Efficient Feature Selection Algorithm for Gene Classification

Narayan Naik[1,2(✉)] and Y. H. Sharath Kumar[1,2]

[1] Canara Engineering College, Mangaluru, India
naik.mtech09@gmail.com
[2] Maharaja Institute of Technology Mysore, Srirangapatna, India

Abstract. Microarray technology was evolved as one of the authoritative mechanisms for an organism to analysis of gene expression level. The microarray gene expression datasets contain a considerably large number (in terms of thousands) of features (genes) and a comparatively small number (in terms of hundreds) of samples. Because of these characteristics, microarray gene expression data analysis is complex. Therefore, efficient feature selection is the immediate requirement. The essential aspects of microarray gene expression data analysis are feature selection and classification. Although many feature selection methods were developed, the SVM, along with recursive component reduced termed as SVM-RFE, was tested to be a promising method. The genes are ranked during SVM classification model training, and critical features are selected with a combination of recursive feature elimination (RFE). The SVM-RFE main drawback was a significant amount of time consumption in the process. Therefore, efficient deployment of linear Support Vector Machine was introduced to overcome this issue. At the same time, Recursive Feature Elimination (RFE) was improvised with the technique known as the variable step size. Along with this, an effective resampling technique was proposed to preprocess the datasets in order to overcome the class imbalance problem. By using this method, the sample became balance from the same distribution that provides better classification result. The recursive feature elimination with variable step size (RFEVSS) with an effective resampling method was used in order to achieve better performance of the classifier that has been presented in this work. The class imbalance problem was addressed by implementation the effective resampling method described in this work. The large-scale linear support vector machine (LLSVM) has also been implemented effectively in order to increase efficiency. The detailed experiments were conducted to test the result with three classifiers on four benchmark microarray gene expression datasets. The results were presented in graphical form for better understanding.

1 Introduction

Microarray technology is one of the benchmark tools that have attracted many researchers for the study of the level of gene expression. This technology can imitate the transcriptome level of an organism's physiological status and gene activities. Cancer has been treated as one of the most dangerous diseases for medical science across the globe, but it can be controlled and treated by medical science if identified in an

D. S. Guru et al. (Eds.): ICCR 2021, CCIS 1697, pp. 165–189, 2022.
https://doi.org/10.1007/978-3-031-22405-8_14

early stage. Typically, microarray datasets contain a large number of features along with a small number of samples and as well as noise (Hira and Gillies [1]). Over the few decades, the microarray dataset's characteristics remain almost the same. A considerably large number of features, a small number of samples along with not proper balance class, are the prime features of microarray datasets, which are challenging issues that need to be addressed (Bolón-Canedo et al. [2]). The feature selection method is used to recognize the gene which are perfectly companion with particular disorder. Usually, the standard of feature selection is measured by evaluating the classification accuracy. Hence, classification is also an essential part of gene recognition. Mostly, the identification of disease in gene expression data is known as classification. Although a considerably huge number of features and a comparatively small number of samples of training data has been treated as the curse for classification, and the generalization capability of the classification model can be faulty (Elkhani and Muniyandi [3]). Taking into ac- count the properties of microarray gene expression datasets, such as large dimensions and comparatively lesser sized, reducing of breadth are very much essential during the classification. Usually, selection of features has been considered as one of the best approaches for dimensions reduction of gene expression data. Therefore, efficient feature selection along with suitable classifiers is essential for the diagnosis of a disease or gene identification of these datasets. On the other hand, classification results can be misleading because of the presence of lousy class imbalance (Chawla et al. [4]). So, an effective resampling technique is necessary for solving this problem.

Feature selection approaches attracted many researchers' attention in the last few years (Liu et al. [5]). Many techniques for feature selection were proposed for identifying the genes that are affected by disease (Ding and Peng [6]). Least square (LS) bound measure was proposed for solving the problem of several redundant genes (Zhou and Mao [7]). Many statistical approaches such as t-test, χ^2, information gain was used extensively as feature selection along with classical classifiers such as SVM (Saeys et al. [8]). These feature selections were classified as three types: Filter, Wrapper, and Embedded approach (Saeys et al. [8]).

The SVM provides a promising performance as a classifier since it has the inbuilt capability of feature selection. Many researchers have taken SVM as a prime interest for a long time. Guyon et al. [9] proposed a novel approach of feature selection as SVM-RFE. The capabilities of SVM were utilized for eliminating one feature recursively, which was the least significant in the position items until the left-out features meets requirements (Guyon et al. [9]). Further, this approach was considered quickly as a benchmark in the area of feature selection in the subsequent studies. However, the probable hidden correlation among features was not considered by SVM-RFE in the procedure of feature selection. This was taken as one of the limitations of SVM- RFE. The combined approach of mRMR with SVM-RFE was proposed as a hybrid method for the selection of important genes to solve this problem (Mundra and Rajapakse [10]). SVM-RFE was further modified and proposed as another variant, that works based on mutual information (Yoon and Kim [11]). SVM-RFE is extensively time-consuming; that is another drawback. For the process of increase attribute selection, Tang et al. [12] presented two phase SVM-RFE. The improvised version of Recursive Feature Elimination (RFE) was proposed, in which a quantity of features to be completely removed

keeps varying during every iteration (Ding and Wilkins [6]). Here, 1 is divided by j+1 of the leftover component was eliminated in the j^{th} repetition. Now, it could be said that the micro-array dataset has 25000 genes, out of the 12500 genes cab be eliminated. Then 4166 genes can be eliminated in initial and next iteration respectively, and so on, which has been found to be" too rude" for feature selection though this process guaranty better speed, but the quality of the feature selection process may be compromised with this type of procedure. Yin et al. [13] also put forward to better RFE. Those methods reduce time utilization, up to some extent, and obtained better performance. The main objectives were targeted in this work are to improve the speed and address the feature selection issue for improving the quality of feature selection. The RFEVSS method was proposed, which is the improved version of RFE in which step size keeps varying. The step size has been considered as the quantity of features to be completely removed in every reptation activity. That is, step size gets reduced when the quantity of features in the selection process gets reduced. Later, the former remains unchanged with one when it reaches a certain point. The systematic execution of large linear SVM was also proposed & used instead of SVM that has been combined with improvised RFE for further improvising the speed of feature selection.

The main challenges in microarray data analysis are a considerably huge number of features and small sample sizes, at the same time because of class imbalance case becomes worse. The vast difference between the samples belong to different classes is termed as a class imbalance. Unpredictable classification may be produced because of class imbalance. For example, suppose that test set has a sample of binary classes, which can be distributed the same as that, sample X is two times that of Y. When all samples estimated in the test dataset as X, next the accuracy is 66.67% that is greater than 50%. Therefore, it may be concluded; the class imbalance will affect the credibility of the classifier. Hence, to address these problems, many researchers have suggested re-sampling approaches (Chawla et al. [4], Zhu et al. [14], Galar et al. [15], Qian et al. [16]). There are two traditional re-sampling methods named as over and under re-sampling method. The running concept of current techniques sample are, selected randomly from minority classes, or samples are eliminated randomly from minority, then after replicated. However, this technique may result in loss of information or over-fitting (Yoon and Kim [11]). Zhu et al. [14] deployed Synthetic Minority Over-sampling Technique (SMOTE), that gives effective result. However, synthesize the value of generated samples was the main feature of SMOTE. Therefore, the SMOTE technique may not be suitable for microarray gene expression data, particularly when gene recognitions is the main objective. Later, ensemble approaches gained significant attention from many researchers for their competitive results (Galar et al. [17]). However, the complexity of ensemble techniques was on the higher side for microarray datasets since a small sample size was available. An effective re-sampling method was proposed in this work that select component value randomly in place of picking a unknown sample, & further new samples were constructed to overcome the class imbalance problem.

Classification is being treated as the core unit in order to analyze the microarray datasets. However, the classifiers were not built; perhaps the existing and proven classifier was used. In this work, the four most frequently used and benchmark microarray datasets have been considered, data were pre-processed with the help of the proposed re- sampling

method, and then the important features were selected with proposed feature selection method. Finally, three popular classifiers such as SVM, k-Nearest Neighbors, as well as Logistic Regression (Yu et al. [18]) were used to perform the classification task.

2 Proposed System

The figure 1 illustrates the proposed system architecture. Microarray gene expression datasets are the input data. Since the data may be inconstant and noisy, so it is pre-processed first. Then the balanced datasets are created with the use of the proposed resampling technique. Next, the proposed feature selection method is used for selecting important features (genes). Finally, efficiency and effectiveness are measured with the application of different classifiers.

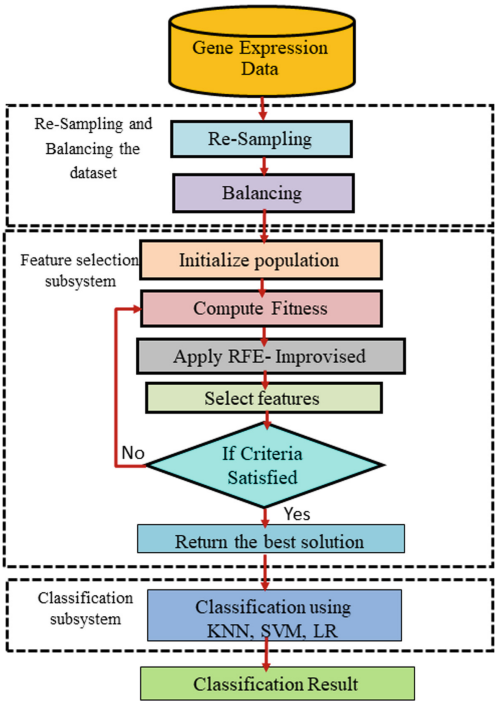

Fig. 1. System architecture of proposed method

The proposed method is broadly divided into 3 components:

i. Re-sampling and balancing the datasets
ii. Feature selection
iii. Classification

Details are elaborated in the following sections of this work.

Proposed Method of Re-Sampling Based on Random Value (RSRV)

The imbalance of class problem has been addressed using this method. The data on gene expression is biologically determined &, therefore, should not be changed arbitrary lily. Hence, proposed method aims to address the imbalance of class for microarray data with the preservation of inspired biological value and hence no over-fitting of model and loss of information. In this method, it is assumed that the samples of the same class label are subjected to similar distributed. A data matrix with a minority class was constructed under this assumption. Then one value has been chosen randomly from every column to determine the value of the new sample of the respective position. The current sample was saved, and this process was repeated for k times for equalizing the sample of both the classes. Finally, k number of samples were obtained that were from the same distribution but separate from the actual dataset. The RSRV method is illustrated below as Algorithm 1:

Algorithm 1: RSRV: Resampling based on Random Value

Input: X - Given minority as a sample data matrix, k be the quantity of first samples
Output: X - New data matrix
while (k > = 1): **do**
  for j = 1, 2, ..., n (n is the column size of ***X):*** **do** Random value V chosen from $\mathbf{X_j}$ (j^{th} column of **X**);Save V to respective position of new sample;
  end
  Update new sample to **X;**
  k = k −1;
end
return **X;**

Here, **X** is the matrix of specified data, that denotes the small quantity class of microarray data, wherein rows and columns are represented as samples and genes(features), respectively.

Proposed Method Recursive Feature Elimination (RFE) Using Variable Step Size (RFEVSS)

Recursive feature elimination (RFE) technique put forward by Guyon et al. (2002) was a concrete example of a backward removal of features. The weights to the features were assigned depending on the external estimator. The main objective of RFE is eliminating the mostly irrelevant features, and features subsets are arranged. Hence, first, with an initial set of features, the estimator was trained, and weight has assigned to each feature. Then, these weights were sorted in descending order as per their particular value. The left-out features of the feature were finally eliminated. This process was repeated till the required number of features gets selected on the pruned set.

The problem of a considerably large amount of time consumption is the main drawback of RFE, in particular when input datasets contain very large dimensions. There-

fore, in order to decrease the number of iterations, it was important to increase the step size. Some researchers yet, essentially stated that there would be a negative impact on the feature selection result for large step size, in particular when the RFE process is almost complete (Yin et al. [19]). The improvised method of RFE named RFEVSS has been proposed to minimize the time consumption of RFE and mitigate the negative impact on feature selection simultaneously. In this method particular, first, a large initial value was initialized to the step size, then reduce the value to its half when feature's number that have to eliminated becomes half of their original size; the process was repeated till the step size turns into one. It can be clarified in detail with two aspects: one, size of the step that changes through larger to smaller and does not alter each time that is depending on the state of updating the order, & also the quantity of features to be deleted. Moreover two, the feature elimination process is moderately filtered. Often, microarray gene expression datasets contain a considerably greater quantity genes (component); among them, only a hardly any of genes are in a very important related to the disease (class labels). Therefore, there is a strong reason to conclude that comparatively a greater number of genes were removed in the beginning were more irrelevant to the class labels. Conversely, the genes are eliminated in a later stage are more significant to the class labels. Hence, in the beginning stage of feature selection, the step size was larger for decreasing the number of iterations, and then step size was reduced progressively in the later stage of the process of feature selection. So, features are selected more carefully, thus feature selection quality is guaranteed. This has been taken as the base to improvise the RFE; in addition to that, the initial value of step size was set as a key parameter that relates to specific datasets. The detailed procedures have been shown in Algorithm 2.

Algorithm 2: RFEVSS: RFE using Variable Step Size

Input: **X** - Set of genes, **Y**- labels of sample, n selected = quantity of genes to select, starting_step size

Output: **X** - Matrix with total number of genes selected from Xtemp = n

total, N= n initial, S= s- initial;

while *(N > n selected):* **do**

$N = N - S;$

if *(temp/N* $= 2$ *and* $S > 1$*):* **then**

temp = N;

S = S / 2;

end

Train LLSVM with **X** and **Y** and get sorted weights vector **W**;Remove features according to **W** and S, and update **X**;

end

Return **X**

Large Scale Linear Support Vector Machine (LLSVM)

Many researchers have considered SVM as one of the best choices in their study as a selection of features & frequently implemented as a classifier for the microarray gene

expression datasets. Though, most of the Support Vector Machine depends on the kernel techniques (usually, linear kernel) and Lagrange dual solver. The large scale linear SVM (LLSVM) have been implemented in this study for accelerating the process of weights allocation instead of SVM (Yuan et al. [19]). LLSVM were designed spicily to perform classification task on large-scale datasets, for example, text data classification. The microarray datasets also contain very large dimensions alike text data. Therefore, LLSVM will also be suitable for microarray datasets.

The large-scale liner SVM objective function is defined as:

$$\underset{w}{Min}\, f(w) = \|w\|_1 + C \sum_{i \in I(w)} b_i(w)^2 \tag{1.1}$$

where

$$b_i(w) = 1 - y_i w^T x_i \tag{1.2}$$

$$I(w) = \{i | b_i(w) > 0\} \tag{1.3}$$

Here, feature vector is represented as $\mathbf{x}_i$ for i^{th} sample, y_i is the respective label and the weight vector of the feature is represented as **w**. Therefore, the loss function of large scale linear SVM is a square hinged that is L1 regularized. The penalty factor C > 0, which determines the sparseness of the weight vector (**w**). Less significant genes that have more weights get penalized to 0, as C gets bigger, that is, weight vector (**w**) gets sparser. Just like other linear SVMs, the final decision function has the same form as shown in Eq. 1.4:

$$f(x*) = sign(w.x*) \tag{1.4}$$

The unknown sample feature vector is denoted by x*.

The overview of cyclic coordinate descent technique for LLSVM (Yuan et al. 2010, Fan et al. 2008) is depicted in Algorithm 3.

Algorithm 3: Cyclic coordinate descent method for large scale linear SVM

Input: $\mathbf{w}^1$
Output: $\mathbf{w}^{m+1}$
Given $\mathbf{w}^1$;
for $(k = 1, 2, 3, \cdots m;)$ **do**
$\mathbf{w}^{k,1} = \mathbf{w}^1$;
  for $(j = 1, 2, \cdots n)$ **do**
    Obtain z^* by solving the sub-problem 4.6;
    $\mathbf{w}^{k,\, j+1} = \mathbf{w}^{k,j} + z^* e_j$;
  end
  Return $\mathbf{w}^{m+1}$
end

3 Experimental Evaluation

The experimental verification of the proposed methods has been emphasized in this section. These experiments have been conducted on the four most frequently used benchmarked microarray gene expression datasets. Dataset's descriptions, Experimental settings including data preprocessing, parameter estimation, and evaluation of performance measures are described in the following subsections.

3.1 Datasets

The four most frequently used cancer benchmarked microarray datasets such as Colon, Leukemia, Ovarian, and Breast cancer dataset have chosen for conducting extensive experiments. All these datasets were frequently used by many researchers in the bioinformatics field of study and have been made available publicly for researchers. Colon and Leukemia (ALL AML) datasets are available at http://featureselection.Asu.edu/datasets.php. Moreover, Breast and Ovarian datasets are available at http://csse.szu.edu.cn/staff/zhuzx/Datasets.html. The considered datasets for these experiments are of binary classes, where the problem of class imbalance is commonly available. Table 1 describe the details of these datasets' characteristics. SDR referred to as the Sample-to-dimension ratio, i.e., (No of class 1 + No of class 2)/No of Features. IR referred to the Imbalance ratio, i.e. (No of class 2/No of class 1).

Table 1. Characteristics of raw datasets

Dataset	Number of Class 1	Number of Class 2	Number of Features	SDR	IR	Class description	Reference
Colon	22	40	2000	3.1%	1.82	22 Normal 40 Cancer	Alon et al. (1999)
Leukemia	25	47	7129	1.01%	1.88	25 AML 47 ALL	Pomeroy et al. (2002)
Ovarian	91	162	15154	1.67%	1.78	91 Normal 162 Cancer	Petricoin et al. (2002)
Breast	46	51	24481	0.40%	1.11	46 Normal 51 Cancer	Van't Veer et al. (2002)

The RSRV algorithm, which was proposed, has been used separately to address the class imbalance problem on four datasets that were considered for experiments. New samples of datasets were obtained from class 1 as the quantity of class 1 samples are equal to a quantity of class 2 samples. As the outcome, IR is 1.0 for all datasets, and SDR adjusts accordingly.

3.2 Data Pre-processing

Mathematically, each dataset (including balanced datasets and raw datasets) was standardized as unit variance and zero mean. Accordingly, the conflicting outcome generated

be- cause of different genes with significant gaps in expression levels were mitigated. The given mRMR approach has been used, which is based on mutual information, so in particular, these datasets need to be discretized. The measure proposed by Guyon et al. (2002) were used, as described below:

$$\tilde{x} = \begin{cases} +2, \ if \ x > \mu + \sigma/2 \\ -2, \ if \ x > \mu - \sigma/2 \\ 0, \ Otherwise \end{cases}$$

where, μ and σ denotes the mean value and standard variance respectively. So, two types of datasets, discreet and continuous, are obtained, which all are standardized. The mRMR approach is employed on discrete datasets while other feature selectors are used on continuous datasets.

4 Classifiers

Classification of the learning can be obtained according to the representation of knowledge used to emulate the output. The most common representations of knowledge that are being used in this study as supervised learning are:

Support Vector Machine (SVM)
SVM belongs to the linear model family, a group of model-based learning methods representing feedback as a linear combination of input attributes. It is a statistical learning theory based on classifier (Vapnik and Vapnik [20]). The Input data space is mapped to a high dimension feature. The mapping of the input space vector is done by the kernel function. The SVM is based on the concept of decision plans defining boundaries for decisions, a decision plane attempt to isolate a set of instances belonging to various classes, also known as a hyperplane. Therefore, the SVM aims to construct hyperplanes separating the samples while optimizing the range, i.e., the distance between data points from distinct classes. Figure 2 represents an example for SVM.

The hyperplane can be defined as given in equation 1.5, which separates the instances.

$$f(x) = (w^T.x) + b \tag{1.5}$$

where ω refers to a vector of d-dimensional coefficient, which is normal to the hyperplane, and b is the offset from the origin. The margin (W) of the hyperplane can be maximized with the help of linear SVM by solving the optimization task, as shown below.

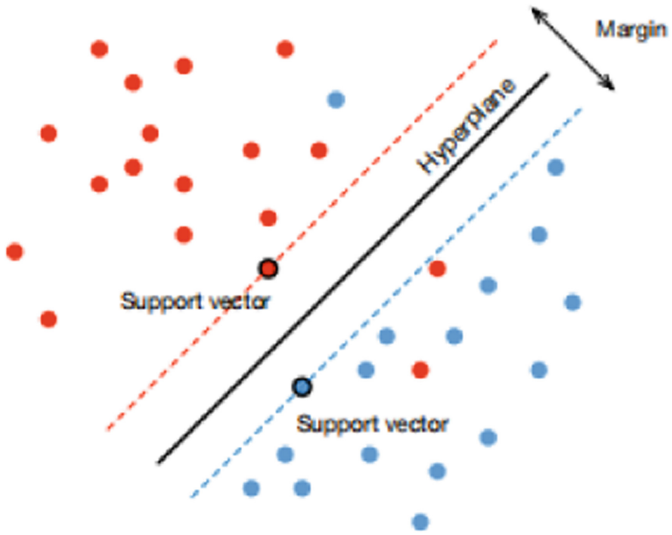

Fig. 2. Support vector machine

K-Nearest Neighbor(kNN)

It is one of the commonly used machine learning algorithms. The kNN is easy to understand, very simple, and versatile algorithms. There is a wide range of applications such as finance, political science, health care, bioinformatics, handwriting, and video recognition, wherein kNN being used effectively. The Principle of the kNN algorithm is on the basis of feature similarity. The kNN's is a non-parametric and lazy learning algorithm. Non-parametric means the underlying distribution of data is not assumed. In practice, this is very helpful when mathematical assumptions are not followed in most real-world datasets. Lazy learning means that it needs no model generation training data points; in the testing phase, all training data are used. Due to this, training makes faster, whereas the testing phase becomes slower and expensive (consume more memory and time).

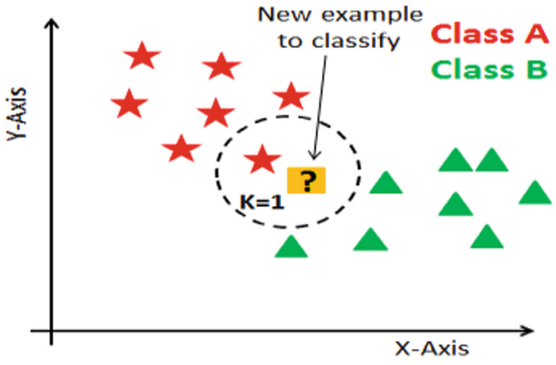

Fig. 3. K-nearest neighbor example

K is the number of nearest neighbors in kNN. The core deciding factor of KNN is the number of neighbors i.e., the value of K. When a number of the class is 2, then generally the value K an odd number. The algorithm is called as the nearest neighbor when K=1. For example, P_1 is the point, for that label has to be predicted. Consider one point nearest to P_1 in the beginning, and then P_1 is assigned as the label of the closest point. This is illustrated in Fig. 3.

Suppose the label needs to predict for a point P_1. Consider the k nearest to P_1 in the beginning and then categorize the k neighbors' points by majority vote. The prediction is taken for every entity to vote for its class and the most voting class. The distance between the points is computed by distance measure, for example, Manhattan distance, Euclidean distance, Minkowski distance, and Hamming distance in order to determine similar nearest points. The following simple steps are followed by KNN, as illustrated in figure 4.

- Calculate distance
- Find closest neighbors
- Vote for labels

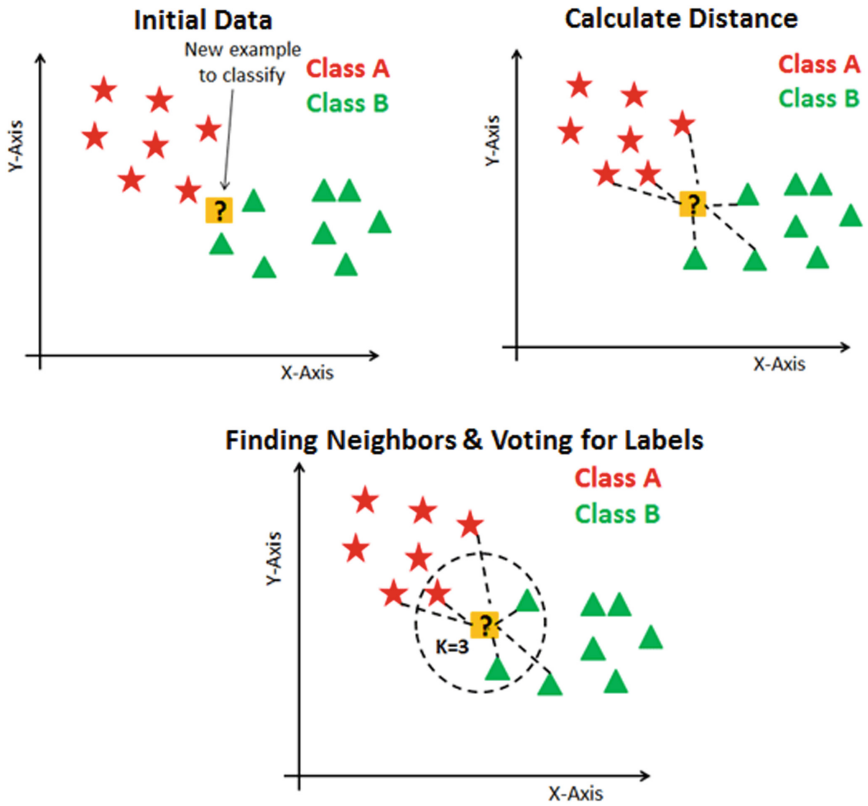

Fig. 4. Steps in K-nearest neighbor

Logistic Regression (LR)

This is one of the simplest and most frequently used approaches to classify binary class problems in machine learning algorithms. There are several classification problems such as diabetes prediction, spam detection, customer identification, churn prediction, advertisement prediction, available in the bucket. LR defines and estimates the relation- ship between a binary variable dependent and independent. The statistical method for binary class classification is logistic regression. The target variable or outcome is dichotomous in nature. When there are only two possible target classes, then it is known as dichotomous; for instance, it can be used to detect the occurrence of cancer. The probability of occurrence of an event is computed. This type of linear regression is a special case in which the target class happens to be categorical variables. As the de- pendent variable, it uses a log of odds. Logistic regression estimates the likelihood of a binary event using a logit function.

Equation 1.7 describes the Linear Regression:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{1.6}$$

Here, x_1, $x_2 \cdots and\ x_n$ are explanatory variables and y is dependent variable.

Sigmoid function is illustrated in Eq. 1.7:

$$p = 1/1 + e^{v} \tag{1.7}$$

After applying Sigmoid function on linear regression the following Eq. 1.8 is obtained:

$$p = 1/1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)} \tag{1.8}$$

Logistic Regression properties are as follows:

- Bernoulli distribution is followed by the dependent variable in logistic regression.
- Using maximum likelihood estimation is done.
- No R Square, Model fitness is computed through Concordance, KS-Statistics.

Linear Regression vs. Logistic Regression

The discrete output is produced by logistic regression, whereas Linear regression produces a continuous output. The stock price and house price are examples of continuous. Predicting customer churn and the patient has cancer are examples of the discrete output. Maximum Likelihood Estimation (MLE) technique is used for estimating logistic regression, whereas the Ordinary Least Squares (OLS) used for estimating Linear regression, illustrated in Fig. 5.

Estimation of Parameter

The parameters of the classifiers such as SVM, kNN, and LR needs to be determined,

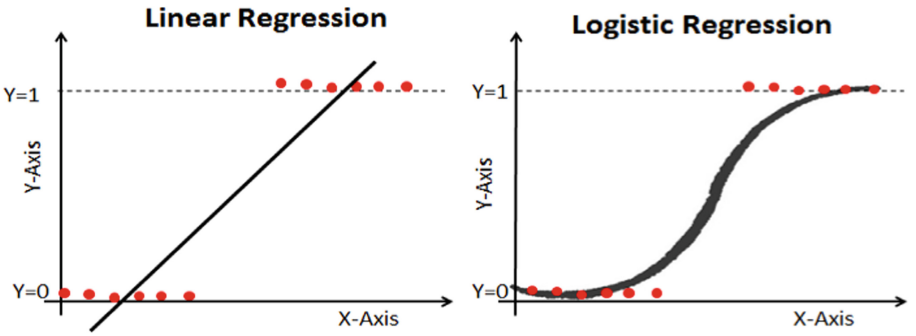

Fig. 5. Linear regression vs. logistic regression

before selecting features using LLSVM and SVM and classifying the transformed datasets. C was the penalty factor for SVM, large scale linear SVM, and LR, which is a crucial parameter. The C value affects the result of the selection of features as well as the complexity of classification model such as SVM, LLSVM, and LR. The number of nearest neighbors to be selected is denoted by K for the kNN. The too small or too big value of K is not the right choice, so the value of K needs to be tuned carefully.

The parameters were estimated for the corresponding model separately because these models are used as feature selection or classifiers. Further, one more parameter step size(S) input is necessary to be determined before applying RFEVSS along with large scale linear SVM (LLSVM) or SVM, which are represented as LLSVM-RFEVSS and SVM-RFEVSS respectively. The Stratified 5-fold cross-validation and grid search have been utilized in the process of specifying these parameters, and the best result has been achieved, as shown in Table 2.

Table 2. Parameters of feature selectors and classifiers for balanced datasets

		Parameter	Leukemia	Ovarian	Breast	Colon
Feature selectors	LLSVM	C	0.1	0.3	0.3	0.9
	SVM	C	0.1	0.5	0.1	0.1
Step size	LLSVM	S	600	1000	800	100
	SVM	S	400	1000	800	200
Classifiers	SVM	C	9	3	0.09	7
	kNN	K	1	7	7	6
	LR	C	19	7	3	9

In this work, balanced datasets that were obtained after allying RSRV were used to conduct most of the experiments, but the value of C and S for SVM-RFEVSS were tuned on raw data sets for validating the performance of RSRV algorithm. Table 3 describes the details.

It can be seen in Tables 2 and 3 that the starting step size is quite different for different datasets. The step size becomes larger when datasets have more genes (Breast and Ovarian). On the other hand, when datasets have fewer genes (Leukemia and Colon), the starting step size becomes smaller. This confirms exactly the gene importance assumption as well as the basis for improvising RFE that has been outlined in RSRV.

Table 3. Parameters for SVM-RFEVSS on raw datasets

Feature selectors	Parameters	Leukemia	Ovarian	Breast	Colon
SVM	C	0.3	0.5	0.1	0.5
RFEVSS	S	100	1000	1000	60

Measures for Performance Evaluation

In this study, frequently used three types of measures have been chosen as the performance evaluation measure such as ACC, AUC, and MCC. All these measures were widely used for evaluating the classification task, out of the ACC and MCC are defined as below:

$$ACC = \frac{TN + TP}{TP + TN + FP + FN} \tag{1.9}$$

$$MCC \frac{(TPXTN) - (TPXFN)}{\sqrt{(TP + TN)}(TP + FP)(TN + FP)(TN + FN)} \tag{1.10}$$

where

ACC → Accuracy, MCC → Matthew's correlation coefficient
AUC → Area under ROC curve
TP → True Positive
TN → True Negative
FP → False Positive
FN → False Negative

ACC is the most commonly used standard for evaluation, but using this alone may be sufficient. MCC is generally selected as one of the best options since MCC can still deliver a good evaluation result even when the dataset is class imbalanced. Mostly, the correlation coefficient between the observation and target (predicted) value was measured as MCC, and its value ranges in-between −1 and +1. When the coefficient value obtained as +1 that refers to the perfect prediction, whereas 1 refers to the worst pre- diction. True Positive Rate (TPR) and False Positive Rate (FPR) both are taken into account for AUC computation that is described as below:

$$TPR = TP/TP + FN, FPR = FP/FP + TN$$

AUC is referred to as a probability value that classified correctly one sample, the greater the value of probability is the better.

5 Result and Discussion

Four sets of comparative experiments have been performed in this section for model evaluation. The proposed RSRV, RFEVSS, and LLSVM algorithms have been verified in the first three sets of comparative experiments, respectively. Then the fourth set of experiments has been conducted for evaluating the outcome of the three standard classifiers and discussed the suitability of as classifier for microarray datasets. Moreover, finally, the desired experiments were conducted to evaluate the generalization capability of the classifiers. All experiments were done with stratified 5-fold cross-validation since it is guaranteed by the stratified cross-validation technique which are the instances proportion that belong to two classes that is in both the train and test set are equal.

Comparative Analysis of Balanced Datasets with RSRV and Raw Datasets
The proposed RSRV has been used in this section for balancing the raw datasets; then, experiments were conducted on four raw datasets with SVM-RFEVSS and balanced for gene selection task. The SVM-RFEVSS method has chosen as a feature selector be- cause SVM-RFEVSS consumes less processing time than SVM-RFE for attaining the same purpose. Since SVM happens to be the natural choice with SVM-RFEVSS, so linear SVM (with C = 1) has been used as a classifier, and all the datasets were performed 128 times for selecting 1 to 128 genes.

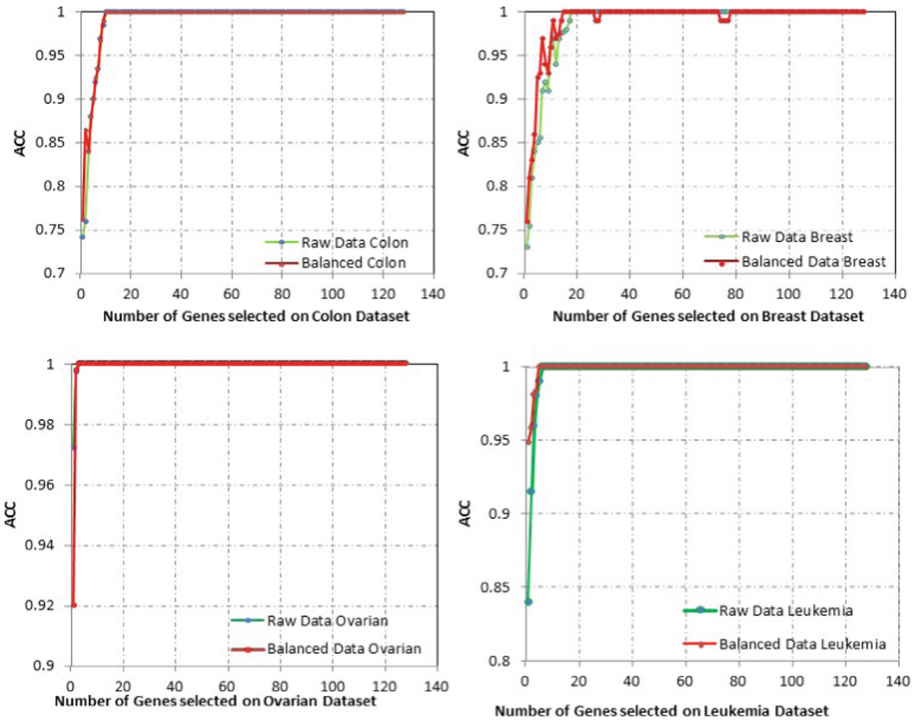

Fig. 6. ACC comparison on raw and balanced datasets

Figures 6, 7 and 8 represent the comparative analysis of performance of three evaluation measures (ACC, MCC and AUC) on balanced and raw datasets. It has been observed that the equal weighted Leukemia given well showing on all count. The balanced Colon and Breast perform better on MCC and ACC, whereas on AUC, it was closely similar. It has also been noted that the outcome of balanced ovarian was undesirable, but it takes place when fewer genes were selected. Moreover, the results achieved on the balanced datasets get good enough the raw datasets as the quantity of genes raises. From this solution to the class imbalance issue of microarray datasets, RSRV takes place a good choice.

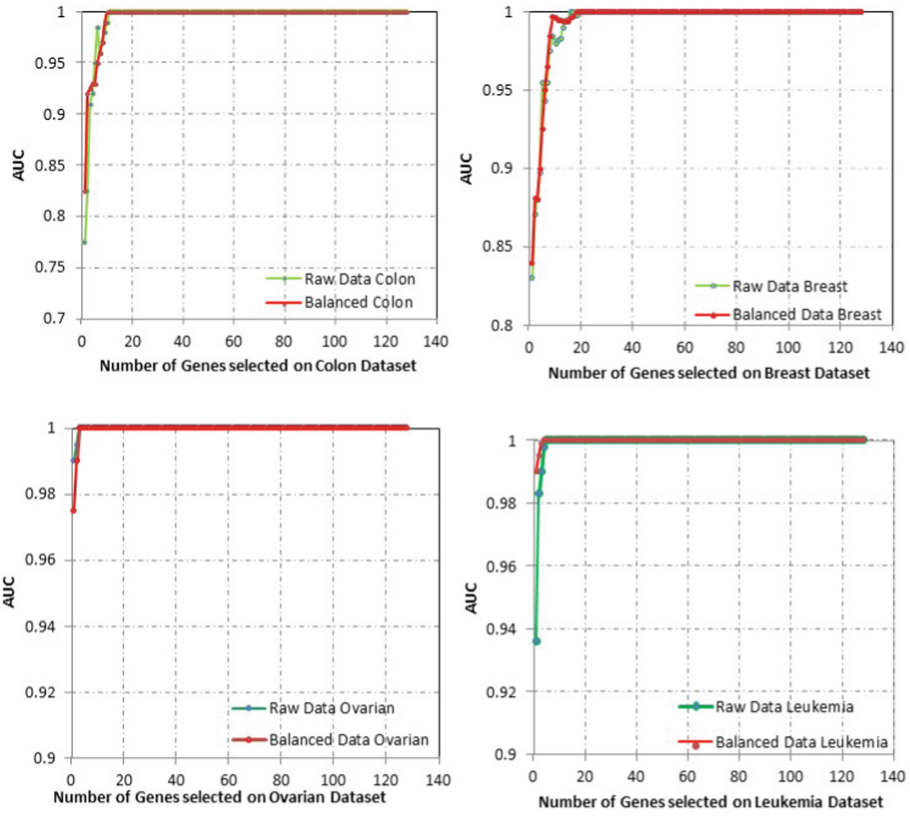

Fig. 7. AUC comparison on raw and balanced datasets

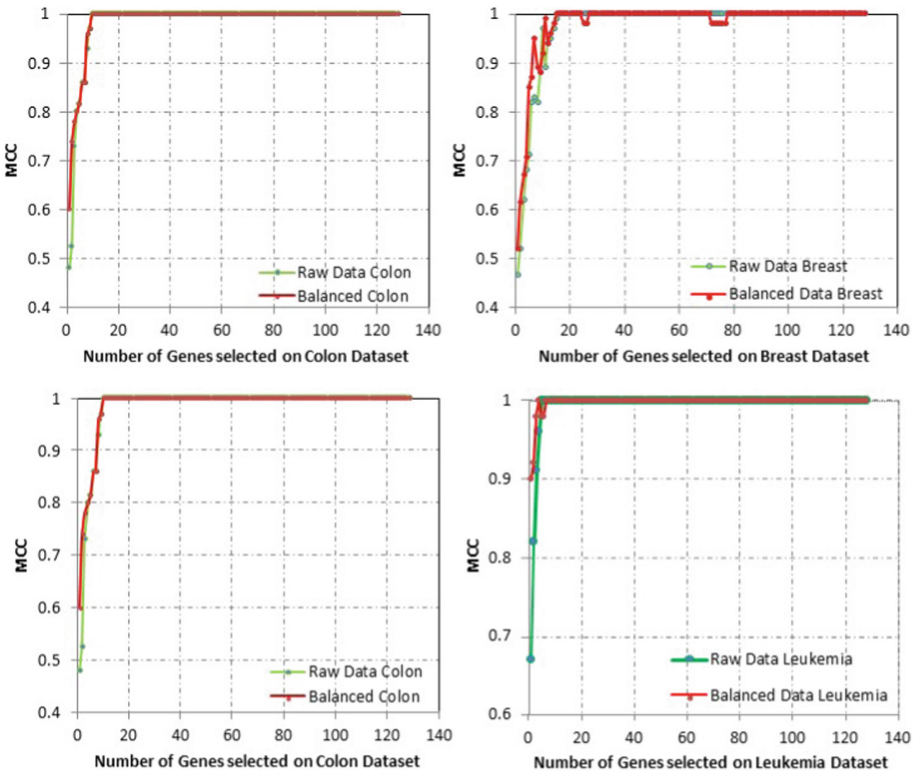

Fig. 8. MCC comparison on raw and balanced datasets

Comparative Analysis of RFEVSS and RFE

The outcome of RFEVSS was rectified in this section. The traditional linear SVM is applied as a primary feature selection method and combined along with RFE and RFEVSS separately for conducting the experiments on IV balanced datasets. In the same condition, except for the step size of RFE, there are two sets of experiments conducted. It was set to 1 in one-case, & in other, it has to be find by the initial input value along with a quantity of component that are to be completely removed. In addition, a number of genes to select is chosen as 4 in this experiment because four (4) is relatively small. Linear SVM with $C = 1$ has been applied as the classifier.

Feature Selection Performance Analysis of LLSVM-RFEVSS with Three Other Feature Selectors

The efficiency of LLSVM was verified in this section. LLSVM combined with VSS-RFE termed as LLSVM-RFEVSS and their outcome as a feature selector were compared with three typical feature selectors such as relief (Kononenko [21]), mRMR and SVM-RFEVSS. The SVM-RFEVSS, instead of SVM-RFE, has chosen as a feature selector because of its huge time consumption. The linear SVM (with $C = 1$) was used as a classifier introduced in the previous section, and each balanced datasets are implemented 128 times orderly select 1 to 128 genes.

The time utilization by LLSVM-RFEVSS and SVM-RFEVSS are shown in Table 3. This depicts that the time utilization of LLSVM-RFEVSS is significantly lessened as a feature selector compared to SVM-RFEVSS (bold faces shows the best performances), in particular for high dimensional datasets (e.g., Breast).

The quality of four feature selector are shown in Figs. 9, 10 and 11. It can be observed that the curve of relief on all four datasets are unstable, and evaluation measures values on some datasets (Breast, Leukemia) are lowest. The mRMR curves are more stable compared to reliefF, but values of evaluation measure are much lower in comparison with SVM-RFEVSS and LLSVM-RFEVSS. In fact, both SVM-RFEVSS and LLSVM-RFEVSS can produce the evaluation values of classifier either 100% or very close to that for Breast, Leukemia, and Ovarian datasets. In the case of the Colon dataset, SVM-RFEVSS performed slightly better than LLSVM- RFEVSS; otherwise, LLSVM-RFEVSS outperforms other feature selectors.

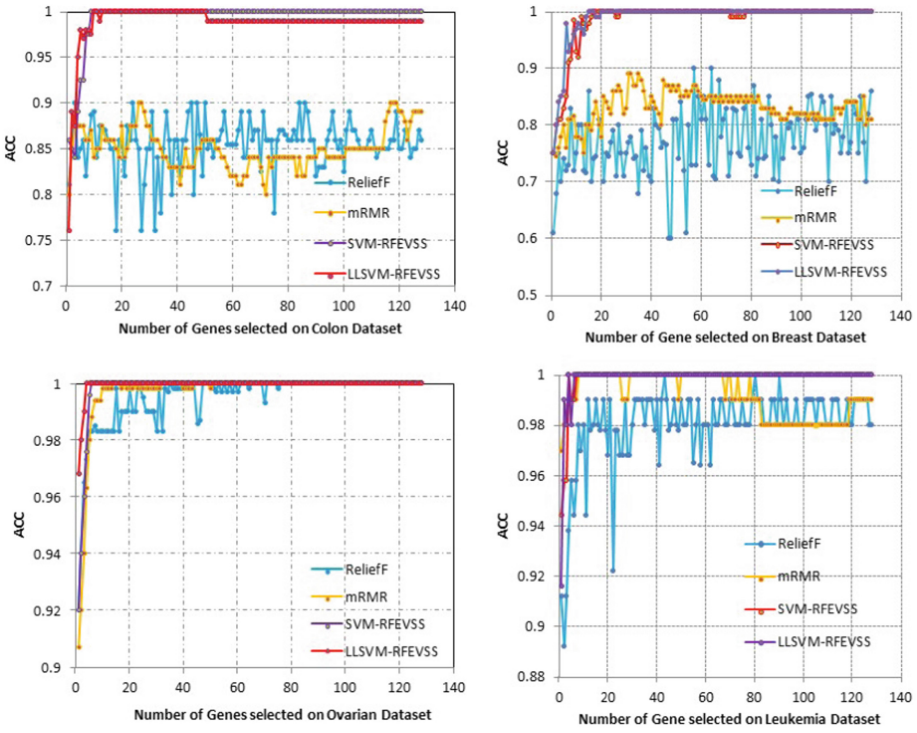

Fig. 9. ACC comparison obtained by four feature selectors

Three common CLASSIfier's Comparative study

The validation of three typical classifiers, such as k-Nearest Neighbors (kNN), Linear SVM, and Logistic Regression (LR), was carried out in this section. LLSVM-RFEVSS has been deployed to select 1 to 32 genes as the feature selector from the obtained balanced datasets, and then selected genes were evaluated with classifiers that are well-tuned. Also, LLSVM-RFEVSS has been utilized as a feature selector with LR as a classifier in order to conduct experiments on balanced datasets. The training and testing scores were determined for evaluating the model's generalization capability.

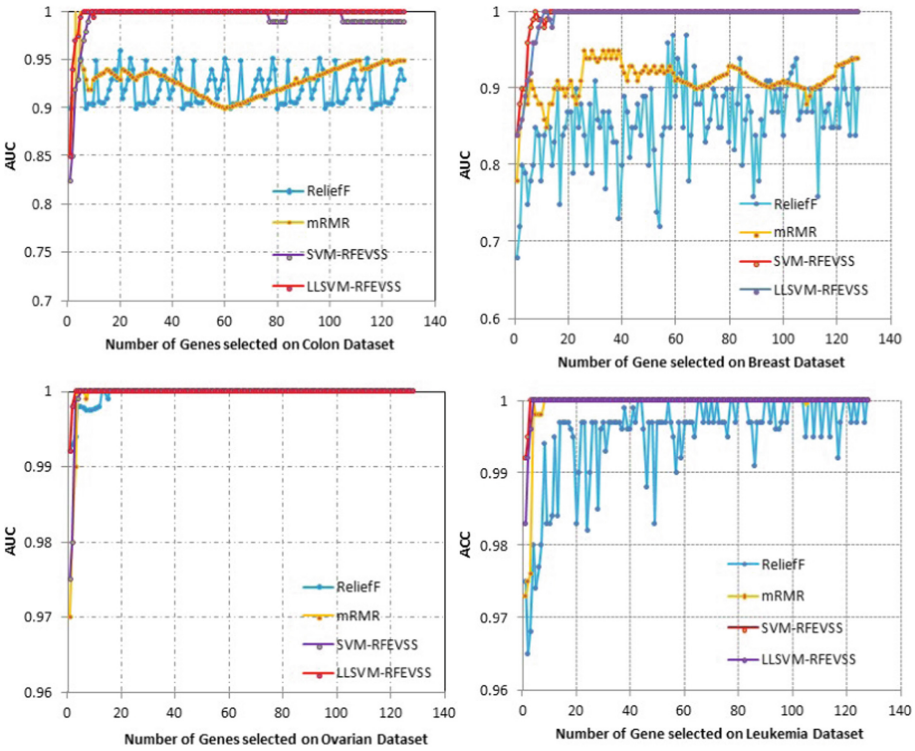

Fig. 10. AUC comparison obtained by four feature selectors.

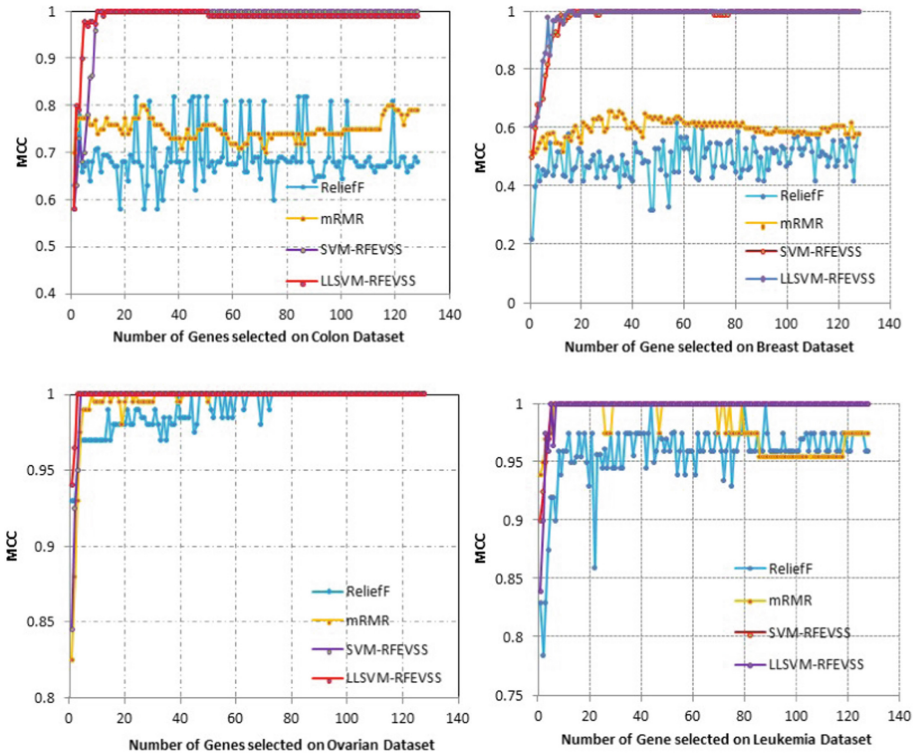

Fig. 11. MCC comparison obtained by four feature selectors

The effect of different classifiers on the performance of classification are shown in Figs. 12, 13 and 14. It can be seen that the outcomes obtained on the same datasets from various classifiers can be so specific, and overall, the datasets, SVM and LR, outperform the three evaluation tests as tabulated in Table 4. There are many variations of the expression level of each gene for microarray data; even the samples belong to the same category that is the disadvantage of kNN. The kNN algorithm was acted upon by the interval directly between the data points, that were find by the component value. On the other hand, SVM and LR models are suitable for microarray datasets because they are linearly separable. That is the reason why these two classifiers are widely used in these areas of research.

Table 4. Classifier's comparative study

	kNN			SVM			LR		
	ACC	AUC	MCC	ACC	AUC	MCC	ACC	AUC	MCC
Breast	0.942	0.964	0.893	**0.963**	0.978	**0.929**	0.960	**0.980**	0.923
Colon	0.939	0.980	0.889	0.967	0.991	**0.965**	**0.979**	**0.993**	0.961
Ovarian	0.998	0.999	0.996	**1.0**	**1.0**	**0.997**	**1.0**	**1.0**	**0.997**
Leukemia	0.995	0.996	0.993	**0.996**	0.996	**0.996**	**0.996**	**0.999**	0.993

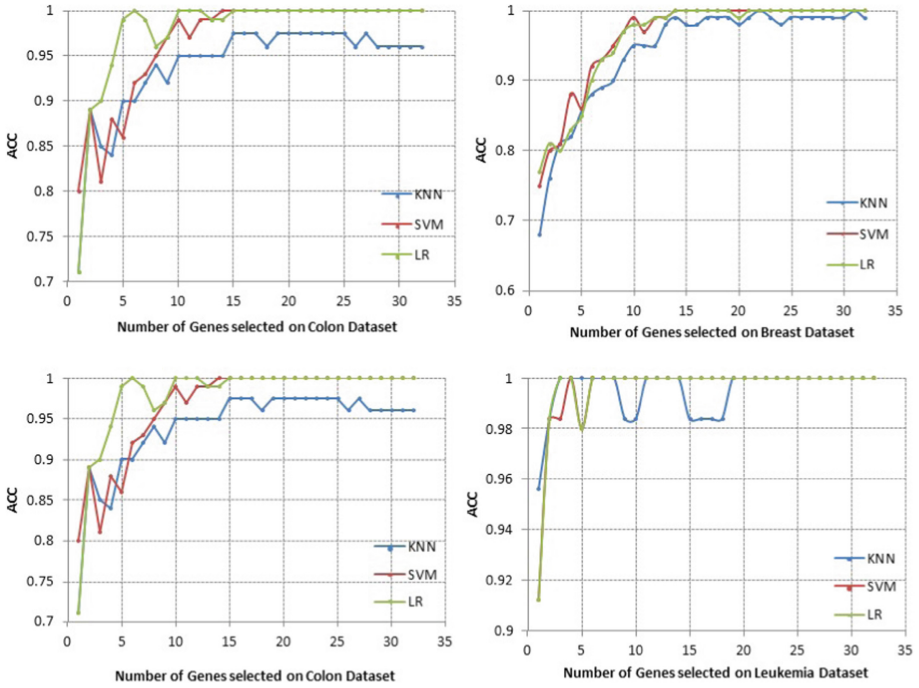

Fig. 12. ACC comparison acquired by three classifiers

Furthermore, it can be noted that LR's curves are smoother than SVM's, so the LR's performance is stable. It is worth noting that the LR classifier is simple and easy to implement; that means to say that for a dataset with a small sample size such as microarray datasets. Hence, it can strongly believe that Logistic Regression to paid more attention as a classifier for microarray datasets.

The estimation outcome of classification model are shown in figure .15 for the number of genes selected respectively are 1, 2, 4, 8, 16, 32, 64, and 128. The training scores are very close to testing scores, as shown in Fig. 15, in particular, when more genes are selected. That is to say that the classification model has good generalization capability (Hawkins 2004). The Classification model learning capability on the given data and applied to the unseen data is referred to as generalization. Therefore, the good generalization capability of the model is nothing, but it guarantees the quality outcome.

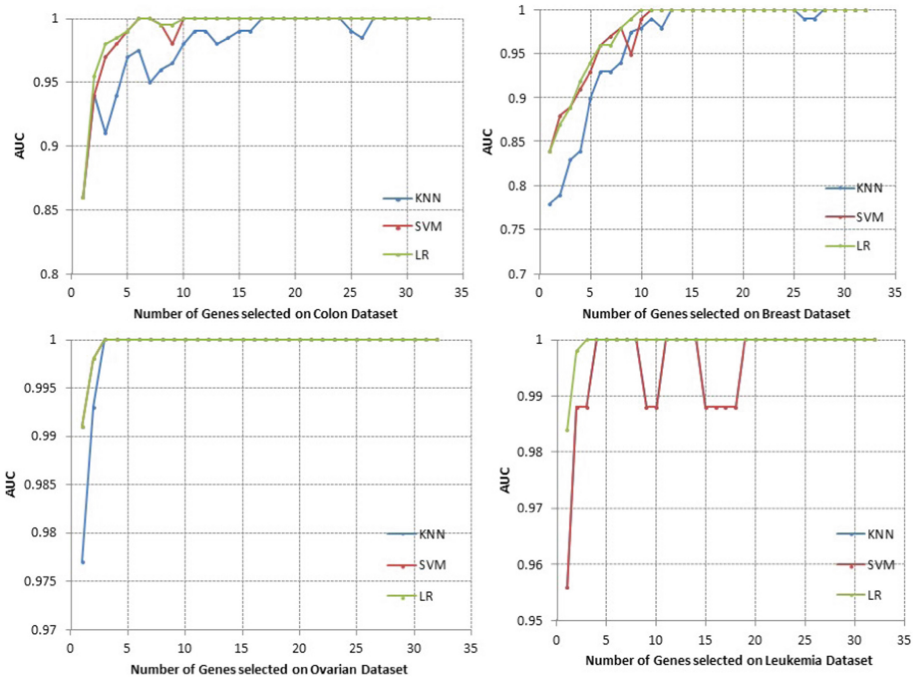

Fig. 13. AUC comparison acquired by three classifiers

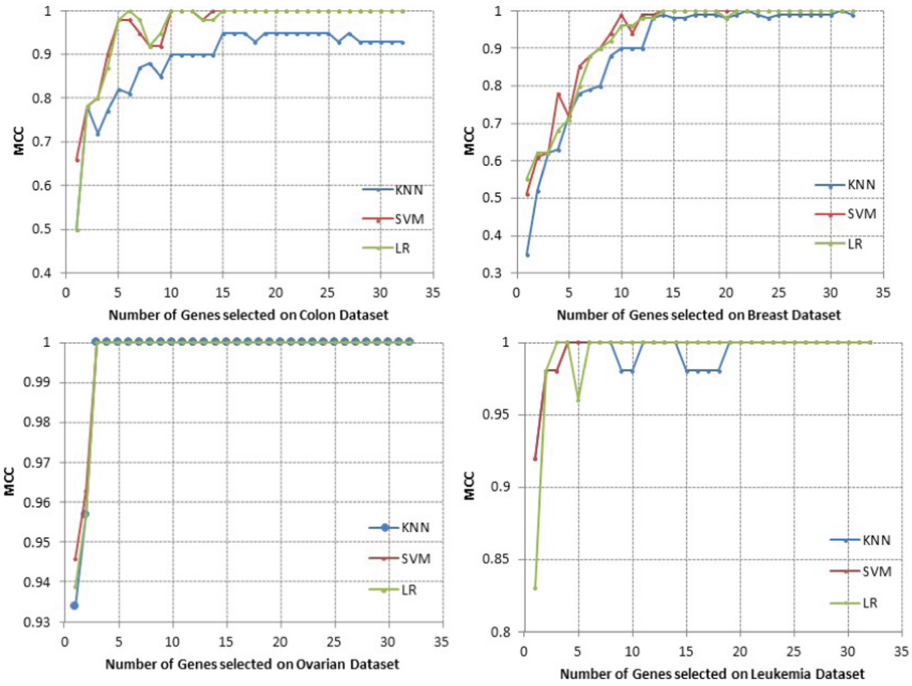

Fig. 14. MCC comparison acquired by three classifiers

6 Summary

In this section, the summary of the work has been presented. The complex and deadly diseases, for example, cancer, continue to represent the biggest threat for human. The advancement and growth in microarray data along with the statistical methods and machine learning techniques were contributed in contemporary dimension for the diagnosis and prognosis of these diseases. The core technologies of microarray data analysis are feature selection and classification. Both of these techniques play a very important role in gene recognition leads to the diagnosis of diseases. The special attention and careful utilization of feature selection and machine learning techniques are required because of the challenging characteristics of microarray gene expression datasets.

SVM-RFE is a standard approach that is commonly deployed in this field by many researchers. The improved version RFE, known as RFEVSS, was proposed for reducing the consumption of time by SVM-RFE. The recursion time was reduced with the help of larger step size initially, continue reducing the step size when the features are to be eliminated is reduced, hence the quality of meaningful gene selection is assured. There is a huge number of genes available in the human body, and very few are responsible for causing these diseases. Therefore, it is necessary to deploy efficient feature selection. Although the structured execution of linear SVM was introduced which is known as LLSVM. Large Linear Support Vector Machine (LLSVM) is a pure linear classifier based on a vector that acquire the benefit of SVM and reduced the cost of computational for

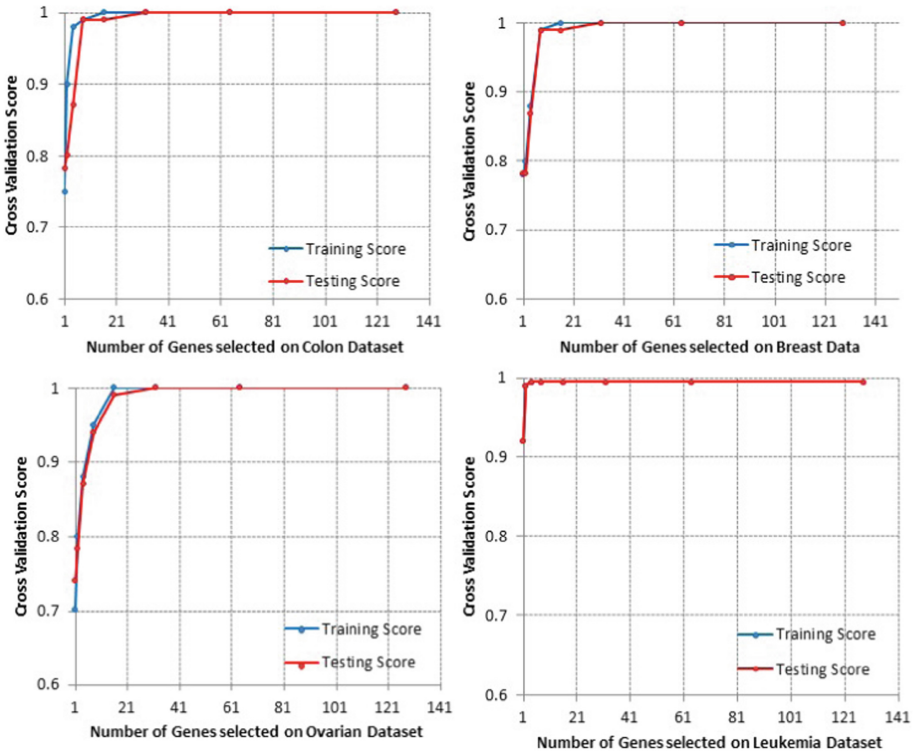

Fig.15. Evaluation of classification model

microarray datasets (large scale linearly separable data). These methods were combined with RFEVSS known as LLSVM-RFEVSS that happens to be an effective and efficient feature selector compared to other existing feature selectors, as shown in the results section. Finally, the experiments were conducted to identify the impact of dissimilar classifiers on the obtained outcomes and have been observed that the Logistic Regression performed finer in most of the cases.

References

1. Hira, Z.M., Gillies, D.F.: A review of feature selection and feature extraction methods applied on microarray data. Adv. Bioinform. (2015)
2. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: Feature selection for high-dimensional data. Progr. Artif. Intell. **5**(2), 65–75 (2016)
3. Elkhani, N., Muniyandi, R.C.: Review of the effect of feature selection for microarray data on the classification accuracy for cancer data sets. Int. J. Soft Comput. **11**(5), 334–342 (2016)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
5. Li, J., et al.: Feature selection: a data perspective. ACM Comput. Surv. (CSUR) **50**(6), 94 (2018)

6. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J. Bioinform. Comput. Biol. **3**(02), 185–205 (2005)
7. Zhou, X., Mao, K.: Ls bound based gene selection for DNA microarray data. Bioinformatics **21**(8), 1559–1564 (2004)
8. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007)
9. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**(1–3), 389–422 (2002)
10. Mundra, P.A., Rajapakse, J.C.: SVM-RFE with MRMR filter for gene selection. IEEE Trans. Nanobiosci. **9**(1), 31–37 (2009)
11. Yoon, S., Kim, S.: Mutual information-based SVM-RFE for diagnostic classification of digitized mammograms. Pattern Recogn. Lett. **30**(16), 1489–1495 (2009)
12. Tang, Y., Zhang, Y.-Q., Huang, Z.: Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. IEEE/ACM Trans. Comput. Biol. Bioinf. **4**(3), 365–381 (2007)
13. Yin, J., Hou, J., She, Z., Yang, C., Yu, H.: Improving the performance of SVM-RFE on classification of pancreatic cancer data. In: 2016 IEEE International Conference on Industrial Technology (ICIT), pp. 956–961. IEEE (2016)
14. Zhu, B., Baesens, B., vanden Broucke, S.K.: An empirical comparison of techniques for the class imbalance problem in churn prediction. Inf. Sci. **408**, 84–99 (2017)
15. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybridbased approaches. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **42**(4), 463–484 (2011)
16. Qian, Y., Liang, Y., Li, M., Feng, G., Shi, X.: A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing **143**, 57–67 (2014)
17. Galar, M., Fernandez, A., Barrenechea, E., Herrera, F.: Eusboost: enhancing ensembles for highly imbalanced datasets by evolutionary undersampling. Pattern Recogn. **46**(12), 3460–3471 (2013)
18. Yu, H.-F., Huang, F.-L., Lin, C.-J.: Dual coordinate descent methods for logistic regression and maximum entropy models. Mach. Learn. **85**(1–2), 41–75 (2011)
19. Yuan, G.-X., Chang, K.-W., Hsieh, C.-J., Lin, C.-J.: A comparison of optimization methods and software for large-scale l1-regularized linear classification. J. Mach. Learn. Res. **11**(Nov), 3183–3234 (2010)
20. Vapnik, V., Vapnik, V.: Statistical Learning Theory, vol. 1. Wiley, New York (1998)
21. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-57868-4_57
22. Yang, D., Zhu, X.: Gene correlation guided gene selection for microarray data classification. BioMed. Res. Int. **2021**, Article ID 6490118, 11 p. (2021). https://doi.org/10.1155/2021/6490118
23. Ramadhani, P.T., Nasution, B.B.: Neural network as a preferred method for microarray data classification. In: 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), pp. 337–340 (2021). https://doi.org/10.1109/ICSECS52883.2021.00068