https://doi.org/10.31436/iiumej.v22i1.1447

A NOVEL DIMENSIONALITY REDUCTION APPROACH TO IMPROVE MICROARRAY DATA CLASSIFICATION

MOHAMMED HAMIM[1*], ISMAIL EL MOUDDEN[2], MOUNIR OUZIR[3], HICHAM MOUTACHAOUIK[1] AND MUSTAPHA HAIN[1]

[1]I2SI2E Laboratory, ENSAM- Casablanca, University Hassan II, Casablanca, Morocco
[2]EVMS-Sentara Healthcare Analytics and Delivery Science Institute, Eastern Virginia Medical School, Norfolk, VA, USA
[3]Group of Research in Physiology and Physiopathology, Department of Biology, Faculty of Science, University Mohammed V, Rabat, Morocco

[*]Corresponding author: mohamed.hamim@gmail.com

(Received: 12th April 2020; Accepted: 30th September 2020; Published on-line: 4th January 2021)

ABSTRACT: Cancer tumor prediction and diagnosis at an early stage has become a necessity in cancer research, as it provides an increase in the treatment success chances. Recently, DNA microarray technology became a powerful tool for cancer identification, that can analyze the expression level of a different and huge number of genes simultaneously. In microarray data, the large genes number versus a few records may affect the prediction performance. In order to handle this "curse of dimensionality" constraint of microarray dataset while improving the cancer identification performance, a dimensional reduction phase is necessary. In this paper, we proposed a framework that combines dimensional reduction methods and machine learning algorithms in order to achieve the best cancer prediction performance using different microarray datasets. In the dimensional reduction phase, a combination of feature selection and feature extraction techniques was proposed. Pearson and Ant Colony Optimization was used to select the most important genes. Principal Component Analysis and Kernel Principal Component Analysis were used to linearly and non-linearly transform the selected genes to a new reduced space. In the cancer identification phase, we proposed four algorithms C5.0, Logistic Regression, Artificial Neural Network, and Support Vector Machine. Experimental results demonstrated that the framework performs effectively and competitively compared to state-of-the-art methods.

ABSTRAK: Ramalan tumor kanser dan diagnosis pada peringkat awal telah menjadi keperluan dalam kajian kanser, kerana ia membuka peluang peningkatan kejayaan dalam rawatan. Kebelakangan ini, teknologi mikrotatasusunan DNA menjadi alat berkuasa bagi mengenal pasti kanser, di mana ia mampu menganalisa level ekspresi yang pelbagai dan gen-gen yang banyak secara serentak. Dalam data mikrotatasusunan, gen-gen yang banyak ini bakal menentukan ramalan prestasi berbanding analisa melalui rekod-rekod yang sebilangan. Fasa pengurangan dimensi adalah perlu bagi mengawal kakangan "penentuan kedimensian" dataset mikrotatasusunan, sementara itu ia memantapkan lagi keberkesanan kenal pasti kanser. Kajian ini mencadangkan rangka kombinasi kaedah pengurangan dimensi dan algoritma pembelajaran mesin bagi mencapai prestasi ramalan kanser terbaik dengan menggunakan pelbagai dataset mikrotatasusunan. Dalam fasa pengurangan dimensi, kombinasi pemilihan ciri dan teknik pengekstrakan ciri telah dicadangkan, Pengoptimuman Pearson dan Koloni Semut bagi memilih gen yang paling penting, Analisis Komponen Prinsipal dan Analisis Komponen Prinsipal Kernel, bagi menukar gen terpilih yang linear dan tak linear kepada ruang baru yang dikurangkan. Dalam

menentukan fasa mengenal pasti kanser, kajian ini mencadangkan empat algoritma iaitu C5.0, Regresi Logistik, Rangkaian Neural Buatan dan Mesin Vektor Sokongan. Dapatan kajian menunjukkan rangka ini adalah berkesan dan kompetitif berbanding kaedah semasa.

KEYWORDS: *gene selection; metaheuristic-ant colony optimization; feature extraction; pattern recognition; microarray data analysis*

1. INTRODUCTION

According to a recent publication by the World Health Organization (WHO) in 2018, cancer is considered the second most lethal factor for human beings. Knowing that early diagnosis is a mandatory and a crucial step in cancer treatment, the chance to get an appropriate treatment may require further measurements to increase the accuracy of cancer diagnosis combined with other clinical tests. With the development of machine learning techniques and microarray technology, the DNA analysis microarray data brings a great opportunity in cancer diagnosis. However, the presence of a large number of irrelevant or redundant genes (features) in gene expression data may increase the search space size, which makes pattern detection more difficult and makes it complex to capture the necessary rules for classification [1]. To overcome this "curse of dimensionality", a dimensional reduction process is strongly recommended. Dimensional reduction refers to a process that removes redundant and noisy features from the data, thus maximizing prediction performance. Dimensional reduction can be divided into feature selection (FS) and feature extraction (FE). FE methods create a subset of new features by combinations of existing features. The new features are low-dimensional features with the same or better performance in terms of prediction accuracy. In the literature, some proposed FE methods for cancer classification using gene expression data include Principal Component Analysis (PCA) [2] and kernel PCA [3]. On the other hand, the FS process focuses simply on the relevant features in the dataset by removing any redundant, irrelevant, or noisy features, which leads to better learning performance. The frequently used FS methods are divided into filter and wrapper. In the filter approach, features are scored based on statistical criteria such as Pearson correlation coefficients (P) [4]. In the wrapper approach [5], FS is combined with classification algorithms. Examples of wrapper algorithms include Ant Colony Optimization algorithm (ACO), Genetic Algorithm (GA), and others. When the number of features becomes very large, the filter methods are usually chosen due to their computational efficiency and simplicity [6]. In this paper, in addition to the Pearson correlation-based filter, a hybrid approach of feature selection has also been proposed that takes advantage of filter and wrapper methods. The proposed hybrid approach combines correlation-based feature selection with the ACO algorithm.

In this study, our aim is to improve the performance of cancer tumor modeling using a framework that combines FS and FE as dimension reduction methods with machine learning algorithms.

2. RELATED WORKS

The importance of classifying cancer patients into high or low risk groups has led to study the application of machine learning methods. Different strategies exist focusing on modifying the data for better fitting in a specific machine learning method; among them, we have dimensionality reduction, FS, and FE [7]. Several DNA microarray experiments have marked the power of datamining methods over clinical criteria for cancer diagnosis [8,9]. These studies accentuate the improvement of prediction performance based on gene

expression data by combining dimensional reduction techniques with machine learning algorithms.

To improve prostate cancer performance modeling and mining, Hicham and al. have proposed a new framework combining feature selection using Pearson and feature extraction using PCA in conjunction with machine learning algorithms. The most important result achieved in this study is obtained by the Pearson-PCA-C5.0 model with 94.05% classification accuracy and five selected features [10]. Kar et al. proposed a combination of filter method based on t-test and wrapper method based on particle swarm optimization (PSO) to find the most relevant genes in the SRBCT microarray dataset. The study achieved 100% accuracy for 14 selected genes [11].

Atiyeh and Mohammad implemented an innovative feature selection approach Based on Cooperative Game Theory and Qualitative Mutual Information (QMT). The classification accuracy on 11 microarray datasets, namely Leukemia1, SRBCT, Lung, and prostate cancer, shows that the proposed approach improves both accuracy and stability compared to other methods [12]. Chandra proposed an efficient feature selection technique that removes the drawbacks of [13], by taking into account the redundancy between features. the research study shows that the classification accuracy form using the proposed algorithm Inter Feature Effective range overlap (IFERO), for many cancers, is much superior compared to other feature selection algorithms. The proposed technique has been applied to 8 benchmark cancer datasets [14].

Shun Guo et al. formulated the feature selection problem as an optimization one based on a newly defined linear discriminant analysis criterion. The experiment was applied to 10 publicly available microarray datasets, and the results show that the proposed gene selection is an effective method for improving the accuracy of tumor classification [15].

The present paper aims to improve the classification performance for four benchmark cancer datasets. For this purpose and in order to handle the curse of dimensionality problem of the Microarray dataset, we propose a framework that combines FS and FE methods in conjunction with machine-learning algorithms.

3. MATERIALS AND METHODS

Figure 1 summarizes the main steps of our proposed framework, which is based on feature selection using Filter and Hybrid approaches, FE using linear and non-linear PCA, and cancer identification (classification) using Logistic Regression (LR), C5.0 Decision Tree algorithm, Support-Vector Machines SVM, and Artificial Neural Network (ANN). The main structure of the proposed Framework is described in Algorithm 1.

3.1 Feature Selection Methods

The feature selection or gene selection in the context of microarray data analysis is a useful technique that can reduce dimensionality by removing any redundant, irrelevant or noisy genes, which can lead to improve the classification performance and reduce the cost of computation [16]. As shown in Fig. 2, the feature selection process can be reformulated as follows: given an original set, $X = (X_1, X_{1,}, \cdots, X_p)$, of p features, find the subset which consists of k features (where $k \ll p$), such that the most informative features are selected.

The proposed framework in the present paper implements two feature selection techniques, the filter method based on statistic tools and the hybrid method that combines the filter approach with ACO.

https://doi.org/10.31436/iiumej.v22i1.1447

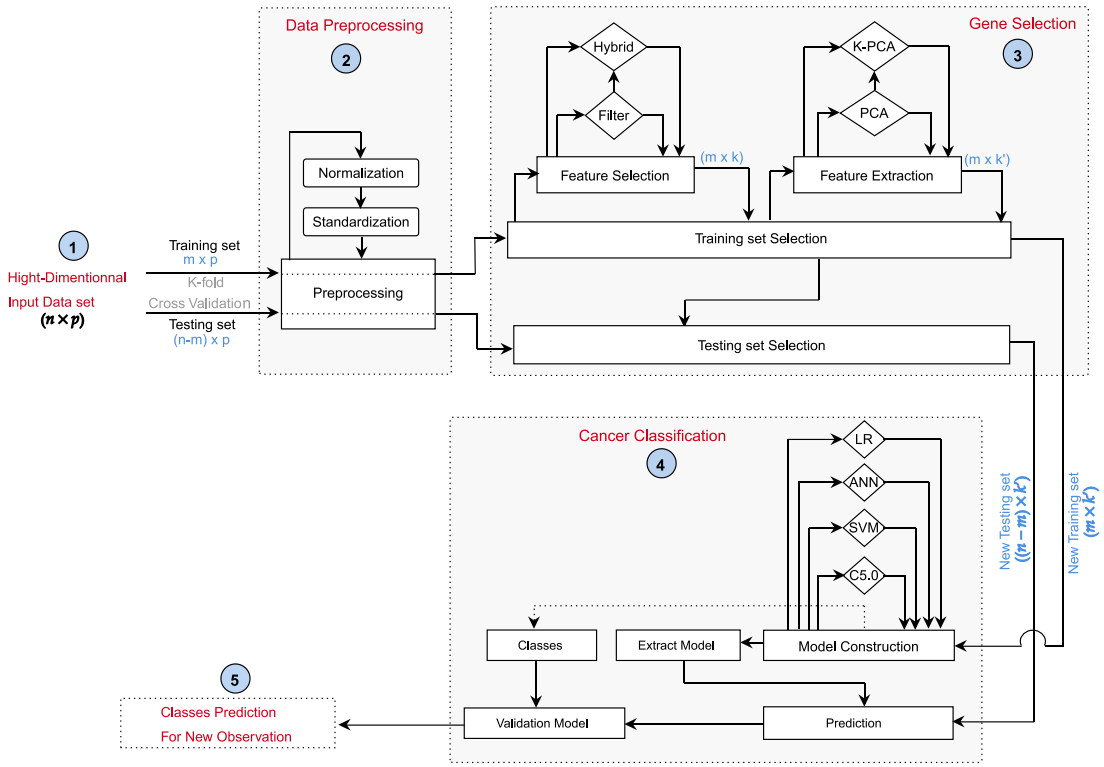


Fig. 1: Our proposed Framework.

Fig. 2: Feature Selection process.

3.1.1 Filter Method using Pearson's Correlation Coefficient

Because they act independently of any classification process, filters are considered to be faster than the wrapper approach. This is because this model is frequently used when it comes to working with a large number of features [17]. To measure feature relevance using filter methods, some statistical techniques are applied for each feature, such as Pearson's correlation coefficient, Spearman's rank correlation, Pearson 's Chi-square, Cramer's v, ...

In the present paper, Pearson's correlation in Eq. (1), denoted by r, was applied to recognize features (X) showing a strong linear relationship with the target (Y).

Algorithm 1: Framework

```
/* Function to preprocess data */
Function DataPreprocessing (Training_set, Testing_set):
    Training_set, Testing_set ←Log_Transformation ( Training_set, Testing_set ) ;   // Equation (15)
    Training_set, Testing_set ←Standardization( Training_set, Testing_set);         // Equation (16)
    return Training_set, Testing_set

/* This function returns an new subset of k genes from the original set */
Function FeatureSelection (Method, Training_set, Testing_set):
    if Method = Filter then
        Subset_train ←Select relevant genes from Training_set using Pearson correlation based Filter method. ;   // Equations (1, 2)
    else if Method = Hybrid then
        Subset_train ←Select relevant genes from Training_set using PACO based Hybrid method. ;                   // Algorithm (2)
    Subset_test ←Select from Testing_set the same genes selected from Training_set.
    Return Subset_train, Subset_test

/* This function transforms an input set to a new set of k' predictors */
Function FeatureExtraction (Method, Training_set, Testing_set):
    if Method = PCA then
        Construct a projection matrix using Principal Component Analysis .;          // Algorithm (3)
    else if Method = KPCA then
        Construct a projection matrix using kernel Principal Component Analysis .;   // Algorithm (4)
    Subspace_train, Subspace_test ←Use the projection matrix to transform the Training_set and Testing_set into a k'-dimensional feature subspaces.
    Return Subspace_train, Subspace_test

Function Classification (Training_set, Testing_set):
    List_Classifiers = [SVM, LR, C5.0, ANN] /* iterate over all classifiers */
    for each Classifiers in List_Classifiers do
        Model ←Train classifier on the Training_set
        Test the Model on the Testing_set
        Calculate average performance (Accuracy and AUC).
    Return All obtained Models with their average performances

/* Main program */
Input  :-A p-diemsional DNA microarray dataset D = [y, x1, x2, ..., xp]n×1 , n is the number of patterns, x is the feature vector , y is the target vector and p is the number of features
Output : List of generated models with average performance and running time for each one.
Split dataset D into K-folds using Stratified K − fold cross-validation technique.
for each fold in D do
    D_test ← fold
    D_train ← remaining (K − 1) − folds
    /* Step aims at separately preprocessing the Training and Testing sets */
    D_train, D_test ← DataPreprocessing(D_train, D_test).
    /* Step aims at generating all models based on pristine data */
    Pristine_Models ← Classification(D_train, D_test)
    Pristine_PCA_Models ← Classification(FeatureExtraction(PCA, D_train, D_test))
    Pristine_KPCA_Models ← Classification(FeatureExtraction(KPCA, D_train, D_test))
    /* Generate all models based on selected genes using Pearson correlation based filter method */
    SubF_train, SubF_test ← FeatureSelection(Filter, D_train, D_test)
    Pearson_Models ← Classification(SubF_train, SubF_test)
    Pearson_PCA_Models ← Classification(FeatureExtraction(PCA, SubF_train, SubF_test))
    Pearson_KPCA_Models ← Classification(FeatureExtraction(KPCA, SubF_train, SubF_test))
    /* Generate all models based on selected genes using PACO based Hybrid approach */
    SubH_train, SubH_test ← FeatureSelection(Hybrid, D_train, D_test)
    PACO_Models ← Classification(SubH_train, SubH_test)
    PACO_PCA_Models ← Classification(FeatureExtraction(PCA, SubH_train, SubH_test))
    PACO_KPCA_Models ← Classification(FeatureExtraction(KPCA, SubH_train, SubH_test))
Return all generated models
```

$$r_{(X,Y)} = \frac{\sum_{i=1}^{N}(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{X})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{Y})^2}} \quad (1)$$

Where N is the total number of samples in the training set. $\bar{X}$, $\bar{Y}$ are, respectively, the overall mean of the X and the Y. x_i and y_i are, respectively, the i-th observation in X and Y. The $r_{(X,Y)}$ always lies between ±1, where 1 indicates a perfect relationship between X and Y, and the 0 indicates no relationship between them.

Then, the relevance value of each feature X is measured as $(1 - \mathrm{p_{Value}}) \times 100\%$, where $\mathrm{p_{Value}}$ based on the t-statistic with $\mathrm{df} = \mathrm{N} - 2$ degree of freedom is computed using the Eq. (2).

$$p_{Value} = Probability(T(df) > t) \quad (2)$$

With $t = \mathrm{r}\sqrt{df/(1 - r^2)}$, and $\mathrm{T(df)}$ is a random variable that follows a Student's t-distribution with df.

In this study, all features (genes) in the training set with relevance greater than 95% were selected.

3.1.2 Ant Colony Optimization Hybrid Approach

Hybrid methods attempt to combine the straightness of two feature selection methods. The most frequently used combination is the filter with a wrapper approach [17]. The present framework proposes a new hybrid technique (Fig. 3) that combines Pearson's correlation and Ant Colony Optimization (PACO). The filter step in the framework consists of reducing the number of genes by removing non-informative genes in the original training set, and then the number of pre-selected genes is given to the ACO to select the optimal subset in the original training set.

Proposed by Marco Dorigo [18], ACO is a nature-inspired metaheuristic approach. The idea behind ACO is to represent the search space of a problem in the form of a graph, then the solution of the problem is to find the optimal path in this graph using artificial ants. As in real ant colonies, each ant deposits the pheromone trail with the same rate on the components of the graph that it chooses to cross. The chosen path to cross by an ant is usually based on the accumulated pheromone trail. Thus, the accumulated pheromone is considered to be an indicator of the quality of the chosen path, which can attract ants in the next iterations to the corresponding areas in the search space [19].

The ACO has been a powerful tool in many optimization problems [20-22], and for many reasons, it was recently used as a powerful tool for gene selection [23-25]. In feature selection using ACO, each node in the graph is viewed as a feature (gene), and edges between nodes (features) represent the choice of the next node to be selected. Thus, searching the optimal subset of features is to find the optimal path in the graph until such a stopping criterion is satisfied. The problem of feature selection using PACO can be reformulated as follows: given an original training set $X = (X_1, X_1, \cdots, X_p)$, of p features, find the subset that consists of k features (where $k \ll p$), such that a maximum number of iterations is reached.

According Fig. 3, before starting any iteration, the number of genes in the optimal subset to select is initialized using a Pearson correlation-based filter, and the amount of pheromone in the search space is initialized to a constant value. Then, at the start of each iteration t, each ant k starts in a randomly selected feature. To select (to visit) the next feature (nodes) from unselected ones, each ant must respect the probabilistic "transition rule" [19] using the Eq. (3).

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t) \cdot \eta_{ij}^{\beta}(t)}{\sum_{l \in S_i^k} (\tau_{il}^{\alpha}(t) \cdot \eta_{il}^{\beta}(t))}, & \forall j \in S_i^k \\ 0, & otherwise \end{cases} \tag{3}$$

Where:

S_i^k : features set that have not been visited yet.
$\tau_{ij}(t)$: the amount of pheromone trail between feature i and j.
$\eta_{ij}(t)$: the heuristic desirability of choosing feature j when the ant k at feature i.
$\alpha \geqslant 0$: adjustable parameters deciding the relative influence of pheromone.
$\beta \geqslant 1$: adjustable parameters controlling the influence of η_{ij}.

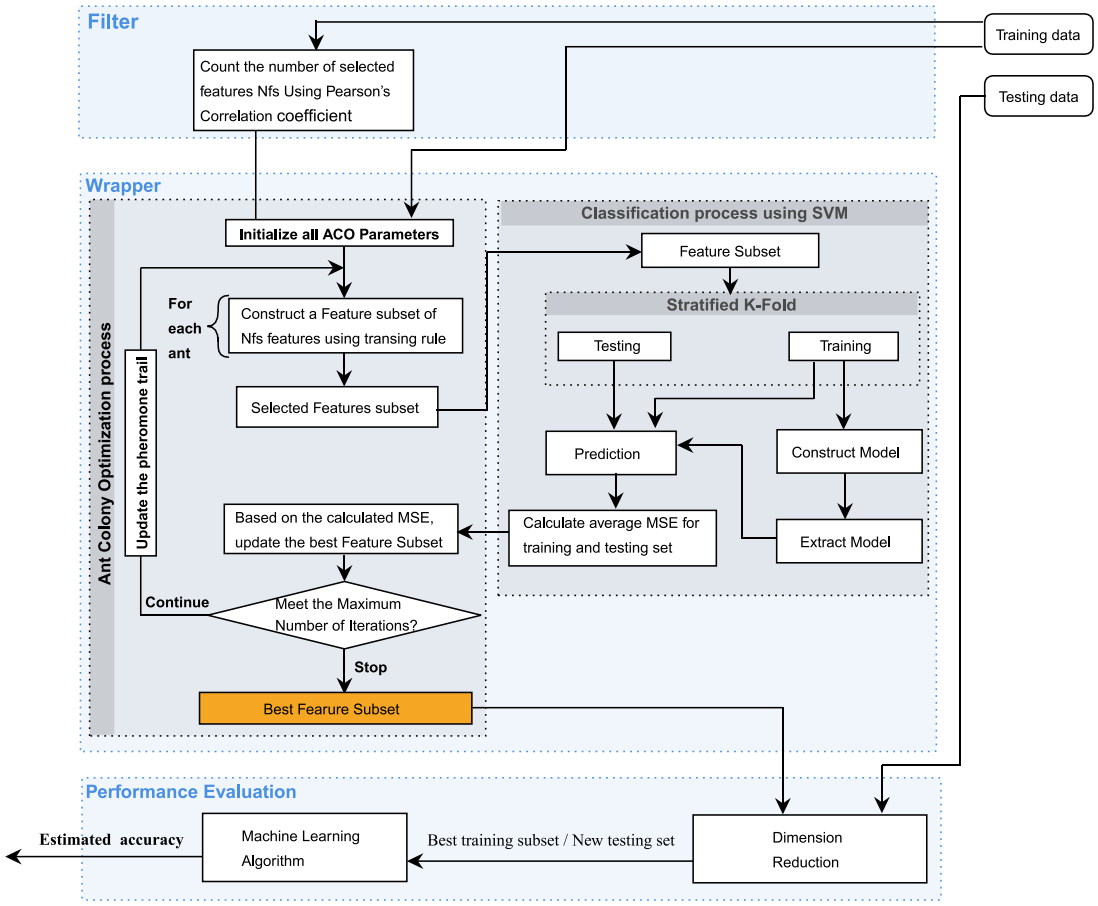


Fig. 3: Ant Colony Optimization hybrid approach.

The constructed subset S^k by the k-th ant is then evaluated using an SVM classifier, and the estimated Mean Square Error MSE_k of the classification results will decide if the current subset is the best one. The MSE is computed by applying a stratified $5-fold$ Cross Validation method. The constructed subset is split into five-folds, and at each time, one of the five folds is used for the test, and the remaining folds form the training data. Then the average MSE for the five trials is calculated using the Eq. (4).

$$MSE = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{2}(MSE_i^{Train} + MSE_i^{Test}) \tag{4}$$

The subset giving the lowest MSE is known as the best one related to the best ant and denoted by S_{best}. At the end of each iteration, the amount of pheromone in the search space is updated according to the Eq. (5) [19].

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$
$$\text{With}: \Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{MSE_k} & if\ the\ edge(i,j)\ is\ spart\ of S^k \\ 0 & otherwise \end{cases} \tag{5}$$

Where:

m : is the number of ants.

S^k : represents the constructed subset corresponding to the ant k.
ρ : denotes the pheromone evaporation coefficient.
Q: a constant multiplier that defines the amount of pheromones that should put each ant.
MSE_k: denote the Mean Square Error corresponding to the constructed subset by the ant k.

The overall pseudocode of the proposed PACO gene selection approach is illustrated in Algorithm 2.

Algorithm 2: PACO algorithm

Input : $X = (X_1, X_2, ..., X_p)$:is the training set, where p is the total number of features
Nfs : the number of features to select
m : define the number of ants
$Nits$: number of iterations that algorithm repeated
Output: S_{best}: the best set of selected features from X

-Initialize the pheromone trails τ and heuristic desirability η for each edge in the search space.
-Initialize system parameters $(\alpha, \beta, \sigma, \rho, Q)$
for $i \leftarrow 1$ **to** $Nits$ **do**
  for $k \leftarrow 1$ **to** m **do**
    -Randomly select the first feature from the original set X
    -Select the remaining $(Nfs - 1)$ features according to Equation (3)
    -Evaluate the constructed subset S using SVM classifier
    -If S has the best Mean Square Error (MSE), store the corresponding ant as the best one
  end
  -Update the pheromone trail according to Equation (5)
end
Return S_{best} the best constructed subsets corresponding to the best ant

3.2 Feature Extraction Methods

Feature extraction (FE) is the process of transforming original data with a large number of features into a reduced representation of a set of features. As shown in Fig. 4, the FE is achieved by transforming $X = (X_1, X_1, \cdots, X_p)$, of p features to a new set of k predictor variables called components (where $k \ll p$).

Among linear and nonlinear methods, PCA and Kernel PCA are the most commonly used FE techniques for dimensionality reduction. In this paper, we attempt to use FE methods combined with FS ones in order to handle the curse of dimensionality of cancer datasets.

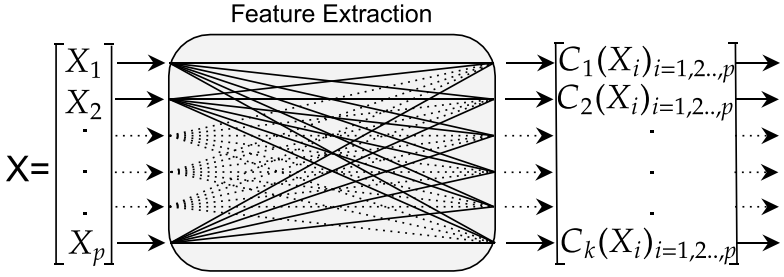


Fig. 4: Feature Extraction process.

3.2.1 Principal Components Analysis (PCA)

PCA is a classical dimension-reduction technique used to reduce large sets of variables (features) into new small ones without much loss of information from the large sets [26]. Mathematically, PCA attempts to transform a number of linearly correlated variables into a smaller number of new ones called components. In other words, PCA aims to find a linear subspace of lower dimensionality than the large variable space, where the new linear

subspace has the largest variance (has the most of the information in the large space). FE using PCA can be reformulated as follows:

Given a p-dimensional training set: $[x_1, x_2, \dots, x_p]_{n\times 1}$, Where: p denotes the number of features and n the number of patterns. We want to find Ψ, the matrix of new components where the number of principal components that should be retained is decided using the percentage of total variance explained. The pseudocode of the PCA method is illustrated in Algorithm 3.

Algorithm 3: PCA algorithm

Input :-A p-diemsional training set $X = [x_1, x_2, \dots, x_p]_{n\times 1}$, n is the number of patterns, x is the feature vector and p is the number of features
Output: ψ the matrix of principal components
1 Compute the mean $\overline{x} \leftarrow \frac{1}{n}\sum_{i=1}^{n} x_i$
2 compute the standard deviation $s_x \leftarrow \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$
3 Compute Z the standardized matrix : $z_i \leftarrow \frac{x_i - \overline{x}}{s_x}$
4 Compute the Correlation matrix $R \leftarrow \frac{1}{n} ZZ^T$, T is the matrix transposition
5 Compute eigenvalues λ_i and eigenvectors u_i of matrix R, $Ru_i \leftarrow \lambda_i u_i$ where $i \in (1, 2, \dots, p)$
6 Arrange all eigenvalues (λ_i) in descending order
7 keep the d (where $d << p$) eigenvalues showing $\alpha \times 100\%$ of the variance $\frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{p}\lambda_i} \geqslant \alpha$
8 For each u_i where $i \in (1, .., d)$ compute $\psi_i \leftarrow Zu_i$
9 **Return** Ψ the matrix of principal components

3.2.2 kernel-PCA

While PCA is a dimension reduction technique that assumed to find linear transformation to represent the data in a lower dimension, kernel-PCA is used when we deal with complex structure data where linear subspace is not very useful [27]. In this paper, the kernel-PCA is used as an alternative to PCA when there is no linear correlation between features, which can affect classification accuracy.

Introduced as a nonlinear generalization of standard PCA [27], in the kernel PCA, the original input matrix $X_1, X_2, \dots, X_p \in R^n$ is mapped into a new feature space $\Phi(X_1), \Phi(X_2), \dots, \Phi(X_p) \in F$ and then the standard PCA is performed using this new feature space. However, computing $\Phi(X)$ explicitly before extracting the principal components is extremely costly [28]. The best practice is to directly construct a kernel matrix using X instead of computing $\Phi(X)$ explicitly [29]; thus, the mapping $\Phi(X)$ is implicitly specified by the kernel function. The most commonly used kernel function is the Radial Basis Function kernel (RBF) in Eq. (6)

$$k(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2) \tag{6}$$

Where:
$\parallel x_i - x_j \parallel^2$: denotes the squared Euclidean distance
$\gamma > 0$: a parameter that sets the spread of the kernel
d : the degree of the kernel.

If the new feature space is not centered, a centering transformation can be applied directly to the kernel matrix using Eq. (7) [30].

$$K_c = K - KI_{1/n} - I_{1/n}K + I_{1/n}KI_{1/n} \tag{7}$$

Where $I_{1/n}$ is the $n \times n$ matrix with all elements equal to $1/n$ and n is the number of patterns.

The overall pseudocode of the Kernel Principal Components Analysis method is illustrated in Algorithm 4.

Algorithm 4: KPCA algorithm

Input :-A p-diemsional data set $X = \{x_1, x_2, ..., x_n\}_{n\times 1}$ n is the number of patterns, x is the feature vector and p is the number of features
-Number of principal components d
Output: ψ the matrix of principal components
Compute $K_{(p\times p)}$ the kernel matrix using the Equation (6)
Centring transformation of K using the Equation (7)
$U \leftarrow TopEigenValues(K, d)$
$\lambda \leftarrow TopEigenVectors(K, d)$
for $i \leftarrow 1$ **to** n **do**
  for $k \leftarrow 1$ **to** d **do**
    $[\psi]_{k,i} \leftarrow \frac{1}{\sqrt{\lambda_k}} \sum_{t=1}^{n} K(x_i, x_t) U_{t,k}$
  end
end
Return ψ

4. CANCER IDENTIFICATION AND CLASSIFICATION METHODS

4.1 C5.0 Decision Tree

C5.0 is a new decision tree algorithm developed from C4.5 by [31], which has proven its high detection accuracy in many research fields [32-34]. Compared to C4.5, C5.0 can handle different types of data, deal with missing values and support boosting to improve classifier accuracy [35]. In C5.0 algorithm, samples are split into sub-samples by using a recursive method based on information gain ratios. Each sub-sample received from the first split will be split again. The split process is repeated until there is no more split that makes a difference in terms of information gain ratios. At the end of the process, any split which doesn't have a significant contribution to the model is rejected [36].

4.2 Support Vector Machine

The Support Vector Machine (SVM) is a binary classifier algorithm that has been successfully applied in many pattern recognition areas. In linear classification, SVM constructs a classification hyper-plane that separates the data into two sets by maximizing the margins and minimizing the classification error. The hyper-plane is constructed in the middle of the maximum margin. Thus, samples above the hyper-plane are classified as positives. Otherwise, they are classified as negatives (Fig. 5). The classification function is given with Eq. (8) [37].

$$y = sign(\sum_{i=1}^{n} w_i x_i + b) \tag{8}$$

Where y denotes the class label, w and b are the parameters of the hyper-plane, and $sign$ denotes the sign function.

However, in a real classification problem, datasets are often linearly non-separable. Therefore, Eq. (8) will allow some of the samples to be on the wrong side of the hyper-plane. To overcome this problem of non-linearity, a nonlinear transformation of the input vectors into a new feature space is performed, and then a linear separation is performed using this new feature space [37]. To perform a nonlinear SVM, the product (x, y) is replaced by a kernel function (Eq. (9)).

$$y = sign(\sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b) \tag{9}$$

https://doi.org/10.31436/iiumej.v22i1.1447

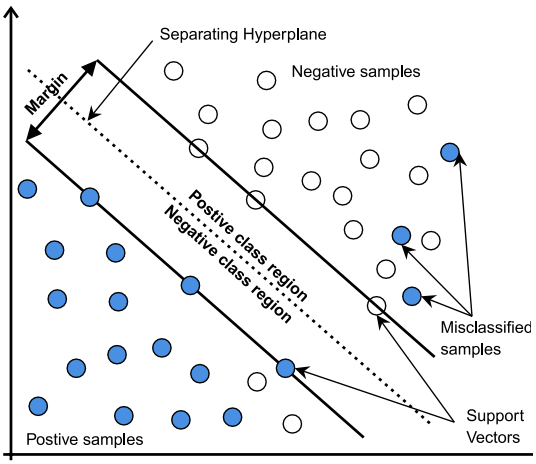


Fig. 5: Support Vector Machines diagram

In this paper a Gaussian kernel (Eq. (6)) was used to deal with the problem of non-linearity.

4.3 Artificial Neural Network

Introduced by [38], the Artificial neural network (ANN) is a form of distributed computation inspired by networks of human biological neurons. As shown in Fig. 6(a), An ANN consists of a set of interconnected artificial neurons that are organized in a minimum of three layers: the input layer, hidden layer, and output layer. All nodes (neurons) in each layer of the network are connected to the nodes of the next layer with no connection back, and all the connections are defined by weight values denoted by w. In the input layer, all nodes get information from the outside and pass it to the nodes of the next layer weighted by w. If we take a look at one of the hidden or output neurons (Fig. 6 (b)), we find that each node computes the weighting sum of all the N neurons of the previous layer and passes it through an activation function [39]. Equation (10) represents the equation for a given neuron.

$$z_j = f\left(\sum_{i=1}^{N} w_{ij}x_i + bj\right) \tag{10}$$

A Common choice for the activation function is non-linear functions such as the logistic sigmoid function given by the Eq. (11).

$$f(x) = \frac{1}{1+e^{-x}} \quad with \quad 0 \leqslant f(x) \leqslant 1 \tag{11}$$

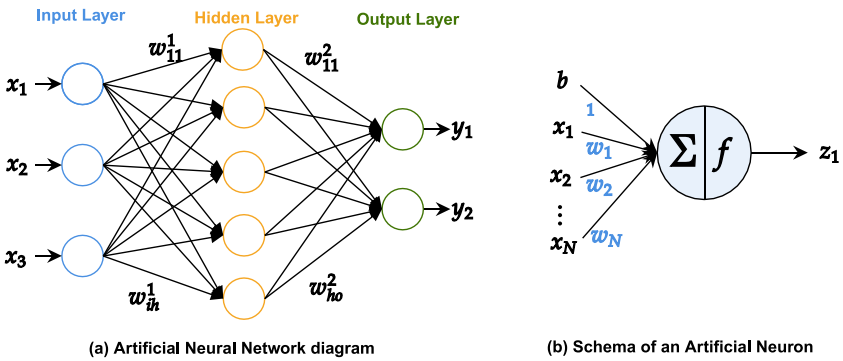


Fig. 6: Artificial Neural Network diagram.

4.4 Logistic Regression

As an extension of the linear regression algorithm for classification problems, Logistic Regression aims to find the best fitting model, which squeezes the output of a linear equation between 0 and 1 using the logistic function (Eq. (11)). In linear regression, the relationship between output and features is modeled using a linear equation (Eq. (12)). However, in a classification problem, it is strongly recommended to have probabilities between 0 and 1, which can force the outcome to be only between 0 and 1 (Eq. (13)).

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n \tag{12}$$

$$f(x) = \frac{1}{1+e^{-(b_0+b_1 x_1+b_2 x_2+\ldots+b_n x_n)}} \tag{13}$$

4.5 Performance Evaluation

The datamining process has several ways to check the performance of any classification model. The quality of any classification model is built from the confusion matrix (Table 1), which summarizes the comparison between predicted and observed classes for all observations.

Table 1: Confusion matrix

		Predicted classes	
		True	False
Observed classes	True	True positives (TP)	True negatives (TN)
	False	False positives (FP)	False negatives (FN)

Different types of evaluation measure are available, and the most commonly used in practice is the classification accuracy (Eq. (14)), which evaluates the classification performance by the percentage of correct predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

Another common evaluation metric used in Machine Learning is the receiver operating characteristic (ROC) curve which is created by plotting the True Positive Rate ($TPR = TP/(TP + FN)$) against the False Positive Rate ($FPR = FP/(FP + TN)$). The Area Under the ROC Curve (AUC) provides a good idea about model performance. The model that gives 100% of correct predictions has an AUC of 1, while the model that gives 100% of wrong predictions has an AUC of 0.

In the present paper, both accuracy and ROC curve were used to evaluate the performance of each generated model.

5. EXPERIMENTAL RESULTS

5.1 Dataset Description

In order to evaluate the ability of our framework to adapt to different situations, experiments are achieved on several public high-dimensional microarray datasets with different properties (number of genes, number of patterns, and the number of classes). Description of the datasets used in the present work is provided in Table 2.

Table 2: Microarray Datasets Characteristics

Dataset	Description	Genes	Patterns	Classes	Missing Values	Ref.
Prostate	Is the development of cancer in the prostate, a gland in the male reproductive system	10509	102	2	No	[40]
Leukemia1	It is one of the most common childhood cancers that affect leukocytes, but it most often occurs in older adults.	5327	72	3	No	[41]
SRBCT	Small-round-blue-cell tumors are four different childhood tumors; Ewing's family of tumors (EWS), neuroblastoma (NB), non-Hodgkin lymphoma and rhabdomyosarcoma (RMS)	2308	83	4	No	[42]
Lung	It is a malignant tumor caused by uncontrolled cell growth in lung tissues	12600	203	5	No	[43]

5.2 Partitioning

In order to avoid the overestimating prediction, a stratified $5-fold$ cross-validation technique was employed. Using this technique, samples are split into five equal folds (subset) of samples. One of the five folds is used as a testing step, and the remaining four folds are put together to form the training data. This process is repeated five times. The stratification process was used to ensure that all folds are made by preserving the same percentage of samples for each class.

5.3 Data Preprocessing

Before supplying the datasets to our analysis system, it was necessary to perform data preprocessing, as it is an important step in the data analysis process. In the present paper, gene expression datasets were preprocessed using the standard procedure, which includes log transformation and standardization.

a) Data Transformation

The main motivation for using the log transformation is due to the asymmetric distribution of the derived expression levels [44], which can affect the identification of expression Patterns, and prospective Classification in Human Cancer Genomes [45].

In the present work, before transforming our data (training or testing set) using Eq. (15), a test of normality using Shapiro-Wilk [46,47] is used to evaluate whether the distribution of the data agrees with normal distribution. The calculated P-value of the Shapiro–Wilk test for each gene shows a strong significance, which indicates a deviation from the normal distribution for most of the gene expressions.

$$X = Log_{10}(X - 1 + Min(X)) \tag{15}$$

b) Data Standardization

Gene expression levels for each gene were standardized using the Eq. (16). The result is that expression levels for each feature have a mean 0 and variance 1.

$$X = \frac{X-\overline{X}}{\sigma} \tag{16}$$

Where: $\overline{X}$ is the overall mean of the feature X and σ, its standard deviation.

5.4 Experimental Settings

a) System Configuration

Using parallel processing, our proposed framework was implemented in Python 3.7 language. All of the experiments were carried out using an Intel Xeon E5-2637 v2 3.5 GHz PC with 64 GB of RAM.

b) Parameters Settings

The parameters used in our Framework are shown in Table 3:

Table 3: Parameters settings used in our proposed Framework

Parameter	Value
PACO parameters	
Number of ants	10
Number of selected features	Initialized using the filter method
α	1
β	5
Q	100
ρ	0.5
SVM Kernel	RBF
PCA parameters	
α	0.8
Kernel-PCA parameters	
d	Initialized using standard PCA
Kernel	RBF
ANN classifier parameters	
The number of hidden nodes	Calculated by using Geometric-Pyramid-Rule [48]
The number of hidden layers	1
Activation function	Logistic
Solver (optimization algorithm)	adam
Max iterations	1000
SVM classifier parameters	
Kernel	RBF

6. RESULTS AND DISCUSSION

Table 4, which is also presented as graphs in Fig. 7, shows the overall performance of dimension reduction techniques and their respective classification algorithms used in the four public microarray datasets described in Table 2. 36 different models were generated for each microarray dataset. The quality of each model is measured by the number of selected genes k, the dimension of the new subspace k'generated by using the FE process, the running time t (the running time reported here includes both the dimension reduction and classification stages), and the cancer prediction performance which represents the average accuracy of the training and testing sets. The results of each dataset are as follows:

For the SRBCT dataset, Table 4 and Fig. 7 show that the shrinkage models P-PCA-SVM, P-PCA-ANN, and P-PCA-LR provide an excellent accuracy of **100%**. The power of these models resides in the fact that the number of genes was reduced twice. The first reduction was obtained using Pearson correlation-based feature selection (Line 9 in Algorithm 1), the dimension of dataset passed from $p = 2308$ (the original number of genes in the dataset as reported in Table. 2) to a new subset of $k = 727$ while selecting only the

more representative genes with relevance greater than 95%. Then, the new k-dimension subset was transformed to a linear subspace using PCA where only the first $k' = 22$ predictors (components) that explain approximately 80% of the total variation of genes subset (the cumulative variance equal to 80%) were retained. The same results were achieved by P-KPCA-SVM, P-KPCA-ANN, and P-KPCA-LR models with the nonlinear transformation (KPCA) for the feature extraction method. The classification rate was close to **100%** for most shrinkage models using the PACO-based feature selection method with a more significant response time increase. Fortunately, there is almost no significant accuracy loss for the rest of the generated models.

Table 4: Performance measurement using our Framework on the four datasets

Feature Selection	Feature Extraction	Model	Datasets																		
			SRBCT					Lung					Prostate					Leukemia1				
			p	k	k'	t(s)	Accuracy (%)	p	k	k'	t(s)	Accuracy (%)	p	k	k'	t(s)	Accuracy (%)	p	k	k'	t(s)	Accuracy (%)
Pristine	Pristine	SVM	2308		2308	1	100	12600		12600	2	95.42	10509		10509	1	93.17	5327		5327	1	91.46
		LR			2308	1	100			12600	12	98.75			10509	1	97.62			5327	1	100
		ANN			2308	6	100			12600	60	98.75			10509	27	97.62			5327	13	100
		C50			3	1	96.45			4	21	96.74			2	6	94			2	4	95.57
	PCA	PCA-SVM			24	1	100			53	2	95.61			15	1	85.79			30	1	93.61
		PCA-LR			24	1	100			53	2	97.13			15	1	88.29			30	1	100
		PCA-ANN			24	1	100			53	2	97.44			17	1	93.39			30	1	100
		PCA-C50			6	1	92.6			8	1	96.03			4	1	88.78			5	1	89.77
	KPCA	KPCA-SVM			24	1	100			53	1	92.36			15	1	85.79			30	1	92.46
		KPCA-LR			24	1	96.3			54	1	80.44			15	1	85.18			28	1	80.53
		KPCA-ANN			24	1	92.6			53	2	81.82			16	1	53.05			30	1	86.7
		KPCA-C50			5	1	89.26			11	1	87.52			6	1	90.67			6	1	84.91
Pearson	Pearson	P-SVM	2308	727	727	1	100	12600	4566	4566	6	96	10509	2824	2824	3	97	5327	1120	1120	4	100
		P-LR		727	727	1	100		4534	4534	13	97.5		2640	2640	6	97.5		985	985	3	100
		P-ANN		727	727	3	100		4534	4534	15	98.75		2640	2640	8	97.5		985	985	5	100
		P-C50		727	4	2	96.3		4679	4	10	96.74		2824	2	5	91.01		988	3	3	96.67
	PCA	P-PCA-SVM		727	22	1	100		4264	59	4	97.5		2824	17	3	95.15		1035	26	2	100
		P-PCA-LR		727	22	1	100		4534	57	8	97.5		1801	18	4	94.62		1035	26	2	100
		P-PCA-ANN		727	22	2	100		4446	59	7	98.75		2824	17	4	94		1035	26	3	100
		P-PCA-C50		756	5	1	96.45		4566	7	6	92.67		2824	6	3	97		1035	3	2	100
	KPCA	P-KPCA-SVM		727	22	1	100		4566	60	5	95.72		2640	17	5	95.06		985	28	3	100
		P-KPCA-LR		727	22	1	99.24		4446	59	6	84.79		1801	18	4	93.39		1035	26	2	88.07
		P-KPCA-ANN		756	22	2	99.23		4566	60	6	79.8		1801	18	4	90.39		1035	26	3	92.46
		P-KPCA-C50		756	3	1	93.68		4534	9	7	92.19		2640	4	5	91.34		1041	3	2	93.33
PACO	PACO	PACO-SVM		727	727	214	100		4566	4566	4502	95.42		2640	2640	3298	93.17		1120	1120	1081	92.31
		PACO-LR		727	727	214	100		4534	4534	4705	98.75		1801	1801	3591	97.62		985	985	1290	100
		PACO-ANN		727	727	215	100		4534	4534	4709	99		2640	2640	3301	97.5		985	985	1291	100
		PACO-C50		750	4	156	96.13		4446	4	6366	96.27		1801	4	3591	95.24		1035	3	1163	100
	PCA	PACO-PCA-SVM		727	23	214	100		4264	52	6241	95.61		2640	15	3298	90.12		1035	28	1163	95.79
		PACO-PCA-LR		727	23	214	100		4264	52	6241	97.13		1801	17	3591	91.53		985	30	1289	100
		PACO-PCA-ANN		727	23	215	100		4566	52	4503	97.56		2824	16	3620	92.15		988	30	1157	96.67
		PACO-PCA-C50		750	5	156	92.26		4264	8	6241	96.52		1801	6	3591	88.62		985	3	1289	91.13
	KPCA	PACO-KPCA-SVM		727	23	214	100		4264	52	6240	94.63		2640	15	3298	87.62		1120	30	1081	96.15
		PACO-KPCA-LR		727	23	214	96.3		4446	54	6362	80.75		3879	16	3456	87.68		988	30	1157	80.53
		PACO-KPCA-ANN		756	23	192	97.22		4264	52	6241	84.26		2824	16	3620	55.82		985	30	1290	89.29
		PACO-KPCA-C50		847	6	158	90		4446	11	6362	87.22		3879	5	3456	93.17		1120	5	1081	87.61

https://doi.org/10.31436/iiumej.v22i1.1447

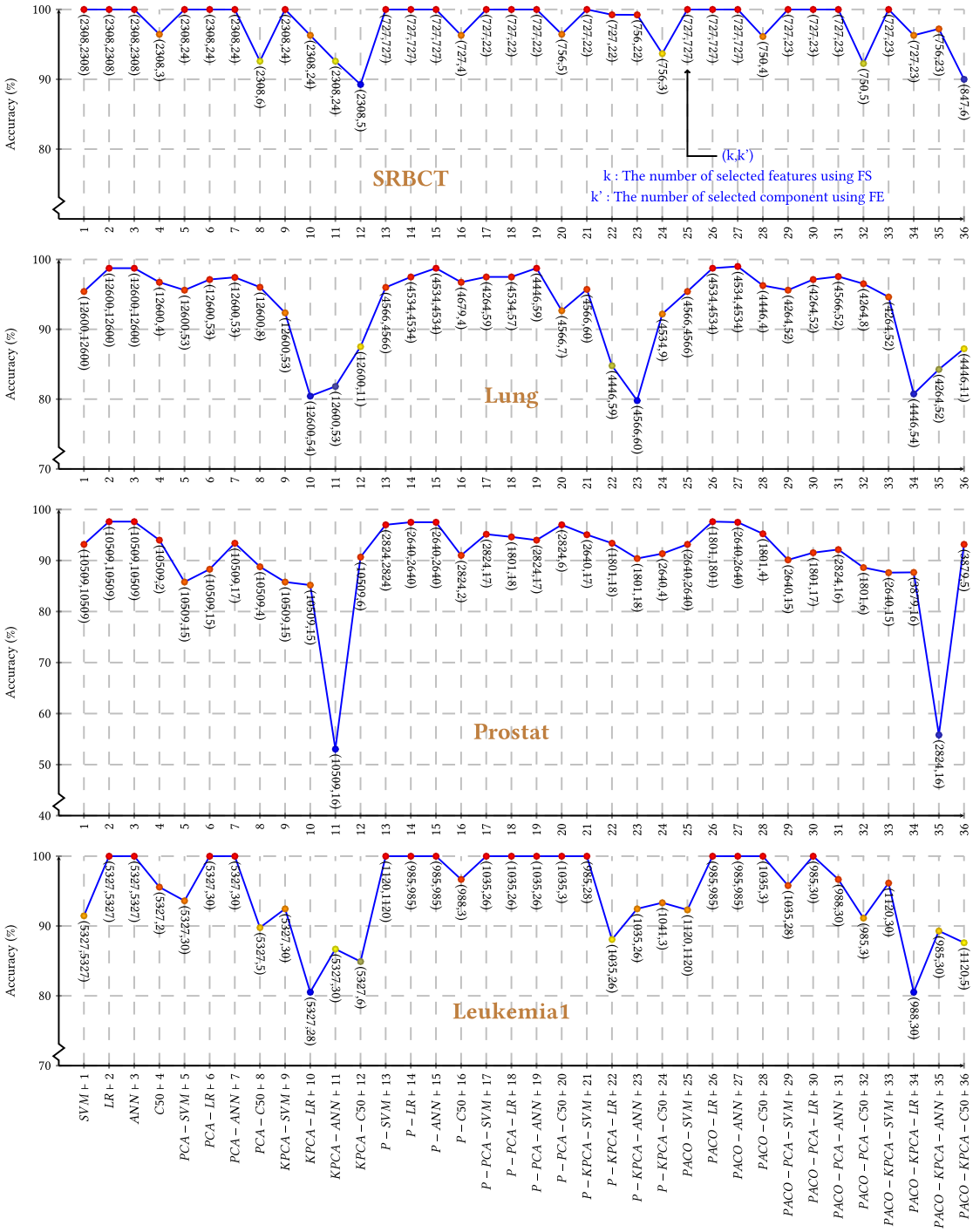


Fig. 7: Performance measurement using our Framework on the four datasets.

For the Lung dataset, The PACO-ANN model achieved the highest classification performance of 99**%** over the entire set of generated models while using only 36% of genes ($k = 4534$ most significant genes) from the original pre-processed dataset by using PACO-based feature selection (Line 13 in Algorithm 1). As there is no feature extraction process in this model, the selected subset of k genes PACO-based was used as an input layer for the ANN classifier, thus $k = k' = 4534$. In contrast, this model took about 4709 seconds, which is considered to be a significant response time compared to the other generated models.

For the Prostate dataset, the best performance was achieved by the PACO-LR model since the average classification accuracy of LR (Logistic Regression algorithm) reached 97.62% when involving only $k = 1801$ genes from $p = 10509$ by using the PACO feature selection method. The same average accuracy was achieved (97.62%) when applying LR and ANN on the p original genes without any dimensionality reduction process.

For the Leukemia1 dataset**,** with the different values of k and k', almost all generated models produced high average classification performance close to 100%.

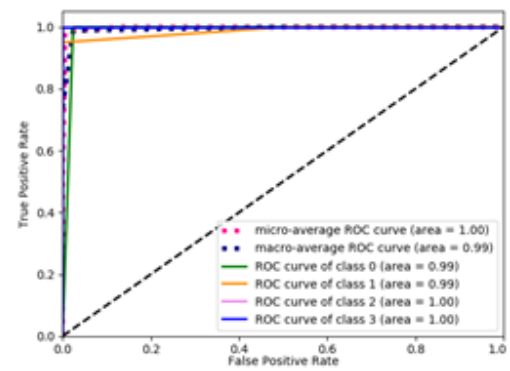
The power of our framework resides in the fact that it proposes a large number of models that combine different dimensionality reduction techniques with the classification process. The major aim behind this combination is to use a bare minimum of dimensions while maximizing the classification performance. Regarding Table 5, the most interesting result concerned the Leukemia1 dataset since the best model, P-PCA-C5.0, achieved an excellent accuracy of 100% with only $k' = 3$ selected dimensions and a response time of 2 seconds, which is much better than what was reported in [11-12], [14-15]. The classification accuracy of 100% was obtained by computing the percentage of correct predictions from the confusion matrix (c) shown in Fig. 9. The strongest point of the P-PCA-C5.0 model resides in the fact that the original number of genes $p = 5327$ was reduced three times to arrive at $k' = 3$ dimensions. The first reduction was obtained by selecting only the $k = 1035$ most relevant genes by using Pearson correlation-based feature selection. Then, using the PCA based FE, the new subset of k genes was converted into a new subspace of 26 dimensions (components), which in turn reduced into $k' = 3$ dimensions by using the innate feature selection capacity of C5.0 algorithm [49,50]. The quality of the P-PCA-C5.0 model in terms of classifications performance was validated by the area under the ROC curve (AUC). As we can notice from ROC curves (g) and (h) drawn in Fig. 8, the average AUC of testing and training set shows a maximum value of $AUC = 1$ which can confirm the quality of our favorite model. With a significant increase of consumed time, PACO-C5.0 model achieved exactly the same results (in terms of classification accuracy and dimension reduction degree), except that the $k' = 3$ in this model represents the numbers of genes instead of dimensions (components) obtained by P-PCA-C5.0.

Table 5: Table summarizing our favorite models for each gene expression dataset

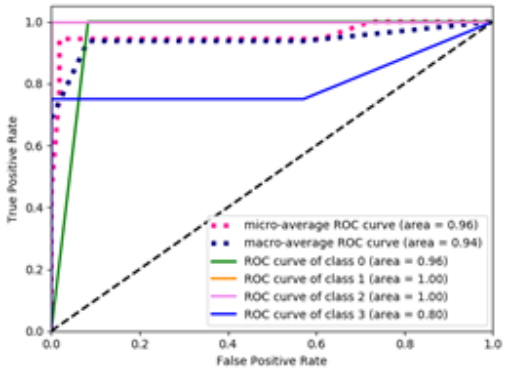
Dataset	Model	p	k	k'	Accuracy (%)	$AUC = \frac{AUC_{Train} + AUC_{Test}}{2}$	t(s)
SRBCT	C5.0	2308	2308	3	96.45	0.98	1
Lung	P-C5.0	12600	4679	4	96.74	0.985	10
Prostate	C5.0	10509	10509	2	94	0.935	6
Leukemia1	P-PCA-C5.0	5327	1035	3	100	1	2
	PACO-C5.0	5327	1035	3	100	1	1163

For the Prostate dataset, the power of the C5.0 algorithm in terms of FS and classification performance was enough to achieve the best result compared to the other generated models. As we can notice from confusion matrix (d) in Fig. 9, 4 out of 102 samples are incorrectly classified by the C5.0 model, resulting an average accuracy of

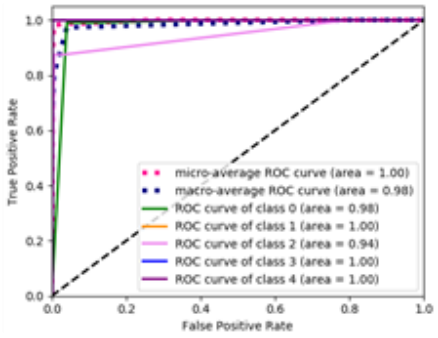
https://doi.org/10.31436/iiumej.v22i1.1447



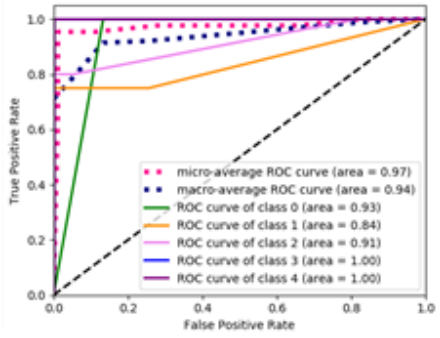
(a) : Performance of C5.0 model on the training set of SRBCT



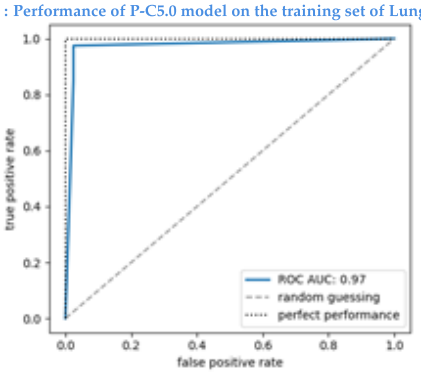
(b) : Performance of C5.0 model on the testing set of SRBCT



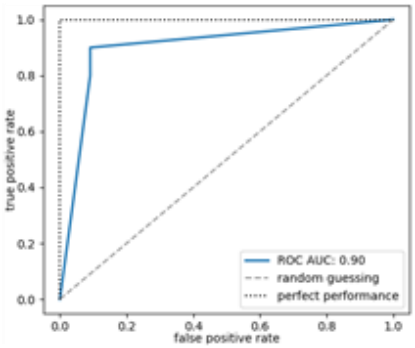
(c) : Performance of P-C5.0 model on the training set of Lung



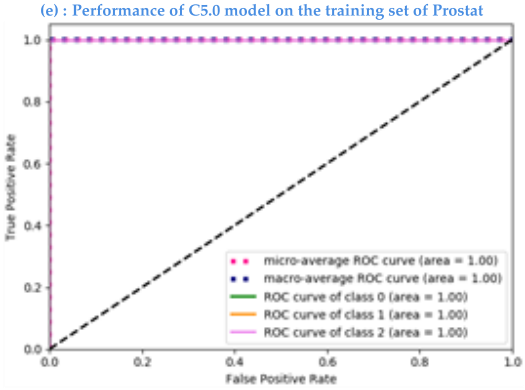
(d) : Performance of P-C5.0 model on the testing set of Lung



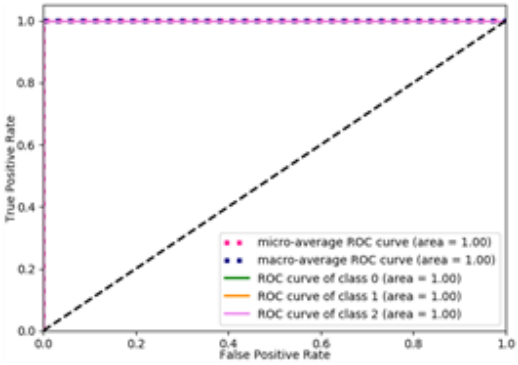
(e) : Performance of C5.0 model on the training set of Prostat



(f) : Performance of C5.0 model on the testing set of Prostat



(g) : Performance of P-PCA-C5.0 model on the training set of Leukemia1



(h) : Performance of P-PCA-C5.0 model on the testing set of Leukemia1 dataset

Fig. 8: ROC curves showing the performance of our favorite model on the training and testing sets of each dataset.

94% with only $k' = 2$ genes such as reported in Table 5, which is better than what is reported in [10]. The average AUC of 0.935 obtained from the ROC curves (e) and (f) confirms the quality of our model in terms of classification performance.

https://doi.org/10.31436/iiumej.v22i1.1447

For the Lung dataset, as we can notice from Table 5, the **P-C5.0** model achieved a classification accuracy of 96.74% (calculated from confusion matrix (b) in Fig. 9) by involving only 4 genes. This model benefitted from both Pearson correlation-based feature selection to reduce the number of genes from $p = 12600$ to $= 4679$, and from the innate power of C5.0 to reduce a second time the number of genes from k to $k' = 4$. The average AUC of 0.99 could validate the choice of our model.

For the SRBCT dataset, our favorite model, C5.0, achieved almost the same accuracy on the Prostate dataset using only $k' = 3$ genes from $p = 2308$.

According to the results reported in Table 4 and Table 5, our framework could improve both the accuracy and degree of dimensionality reduction compared with state-of-the-art methods.

Fig. 9: Confusion Matrix of our favorite models for each dataset. (a) C5.0 model for SRBCT dataset; P-C5.0 model for Lung cancer dataset; (c) P-PCA-C5.0 and PACO-C5.0 models for Leukemia1 dataset; (d) P-PCA-C5.0 and PACO-C5.0 models for Prostate dataset.

7. CONCLUSIONS AND FUTURE WORK

Genes expression data analysis is challenging the conventional prediction techniques, since limited labeled samples versus a large number of genes may significantly affect the classification performance. To overcome this issue, a new generic approach combining dimensional reduction techniques with machine learning algorithms was proposed. The main objective behind this approach is to improve prediction performance for microarray datasets while involving a bare minimum number of predictors. The dimensional reduction process used in this paper is a combination of FS and FE techniques. The FS using Pearson correlation or PACO aims at selecting the most relevant genes, while FE using PCA or kernel-PCA aims at transforming the original genes space into a new linear or non-linear subspace. Dimensional reduction techniques were combined with four classifiers SVM, ANN, LR, and C5.0. We conducted the experiments on four public microarray gene expression data sets, SRBCT, leukemia1, lung, and Prostate cancer. Experimental results

show that the number of genes was efficiently reduced to reach two genes, with a high classification accuracy that reached up to 100% (Table 5), making our framework very effectively competitive with the reference approaches. Moreover, our experiment confirms that our coupling of dimensionality reduction with classification makes our framework powerful in terms of its ability to adapt with different kinds of microarray datasets.

Our future work includes experimentation of our proposed approach on new gene expression datasets, and study of new data mining techniques that can enhance our framework in many different aspects in the aim of identifying, with high performance, previously unknown cancer-related genes, which may guide further cancer research.

REFERENCES

[1] Gheyas, I., & Smith, L. (2010). Feature subset selection in large dimensionality domains. Pattern Recognition, 43, 5–13.
[2] Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433–459.
[3] Liu, Z., Chen, D., & Bensmail, H. (2005). Gene Expression Data Classification With Kernel Principal Component Analysis. Journal of Biomedicine and Biotechnology, 2005, 155–159.
[4] Biesiada, J., & Wlodzislaw, D. (2007). Feature Selection for High-Dimensional Data—A Pearson Redundancy Based Filter. In Advances in Soft Computing (Vol. 45, pp. 242–249).
[5] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial Intelligence, 97(1), 273–324.
[6] Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. Proceedings, Twentieth International Conference on Machine Learning, 2, 856–863.
[7] Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, 13, 8–17.
[8] Sayed, S., Nassef, M., Badr, A., & Farag, I. (2019). A Nested Genetic Algorithm for feature selection in high-dimensional cancer Microarray datasets. Expert Systems with Applications, 121, 233–243.
[9] Ghosh, M., Begum, S., Sarkar, R., Chakraborty, D., & Maulik, U. (2019). Recursive Memetic Algorithm for gene selection in microarray data. Expert Systems with Applications, 116, 172–185.
[10] Moutachaouik, H., & El Moudden, I. (2018). Mining Prostate Cancer Behavior Using Parsimonious Factors and Shrinkage Methods.
[11] Kar, S., Sharma, K. D., & Maitra, M. (2016). A particle swarm optimization based gene identification technique for classification of cancer subgroups. 2016 2nd International Conference on Control, Instrumentation, Energy Communication (CIEC), 130–134. https://doi.org/10.1109/CIEC.2016.7513800
[12] Mortazavi, A., & Hossein Moattar, M. (2016). Robust Feature Selection from Microarray Data Based on Cooperative Game Theory and Qualitative Mutual Information. Advances in Bioinformatics, 2016, 1–16. https://doi.org/10.1155/2016/1058305
[13] Chandra, B., & Gupta, M. (2011). An efficient statistical feature selection approach for classification of gene expression data. Journal of Biomedical Informatics, 44, 529–535. https://doi.org/10.1016/j.jbi.2011.01.001
[14] Chandra, B. (2018). An efficient feature selection technique for gene expression data. 1–6. https://doi.org/10.1109/CIBCB.2018.8404977
[15] Guo, S., Guo, D., Chen, L., & Jiang, Q. (2016). A Centroid-based Gene Selection Method for Microarray Data Classification. Journal of Theoretical Biology, 400. https://doi.org/10.1016/j.jtbi.2016.03.034

[16] Chu, C., Hsu, A.-L., Chou, K.-H., Bandettini, P., & Lin, C. (2012). Does feature selection improve classification accuracy? Impact of sample size and feature selection on classification using anatomical magnetic resonance images. NeuroImage, 60(1), 59–70.
[17] Ang, J., Mirzal, A., Haron, H., & Hamed, H. (2016). Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 13(05), 971–989. https://doi.org/10.1109/TCBB.2015.2478454
[18] Dorigo, Marco. (1992). Optimization, Learning and Natural Algorithms [PhD Thesis]. Politecnico di Milano.
[19] Bullnheimer, B., Hartl, R., & Strauss, C. (1999). A New Rank Based Version of the Ant System—A Computational Study. Central European Journal of Operations Research, 7, 25–38.
[20] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), 29–41.
[21] Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation, 6(4), 321–332.
[22] Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed Stigmergetic Control for Communications Networks. J. Artif. Int. Res., 9(1), 317–365.
[23] Aldryan, D. P., Adiwijaya, & Annisa, A. (2018). Cancer Detection Based on Microarray Data Classification with Ant Colony Optimization and Modified Backpropagation Conjugate Gradient Polak-Ribiére. 2018 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), 13–16. https://doi.org/10.1109/IC3INA.2018.8629506
[24] Wichaidit, S., Wardkean, P., Chaiwong, K., & Wettayaprasit, W. (2012). New hybrid adaptive Ant Colony Optimizaion and Self-Organizing Map for DNA microarray group finding. 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), 3, 444–447.
[25] Yu-Min Chiang, Huei-Min Chiang, & Shang-Yi Lin. (2008). The application of ant colony optimization for gene selection in microarray-based cancer classification. 2008 International Conference on Machine Learning and Cybernetics, 7, 4001–4006.
[26] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. Phil. Trans. R. Soc. A, 374(2065), 20150202.
[27] Schölkopf, B., & Smola, A. J. (2001). Smola, A.: Learning with Kernels—Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge, MA. In Journal of The American Statistical Association—J AMER STATIST ASSN (Vol. 98).
[28] Wang, Q. (2012). Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models.
[29] Weinberger, K., Sha, F., & K. Saul, L. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. https://doi.org/10.1145/1015330.1015345
[30] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
[31] R, Q. J. (2007). C5. http://rulequest.com
[32] Bujlow, T., Riaz, T., & Pedersen, J. M. (2012). A method for classification of network traffic based on C5.0 Machine Learning Algorithm. 2012 International Conference on Computing, Networking and Communications (ICNC), 237–241. https://doi.org/10.1109/ICCNC.2012.6167418
[33] Ranjbar, S., Aghamohammadi, M., & Haghjoo, F. (2016). Determining Wide Area Damping Control Signal (WADCS) based on C5.0 classifier.
[34] Agaoglu, M. (2016). Predicting Instructor Performance Using Data Mining Techniques in Higher Education. IEEE Access, 4, 1–1. https://doi.org/10.1109/ACCESS.2016.2568756
[35] Rathinasamy, R., & Raj, L. (2019). Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data. International Journal of Innovative Research in Computer and Communication Engineering, 5, 2017.
[36] Elsalamony, H., & Elsayad, A. (2013). Bank Direct Marketing Based on Neural Network. International Journal of Engineering and Advanced Technology, 2, 392–400.

[37] Marjanović, M., Kovačević, M., Bajat, B., & Voženílek, V. (2011). Landslide susceptibility assessment using SVM machine learning algorithm. Engineering Geology - ENG GEOL, 123, 225–234.
[38] McCulloch, W. S., & Pitts, W. H. (1988). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology, 52, 99–115.
[39] Cachim, P. (2011). Using artificial neural networks for calculation of temperatures in timber under fire loading. Construction and Building Materials - CONSTR BUILD MATER, 25, 4175–4180. https://doi.org/10.1016/j.conbuildmat.2011.04.054
[40] Singh, D., Febbo, P., Ross, K., G Jackson, D., Manola, J., Ladd, C., Tamayo, P., A Renshaw, A., V D'Amico, A., P Richie, J., S Lander, E., Loda, M., Kantoff, P., R Golub, T., & Sellers, W. (2002). Gene Expression Correlates of Clinical Prostate Cancer Behavior. Cancer Cell, 1, 203–209. https://doi.org/10.1016/S1535-6108(02)00030-2
[41] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M., Downing, J. R., Caligiuri, M., D Bloomfield, C., & S Lander, E. (1999). Molecular classification of cancer: Class discovery and class prediction by gene monitoring. Science (New York, N.Y.), 286, 531–537.
[42] Westermann, F., Wei, J. S., Ringner, M., Saal, L., Berthold, F., Schwab, M., Peterson, C., Meltzer, P., & Khan, J. (2002). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. GBM Annual Fall Meeting Halle 2002, 2002. https://doi.org/10.1240/sav_gbm_2002_h_000061
[43] J Gordon, G., Jensen, R., Hsiao, L.-L., R Gullans, S., Blumenstock, J., Ramaswamy, S., G Richards, W., Sugarbaker, D., & Bueno, R. (2002). Translation of Microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research, 62, 4963–4967.
[44] Li, W., Suh, Y. J., & Zhang, J. (2006). Does Logarithm Transformation of Microarray Data Affect Ranking Order of Differentially Expressed Genes? Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, Suppl, 6593–6596. https://doi.org/10.1109/IEMBS.2006.260896
[45] Marko, N., & Weil, R. (2012). Non-Gaussian Distributions Affect Identification of Expression Patterns, Functional Annotation, and Prospective Classification in Human Cancer Genomes. PloS One, 7, e46935. https://doi.org/10.1371/journal.pone.0046935
[46] Shapiro, S. S., & Wilk, M. B. (1965). An Analysis of Variance Test for Normality (Complete Samples). Biometrika, 52(3/4), 591–611.
[47] Royston, J. P. (1982). An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. Applied Statistics, 31(2), 115. https://doi.org/10.2307/2347973.
[48] Masters, T. (1993). Practical Neural Network Recipes in C++. Academic Press Professional, Inc.
[49] Deng, L., Yan, Y., & Wang, C. (2015). Improved POLSAR Image Classification by the Use of Multi-Feature Combination. Remote Sensing, 7, 4157–4177.
[50] McIver, D. K., & Friedl, M. A. (2002). Using Prior Probabilities in Decision-Tree Classification of Remotely Sensed Data. Remote Sensing of Environment, 81, 253–261.