variable X. After observing values of another variable Y, the entropy of X is defined as:

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_j) log_2(P(x_i|y_j \quad (2)$$

Where $P(X)$ and $P(X|Y)$ denote the prior probabilities for all value of X and the posterior probabilities of X given the value of Y, respectively. Information gain between two variables X and Y reflects additional information about X provided by Y, by which the entropy of X decreases. Information gain is the decrease of the uncertainty of X after observing Y, and is defined as:

$$IG(X|Y) = H(X) - H(X|Y) \quad (3)$$

Where $H(X)$ and $H(X|Y)$ have been defined in equation 1 and 2. If $IG(X|Y) > IG(Z|Y)$, the correlation between features Y and Z is more than correlation between features Y and X. Information gain is a symmetrical measure and therefore for two features X and Y, the order of them does not affect the value of IG ($IG(X|Y) = IG(Y|X)$). As IG is symmetry, it can be used for measuring the correlation between two features.

For normalizing the value of IG, IG should be divided by feature entropies and is known as Symmetric Uncertainty (SU). In some researches, SU is used to evaluate the goodness of features for classification.

The symmetric uncertainty is defined as follows:

$$SU(X,Y) = \frac{2 \times IG(X|Y)}{H(X)+H(Y)} \quad (4)$$

SU value is in range [0, 1]. For two features X and Y, if the value of SU is 1, it indicates that knowing the value of each feature, the value of other feature is predictable. If the value of SU is 0, it indicates the two variables are independent.

For calculating SU in continues features, the discretization process should be done before [35]. In [27], F-Correlation and T-Relevance are defined as following:

F-Correlation: The correlation between any pair of features F_i and F_j ($(F_i, F_j \in F) \wedge (i \neq j)$) is called the F-Correlation of F_i and F_j and denoted by $SU(F_i, F_j)$.

T-Relevance: The relevance between the feature $F_i \in F$ and the target concept C is referred to as T-Relevance of F_i and C and denoted by $SU(F_i, C)$.

Following these definitions, the below items are realized:

1. The correlation between irrelevant feature and target concept is very weak.
2. For redundant features, the value of F-Correlation will be near 1.

3.2 Community Detection

In real system, the graph representation of nodes and vertices is not regular. Using clustering approaches, in the real systems, improves our perception of the relations between patterns. In past decades, some clustering approaches have been introduced for detecting clusters (communities) in complex patterns. k-means is one of the classical clustering approaches that is very sensitive to initialized parameters. To compensate for its weakness, the novel clustering approaches are focused on community detection ([36]). Community detection is used for grouping nodes, which are shared common or similar property to one community (also called cluster or module). Therefore, community detection field refers to finding groups of nodes that are more internally connected that externally. Based on [37], using community detection leads to information regarding the network structure, its functionality and its compact representation. Community detection concept is used in some fields such as social networks, recommendation systems, Ad-hoc networks and so on.

Louvain community detection method is one the popular community detection methods that uses [38]. This method is a heuristic method that works based on modularity function maximization. Modularity has been used to compare the partition quality in different methods, and is an objective function, which its optimization is computationally difficult.

Louvain method is used for finding high modularity partitions in a short period of time. The algorithm steps are intuitive and easy to implement. The computational complexity of the algorithm, for n nodes in the graph, is $O(n\log n)$, therefore for large scale networks with some nodes, the algorithm will be rather quick.

Iterative Louvain method is started with a weighted network as input. In the first step, each node of the network is assigned to a community (community No. has the same value of node No. in the beginning). Then, for each node, the gain modularity is calculated based on removing a node from its community to the other community. In the second phase, a new network, based on modularity, is generated. These two steps are repeated till the significant improvement of the network modularity is obtained. The below equation is used for calculating the gain in modularity by moving one node (i^{th} node) into a community C:

$$\Delta Q = \left[\left(\frac{\Sigma_{in}+k_{i,in}}{2m}\right) - \left(\frac{\Sigma_{tot}+k_i}{2m}\right)^2\right] - \left[\left(\frac{\Sigma_{in}}{2m}\right) - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (5)$$

Where $\sum_{in}$ is the summation of all the weights of the links inside the community C, where i is moving into, $\sum_{tot}$ is the sum of all the weights of the links to nodes in the community C, where i is moving into, k_i is the weighted degree of i, $k_{i,in}$ is the sum of the weights of the links between i and other nodes in the community that i is moving into, and m is the sum of all of the edge weights in the network.

3.3 Identifying influential nodes

In complex networks, node importance is a basic measure for characterizing and identifying the structure of them, which is also an open issue ([39]). Centrality is a common measure for ranking the nodes of graphs [40]. Different centrality measures have been proposed such as degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality [41]. Although these centrality measures are used in complex networks, however there are some disadvantages [42].

In [42], a new evaluation method is introduced for determining the important nodes based on Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) approach. In this method, Multiple Attribute Decision Making (MADM) for exploring how to identify important nodes is introduced. TOPSIS method chooses the alternatives that have the shortest distance from the positive ideal solution and the farthest distance from the negative-ideal solution, and it operates in four steps. In the first step, the network is constructed based on the connection of nodes. After that, different centrality values are calculated based on network topology. In the third step, based on Euclidean distance, the separation from the positive ideal alternative S_i^+ and the separation from the negative ideal alternative S_i^- of nodes is calculated as follow:

$$S_i^+ = \sqrt{\sum_{j=1}^{n} (v_j^+ - v_{ij})^2}, \quad i = 1,\dots,m; j = 1,\dots,m \quad (6)$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_j^- - v_{ij})^2}, \quad i = 1,\dots,m; j = 1,\dots,m \quad (7)$$

In the final step, the relative closeness to the ideal alternatives (C_i) is calculated and the alternatives with higher C_i are considered as important alternatives.

$$C_i = \frac{S_i^-}{S_i^- + S_i^+}, \quad i = 1,\dots,m \quad (8)$$

4. Proposed Method

In this paper, a novel gene selection method is introduced which can remove irrelevant and redundant genes in the selection process. The proposed method consists of three steps. In the following sections, these three steps are described in more detail.

4.1 Graph Representation

In all graph-based feature selection methods, the first step is to represent the search space as an undirected graph. In the constructed graph, each feature represents a

node in graph. The gene search space is mapped to a fully connected undirected weighted graph $G = \langle F, E, w_F \rangle$. In this graph, $F = \{F_1, F_2, \ldots, F_n\}$ denotes a set of original genes (features) and $E = \{(F_i, F_j); F_i, F_j \in F\}$ is the graph edge and w_{ij} contains the similarity between two features F_i and F_j, which are connected by edge (F_i, F_j).

So far, the methods expressed in different papers have been emphasizing on improving performance of graph-based gene selection algorithm. In this paper, on the other hand, a new measurement based on Symmetrical Uncertainly (SU) is proposed. As it was already explained in 3.1, in SU the similarity value between genes and between genes and their class labels are measured. According to SU, for two features, T-Relevance should be as great as possible, and F-Correlation should be as small as possible. In this case, w_{ij} is defined as follows:

$$w_{ij} = \begin{cases} \beta SU(F_i, C) + \gamma SU(F_j, C) - \alpha SU(F_i, F_j), \\ \alpha + \beta + \gamma = 1, \text{if} \quad i \neq j \\ 1, \quad \text{otherwise} \end{cases} \tag{9}$$

Where $SU(F_i, F_j)$ is F-Correlation between two genes (F_i and F_j) and $SU(F_i, C)$ and $SU(F_j, C)$ are T-Relevance values between feature F_i and F_j and target class C, respectively.

Based on [43], the measure that is sensitive to rotation is not desirable in many applications. As above formula, if $\beta \neq \gamma$, therefore the value of w_{ij} is not equal to w_{ji} ($w_{ij} \neq w_{ji}$). For this reason, γ should be equal to β ($\gamma = \beta$). Therefore, the equation 8 could be changed as follows:

$$w_{ij} = \begin{cases} \frac{(1-\alpha)}{2}(SU(F_i, C) + SU(F_j, C)) - \alpha SU(F_i, F_j), \\ 0 \leq \alpha \leq 1, \text{if} \quad i \neq j \\ 1, \quad \text{otherwise} \end{cases} \tag{10}$$

w_{ij} is in the range [-1,1]. For scaling the edge weight into the range [0,1], softmax scaling is used as follows [6]:

$$\widehat{w}_{ij} = \frac{1}{1+\exp(-\frac{w_{ij}-\bar{w}}{\sigma})} \tag{11}$$

Where w_{ij} is the edge weight between node F_i and F_j and, $\bar{w}$ and σ are the mean and variance of all edge weights in the graph, respectively. After applying this equation, $\widehat{w}_{ij}$ is the normalized edge weight that represents the similarity between two genes, as well as each gene and its class label on the two sides of the edge.

4.2 Gene Clustering

The clustering methods try to group the existing data into different clusters. Therefore, the similarity array is inputted to the proposed algorithm, and the output of clustering method is some clusters that each of them contains the seeds with maximum similarity. Over the past few years, some community detection methods are introduced, which could apply the clustering as well. As mentioned in 3.2, Louvain community detection algorithm, performance wise, is one of the fastest community detection algorithms that has been used in the past. Additionally, this method is reasonably easy to implement. Due to these advantages, the method is used in this paper in the gene clustering step. Therefore, in the second step, after constructing the graph, Louvain community detection algorithm is applied to the graph for grouping the nodes (genes) to different clusters based on the normalized weight of graph edges.

4.3 Selecting the Best Genes

After grouping the genes in different clusters, selecting the best genes in each cluster is a major step by removing irrelevant and redundant genes. Therefore, in this step, the main goal is removing the features that have no influence on the result.

According to subsection 3.3, the technique for order performance by similarity to ideal solution (TOPSIS) approach is introduced for determining the important nodes in a graph. In the proposed method, after constructing the graph in the first step and applying community detection in the second step, for selecting the influential nodes in each cluster, TOPSIS is used. TOPSIS method is very efficient and practical for evaluating the importance of nodes.

For using TOPSIS, a new ranking method is used to rank each gene. After ranking each gene, the best genes are selected and used as final gene subset. For selecting the best genes, two rank parameters are used:

1. Ranking each gene without considering its cluster (Total Rank Score - TRS) and
2. Ranking the genes with considering its cluster (Cluster Rank Score - CRS)

TRS is independent of the second phase and CRS is dependent to the result of the second phase.

For calculating the first rank value (TRS), TOPSIS runs over all genes, based on the graph that is constructed in the first step. By running TOPSIS over all genes, the rank of each gene is compared to the other genes. After that, the total rank score is calculated by dividing the distance between [0, 1] to equal intervals, depending on the number of genes. For example, assuming there are five genes F_1, F_2, F_3, F_4 and F_5 in the dataset. If after applying TOPSIS, the gene ranks are 4, 2, 5, 1 and 3, respectively, then the total rank score (TRS) of each gene is calculated as Table 1.

For calculating the second rank value (CRS), TOPSIS runs over each cluster. In this calculation, the subgraphs, which were constructed after applying community detection in the second phase, are used. Applying TOPSIS in each subgraph results the rank of each gene in comparison to the other genes, within each community. Thereafter, for each cluster, the rank of each gene is calculated independent of other clusters and, looking like

Table 1: Calculating Total Rank Score (TRS)

Gene	F_1	F_2	F_3	F_4	F_5
Gene rank	4	2	5	1	3
TRS	0.4	0.8	0.2	1	0.6

Algorithm 1 shows the pseudo-code of calculating CRS and TRS.

For selecting the best genes, two methods are used in this paper. The first method is selecting based on TRS and CRS (TC). The second method is based on TRS, CRS and Simulated Annealing (TCS).

1. Selecting the best genes based on only TRS and CRS (TC):

As it was described in 3.2, the features in one cluster have maximum similarity. Therefore, in this step, selecting genes from one cluster leads to selecting the genes with maximum redundancy. To reach such a target, a gene with maximum TRS value is selected. In this case, the selected gene would not have lower CRS in compare to other genes from the same cluster. If CRS of the new candidate gene is less than the CRS of other selected genes in its cluster, the new gene, depends on the comparison result of ΔTV_{norm} and θ. Term Variance (TV) is to represent valuable information; the larger value this parameter has, the more valuable information the gene contains. TV for gene F_i is defined as follows:

$$TV(F_i) = \frac{1}{|n|}\sum_{j=1}^{|n|} (F_{ij} - \bar{F}_i) \quad (12)$$

Where $|n|$ is the number of samples, F_{ij} shows the value of gene i for sample j and $\bar{F}_i$ is the average value of all samples for gene F_i.

ΔTV is the result of dividing the TV of picking a gene from its cluster by TV of not picking it at all. ΔTV for gene i in cluster C is calculated as:

$$\Delta TV(F_i) = \frac{\sum_k TV(F_k)}{\sum_l TV(F_l)} \quad (13)$$

Where $|m|$ is the number of selected genes in cluster C, $\sum_k TV(F_k)$ and $\sum_l TV(F_l)$ are the summation of TVs of the genes in the cluster, $k \in \{1 \dots |m|\}$ and $l \in \{1 \dots |m|\}$ with the condition of $k \neq i$.

By normalizing $\Delta TV(F_i)$ in range [0 1], $\Delta TV_{norm}(F_i)$ is obtained.

For selecting the gene with lower CRS compared to another CRS in its cluster, $\Delta TV_{norm}(F_i)$ is calculated for the gene and if $\Delta TV_{norm}(F_i)$ is greater than θ, F_i will be selected.

Algorithm 2 shows the pseudo-code of using TRS and CRS (TC) to select the final subset.

2. Selecting the best genes based on TRS, CRS and Simulated Annealing (TCS):

Simulated Annealing (SA) is a probabilistic non-greedy algorithm that explores search space of a problem by annealing from a high to a low temperature state [44]. In this algorithm, moving to better state is accepted in any case but moving to the worse state is accepted only with a variable probability. This probability is high at the beginning, but is getting decreased along with the temperature, and thus the algorithm becomes greedier. If a solid heated past melting point and then cooled down, the solid's structural properties will vary depending on the rate of cooling. By cooling down the liquid slow enough, large crystals will be formed. On the other hand, if the cooling is done quickly (quenched) the crystals will contain defects and imperfections. The algorithm terminates when the final temperature reaches, or sufficient number of consecutive moves have been rejected.

Algorithm 1: CRS-TRS-Calculating Algorithm

input:
- F: $p \times n$ matrix, p patterns of n features;
- α: Relevance threshold.

output:
- CRS: Set of CRS.
- TRS: Set of TRS.

for $i = 1$ to n **do**
  for $j = 1$ to n **do**
    $w_{ij} \leftarrow \frac{(1-\alpha)}{2}\left(SU(F_i, C) + SU(F_j, C)\right) - \alpha SU(F_i, F_j)$
  end for
end for
$\hat{w} \leftarrow$ Softmax-scaling(w)
$\{cluster_1, cluster_2, \dots, cluster_c\} \leftarrow$ Louvian($\hat{w}$)
for $i = 1$ to n **do**
  $TRS_i \leftarrow$ Scale(TOPSIS(F_i))
end for
for $k = 1$ to C **do**
  for $j = 1$ to nC **do**
    $CRS_{kj} \leftarrow$ Scale(TOPSIS(F_{kj}))
  end for
end for

Algorithm 2: Selecting the best genes using TRS and CRS - TC

input:
- CRS: Set of CRS;
- TRS: Set of TRS.

output:
- S_{OPT}: Set of final selected features.

$TRS_S \leftarrow$ Sort-Descending(TRS)
for $i = 1$ to n **do**
  $temp \leftarrow TRS_{S_i}$
  $k \leftarrow$ Check-Cluster($temp$)
  if CRS($temp$) > CRS(all features in cluster k) **then**
    $S_{OPT} \leftarrow S_{OPT} \cup temp$
  else
    if $\Delta TV_{norm}(temp) > \theta$ **then**
      $S_{OPT} \leftarrow S_{OPT} \cup temp$
    end if
  end if
end for

In SA, at first an initial solution is randomly selected, and it is assumed to be the optimal solution. Subsequently, the cost of the initial solution is computed using the $Cost$ function. While temperature T does not satisfy the termination condition, a neighboring solution of the current optimal solution is selected, and its cost is calculated. If the cost of the newly selected neighboring solution is less than or equal to current optimal solution, the current optimal solution is replaced with a newly selected neighbor solution. If the cost of the neighboring solution is greater than the current optimal solution, a random value q is chosen in the range of [0, 1]. In this case, the replacement of the optimal solution is permitted only if a random value q is less than $e^{-\frac{Cost(v_n)-Cost(v_b)}{T}}$.

After temperature T is reduced based on $g(T)$, the same process is continued until T satisfies the termination condition.

Definition of $Cost$ function is the main step of SA. For this purpose, Sum CRS($SCRS$) is used as $Cost$ function in this paper , which is calculated as follows:

$$SCRS = \sum_{i=1}^{c} ICRS_i \quad (14)$$

Where

$$ICRS = \frac{\sum_{j=1}^{m} CRS}{m} \quad (15)$$

Where c and m are the number of cluster and the number of genes in each cluster, respectively and $ICRS$ is summation of CRSs in each cluster.

The goal of this algorithm is reaching maximum value on $SCRS$ applied to the sorted TRS.

For using SA, the inputs are FB and T_0. FB is an array of 0 and 1. fb_i is 0 when the gene is not selected and 1 when gene is selected. For calculating FB, TRSs are sorted in descending manner, and fb_i represents the index of each variable in FB, which is calculated based on the sorted TRSs.

T is the initial temperature. This initial temperature should be should be large enough to allow sufficient transitions to be accepted. The temperature reduction function is defined as a simple iterative function which is the product of T multiplied by a constant r ($g(T) = r \times T$). In some references, this value is in range [0.5 0.99].

Algorithm 3 shows the pseudo-code of using SA, TRS and CRS (TCS) for selecting the final subset.

In the last step, for selecting the final subset, two mentioned methods could be used. When using only TRS and CRS in the selection process, θ parameter should be set, and when using SA, then it is added to the selection process. The main advantages of using SA are the flexibility and ability to approach global optimality in this step. However, the selection process is slower comparing to when using only CRS and TRS. On the other hand, selecting the best value for θ in the first method, makes it very sensitive and could affect the results.

Fig. 1 shows the schematic diagram of the proposed method.

5. Experimental Results and Discussion

In this section, the performance of the proposed method is compared to twelve other frequently used gene/feature selection methods upon twelve well-known microarray datasets.

For comparing the proposed method with other methods, twelve different DNA microarray datasets are selected. The selected datasets cover a wide spectrum of cancer types and are already used in some other papers. Datasets include nine binary classification types (Blastom, Colon, Gastric, Central Nervous System (CNS), Leukemia 2-class, Diffuse Large-B-Cell Lymphoma (DLBCL), Prostate Tumor, Ovarian, and Brain) and three multi-class (Lymphoma, SRBCT, and Lung Cancer) that are available at [45-47]. Table 2 displays the brief description of used datasets.

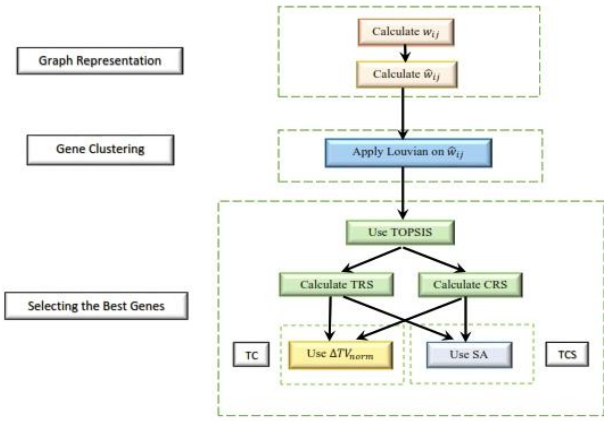

Fig. 1: Illustrating the proposed method on a schematic diagram

Algorithm 3: Selecting the best genes using TRS, CRS and SA – TCS

input:
- $FB = \langle fb_1 \dots fb_n \rangle$ where $fb_i \in \{0,1\}$;
- T_0: The initial temperature;
- r: temperature reduction constant.

output:
- v_b: Combination of genes.

1: Generate an initial solution, v_i
2: $v_b \leftarrow v_i$
3: Calculate the cost of initial solution, $Cost(v_b)$
4: **while** $(T > T_{min})$ or $(iteration < itr_{max})$ **do**
5: Randomly select a neighbor solution, v_n, of v_b which have one bit different from v_b
6: $\Delta Cost = Cost(v_b) - Cost(v_n)$
7: **if** $(\Delta Cost \leq 0)$ **then**
8: $v_b \leftarrow v_n$
9: **else**
10: Generate a random number q uniformly in the range [0,1]
11: **if** $(q < e^{-\frac{\Delta Cost}{T}})$ **then**
12: $v_b \leftarrow v_n$
13: **end if**
14: **end if**
15: $T \leftarrow g(T)$
16: **end while**

Table 2: Description of the datasets used in the experiments

Dataset	Sample (Pattern) NO	Feature (Gene) NO	Class NO
Blastom	23	1,465	2
Colon	62	2,000	2
Gastric	30	4,522	2
CNS	60	7,129	2
ALL-AML	72	7,129	2
DLBCL	77	7,129	2
Prostate Tumor	102	10,509	2
Ovarian	253	15,154	2
Brain	21	12,625	2
Lymphoma	62	4,026	3
SRBT	83	2,308	4
Lung Cancer	203	12,600	5

The other gene/feature selection methods that are used for comparing the results are Mutual correlation (MC) [48], FAST [27], Graph Clustering with Node Centrality for unsupervised feature selection (GCNC) [49], Unified-Feature Association Map (U-FAM) [50], Feature Selection

method with Joint Maximal Information Entropy between features and class (FS-JMIE) [34], A Correlation based Memetic Algorithm (MA-C) [33], Dense Subgraph Finding with Feature Clustering (DSFFC) [29], Distributed dCor-based FS (D^2CORFS) [51], a ReliefF and ACO-based gene selection (RFACO-GS) [52], A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm (FICA-IWSSr) [53], Greedy Randomized Adaptive Search Procedure (GRASP) [54], and Support Vector Machine Recursive Feature Elimination (FCSVM-REF) [55].

Since the proposed method is a filter-based gene selection and is independent of using any classifier in the gene selection process, for evaluating the performance, the different types of classifiers are used in experiment phase. Different classifiers that are considered in this paper are Support Vector Machine (SVM) [56], Decision Tree (DT) [57], Naïve Bayes (NB) [6] and k-Nearest Neighbor (kNN) [58].

For obtaining the accurate and stable experiment results, the average classification accuracy rate (%), over ten independent runs, is considered. In each run, for training/testing the classifier, 70%/30% of the dataset is used in train/test phase. In the training phase, the final feature subsets are selected using training sets and the learning model is evaluated in the test phase, using testing sets. All experiment results are obtained on a machine with 2.70 GHz CPU and 4GB of RAM.

5.1 Experimental Results

To demonstrate the performance of the proposed method, in this section, the experimental results have been presented in terms of classification accuracy term, feature reduction, and the parameters' effect and the execution time. In all results, α is 0.4 when calculating the weights of the graph.

The last step in the proposed algorithm is where selecting the best genes by using two methods; TC and TCS. Where ever TC is used, θ is set to 0.7, and where TCS is applied initial temperature (T_0), temperature reduction constant (r) and termination condition of SA are 100,000, 0.95 and $T < 0.01$, respectively.

The accuracy in the selection process of FCSVM-REF is reported by SVM, since that is the mechanism used in the selection process.

5.1.1 Classification accuracy

To demonstrate the advantages of the proposed methods (TC and TCS), the average classification accuracy of it has been compared with other well-known gene selection methods. Table 3 shows the comparison results between the proposed method with twelve gene selection methods. The results are averaged over ten runs of different gene selection methods, using SVM, DT, NB and kNN classifiers. For each classifier in Table 3, the rank of each algorithm in comparison to other methods is presented in parentheses.

As it is indicated in the results, in average, the proposed method using TC achieves the second rank with 87.33%, 87.29%, and 86.03%, using SVM, DT, and kNN, respectively. In average, the proposed method using TCS, obtains the best rank with 89.08%, 87.5%, and 87.14%, using SVM, DT, and kNN classifiers, respectively.

5.1.2 Feature reduction

In Table 4, feature reduction of the proposed method (TC and TCS) has been compared with other feature selection methods over different datasets. As it is presented, that the proposed method (TC or TCS) feature reductions are 83.81%, 84.75%, 83.19%, 84.78%, 81.90%, 75.65%, 89.25%, 83.36%, 83.74%, 94.97%, 85.23%, 94.18% and 85.27% over Blastom, Colon, Gastric, CNS, Leukemia, DLBCL, Prostate Tumor, Ovarian, Lymphoma, Brain, SRBT, Lung Cancer and in average. The average reduction for the proposed method is the highest one with 83.80% for TC and 85.27% for TCS.

5.1.3 Parameters

In the proposed method, α is a user-specified parameter that is used for calculating the graph weights in equation 10. If $\alpha = 0$, the equation10 will be changed to:

$$w_{ij} = \frac{1}{2} \times (SU(F_i, C) + SU(F_j, C)) \quad (16)$$

In this situation, only T-Relevance (the correlation between each gene and its target class) affects the graph weights, and so, the method will be will be a supervised graph-based gene selection. While $\alpha = 1$, the graph weights are calculated as follow:

$$w_{ij} = -SU(F_i, F_j), \quad (17)$$

Therefore, only F-Correlation between two genes determines the weight of the edge between two nodes and the effect of its target class will be ignored. This leads to changing the proposed method to an unsupervised graph-based gene selection.

In other situation, the value of $\frac{1-\alpha}{2}$ depends on the value of α, and so if the value of α is 0.3, the value of $\frac{1-\alpha}{2}$ is 0.35. Figure 2 and Figure 3 show the average classification accuracy of the proposed method using TCS over ten runs with different α values using SVM and DT classifiers. In these figures, for different values of α (which cause change on the $\frac{1-\alpha}{2}$ value), the average classification accuracy is calculated and represented. For example, using SVM, and $\alpha = 0.4$, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and average, the classification accuracy is 88.53%, 90.54%, 89.67%, 80.70%, 77.73%, 82.31% and 84.91%, respectively. Using DT and $\alpha = 0.4$, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and average, the classification accuracy is 78.48%, 88.92%, 83.45%, 80.67%, 82.88%, 88.27% and 83.78%, respectively.

In Fig. 2, the maximum classification accuracy, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer andin average is 88.53% ($\alpha = 0.4$), 90.54% ($\alpha = 0.4$), 89.67% ($\alpha = 0.4$), 81.61% ($\alpha = 0.5$), 78.45% ($\alpha = 0.3$), 84.48% ($\alpha = 0.5$) and 84.91% ($\alpha = 0.4$), respectively.

Table 3: Comparing average classification accuracy of the proposed method (TC and TCS) and twelve gene selection methods over ten runes using different classifiers. The first best result is shown in bold face. The rank of each algorithm is shown by the number in the parentheses.

Dataset	Classifier	TC	TCS	MC	FAST	GCNC	U-FAM	FS-JMIE	MA-C	DSFFC	D^{2}CORFS	RFACO-GS	FICA-IWSSr	GRASP	FCSVM-REF
Blastom	SVM	91.52% (2)	**93.57% (1)**	66.77% (14)	73.42% (13)	78.93% (11)	80.53% (10)	76.29% (12)	81.59% (8)	82.93% (7)	81.25% (9)	84.79% (6)	86.19% (4)	89.52% (3)	84.92% (5)
	DT	**93.75% (1)**	92.84% (2)	65.97% (13)	71.87% (12)	79.43% (10)	81.88% (9)	77.47% (11)	82.33% (8)	83.06% (7)	85.97% (5)	86.31% (4)	84.65% (6)	89.47% (3)	—
	NB	**92.49% (1)**	92.19% (2)	69.49% (13)	71.79% (12)	78.67% (10)	83.79% (8)	78.36% (11)	83.91% (7)	84.27% (6)	83.46% (9)	87.40% (4)	86.16% (5)	88.79% (3)	—
	kNN	**94.11% (1)**	93.55% (2)	67.82% (13)	72.37% (12)	77.26% (10)	84.21% (5)	77.18% (11)	82.77% (7)	83.50% (6)	82.07% (9)	82.42% (8)	87.46% (4)	90.14% (3)	—
Colon	SVM	85.46% (4)	88.53% (2)	69.65% (14)	**93.50% (1)**	81.45% (11)	83.96% (9)	75.22% (13)	84.79% (6)	81.18% (12)	85.12% (5)	84.21% (8)	82.49% (10)	85.96% (3)	84.56% (7)
	DT	79.47% (9)	79.48% (8)	78.32% (11)	81.23% (7)	**85.87% (1)**	79.31% (10)	72.08% (13)	84.37% (4)	78.08% (12)	81.72% (6)	85.27% (2)	81.94% (5)	84.91% (3)	—
	NB	82.80% (5)	82.89% (4)	69.45% (13)	82.98% (3)	80.56% (9)	75.36% (11)	79.43% (10)	81.27% (7)	74.97% (12)	80.79% (8)	81.97% (6)	**93.60% (1)**	83.87% (2)	—
	kNN	81.49% (6)	83.37% (2)	68.91% (13)	80.03% (10)	81.85% (4)	78.01% (12)	81.69% (5)	80.73% (8)	78.91% (11)	80.89% (7)	80.49% (9)	82.72% (3)	**83.74% (1)**	—
Gastric	SVM	91.72% (2)	**95.19% (1)**	70.93% (13)	74.88% (12)	70.69% (14)	79.17% (11)	81.89% (7)	82.46% (6)	79.67% (10)	81.31% (8)	80.79% (9)	89.46% (4)	89.86% (3)	82.66% (5)
	DT	**95.49% (1)**	94.55% (2)	69.48% (12)	76.98% (11)	66.22% (13)	79.07% (10)	82.83% (5)	82.79% (6)	80.28% (9)	82.47% (7)	80.59% (8)	88.27% (3)	84.59% (4)	—
	NB	94.33% (2)	**94.41% (1)**	70.89% (12)	75.47% (11)	67.11% (13)	80.45% (10)	81.57% (7)	82.78% (5)	81.45% (8)	81.37% (9)	81.97% (6)	85.63% (4)	86.77% (3)	—
	kNN	94.78% (2)	**95.16% (1)**	70.91% (12)	76.77% (11)	64.72% (13)	81.70% (6)	81.06% (9)	81.25% (8)	81.38% (7)	80.29% (10)	81.99% (5)	85.95% (4)	89.56% (3)	—
CNS	SVM	94.81% (2)	**95.94% (1)**	72.06% (14)	73.92% (13)	81.33% (10)	78.37% (12)	88.34% (6)	85.29% (9)	89.21% (4)	80.11% (11)	90.82% (3)	86.79% (8)	88.38% (5)	88.27% (7)
	DT	93.72% (2)	**94.27% (1)**	70.73% (13)	74.69% (12)	84.55% (9)	76.71% (11)	87.19% (8)	88.37% (7)	90.36% (4)	79.54% (10)	91.75% (3)	88.56% (6)	89.46% (5)	—
	NB	91.62% (2)	**91.69% (1)**	67.20% (13)	78.21% (12)	82.67% (9)	78.84% (11)	85.28% (8)	89.18% (5)	91.17% (3)	81.72% (10)	90.79% (4)	88.71% (6)	88.19% (7)	—
	kNN	93.68% (2)	92.49% (4)	66.49% (13)	77.82% (12)	83.19% (9)	77.99% (11)	86.33% (7)	89.28% (6)	92.93% (3)	80.77% (10)	**94.59% (1)**	85.37% (8)	89.45% (5)	—
Leukemia	SVM	87.30% (7)	90.54% (3)	58.67% (14)	88.90% (5)	73.23% (13)	81.76% (10)	78.19% (12)	82.38% (9)	79.35% (11)	**91.72% (1)**	86.19% (8)	91.10% (2)	88.37% (6)	89.84% (4)
	DT	88.06% (5)	**88.92% (1)**	75.9% (12)	79.92% (8)	74.43% (13)	78.48% (11)	79.44% (10)	88.71% (2)	79.47% (9)	88.21% (4)	87.43% (6)	83.90% (7)	88.24% (3)	—
	NB	87.94% (3)	87.83% (5)	65.32% (13)	89.99% (2)	87.94% (3)	76.03% (12)	81.85% (11)	83.04% (9)	81.94% (10)	85.43% (8)	86.72% (7)	**93.06% (1)**	87.51% (6)	—
	kNN	83.20% (7)	85.92% (3)	61.82% (13)	75.29% (12)	84.67% (5)	82.76% (8)	78.49% (11)	83.32% (6)	80.40% (10)	82.07% (9)	84.71% (4)	**89.19% (1)**	88.41% (2)	—
DLBCL	SVM	92.85% (2)	**92.98% (1)**	75.11% (14)	80.62% (10)	78.79% (12)	80.66% (9)	78.76% (13)	83.12% (8)	83.19% (7)	80.29% (11)	85.12% (5)	90.11% (3)	87.42% (4)	83.57% (6)
	DT	**93.29% (1)**	93.12% (2)	76.33% (13)	78.04% (11)	79.62% (9)	79.51% (10)	77.38% (12)	84.44% (6)	83.10% (8)	85.41% (5)	84.37% (7)	91.16% (3)	86.53% (4)	—
	NB	91.48% (3)	92.49% (2)	79.44% (9)	76.39% (13)	78.89% (11)	78.48% (12)	79.04% (10)	85.29% (4)	84.16% (6)	80.31% (8)	84.98% (5)	**99.12% (1)**	80.85% (7)	—
	kNN	90.11% (2)	**92.67% (1)**	80.59% (9)	75.09% (13)	77.39% (12)	78.25% (11)	79.26% (10)	83.10% (7)	84.55% (6)	82.99% (8)	85.56% (5)	90.01% (3)	88.27% (4)	—
Prostate Tumor	SVM	89.94% (2)	**90.67% (1)**	67.89% (14)	79.89% (10)	76.90% (12)	82.13% (6)	76.48% (13)	81.07% (9)	78.64% (11)	81.83% (8)	82.10% (7)	86.16% (4)	88.14% (3)	84.51% (5)
	DT	84.25% (3)	83.45% (4)	71.27% (12)	70.75% (13)	**86.67% (1)**	84.56% (2)	74.19% (11)	82.79% (6)	76.27% (10)	80.52% (9)	81.04% (8)	82.16% (7)	83.09% (5)	—
	NB	79.91% (6)	81.48% (5)	75.92% (11)	68.34% (13)	81.54% (4)	81.97% (3)	74.42% (12)	79.80% (7)	78.45% (10)	79.07% (8)	82.94% (2)	**92.43% (1)**	78.68% (9)	—
	kNN	79.21% (6)	80.35% (4)	73.42% (12)	67.06% (13)	79.21% (6)	79.58% (5)	79.11% (8)	78.33% (10)	74.17% (11)	**81.66% (1)**	78.47% (9)	81.46% (2)	80.38% (3)	—
Ovarian	SVM	79.82% (7)	82.19% (4)	75.94% (12)	78.57% (9)	**84.95% (1)**	76.15% (11)	78.35% (10)	73.18% (14)	75.88% (13)	82.46% (3)	79.10% (8)	80.18% (6)	81.61% (5)	83.17% (2)
	DT	82.49% (2)	**82.95% (1)**	76.34% (11)	76.88% (9)	81.42% (4)	79.57% (5)	77.26% (8)	74.15% (13)	76.87% (10)	81.82% (3)	75.92% (12)	79.37% (6)	78.28% (7)	—
	NB	85.61% (2)	**86.27% (1)**	73.49% (13)	75.29% (11)	82.71% (3)	82.71% (3)	76.46% (9)	75.38% (10)	76.65% (8)	80.37% (5)	74.39% (12)	79.16% (6)	78.01% (7)	—
	kNN	83.09% (3)	**85.46% (1)**	74.91% (13)	77.91% (8)	81.69% (5)	83.58% (2)	77.17% (10)	76.08% (11)	77.18% (9)	82.49% (4)	75.02% (12)	78.39% (7)	79.31% (6)	—
Lymphoma	SVM	91.72% (2)	**92.66% (1)**	76.53% (11)	76.56% (10)	84.95% (4)	80.89% (7)	76.92% (9)	75.94% (12)	71.03% (14)	83.94% (5)	72.84% (13)	90.93% (3)	80.93% (6)	79.93% (8)
	DT	92.84% (2)	**93.49% (1)**	75.37% (10)	75.85% (9)	83.88% (5)	81.43% (7)	71.04% (13)	76.10% (8)	74.92% (11)	85.67% (4)	73.95% (12)	91.77% (3)	81.71% (6)	—
	NB	90.23% (3)	91.59% (2)	75.11% (11)	74.48% (12)	84.28% (4)	79.67% (7)	72.93% (13)	77.29% (8)	75.72% (9)	83.49% (5)	75.28% (10)	**94.10% (1)**	80.21% (6)	—
	kNN	88.76% (2)	**89.77% (1)**	73.94% (12)	76.01% (10)	85.33% (4)	79.38% (7)	70.14% (13)	78.42% (8)	76.65% (9)	85.22% (5)	74.04% (11)	86.26% (3)	83.10% (6)	—
Brain	SVM	**80.86% (1)**	80.70% (2)	73.90% (11)	65.66% (14)	69.7% (13)	79.43% (6)	78.49% (7)	80.11% (4)	79.72% (5)	74.97% (9)	72.91% (12)	73.92% (10)	80.55% (3)	78.18% (8)
	DT	79.99% (5)	80.67% (3)	67.89% (13)	79.89% (6)	80.60% (4)	81.03% (2)	79.45% (8)	79.48% (7)	78.04% (9)	73.59% (11)	74.74% (10)	72.93% (12)	**81.62% (1)**	—
	NB	**82.04% (1)**	79.95% (3)	68.41% (13)	77.41% (7)	79.51% (4)	78.72% (5)	74.07% (12)	78.66% (6)	76.51% (9)	76.94% (8)	75.82% (10)	74.61% (11)	81.69% (2)	—
	kNN	78.02% (5)	79.63% (4)	67.20% (13)	73.65% (10)	80.61% (2)	80.38% (3)	73.97% (9)	74.18% (7)	73.99% (8)	75.85% (6)	73.18% (11)	72.91% (12)	**83.15% (1)**	—
SRBT	SVM	80.94% (5)	83.73% (2)	69.28% (14)	69.71% (13)	79.49% (7)	78.65% (8)	76.74% (9)	81.78% (3)	71.46% (11)	80.45% (6)	70.46% (12)	73.81% (10)	**84.18% (1)**	81.04% (4)
	DT	**83.77% (1)**	82.88% (2)	65.99% (13)	75.39% (8)	76.52% (6)	76.22% (7)	73.10% (11)	80.48% (5)	74.05% (10)	82.46% (3)	71.38% (12)	74.27% (9)	81.38% (4)	—
	NB	80.35% (5)	81.49% (3)	73.24% (12)	73.80% (11)	**83.02% (1)**	79.47% (7)	75.49% (9)	80.75% (4)	78.66% (8)	79.49% (6)	72.88% (13)	73.92% (10)	81.72% (2)	—
	kNN	79.94% (4)	**81.54% (1)**	69.32% (13)	72.48% (12)	80.31% (3)	78.22% (7)	74.09% (10)	81.27% (2)	74.49% (9)	79.55% (5)	73.93% (11)	76.83% (8)	78.28% (6)	—
Lung Cancer	SVM	81.06% (5)	82.31% (3)	70.40% (14)	76.35% (10)	83.89% (2)	74.19% (11)	80.79% (7)	81.00% (6)	74.03% (12)	80.46% (8)	71.82% (13)	82.19% (4)	**91.38% (1)**	79.93% (9)
	DT	80.31% (5)	83.34% (3)	71.57% (13)	74.65% (10)	80.30% (6)	74.54% (11)	79.61% (8)	80.17% (7)	76.88% (9)	81.42% (4)	72.01% (12)	85.27% (2)	**89.21% (1)**	—
	NB	83.45% (3)	82.07% (4)	71.52% (12)	75.71% (10)	81.67% (5)	80.29% (7)	74.49% (11)	78.41% (8)	75.92% (9)	80.97% (6)	10.27% (13)	**98.91% (1)**	97.24% (2)	—
	kNN	85.93% (3)	85.76% (4)	60.38% (13)	75.01% (12)	79.03% (7)	77.91% (9)	75.07% (11)	79.76% (6)	78.43% (8)	81.76% (5)	77.16% (10)	**96.27% (1)**	89.52% (2)	—
Average	SVM	87.33% (2)	**89.08% (1)**	70.59% (14)	77.67% (13)	78.69% (12)	79.66% (9)	79.18% (10)	81.06% (7)	78.86% (11)	81.99% (6)	80.10% (8)	84.44% (4)	86.36% (3)	83.24% (5)
	DT	87.29% (2)	**87.50% (1)**	72.10% (13)	76.35% (12)	79.96% (8)	79.36% (9)	78.82% (11)	82.02% (6)	79.28% (10)	82.40% (5)	80.41% (7)	83.68% (4)	84.87% (3)	—
	NB	86.85% (3)	87.03% (2)	71.62% (13)	76.65% (11)	80.71% (7)	79.65% (9)	78.82% (10)	81.31% (5)	79.99% (8)	81.12% (6)	75.45% (12)	**88.28% (1)**	84.46% (4)	—
	kNN	86.03% (2)	**87.14% (1)**	69.64% (13)	74.96% (12)	79.6% (10)	80.16% (7)	78.33% (11)	80.71% (6)	79.72% (9)	81.30% (5)	80.13% (8)	84.40% (4)	85.27% (3)	—

* Corresponding Author

Table 4: Comparison of feature reduction of different feature selection methods over different datasets.

Dataset	Proposed Method		MC	FAST	GCNC	U-FAM	MA-C	DSFFC
	TC	TCS						
Blastom	83.8%	75.4%	63.9%	71.4%	69.3%	72.0%	76.9%	70.4%
Colon	84.7%	82.4%	71.3%	87.5%	81.2%	59.2%	79.1%	73.3%
Gastric	83.1%	83.1%	65.4%	80.4%	75.9%	70.4%	74.0%	75.9%
CNS	81.6%	84.7%	72.0%	73.2%	70.9%	70.8%	72.2%	75.4%
Leukemia	79.9%	81.9%	63.9%	73.0%	80.3%	76.4%	85.4%	74.3%
DLBCL	70.2%	75.6%	61.4%	60.8%	66.7%	70.4%	75.2%	58.2%
Prostate Tumor	88.1%	89.2%	76.9%	97.4%	78.9%	81.6%	79.5%	57.1%
Ovarian	81.5%	83.3%	78.4%	80.6%	76.9%	90.1%	79.4%	73.4%
Lymphoma	80.1%	83.7%	68.9%	78.9%	80.4%	73.8%	71.7%	79.1%
Brain	92.7%	94.9%	81.9%	74.1%	75.3%	80.8%	80.4%	71.3%
SRBT	85.1%	85.2%	83.5%	78.3%	87.1%	87.2%	70.4%	59.9%
Lung Cancer	94.1%	93.4%	54.3%	92.4%	83.9%	86.6%	71.0%	80.2%
Average	83.8%	85.2%	70.7%	79.7%	78.0%	77.0%	76.2%	73.5%

In Fig. 3, the maximum classification accuracy, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and in average is 79.35% ($\alpha = 0.3$), 88.92% ($\alpha = 0.4$), 84.28% ($\alpha = 0.3$), 80.67% ($\alpha = 0.4$), 83.92% ($\alpha = 0.6$), 88.27% ($\alpha = 0.4$) and 83.78% ($\alpha = 0.4$), respectively.

θ is other parameter that is used in the proposed method for selecting the best genes in the final step (Algorithm 2). Selecting the best value for θ is very sensitive and could affect the results. Fig. 4 shows the average classification accuracy of the proposed method using TC over ten runs with different θ values using SVM classifier. Therefore, the maximum classification accuracy, for Colon, Leukemia, Prostate Tumor, Brain, SRBT, Lung Cancer and in average is 83.46% ($\theta = 0.7$), 87.30% ($\theta = 0.7$), 89.94% ($\theta = 0.7$), 81.97% ($\theta = 0.8$), 81.74% ($\theta = 0.9$), 81.06% ($\theta = 0.7$) and 83.59% ($\theta = 0.7$), respectively.

5.1.4 Execution time

To analyze the complexity of the proposed method, the computational complexity should be estimated. The proposed method consists of three steps: graph representation, gene clustering and selecting the best genes from each cluster. Therefore, computational complexity of the proposed method could be estimated as follows:

Step 1: For n genes and p samples, calculating the graph edges for all pairs of genes has a computational cost of $O(n^2 p)$.

Step 2: For clustering the genes, Louvain community detection algorithm is used with computational cost of $O(n \log n)$.

Step 3: based on [59], the time complexity of TOPSIS could be calculated as: $O(n^2)$ for the first step, $O(n)$ for the second step, $O(n)$ for the third step and $O(1)$ for the final step. This means the computational cost of TOPSIS is $O(n^2 + n + n + 1) = O(n^2)$.

In the proposed method, the TOPSIS is used for ranking the genes without considering their clusters with $O(n^2)$ computational cost, and considering their clusters

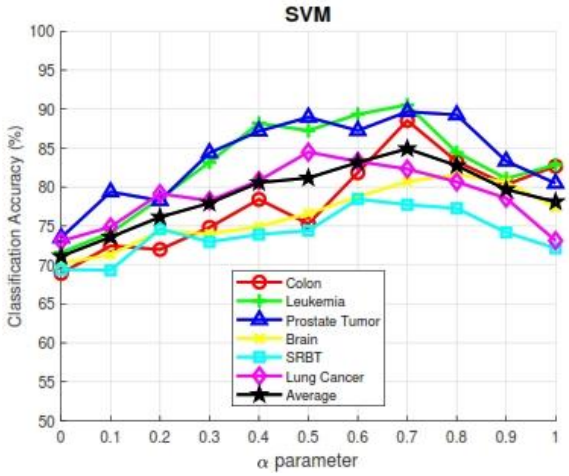

Fig. 2: Average classification accuracy over ten runes with different α values using SVM classifier.

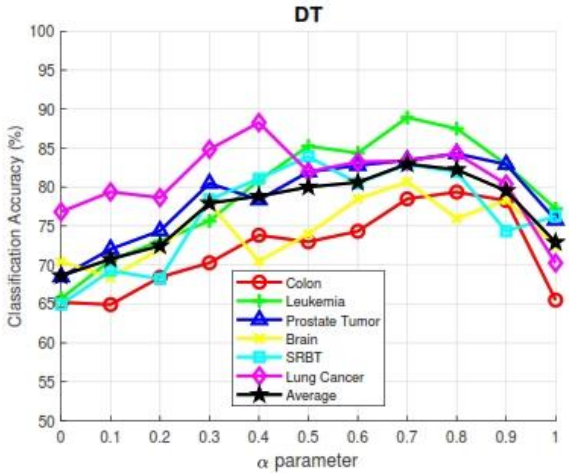

Fig. 3: Average classification accuracy over ten runes with different α values using DT classifier.

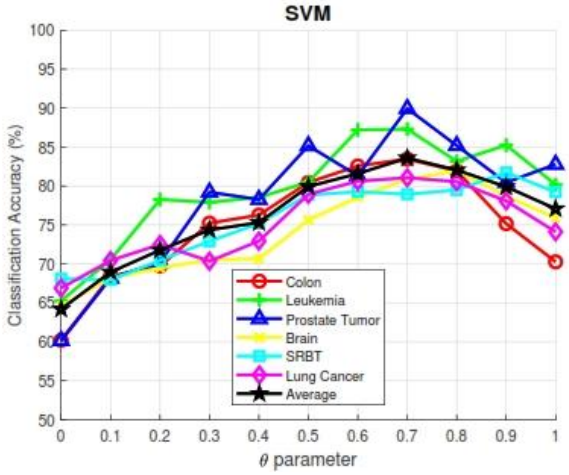

Fig. 4: Average classification accuracy over ten runes with different θ values using SVM classifier.

In the proposed method, the TOPSIS is used for ranking the genes without considering their clusters with

* Corresponding Author

$O(n^2)$ computational cost, and considering their clusters with $O(mk)$ computational cost, where m ($m \leq n$) is the number of genes in each cluster and k is the number of clusters ($O(mk) \leq O(n^2)$).

After applying TOPSIS, for selecting the best genes, two different methods are used. For the first method (Algorithm 2), all genes will be traced based on TRS and CRS values. Therefore, the computational cost of this method is $O(n)$.

For the second method (Algorithm 3), because of using the selected temperature reduction function and the stop condition of SA, the computational cost of this method is $O(\log\frac{T}{T_{min}}) >> O(\log n)$.

Consequently, the total computational complexity of the proposed method is $O(n^2p + n\log n + n^2 + n^2 + (n \vee \log n)) = O(n^2p)$.

For comparing the execution time of the proposed method (TC and TCS) with other gene selection methods, the average execution times (in milliseconds) of different methods are shown in Table 5. As for filter based gene selection, where the gene selection process is independent of the classifier, only the execution time of the feature selection process is reported. As described in section 2, for univariate feature selection, the dependency between genes will be ignored and so each gene is evaluated independently. For multivariate gene selection, on the other hand, the dependency between genes will affect the gene selection process. Therefore, for univariate gene selection (i.e., FS, ReliefF), the execution time is greater than multivariate (i.e., mRMR, and GCNC).

5.2 Discussion

In the past subsections, the performance of the proposed method was compared to other gene selection methods in different experiments. In this section, the reasons behind the efficiency of the proposed method are described briefly.

Increasing the performance of gene selection in microarray data with high-dimensional genes depends on how well the irrelevant and redundant genes are ignored when dealing with a dataset. If the gene selection method handles both the irrelevant and redundant genes efficiently, then it would obtain better results. That is why when using univariate methods (i.e., LS, FS and RelifF), since they ignore the gene dependency in their gene selection processes, they are not successful in removing redundant genes. In the proposed method, the symmetric uncertainty concept takes into consideration the relation between genes and also between each gene and its class label. The higher relation between different genes has a negative impact in the gene selection process, but on the other hand the effect of relation between each gene and class label is positive. Therefore, in the proposed method, using SU leads to ignoring the redundant and irrelevant genes, when selecting the genes in the first phase of the process. But using SU for calculating the graph weights in the first step of the proposed method may be time consuming for the datasets with huge number of features.

When it comes to the second phase by using graph clustering algorithm, not only the similar genes within a cluster are grouped, but also genes with minimum dependency to other genes within a cluster are selected. This helps selecting the genes with minimum redundancy, in the proposed method. By ranking the genes in third step and selecting the genes with highest rank on the global and lowest rank inside the cluster, the genes with similar attributes are ignored. Therefore, the redundant and irrelevant genes have a low chance for being picked up to the final feature subsets. In the proposed method, for selecting the best genes in third step, two different methods are used. The first method selects the genes within a cluster based on its probability. In order to calculate the probability, TV have been used. If a gene has greater TV, it means it contains valuable information. Therefore, using ΔTV (comparing between TV of when the gene is selected and when it is not) as probability of

Table 5: Average execution time (in ms) of different gene selection methods over ten independent runs.

Method	Colon	Leukemia	Prostate Tumor	Brain	SRBT	Lung Cancer	**Average**
TC	109,532	189,472	183,944	247,936	140,368	218,629	181,647
TCS	118,945	210,490	286,701	404,618	172,184	291,309	247,375
ReliefF	1,978	2,591	2,671	2,734	2,080	2,695	2,458
FS	89	103	176	217	91	205	147
LS	1,368	1,498	3,527	8,926	1,395	6,719	3,906
mRMR	9,763	11,261	13,893	14,789	10,428	14,735	12,478
RSM	68	71	128	459	73	193	165
MC	27	64	124	278	59	161	119
FAST	117,591	162,814	168,436	283,612	154,31	220,581	190,607
GCNC	116,831	193,477	205,716	259,332	124,821	238,900	189,846
U-FAM	43,953	61,704	72,136	121,742	51,826	91,491	73,809
FS-JMIE	186,947	261,955	37,827	450,970	190,333	480,729	268,127
MA-C	143,218	205,007	291,476	386,445	178,593	307,906	252,108
DSFFC	121,487	218,706	197,467	390,041	120,963	378,117	237,797
FCSVM-REF	147,281	269,405	293,863	458,052	193,924	394,589	292,852

selecting or ignoring a gene, results picking up only those genes which would lead to improve the method's performance.

In the second method, for selecting the best genes, SA is used, in the proposed method. SA is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a meta-heuristic to approximate global optimization in a large search space. It is often used when the search space is discrete. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, SA may be preferable to alternatives. Using Sum CRS (SCRS) as SA cost function leads to decreasing the redundancy between selected genes and increasing the performance of proposed method. The experimental results show that using SA in the third step of the proposed method (TCS) actually improves the results, however the selection process is slower than TC.

The proposed method uses class label in the first step with parameter α. This parameter is used for determining the effect of the class label and the relation between different genes. By changing the value of α, different conditions will be created and so, different results will be achieved. For example, if $\alpha = 1$, the class label will be ignored, and the algorithm will be changed to an unsupervised algorithm. This condition would be used when the label of the class is not reliable in the dataset. The Figure 2 and Figure 3 show that if the value of α is in range $[0.3,0.5]$ ($\frac{1-\alpha}{2}$ in range $[0.25,0.35]$), the best results are obtained.

5. 3 Statistical Analysis

The Friedman test is a non-parametric equivalent that can be used for illustrating the statistical significance of the results over multiple datasets. The test ranks each classifier accuracy over different datasets, separately. Therefore, the best performing algorithm gets the first rank, the second best gets the second rank and so on. After ranking each classifier over different datasets, χ_F^2 and F_F could be calculated. For N datasets and k methods:

$$\chi_F^2 = \frac{12N}{k(k+1)} \times \left[\sum_{i=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right] \quad (18)$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \quad (19)$$

Where R_j^2 is the average rank of the j^{th} method over all datasets and which is distributed according to the Fisher distribution with $k-1$ and $(k-1)\times(N-1)$ degrees of freedom. In Friedman test, using significance level α, the null hypothesis means all methods perform equivalently at level α.

In our experiments, using SVM, with $N = 12$ and $k = 14$, the critical value of Fisher distribution with 13 and degrees of freedom 143, for $\alpha = 0.05$, $F(13,143)$ is near 1.8. Therefore, if the gene selection method is incorporated with a classifier and the value of F_F is greater than 1.836, the null hypothesis will be rejected, and the result is statistically significant.

For other classifiers, with $N = 12$ and $k = 13$, the critical value of Fisher distribution with 12 and degrees of freedom 132, for $\alpha = 0.05$, $F(12,132)$ is near 1.8. Therefore, if the gene selection method is incorporated with a classifier and the value of F_F is greater than 1.874, the null hypothesis will be rejected, and the result is statistically significant.

For comparing the results of Friedman test between the proposed method (TC and TCS) and other gene selection methods, after ranking different gene selection method, the results of Friedman test are shown in Table 7. For all classifiers, it is demonstrated that when the gene selection methods are incorporated with different classifiers, the value of F_F is greater than 1.8. Therefore, the null hypothesis will be rejected, and it can be concluded that these results are statistically significant.

For demonstrating the distribution of accuracies for different treatments based on the obtained ranks, the distribution of accuracies for twelve datasets are investigated. It conclusion the distribution of all accuracies are near normal. In Fig. 5 and Fig. 6, Quantile-Quantile (Q–Q) plot for five datasets (i.e., Colon, Gastric, Ovarian, SRBT and Lung Cancer) are shown. In this plot the linearity of the points suggests the data are normally distributed [60].

6. Conclusion

In this paper, a new supervised/unsupervised filter gene selection was proposed, which was based on the graph clustering and ranking the genes and it selects the best subset of genes from the microarray data. For selecting the genes with minimum redundancy to other genes and maximum relevance to the class label, some concepts were used; Symmetric Uncertainty for creating graph, community detection for grouping the genes in different clusters, and a new method for ranking the genes. It did not use any learning model in the selection process which leaded to decrease the time complexity. The performance of the proposed method was compared to different gene/feature selection methods including Mutual correlation (MC), fast clustering-based feature selection algorithm (FAST), Graph Clustering with Node Centrality for unsupervised feature selection (GCNC), Unified- Feature Association Map (U-FAM), Feature

Table 7: The results of Friedman test

Classifier	χ_F^2		F_F		Significant	
	TC	TCS	TC	TCS	TC	TCS
SVM	47.30293	45.08423	8.7425	8.0451	+	+
DT	47.38560	48.65455	11.7498	11.7208	+	+
NB	52.84612	50.19547	12.4287	12.8106	+	+
kNN	48.98744	50.04512	11.7298	11.4925	+	+

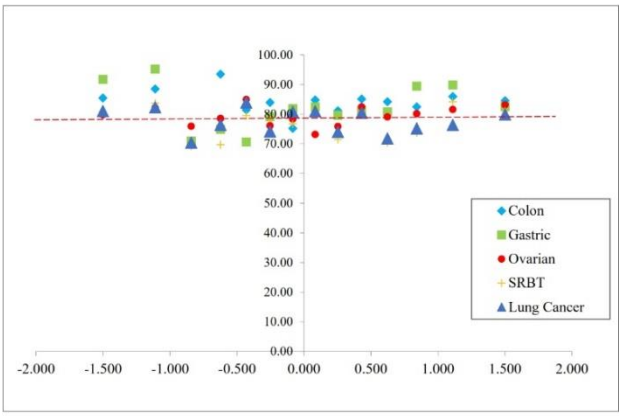

Fig. 5: Distribution of accuracies for different classification accuracies of TCS for five datasets using SVM classifier.

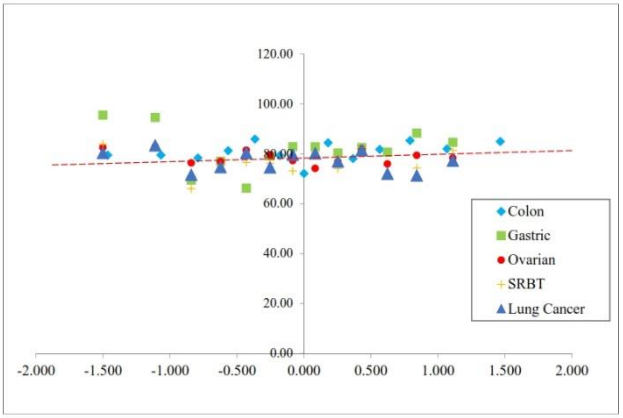

Fig. 6: Distribution of accuracies for different classification accuracies of TCS for five datasets using DT classifier.

Selection method with Joint Maximal Information Entropy between features and class (FS-JMIE), a Correlation based Memetic Algorithm (MA-C), Dense Subgraph Finding with Feature Clustering (DSFFC), Distributed dCor-based FS (D2CORFS), a ReliefF and ACO-based gene selection (RFACO-GS), A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm (FICA-IWSSr), Greedy Randomized Adaptive Search Procedure (GRASP), and Support Vector Machine Recursive Feature Elimination (FCSVM-REF). The performance comparison was done over twelve microarray datasets using four different classifiers including Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB) and k-Nearest Neighbor (kNN).

The experimental results showed that the proposed method using three steps for selecting the best gene subset, was able to select a subset of genes with minimum redundancy between genes and maximum relevance to class lable. The results indicated that the proposed method obtained significantly better result in comparison to the other well-known supervised and unsupervised feature/gene selection approaches. The results also confirmed that the proposed method achieved satisfying performance over different classifiers for different microarray datasets.

References

[1] A. Salimi, M. Ziaii, A. Amiri, M. H. Zadeh, S. Karimpouli, and M. Moradkhani, "Using a feature subset selection method and support vector machine to address curse of dimensionality and redundancy in hyperion hyperspectral data classification," The Egyptian Journal of Remote Sensing and Space Science, vol. 21, no. 1, pp. 27–36, 2018.

[2] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," science, vol. 286, no. 5439, pp. 531–537, 1999.

[3] H. Liu and H. Motoda, Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series). Chapman and Hall, 2007.

[4] A. Jovi´c, K. Brki´c, and N. Bogunovi´c, "A review of feature selection methods with applications," in 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2015, pp. 1200–1205.

[5] B.Liao,Y.Jiang,W.Liang,W.Zhu,L.Cai,andZ.Cao, "Gene selection using locality sensitive laplacian score," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 11, no. 6, pp. 1146–1156, Nov 2014.

[6] S. Theodoridis and K. Koutroumbas, Pattern Recognition, fourth edition. Academic Press, Oxford, 2008.

[7] R. Cai, Z. Hao, X. Yang, and W. Wen, "An efficient gene selection algorithm based on mutual information," Neurocomputing, vol. 72, no. 4, pp. 991 – 999, 2009.

[8] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," Annals of Mathematics and Artificial Intelligence, vol. 41, no. 1, pp. 77–93, May 2004.

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

[10] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," Journal of Machine Learning Research, vol. 5, pp. 1205–1224, 2004.

[11] A. J. Ferreira and M. A. Figueiredo, "Efficient feature selection filters for high-dimensional data," Pattern Recognition Letters, vol. 33, no. 13, pp. 1794 – 1804, 2012.

[12] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," Engineering Applications of Artificial Intelligence, vol. 32, no. Supplement C, pp. 112 – 123, 2014.

[13] C. Lai, M. J. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," Pattern Recognition Letters, vol. 27, no. 10, pp. 1067 – 1076, 2006.

[14] R. J. Manoj, M. A. Praveena, and K. Vijayakumar, "An aco– ann based feature selection algorithm for big data," Cluster Computing, pp. 1–8, 2018.

[15] I. Jain, V. K. Jain, and R. Jain, "Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification," Applied Soft Computing, vol. 62, pp. 203–215, 2018.

[16] S. Maldonado, R. Weber, and J. Basak, "Simultaneous feature selection and classification using kernel-penalized

support vector machines," Information Sciences, vol. 181, no. 1, pp. 115 – 128, 2011.
[17] G. Wang, Q. Song, B. Xu, and Y. Zhou, "Selecting feature subset for high dimensional data via the propositional foil rules," Pattern Recognition, vol. 46, no. 1, pp. 199 – 214, 2013.
[18] J. Canul-Reich, L. O. Hall, D. Goldgof, J. N. Korecki, and S. Eschrich, "Iterative feature perturbation as a gene selector for microarray data," vol. 26, 08 2012.
[19] P. A. Mundra and J. C. Rajapakse, "Svm-rfe with mrmr filter for gene selection," IEEE Transactions on NanoBioscience, vol. 9, no. 1, pp. 31–37, March 2010.
[20] L.-Y. Chuang, C.-H. Yang, K.-C. Wu, and C.-H. Yang, "A hybrid feature selection method for dna microarray data," Computers in Biology and Medicine, vol. 41, no. 4, pp. 228 – 237, 2011.
[21] W. Zhao, G. Wang, H. Wang, H. Chen, H. Dong, and Z. Zhao, "A novel framework for gene selection," vol. 3, pp. 184–191, 04 2011.
[22] G. Forman, "An extensive empirical study of feature selection metrics for text classification," J. Mach. Learn. Res., vol. 3, pp. 1289–1305, march 2003.
[23] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in Proceedings of the Tenth National Conference on Artificial Intelligence, ser. AAAI'92, 1992, pp. 129–134.
[24] I. Kononenko, Estimating attributes: Analysis and extensions of RELIEF.
[25] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in Proceedings, Twentieth International Conference on Machine Learning, vol. 2, 01 2003, pp. 856–863.
[26] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," IEEE Transactions on Neural Networks, vol. 5, no. 4, pp. 537–550, July 1994.
[27] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," IEEE Trans. on Knowl. and Data Eng., vol. 25, pp. 1–14, jan 2013.
[28] M. Mandal and A. Mukhopadhyay, Unsupervised Nonredundant Feature Selection: A Graph-Theoretic Approach. Springer Berlin Heidelberg, 2013, pp. 373–380.
[29] S. Bandyopadhyay, T. Bhadra, P. Mitra, and U. Maulik, "Integration of dense subgraph finding with feature clustering for unsupervised feature selection," Pattern Recognition Letters, vol. 40, no. Supplement C, pp. 104 – 112, 2014.
[30] S. Tabakhi, A. Najafi, R. Ranjbar, and P. Moradi, "Gene selection for microarray data classification using a novel ant colony optimization," Neurocomputing, vol. 168, no. Supplement C, pp. 1024 – 1036, 2015.
[31] P. Moradi and M. Rostami, "Integration of graph clustering with ant colony optimization for feature selection," Knowledge-Based Systems, vol. 84, no. Supplement C, pp. 144 – 161, 2015.
[32] A. Pino Angulo, "Gene selection for microarray cancer data classification by a novel rule-based algorithm," Information, vol. 9, no. 1, p. 6, 2018.
[33] S. S. Kannan and N. Ramaraj, "A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm," Knowledge-Based Systems, vol. 23, no. 6, pp. 580–585, 2010.
[34] K. Zheng and X. Wang, "Feature selection method with joint maximal information entropy between features and class," Pattern Recognition, vol. 77, pp. 20–29, 2018.
[35] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in IJCAI, 1993, pp. 1022–1029.
[36] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, "A link clustering based overlapping community detection algorithm," Data and Knowledge Engineering, vol. 87, no. Supplement C, pp. 394 – 404, 2013.
[37] E. Le Martelot and C. Hankin, "Fast multi-scale detection of relevant communities in large-scale networks," The Computer Journal, vol. 56, no. 9, pp. 1136–1150, 2013.
[38] V. D. Blondel, J. loup Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," pp. 1–12, 2008.
[39] C. Gao, D. Wei, Y. Hu, S. Mahadevan, and Y. Deng, "A modified evidential methodology of identifying influential nodes in weighted networks,"Physica A:Statistical Mechanics and its Applications, vol. 392, no. 21, pp. 5490–5500, 2013.
[40] M. R. G. R. . V. L. V. Nicosia, R. Criado, "Controlling centrality in complex networks," Scientific Reports 2, no. 218, pp. 218–223, 2012.
[41] M. Campiteli, A. Holanda, L. Soares, P. Soles, and O. Kinouchi, "Lobby index as a network centrality measure," Physica A: Statistical Mechanics and its Applications, vol. 392, pp. 5511 – 5515, 2013.
[42] Y. Du, C. Gao, Y. Hu, S. Mahadevan, and Y. Deng, "A new method of identifying influential nodes in complex networks based on topsis," Physica A: Statistical Mechanics and its Applications, vol. 399, no. Supplement C, pp. 57 – 69, 2014.
[43] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 3, pp. 301–312, March 2002.
[44] M. P. V. S. Kirkpatrick, C. D. Gelatt Jr., "Optimization by simulated annealing," American Association for the Advancement of Science, vol. 220, no. 4598, pp. 671–680, may 1983.
[45] K. R. B. D. S. Repository, kent ridge bio-medical dataset, http://datam.i2r.a-star.edu.sg/datasets/krbd/.
[46] B. institute, Cancer Program Data Sets, 2014, http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi.
[47] I. T. G. A. Statnikov, C.F. Aliferis, Gene Expression Model Selector, 2005, http://www.gems-system.org.
[48] M. Haindl, P. Somol, D. Ververidis, and C. Kotropoulos, Feature Selection Based on Mutual Correlation. Springer Berlin Heidelberg, 2006, pp. 569–577.
[49] P. Moradi and M. Rostami, "A graph theoretic approach for unsupervised feature selection," Engineering Applications of Artificial Intelligence, vol. 44, no. Supplement C, pp. 33 – 45, 2015.
[50] A. K. Das, S. Goswami, A. Chakrabarti, and B. Chakraborty, "A new hybrid feature selection approach using feature association map for supervised and unsupervised classification," Expert Systems with Applications, vol. 88, no. Supplement C, pp. 81 – 94, 2017.
[51] A. Brankovic, M. Hosseini, and L. Piroddi, "A distributed feature selection algorithm based on distance correlation with an application to microarrays," IEEE/ACM transactions on computational biology and bioinformatics, 2018.
[52] L. Sun, X. Kong, J. Xu, R. Zhai, S. Zhang et al., "A hybrid gene selection method based on relieff and ant colony

optimization algorithm for tumor classification," Scientific Reports, vol. 9, no. 1, p. 8978, 2019.
[53] M. Moradkhani, A. Amiri, M. Javaherian, and H. Safari, "A hybrid algorithm for feature subset selection in high dimensional datasets using fica and iwssr algorithm," Applied Soft Computing, vol. 35, pp. 123–135, 2015.
[54] P. Bermejo, J. A. Gamez, and J. M. Puerta, "A grasp algorithm for fast hybrid (filter-wrapper) feature subset selection in high dimensional datasets," Pattern Recognition Letters, vol. 32, no. 5, pp. 701–711, 2011.
[55] X. Huang, L. Zhang, B. Wang, F. Li, and Z. Zhang, "Feature clustering based support vector machine recursive feature elimination for gene selection," Applied Intelligence, vol. 48, no. 3, pp. 594–607, 2018.
[56] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," Machine Learning, vol. 46, pp. 389–422, Jan 2002.
[57] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, March 1986.
[58] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," Machine Learning, vol. 6, pp. 37–66, Jan 1991.
[59] Hamdani and R. Wardoyo, "The complexity calculation for group decision making using topsis algorithm," AIP Conference Proceedings, vol. 1755, pp. 0700071 – 0700077, 2016.
[60] T. J. Cleophas and A. H. Zwinderman, "Quantile-quantile plots, a good start for looking at your medical data (50 cholesterol measurements and 58 patients)," in Machine Learning in Medicine-a Complete Overview. Springer, 2015, pp. 253– 259.

Soodeh Bakhshandeh received his BS degree in Computer Engineering (Software) from Alzahra University, Tehran, Iran in 2007. She graduated in Artificial Intelligence from Science and Research Branch, Islamic Azad University, Tehran, Iran. Her research areas are pattern recognition, feature selection, big data analysis, and data mining.

Reza Azmi received his BS degree in Electrical Engineering from Amirkabir University of technology, Tehran, Iran in 1990 and his MS and PhD degrees in Electrical Engineering from Tarbiat Modares University, Tehran, Iran in 1993 and 1999, respectively. He is a faculty member of Computer Engineering Department of Alzahra University of Technology since 2001. He is conducting research activities in the areas of big data analysis, machine learning and cloud computing.

Mohammad Teshnehlab is a faculty member of Electrical Eng. Department of K. N. Toosi University of Technology. His main research interests focus on Intelligent Systems and Control. Professor Teshnehlab is a member of Industrial Control Center of Excellence and founder of Intelligent Systems Laboratory (ISLab). He also is head and Co-founder of Intelligent Systems Scientific Society of Iran (ISSSI) and member of editorial board of International Journal of Information and Communication Technology Research (IJICTR).