mathematics

Article

Combination of Ensembles of Regularized Regression Models with Resampling-Based Lasso Feature Selection in High Dimensional Data

Abhijeet R Patil [1] and Sangjin Kim [2,*]

[1] Computational Science, University of Texas at El Paso, El Paso, TX 79968, USA; arpatil@miners.utep.edu
[2] Department of Mathematical Sciences, University of Texas at El Paso, El Paso, TX 79968, USA
* Correspondence: skim10@utep.edu

Received: 14 October 2019; Accepted: 4 January 2020; Published: 10 January 2020

Abstract: In high-dimensional data, the performances of various classifiers are largely dependent on the selection of important features. Most of the individual classifiers with the existing feature selection (FS) methods do not perform well for highly correlated data. Obtaining important features using the FS method and selecting the best performing classifier is a challenging task in high throughput data. In this article, we propose a combination of resampling-based least absolute shrinkage and selection operator (LASSO) feature selection (RLFS) and ensembles of regularized regression (ERRM) capable of dealing data with the high correlation structures. The ERRM boosts the prediction accuracy with the top-ranked features obtained from RLFS. The RLFS utilizes the lasso penalty with sure independence screening (SIS) condition to select the top k ranked features. The ERRM includes five individual penalty based classifiers: LASSO, adaptive LASSO (ALASSO), elastic net (ENET), smoothly clipped absolute deviations (SCAD), and minimax concave penalty (MCP). It was built on the idea of bagging and rank aggregation. Upon performing simulation studies and applying to smokers' cancer gene expression data, we demonstrated that the proposed combination of ERRM with RLFS achieved superior performance of accuracy and geometric mean.

Keywords: ensembles; feature selection; high-throughput; gene expression data; resampling; lasso; adaptive lasso; elastic net; SCAD; MCP

MSC: 62P10; 62F40; 62F07

1. Introduction

With the advances of high throughput technology in biomedical research, large volumes of high-dimensional data are being generated [1–3]. Some of the examples of what produces such data are microarray gene expression [4–6] data sequencing, RNA-seq [7], genome-wide association studies (GWASs) [8,9], and DNA-methylation studies [10,11]. These data are high dimensional in nature, where the total count of features is significantly larger than the number of samples $(p >> n)$—termed the curse of dimensionality. Although this is one of the major problems, there are many other problems, such as noise, redundancy, and over parameterization. To deal with these problems, many two-stage approaches of feature selection (FS) and classification algorithms have been proposed in machine learning over the last decade.

The FS methods are used to reduce the dimensionality of data by removing noisy and redundant features that help in selecting the truly important features. The FS methods are classified into rank-based and subset methods [12,13]. Rank-based methods rank all the features with respect to their importance based on some criteria. Although there is a lack of threshold to select the optimal number of top-ranked features, this can be solved using the sure independence screening (SIS) [14] conditions. Some of the popular rank-based FS methods used in bioinformatics are information gain [15], Fisher score [16], chi-square [17], and minimum redundancy maximum relevance [18]. These rank-based FS methods have several advantages, such as that they avoid overfitting and are computationally faster because they do not depend on the performances of classification algorithms. However, these methods do not consider joint importance because they focus on marginal significance. To overcome this issue, feature subset section methods were introduced. The subset methods [19] are the ones where the subsets of features are selected with some predetermined threshold based on some criteria, but these methods need more computational time in a high-dimensional data setting and lead to an NP-hard problem [20]. Some of the popular subset methods include Boruta [21] and relief [22].

For the classification of gene expression data, there are non-parametric-based popular algorithms, such as random forests [23], Adaboost [24], and support vector machines [25]. The support vector machines are known to perform well in highly correlated gene expression data compared to the random forests [26]. The random forests and Adaboost are based on the concept of decision trees, and the support vector machines are based on the idea of hyperplanes. In addition to the above, there are parametric machine learning algorithms, such as penalized logistic regression (PLR) models, that have five different penalties which are predominantly popular in high-dimensional data. The first two classifiers are Lasso [27] and ridge [28] that are based on L1 and L2 penalties. The third classifier is a combination of these and is termed as elastic net [29]. The other two PLR classifiers are SCAD [30] and MCP [31], which are based on non-concave and concave respectively. All these individual classifiers are very common in machine learning and bioinformatics [32]. However, in highly correlated gene expression data, the individual classifiers do not perform well in terms of prediction accuracy. To overcome the issue of individual classifiers, ensemble classifiers are proposed [33,34]. The ensemble classifiers are bagging and aggregating methods [35,36] that are employed to improve the accuracy of several "weak" classifiers [37]. The tree-based method of classification by ensembles from random partitions (CERP) [38] showed good performance but is computer-intensive. The ensembles of logistic regression models (LORENS) [39] for high-dimensional data were proven to be better for classification. However, there was a decrease in performance when there were a smaller number of true, important variables in the high-dimensional space because of random partitioning.

To address these issues, there is a need to develop a novel combination of FS with a classification method and compare the proposed method with the other combinations of popular FS with the classifiers through extensive simulation studies and a real data application. In a high dimensional data set, it is necessary to filter out the redundant and unimportant features using the FS methods. This helps in reducing the computational time and helps in boosting the prediction accuracy with the help of significant features.

In this article, we introduce the combination of an ensemble classifier with an FS method—the resampling-based lasso feature selection (RLFS) method for ranking features, and ensemble of regularized regression models (ERRM) for classification purposes. The resampling approach was proven to be one of the best FS screening steps in a high-dimensional data setting [13]. The RLFS uses the selection probability with lasso penalty, and the threshold for selecting the top-ranked features is set using *b*-SIS condition; and these select features were applied to the ERRM to achieve the best prediction accuracy. The ERRM uses five individual regularization models, lasso, adaptive lasso, elastic net, SCAD, and MCP.

2. Materials and Methods

The FS method includes the proposed RLFS method, information gain, chi-square, and minimum redundancy maximum relevance. The classification methods include support vector machines, penalized regression models, and tree-based methods, such as random forests and adaptive boosting. The programs for all the experiments were written using R software [40]. The FS and classification were performed with the packages [41–46] obtained from CRAN. The weighted rank aggregation was evaluated with the RankAggreg package obtained from [47]. The codes for implementing the algorithms are available at [48]. The SMK-CAN-187 data were obtained from [49]; some of the applications of the data can be found in the articles [50,51] where the importance of screening approach in high dimensional data is elaborated.

2.1. Data Setup

To assess the performances of the models, we developed simulation study and also considered a real application of gene expression data.

2.1.1. Simulation Data Setup

The data were generated based on a random multivariate normal distribution where the mean was assigned as 0, and the variance-covariance matrix $\sum_x$ adapts a compound symmetry structure with the diagonal items set to 1 and the off-diagonal items being ρ values.

$$\sum_x = \begin{pmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{pmatrix}_{p\times p} . \tag{1}$$

The class labels were generated using the Bernoulli trails with the following probability:

$$\pi_i(y_i = 1|x_i) = \frac{\exp(x_i\beta)}{1+\exp(x_i\beta)}. \tag{2}$$

The data matrix $x_i \sim N_p(0, \sum_x)$ was generated using the random multivariate normal distribution, and the response variable y_i was generated by binomial distribution, as shown in Equations (1) and (2) respectively. For sufficient comparison of the performance of the model and subsidizing the effects of the data splits, all of the regularized regression models were built using the 10-fold cross-validation procedure, and the averages were taken over 100 partitioning times referred to as 100 iterations in this paper. The data generated are high-dimensional in nature with the number of samples, n = 200 and total features, p = 1000. The true regression coefficients were set to 25, which were generated using uniform distribution with the minimum and maximum values 2 and 4, respectively.

With this setup of high-dimensional data, we simulated three different types of data, each with correlation structures ρ = 0.2, 0.5, and 0.8 respectively. These values show the low, intermediate, and high correlation structures in the datasets which are significantly similar to what we usually see in the gene expression or others among many types of data in the field of bioinformatics [13,52]. At first, the data were divided randomly into training and testing sets with 75% and 25% of samples respectively; 75% of the training data was given to the FS methods, which ranked the genes concerning their importance, and then the top-ranked genes were selected based on b-SIS condition. The selected genes were applied in all the classifiers. For standard comparison and mitigating the effects of the data splitting, all of the regularized regression models were built using the 10-fold cross-validation; the models were assessed for testing the performance with the testing data using different evaluation metrics, and averages were taken over 100 splitting times referred to as 100 iterations.

2.1.2. Experimental Data Setup

To test the performance of the proposed combination of ERRM with RLFS, and compare it with the rest of the combinations of FS and classifiers, the gene expression data SMK-CAN-187 were analyzed. The data include 187 samples and 19,993 genes obtained from smokers, which included 90 samples from those with lung cancer and 97 samples from those without lung cancer. This data is high-dimensional, with the number of genes being 19,993. The preprocessing procedures are necessary to handle these high-dimensional data. At first, the data were randomly divided into training and testing sets with 75% and 25% of samples respectively. As the first filtering step, 75% of the training data were given to the marginal maximum likelihood estimator (MMLE), to overcome the redundant noisy features, and the genes were ranked based on their level of significance. The ranked significant genes were next applied to the FS methods along with the proposed RLFS method as the second filtering step, and a final list of truly significant genes was obtained. These significant genes were applied to all the classification models along with the proposed ERRM classifier. All of the models were built using the 10-fold cross-validation. The average classification accuracy and Gmean of our proposed framework were tested using the test data. The above procedure was repeated for 100 times and the averages were taken.

2.1.3. Data Notations

Let the expression levels of features in ith sample be represented as $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$ for $i = 1, ..., n$, where n is the total number of samples and p is the total number of features. The response variables, $y_i \in \{0, 1\}$, where $y_i = 0$ means that ith individual is in the non disease group and $y_i = 1$ is disease group.

The original data x_i were split into 75% for the training set x_j and 25% for the testing set x_k. The training set $x_j = (x_{j1}, x_{j2}, ..., x_{jp})$ for $j = 1, ..., t$, where t is the number of training samples, the response variable y_j for the training set. The testing set $x_k = (x_{k1}, x_{k2}, ..., x_{kp})$ for $k = 1, ..., v$, where v is the number of testing samples; the response variable is y_k for the testing set. The classifiers are fitted on x_j, and the class labels y_j as training data set to predict the classification of y_k using x_k of the testing set.

The detailed procedure is as follows. The training data x_j were given to the FS methods, and the new reduced feature set $x_r = (x_{j1}, x_{j2}, ..., x_{jf})$ for $j = 1, ..., t$, where t was the samples included in training data, and f was the reduced number of features after the FS step. This reduced feature set x_r was used as new training data for building the classification models.

2.2. Rank Based Feature Selection Methods

With the gain in popularity of high dimensional data in bioinformatics, the challenges to deal with it also grew. In gene expression data, having large p and small n problems, the n represents the samples as patients and p represents the features as genes. Dealing with such a large number of genes that are generated by conducting large biological experiments involves computationally intensive tasks that become too expensive to handle. The performance drops when such a large number of genes are added to the model. To overcome this problem, employing FS methods becomes a necessity. In statistical machine learning, there are many FS methods developed to deal with the gene expression data. But most of the existing algorithms are not completely robust applications to the gene expression data. Hence, we propose an FS method that ranks the features based on some criteria explained in the next section. We also explain some other popular FS methods in classification problems, such as information gain, chi-square, and minimum redundancy maximum relevance.

2.2.1. Information Gain

The information gain (IG) method [15] is simple, and one of the widely used FS methods. This univariate FS method is used to assess the quantity of information shared between the training feature set $x_j = (x_{j1}, x_{j2}, ..., x_{jp})$ for $j = 1, ..., t$, where t is the number of training samples, for $g = 1, 2, ...p$, where g is the feature in p number of features, and the response variable y_j. It provides an ordered ranking of all the features having a strong correlation with the response variable that helps to obtain good classification performance.

The information gain between the gth feature in x_j and the response variable y_j is given as follows:

$$\mathrm{IG}(x_j; y_j) = \mathrm{H}(x_j) - \mathrm{H}(x_j|y_j), \tag{3}$$

where $H(x_j)$ is entropy of x_j and $H(x_j|y_j)$ is entropy of x_j given y_j. The entropy [53] of x_j is defined by the following equation:

$$\mathrm{H}(x_j) = \sum_{g \in x_j} \pi(g) \log(\pi(g)), \tag{4}$$

where g indicates discrete random variable x_j and $\pi(g)$ gives the probability of g on all values of x_j. Given the random variable y_j, the conditional entropy of x_j is:

$$\mathrm{H}(x_j|y_j) = \sum_{y \in y_j} \pi(y) \sum_{g \in x_j} \pi(g|y) \log(\pi(g|y)), \tag{5}$$

where $\pi(y)$ is the prior probability of y_j; $\pi(g|y)$ is conditional probability of g in a given y that shows the uncertainty of x_j given y_j.

$$\mathrm{IG}(x_j; y_j) = \sum_{g \in x_j} \sum_{y \in y_j} \pi(g,y) \log \frac{\pi(g,y)}{\pi(g)\pi(y)}, \tag{6}$$

where $\pi(g,y)$ is the joint probability of g and y . IG is symmetric such that $\mathrm{IG}(x_j; y_j) = \mathrm{IG}(y_j; x_j)$, and is zero if the variables x_j and y_j are independent.

2.2.2. Chi-Square Test

The chi-square test (Chi2) belongs to the category of the non-parametric test, which is used mainly in determining the significant relation between two categorical variables. As part of the preprocessing step, we used the "equal interval width" approach to transform the numerical variables into categorical counterparts. The "equal interval width" algorithm first divides the data into q intervals of equal size. The width of each interval is defined as: $w = (max - min)/q$ and the interval boundaries are determined by: $min + w, min + 2w, ..., min + (q-1)w$.

The general rule in Chi2 is that the features have a strong dependency on the class labels selected, and the features independent of the class labels are ignored.

From the training set, $x_j = (x_{j1}, ...x_{jp})$, $g = 1, 2, ...p$, where g is every feature in p number of features. Given a particular feature g with r different feature values [53], the Chi2 score of that particular feature can be calculated as:

$$\tilde{\chi}^2(g) = \sum_{j=1}^{r} \sum_{s=1}^{p} \frac{(O_{js} - E_{js})^2}{E_{js}}, \tag{7}$$

where O_{js} is the number of instances with the j^{th} feature value given feature g. In addition, $E_{js} = \frac{O_{*s} O_{j*}}{O}$, where O_{j*} indicates the number of data instances with the feature value given feature g, O_{*s} denotes the number of data instances in r, and p is total number of features.

When two features are independent, the O_{js} is closer to the expected count E_{js}; consequently, we will have smaller Chi2 score. On the other hand, the higher Chi2 score implies that the feature is more dependent on the response and it can be selected for building the model during training.

2.2.3. Minimum Redundancy Maximum Relevance

The minimum redundancy and maximum relevance method (MRMR) is built on optimization criteria of mutual information (redundancy and relevance); hence, it is also defined under mutual information based methods. If a feature has uniformly of expressions or if they are randomly distributed in different classes, its mutual information with such classes is null [18]. If a feature is expressed differentially for different classes, it should have strong mutual information. Hence, we use mutual information as a measure of the relevance of features. MRMR also reduces the redundant features from the feature set. For a given set of features, it tries to measure both the redundancy among features and relevance between features and class vectors.

The redundancy and relevance are calculated based on mutual information, which is as follows: We know that, in the training set x_j, $g = 1, ..., p$ represents every feature in x_j and y_j is the response variable.

$$I(g, y) = \sum_{g \in x_j} \sum_{y \in y_j} \log \frac{\pi(g, y)}{\pi(g)\pi(y)}. \tag{8}$$

In the following equation, for simplicity, let us consider the training set x_j as X and response variable y_j as Y. The objective function is shown below:

$$J_{\text{MRMR}}(X_S, Y) = \frac{1}{|S|} \sum_{i \in S} I(X_i, Y) - \frac{1}{|S|^2} \sum_{i,j \in S} I(X_i, X_j), \tag{9}$$

where S is the subset of selected features and X_i is the ith feature. The first term is a measure of relevance that is the sum of mutual information of all the selected features in the set S with respect to the output Y. The second term is measure of redundancy that is the sum of the mutual information between all the selected features in the subset S. By optimizing the Equation (9), we are maximizing the first term and minimizing the second term simultaneously.

2.3. Classification Algorithms

Along with gene selection, improving prediction accuracy when dealing with high-dimensional data has always been a challenging task. There is a wide range of popular classification algorithms used when dealing with high throughput data, such as tree-based methods [54], support vector machines, and penalized regression models [55]. These popular models are discussed briefly in this section.

2.3.1. Logistic Regression

Logistic regression (LR) is perhaps one of the primary and popular models used while dealing with binary classification problems [56]. Logistic regression for dealing with more than two classes is called multinomial logistic regression. The primary focus here is on the binary classification. Given the set of inputs, the output is a predicted probability that the given input point belongs to a particular class. The output is always within [0, 1]. Logistic regression is based on the assumption that the original input space can be divided into two separate regions, one for each class, by a plane. This plane helps to discriminate between the dots belonging to different classes and is called as linear discriminant or linear boundary.

One of the limitations is the number of parameters that can be estimated needs to be smaller and should not exceed the number of samples.

2.3.2. Regularized Regression Models

Regularization is a technique used in logistic regression by employing penalties to overcome the limitations of dealing with high-dimensional data. Here, we discuss the PLR models such as lasso, adaptive lasso, elastic net, SCAD, and MCP. These five methods are included in the proposed ERRM and also tested as independent classifiers for comparing performance with the ERRM.

The logistic regression equation:

$$\log \left(\frac{\pi(y_j = 1|x_j)}{1 - \pi(y_j = 1|x_j)} \right) = \beta_0 + \beta x_j, \tag{10}$$

where $j = 1....t$ and $\beta = (\beta_1 ... \beta_f)^T$.

From logistic regression Equation (10), the log-likelihood estimator is shown as below:

$$l(\beta, y_j) = \sum_{j=1}^{t} \{y_j \log(\pi(y_j = 1|x_j)) + (1 - y_j)\log(1 - \pi(y_j = 1|x_j))\}. \tag{11}$$

Logistic regression offers the benefit by simultaneous estimation of the probabilities $\pi(x_j)$ and $1 - \pi(x_j)$ for each class. The criterion for prediction is $I\{\pi(y_j = 1|x_j) \geq 0.5\}$, where $I(\cdot)$ is an indicator function.

The parameters for PLR are estimated by minimizing above function:

$$\hat{\beta}_{\text{PLR}} = \underset{\beta}{\text{argmin}} \left[-l(\beta, y_j) + p(\beta) \right], \tag{12}$$

where $p(\beta)$ is a penalty function, $l(\beta, y_j)$ is the log-likelihood function.

Lasso is a widely used method in variable selection and classification purposes in high dimensional data. It is one of the five methods used in the proposed ERRM for classification purposes. The LASSO penalized regression method is defined below:

$$\hat{\beta}_{\text{LASSO}} = \underset{\beta}{\text{argmin}} \left[-l(\beta, y_j) + \lambda \sum_{j=1}^{f} |\beta_j| \right], \tag{13}$$

where f is the reduced number of features; λ is the tuning parameter that controls the strength of the L1 penalty.

The oracle property [30] has consistency in variable selection and asymptotic normality. The lasso works well in subset selection; however, it lacks the oracle property. To overcome this, different weights are assigned to different coefficients: this describes a weighted lasso called adaptive lasso. The adaptive lasso (ALASSO) penalty is shown below:

$$\hat{\beta}_{\text{ALASSO}} = \underset{\beta}{\text{argmin}} \left[-l(\beta, y_j) + \lambda \sum_{j=1}^{f} w_j |\beta_j| \right], \tag{14}$$

where f is the reduced number of features, λ is the tuning parameter that controls the strength of the L2 penalty, and w_j is the weight vector based on ridge estimator. The ridge estimator [28] uses the L2 regularization method which obtains the size of coefficients by adding the L2 penalty.

The elastic net (ENET) [57] is the combination of lasso which uses the L1 penalty, and ridge which uses the L2 penalty. The sizable number of variables is obtained, which helps in avoiding the model turning into an excessively sparse model.

The ENET penalty is defined as:

$$\hat{\beta}_{\text{ENET}} = \underset{\beta}{\text{argmin}} \left[-l(\beta, y_j) + \lambda \left(\frac{1-\alpha}{2} \sum_{j=1}^{f} |\beta_j|^2 + \alpha \sum_{j=1}^{f} |\beta_j| \right) \right], \tag{15}$$

where λ is the tuning parameter that controls the penalty, f is the number of features, α is the mixing parameter between ridge $\alpha = 0$ and lasso $\alpha = 1$.

The smoothly clipped absolute deviation penalty (SCAD) [30] is a sparse logistic regression model with a non-concave penalty function. It improves the properties of the L1 penalty. The regression coefficients are estimated by minimizing the log-likelihood function:

$$\hat{\beta}_{\text{SCAD}} = \underset{\beta}{\text{argmin}} \left[-l(\beta, y_j) + \lambda \sum_{j=1}^{f} p_\lambda(\beta_j) \right]. \tag{16}$$

In Equation (16) the $p_\lambda(\beta_j)$ is defined by:

$$|\beta_i| I_{(|\beta_j| \leq \lambda)} + \left(\frac{\{(c^2-1)\lambda^2 - (c\lambda - |\beta_j|)_+^2\} I(\lambda \leq |\beta_j|)}{2(c-1)} \right), \; c > 2 \text{ and } \lambda \geq 0\,. \tag{17}$$

Minimax concave penalty (MCP) [31] is very similar to the SCAD. However, the MCP relaxes the penalization rate immediately, while for SCAD, the rate remains smooth before it starts decreasing. The MCP equation is given as follows:

$$\hat{\beta}_{\text{MCP}} = \underset{\beta}{\operatorname{argmin}} \left[-l(\beta, y_j) + \lambda \sum_{j=1}^{f} p_\lambda(\beta_j) \right]. \tag{18}$$

In Equation (18) the $p_\lambda(\beta_j)$ is defined as:

$$\left(\frac{2c\lambda|\beta_j| - \beta_j^2}{2c} \right) I(|\beta_j| \leq c\lambda) + \left(\frac{c\lambda^2}{2} \right) I(|\beta_j| > c\lambda), \text{ for } \lambda \geq 0 \text{ and } c > 1. \tag{19}$$

2.3.3. Random Forests

The random forest (RF) [23] is an interpretive and straightforward method commonly used for classification purposes in bioinformatics. It is also known for its variable importance ranking in high dimensional data sets. RF is built on the concept of decision trees. Decision trees are usually more decipherable when dealing with binary responses. The idea of RF is to operate as an ensemble instead of relying on a single model. RF is a combination of a large number of decision trees where each tree has some random subset of features obtained from the data by allowing repetitions. This process is called bagging. The majority voting scheme is applied by aggregating all the tree models and obtaining one final prediction.

2.3.4. Support Vector Machines

Support vector machines (SVM) [25] are well known amongst most of the mainstream algorithms in supervised learning. The main goal of a SVM is to choose a hyperplane that can best divide the data in the high dimensional space. This helps to avoid overfitting. The SVM detects the maximum margin hyperplane, the hyperplane that maximizes the distance between the hyperplane, and the closest dots [58]. The maximum margin indicates that the classes are well separable and correctly classified. It is represented as a linear combination of training points. As a result, the decision boundary function for classifying points as to hyperplane only involves dot products between those points.

2.3.5. Adaboost

Adaboost is also known as adaptive boosting (AB) [24]. It improves the performance of a particular weak boosting classifier through an iterative process. This ensemble learning algorithm can be extensively applied to classification problems. The primary objective here is to assign more weights to the patterns that are harder to classify. Initially, the same weights are assigned to each training item. The weights of the wrongly classified items are incremented while the weights of the correctly classified items are decreased in each iteration. Hence, with the additional iterations and more classifiers, the weak learner is bound to cast on the challenging samples of the training set.

2.4. The Proposed Framework

We propose a combination of the FS method and classification method. For the filtering procedure, the resampling-based lasso feature selection method is introduced, and for the classification, the ensemble of regularized regression models is developed.

2.4.1. The Resampling-Based Lasso Feature Selection

From [13], we see that the resampling-based FS is relatively more efficient in comparison to the other existing FS methods in gene expression data. The RLFS method is based on the lasso penalized regression method and the resampling approach employed to obtain the ranked important features using the frequency.

The least absolute shrinkage and selection operator (LASSO) [27] estimator is based on L1-regularization. The L1-regularization method limits the size of coefficients pushes the unimportant regression coefficients to zero by using the L1 penalty. Due to this property, variable selection is achieved. It plays a crucial role in achieving better prediction accuracy along with the gene selection in bioinformatics.

$$\hat{\beta}_{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left[-\sum_{j=1}^{t} \{ y_j \log(\pi(y_j = 1|x_j)) + (1 - y_j) \log(1 - \pi(y_j = 1|x_j)) \} + \lambda \sum_{j=1}^{p} |\beta_j| \right]. \tag{20}$$

The selection probability $S(f_m)$ of the features based on the lasso is shown in the below equation.

$$S(f_m) = \frac{1}{R} \sum_{i=1}^{R} \frac{1}{L} \sum_{j=1}^{L} I(\beta_{ijm} \neq 0), \text{ for } m = 1, 2, ..., p. \tag{21}$$

The b-SIS criteria to select the top k ranked features is defined by,

$$\left\lceil b \times \frac{n}{\log(n)} \right\rceil, \tag{22}$$

where R is defined by the total number of resampling, L is total number of λ values, f_m is the feature indexed as i, p is total number of features, n is total number of samples, and β_{ijm} is defined as regression coefficient of mth feature and $I()$ indicator variable. Each R number of resamples and L number of values of λ are considered to build the variable selection model. The 10-fold cross validation is considered while building the model.

After ranking the features using the RLFS method, we employ the b-SIS approach to select the top features based on Equation (22) where b is set to two. The number of true important variables selected among the top b-SIS ranked features is calculated in each iteration and the average of this is taken over 100 iterations.

2.4.2. The Ensembles of Regularized Regression Models

LASSO, ALASSO, ENET, SCAD, and MCP are the five individual regularized regression models included as base learners in our ERRM. The role of bootstrapped aggregation or bagging is to reduce the variance by averaging over an "ensemble" of trees, which will improve the performance of weak classifiers. $B = B_1^k, ..., B_M^k$ is the number of random bootstrapped samples obtained from reduced training set x_r with corresponding class label y_j. The five regularized regression models are trained on each bootstrapped sample B named sub-training data, leading to 5 × B models. These five regularized models are then trained using the 10-fold cross-validation to predict the classes on the out of bag samples called sub-testing data where the best model fit in each of the five regularized regression model is obtained. Henceforth, in each of the five regularized models, the best model is selected and the testing data x_k is applied to obtain the final list of predicted classes for each of these models. For binary classification problems, in addition to accuracy, the sensitivity and specificity are primarily sought. The E evaluation metrics are computed for each of these best models of five regularized models. In order to get an optimized classifier using all the evaluation measures E is essential, and this is achieved using weighted rank aggregation. Here, each of the regularized models is ranked based on the performance of E evaluation metrics. The models are ranked based on the increasing order of performance; in the case of a matching score of accuracy for two or more models, other metrics such as sensitivity and specificity are considered. The best performing model among the five models is obtained based on these ranks. This procedure is repeated to obtain the best performing model in each of the tree T. Finally, the majority voting procedure is applied over the T trees to obtain a final list of predicted classes. The test class label is applied to measure the final E measures for assessing the performance of the proposed ensembles. The Algorithm 1 defines the proposed ERRM procedure.

The complete workflow of the proposed RLFS-ERRM framework is shown in Figure 1.

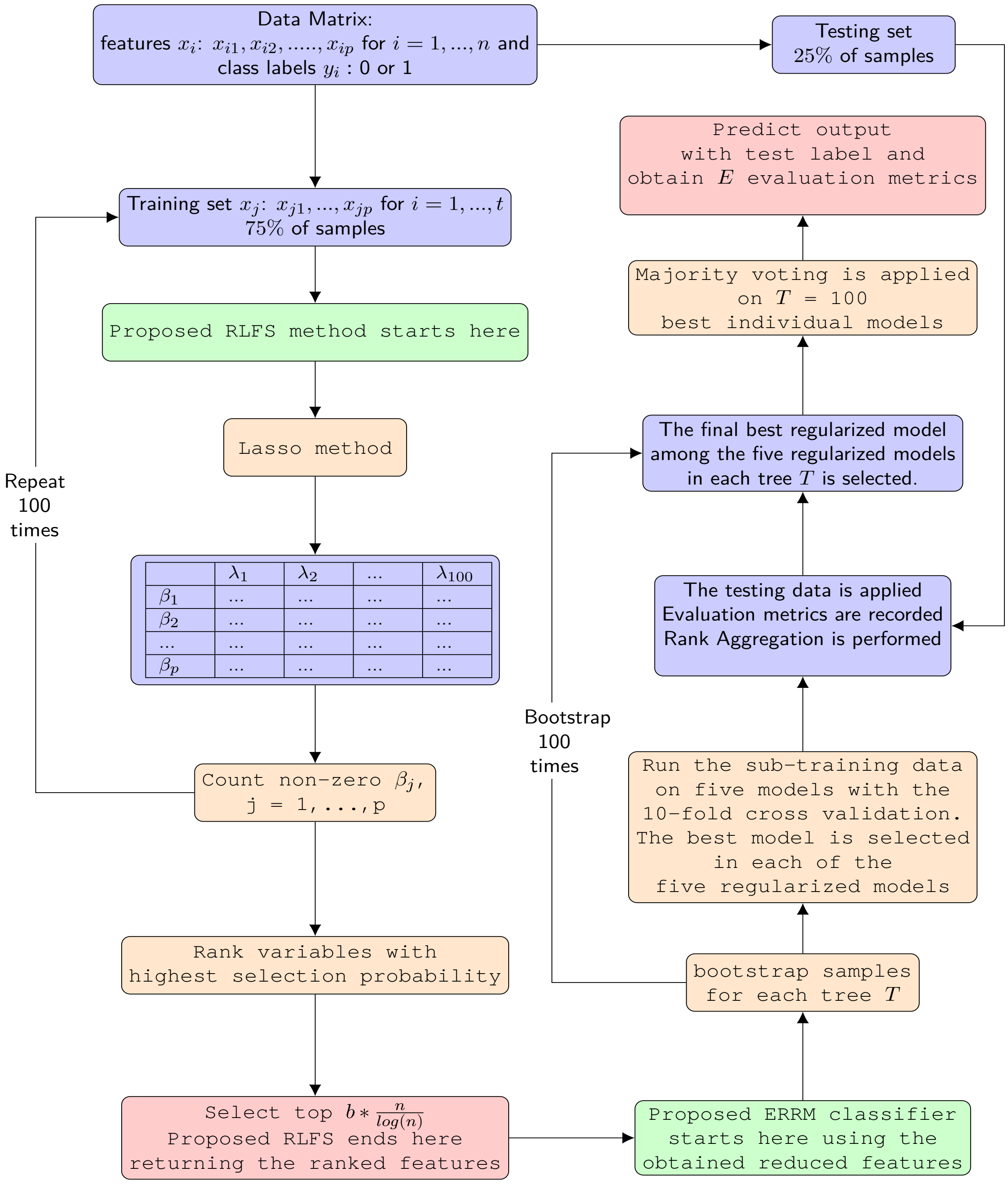
Data Matrix:
features x_i: $x_{i1}, x_{i2}, ..., x_{ip}$ for $i = 1, ..., n$ and
class labels y_i : 0 or 1
Testing set
25% of samples
Predict output
with test label and
obtain E evaluation metrics
Training set x_j: $x_{j1}, ..., x_{jp}$ for $i = 1, ..., t$
75% of samples
Majority voting is applied
on T = 100
best individual models
Proposed RLFS method starts here
Lasso method
The final best regularized model
among the five regularized models
in each tree T is selected.
Repeat
100
times
λ_1
λ_2
...
λ_{100}
β_1
β_2
...
β_p
The testing data is applied
Evaluation metrics are recorded
Rank Aggregation is performed
Bootstrap
100
times
Count non-zero β_j,
j = 1, . . . , p
Run the sub-training data
on five models with the
10-fold cross validation.
The best model is selected
in each of the
five regularized models
Rank variables with
highest selection probability
bootstrap samples
for each tree T
Select top $b * \frac{n}{log(n)}$
Proposed RLFS ends here
returning the ranked features
Proposed ERRM classifier
starts here using the
obtained reduced features

Figure 1. The complete workflow depicting the proposed combination of RLFS-ERRM framework.

Algorithm 1 Proposed ERRM

```
Step 1: Obtain new training data $x_r$ with most informative features using the proposed RLFS method.
Step 2: Draw bootstrap samples from $x_r$ and apply them to each of the regularized methods to be fitted with 10-fold cross validation.
Step 3: Apply out of bag samples (OOB) not used in bootstrap samples to the above fitted models to choose the best model using $E$ evaluation metrics.
Step 4: Repeat steps 2 and 3 until getting 100 bootstrap models.
Step 5: Apply testing set $x_k$ to each of 100 models to aggregate votes of classification.
Step 6: Predict classification of each sample by the rule of majority voting in the testing set.
```

2.5. Evaluation Metrics

We evaluated the results of combinations of FS methods with the classifier using accuracy and geometric mean (Gmean). The metrics are detailed with respect to true positive (TP), true negative (TN), false negative (FN), and false positive (FP). The equations for accuracy and Gmean are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{23}$$

$$\text{Gmean} = \sqrt{\text{Sensitivity} \times \text{Specificity}},$$

where the sensitivity and specificity are given by:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{and} \quad \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{24}$$

3. Results

3.1. Simulation Results

The prediction performance of any given model is largely dependent on the type of the features. The features affecting the classification will help in attaining the best prediction accuracies. In Figure 2, we see the RLFS method with the top-ranked features based on the b-SIS criterion includes a higher number of true important features than other existing FS methods, such as IG, Chi2, and MRMR used for comparison in this study. The proposed RLFS performs consistently better across low, medium, and highly correlated simulated data, and the positive effect of having more true important variables was seen in all three simulation scenarios (further explained in detail).

3.1.1. Simulation Scenario (S1): Low Correlation 0.2

The predictors were generated, having a low correlation structure with ρ = 0.2. The proposed classifier ERRM performs better than the existing classifier on all the FS methods: proposed RLFS, IG, Chi2, and MRMR. Also, the proposed combination of ERRM classifier with RLFS method, with the accuracy and Gmean, each of which is 0.8606 and 0.8626 respectively, is relatively better in comparison to other combinations of FS method and classifier such as RLFS-LASSO, RLFS-ALASSO, RLFS-ENET, and the other remaining combinations, as observed in Figure 3. The combination of the FS method IG with proposed ERRM with an accuracy of 0.8476 is also seen performing better than IG-LASSO, IG-ALASSO, IG-ENET, IG-SCAD, IG-MCP, IG-AB, IG-RF, IG-LR, and IG-SVM. Similarly, the combination of Chi2-ERRM with an accuracy of 0.8538 is seen better than FS method Chi2 with the other remaining classifiers. The results are reported in Table 1. The combination of MRMR-ERRM has an accuracy of 0.8550 and Gmean of 0.8552 is better than the combination of FS method MRMR with the rest of the nine classifiers.

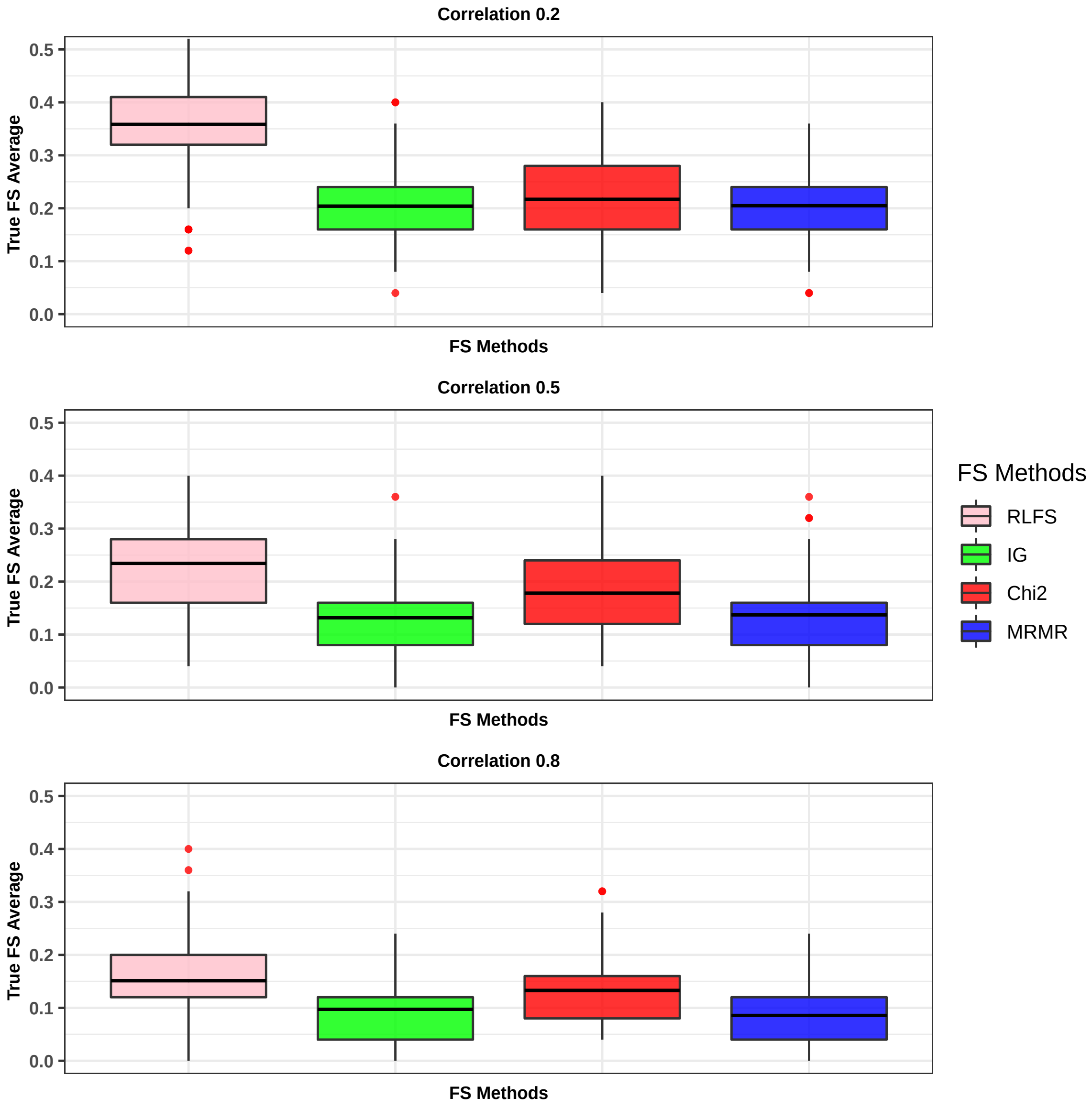

Figure 2. True number of features selected among top *b*-SIS ranked features, and the average of this taken over 100 iterations for three different scenarios. The first horizontal line in the box shows the first quartile and the second horizontal dark line which usually represents the median values are shown as the mean values in this article. The third horizontal line in each of the boxes shows the third quartile. The red dotted circles indicate the outliers in each of the FS methods.

All the classifiers with the features obtained from the RLFS method achieved the best accuracies in comparison to other FS methods, as seen in Figure 3. The combination of RLFS with SVM showed the second-best performance by attaining an accuracy of 0.8582, as seen in Table 1. The ENET method showed the best performance among all the regularized regression models with all the FS methods, and the best accuracy was obtained with the proposed RLFS method.

The proposed combination of RLFS-ERRM has better performance than the other existing combinations of the FS and classifier without the proposed FS method RLFS and classifier ERRM itself. For example, the existing FS methods IG, Chi2, and MRMR with the eight existing individual classifiers' performances are lower than the proposed RLFS-ERRM combination, as shown in Table 1.

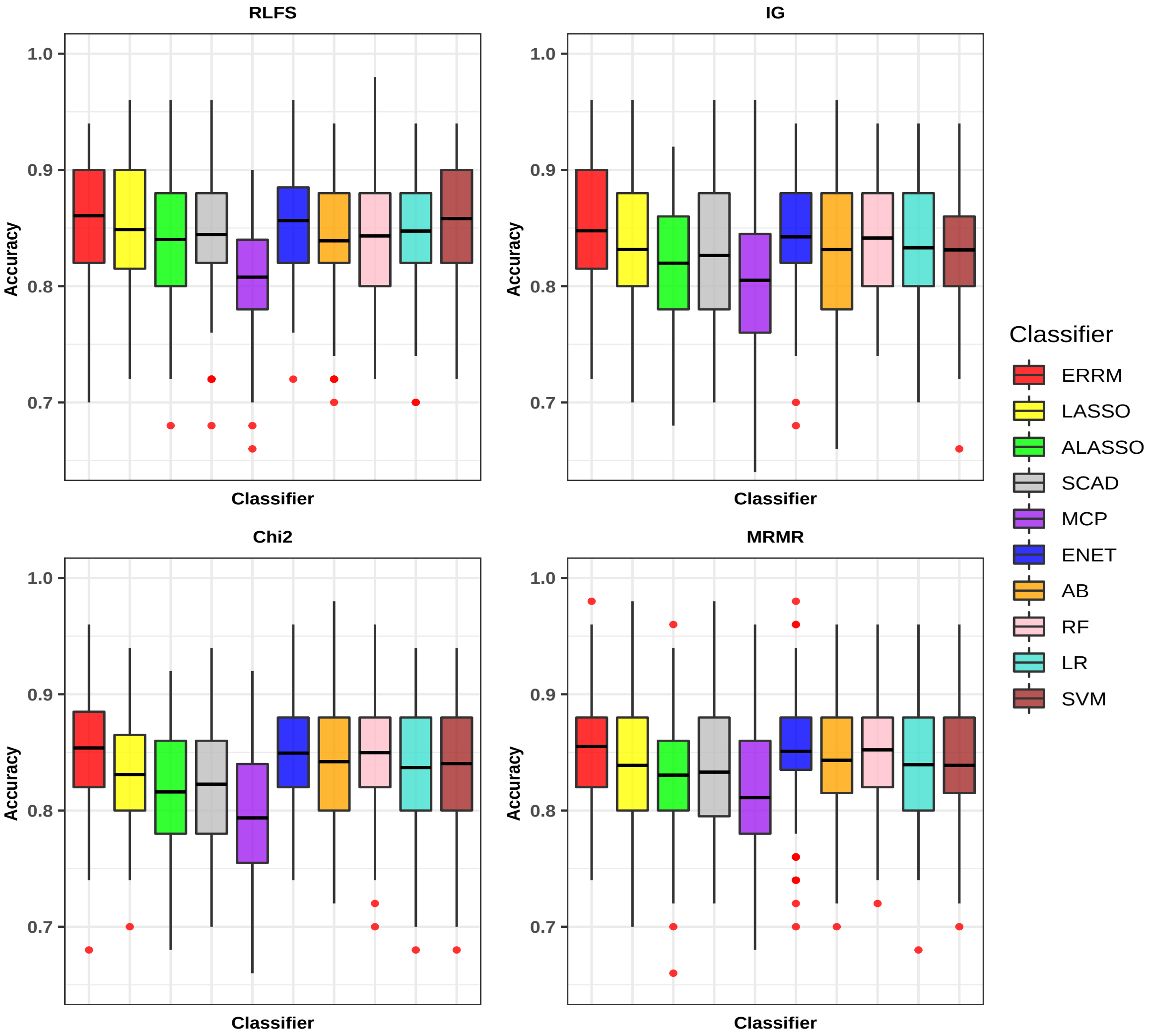

Figure 3. Comparison of accuracies of proposed combination of ensemble of regularized regression models (ERRM) with resampling-based lasso feature selection (RLFS) and other classifiers with feature selection methods when correlation = 0.2.

3.1.2. Simulation Scenario (S2): Intermediate Correlation 0.5

The predictor variables were generated using a medium correlation structure with $\rho = 0.5$. The proposed combination of the RLFS method and ERRM classifier, with the accuracy and Gmean, each of which is 0.9256 and 0.9266, respectively, attained relatively better performance compared to other combinations of the FS method and classifier such as RLFS-LASSO, RLFS-ALASSO, RLFS-ENET and the other remaining combinations. The results are shown in Table 1. From Figure 4, we see that the proposed ensemble classifier ERRM with other FS methods such as IG, Chi2, and MRMR performs best compared to the other nine individual classifiers.

The SVM and ENET classifiers with the RLFS method attained accuracies that are almost similar to the proposed combination of ERRM-RLFS. However, when Gmean is considered, the ERRM-RLFS outperforms the SVM combinations. The average SD of the proposed combination of the ERRM-RLFS is smaller than other combinations of the FS method and classifier. The accuracies of SVM and ENET classifiers with the IG method were 0.9128 and 0.9150 lower compared to the ERRM classifier with the IG method which had an accuracy of 0.9184. Similarly, the ERRM with the Chi2 method showed relatively better performance than the competitive classifiers ENET and SVM. Further, the ERRM classifier with the MRMR method having an accuracy of 0.9174 showed better performance than ENET, SVM, and other top-performing individual classifiers.

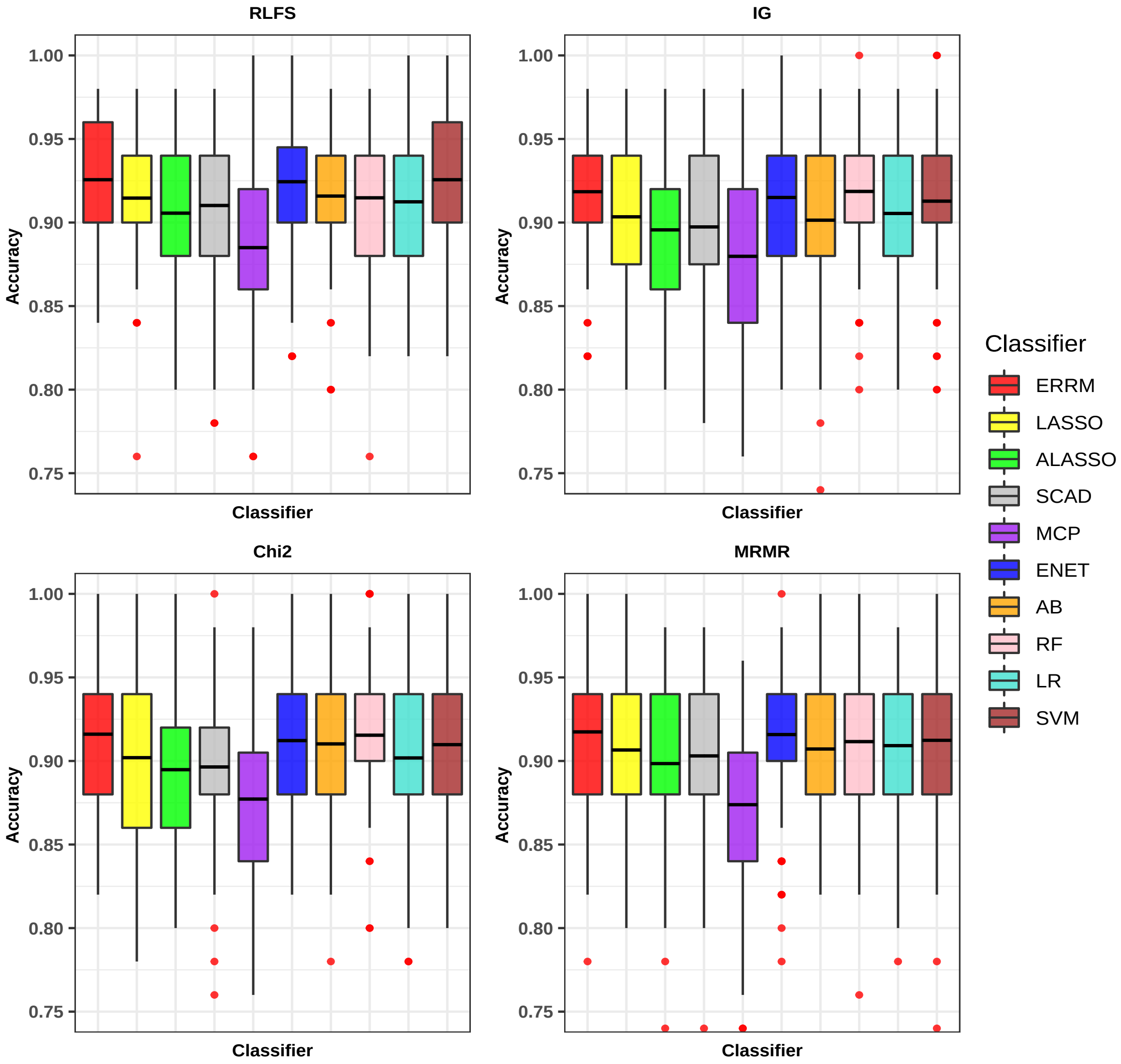

Figure 4. Boxplot showing the accuracies of classifiers with FS methods when correlation = 0.5.

While the SVM and ENET classifiers showed promising performance on the RLFS that had a good number of important features, they failed to show the same consistency on the other FS methods. On the other hand, the ensemble ERRM showed robust behavior, with being able to withstand the noise that helps in attaining better prediction accuracies and Gmean, not only with the RLFS method but also with other FS methods, such as IG, Chi2, and MRMR, as seen in Table 1.

Similar results are also found in the Simulation Scenario (S3): which has the highly correlated data with ρ set to 0.8. The results for this scenario are described in the Appendix A.

3.2. Experimental Results

Figure 5 shows the box plot of average accuracies taken over 100 iterations for all the combinations of FS and classifiers in experimental data. Each of the sub-figures in the figure shows the classifiers with the corresponding FS methods. As seen in Table 2, the performance of all the individual classifiers when applied on the RLFS method—the accuracy and Gmean—are relatively much better than the accuracies of the individual classifiers when applied on the IG, Chi2, and MRMR methods.

Table 1. Classification performance of proposed RLFS with ERRM compared to other combinations of feature selection methods with classifiers over 100 iterations.

		Proposed RLFS		**IG**		**Chi2**		**MRMR**	
Correlation	**Classifier**	**Acc (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**
0.2	Proposed ERRM	0.8606 (0.049)	0.8626 (0.073)	0.8476 (0.052)	0.8483 (0.079)	0.8538 (0.053)	0.8551 (0.071)	0.8550 (0.049)	0.8552 (0.075)
	LASSO	0.8486 (0.052)	0.8504 (0.075)	0.8316 (.054)	0.8335 (0.083)	0.8310 (0.052)	0.8323 (0.071)	0.8388 (0.051)	0.8393 (0.077)
	ALASSO	0.8402 (0.054)	0.8416 (0.077)	0.8198 (0.051)	0.8217 (0.079)	0.8160 (0.053)	0.8171 (0.075)	0.8304 (0.051)	0.8313 (0.079)
	ENET	0.8564 (0.048)	0.8584 (0.072)	0.8424 (0.054)	0.8441 (0.081)	0.8494 (0.046)	0.8509 (0.067)	0.8508 (0.052)	0.8508 (0.077)
	SCAD	0.8440 (0.054)	0.8457 (0.080)	0.8264 (0.057)	0.8283 (0.086)	0.8226 (0.061)	0.8239 (0.077)	0.8330 (0.056)	0.8336 (0.081)
	MCP	0.8078 (0.049)	0.8095 (0.081)	0.8050 (0.062)	0.8074 (0.088)	0.7936 (0.060)	0.7952 (0.085)	0.8110 (0.060)	0.8126 (0.082)
	AB	0.8390 (0.051)	0.8224 (0.077)	0.8314 (0.060)	0.8328 (0.080)	0.8422 (0.054)	0.8435 (0.075)	0.8432 (0.054)	0.8437 (0.075)
	RF	0.8432 (0.057)	0.8467 (0.084)	0.8414 (0.052)	0.8435 (0.078)	0.8498 (0.053)	0.8520 (0.075)	0.8522 (0.051)	0.8534 (0.077)
	LR	0.8474 (0.050)	0.8489 (0.076)	0.8330 (0.053)	0.8346 (0.080)	0.8370 (0.054)	0.8380 (0.073)	0.8394 (0.051)	0.8394 (0.080)
	SVM	0.8582 (0.049)	0.8595 (0.070)	0.8312 (0.052)	0.8320 (0.083)	0.8404 (0.054)	0.8416 (0.074)	0.8388 (0.049)	0.8378 (0.084)
0.5	Proposed ERRM	0.9256 (0.037)	0.9266 (0.053)	0.9184 (0.039)	0.9195 (0.059)	0.9160 (0.038)	0.9165 (0.056)	0.9174 (0.042)	0.9176 (0.056)
	LASSO	0.9146 (0.037)	0.9155 (0.053)	0.9034 (0.045)	0.9046 (0.061)	0.9020 (0.043)	0.9029 (0.063)	0.9066 (0.045)	0.9065 (0.062)
	ALASSO	0.9056 (0.039)	0.9062 (0.056)	0.8956 (0.044)	0.8966 (0.065)	0.8948 (0.046)	0.8954 (0.065)	0.8984 (0.046)	0.8982 (0.062)
	ENET	0.9244 (0.038)	0.9253 (0.052)	0.9150 (0.044)	0.9163 (0.061)	0.9122 (0.039)	0.9130 (0.060)	0.9158 (0.043)	0.9155 (0.058)
	SCAD	0.9102 (0.041)	0.9110 (0.060)	0.8974 (0.046)	0.8986 (0.063)	0.8964 (0.045)	0.8972 (0.065)	0.9090 (0.045)	0.9090 (0.059)
	MCP	0.8850 (0.047)	0.8855 (0.066)	0.8798 (0.050)	0.8813 (0.068)	0.8772 (0.045)	0.8782 (0.065)	0.8738 (0.049)	0.8738 (0.070)
	AB	0.9158 (0.035)	0.9166 (0.050)	0.9014 (0.046)	0.9027 (0.065)	0.9102 (0.040)	0.9112 (0.060)	0.9072 (0.047)	0.9075 (0.062)
	RF	0.9148 (0.039)	0.9166 (0.055)	0.9186 (0.041)	0.9199 (0.059)	0.9154 (0.042)	0.9167 (0.060)	0.9116 (0.043)	0.9127 (0.060)
	LR	0.9124 (0.037)	0.9127 (0.054)	0.9054 (0.043)	0.9063 (0.061)	0.9018 (0.045)	0.9024 (0.063)	0.9092 (0.043)	0.9084 (0.060)
	SVM	0.9256 (0.038)	0.9261 (0.054)	0.9128 (0.038)	0.9135 (0.056)	0.9080 (0.043)	0.9099 (0.061)	0.9126 (0.045)	0.9120 (0.062)

When we look at the performances of all the classifiers with the IG method in comparison to other FS methods, there is much variation in the accuracies, as seen in Figure 5. The SVM classifier, which attained the accuracy of 0.7026 with the RLFS method, dropped to 0.6422 with the IG method.

The proposed combination of the ERRM classifier with the RLFS method achieved the highest average accuracy of 0.7161, and the Gmean of 0.7127 outperformed the rest of the combinations of classifier with the FS method. The RLFS method is also a top-performing FS method on all individual classifiers. However, among the other FS methods, the MRMR method, when applied to all the individual classifiers, showed relatively much better performance than the application of IG and Chi2 methods to the individual classifiers. The second best-performing method is the ENET-RLFS combination, which had an accuracy of 0.7138. The SVM-IG combination showed the lowest performance with an accuracy of 0.6422 among all the combinations of the classifier with FS methods, as shown in Table 2.

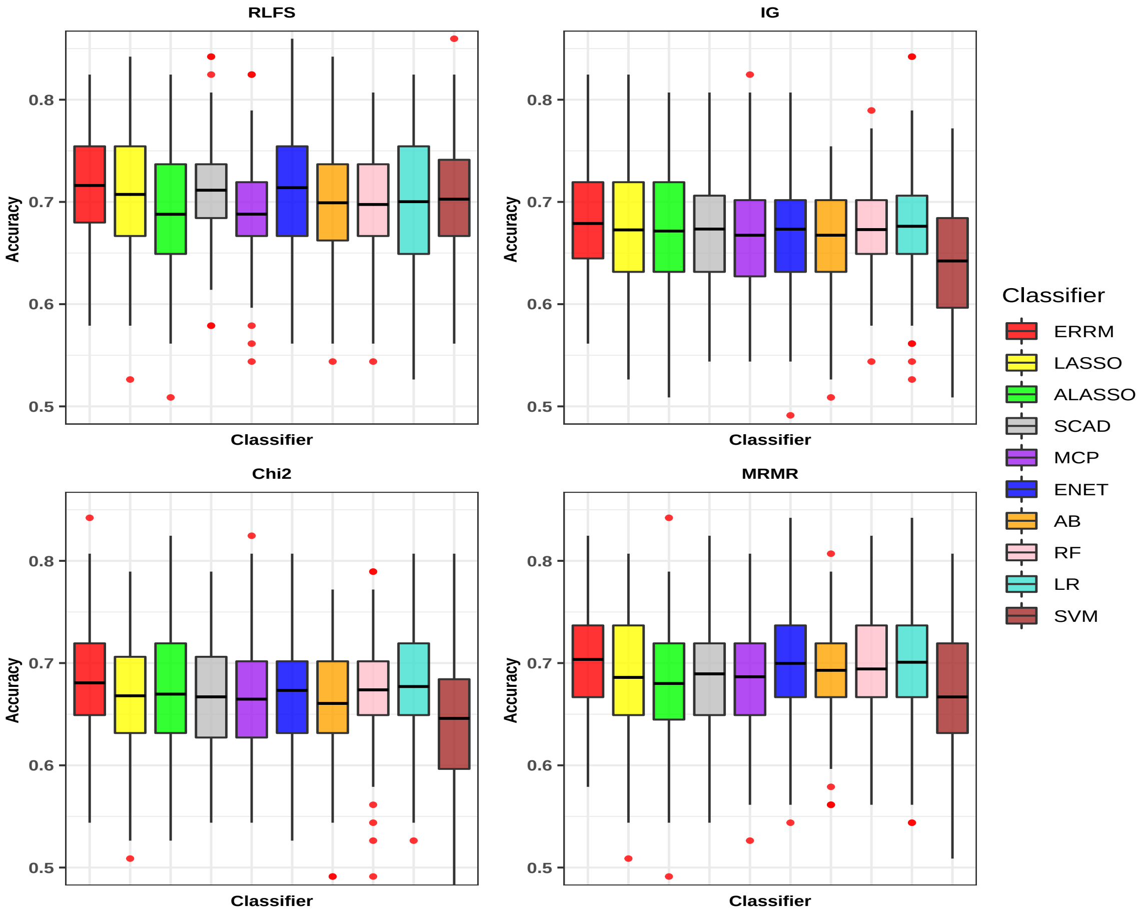

Figure 5. Boxplot showing the accuracies of classifiers with FS methods in experimental data SMK-CAN-187.

Table 2. Average values taken over 100 iterations in experimental data: SMK-CAN-187.

Classifier	**Proposed RLFS**		**IG**		**Chi2**		**MRMR**	
	Accuracy (SD)	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**
Proposed ERRM	0.7161 (0.053)	0.7127 (0.082)	0.6789 (0.056)	0.6791 (0.091)	0.6807 (0.056)	0.6808 (0.091)	0.7035 (0.056)	0.7024 (0.087)
LASSO	0.7073 (0.064)	0.7058 (0.087)	0.6726 (.060)	0.6725 (0.095)	0.6680 (0.057)	0.6680 (0.090)	06859 (0.061)	0.6871 (0.097)
ALASSO	0.6878 (0.065)	0.6869 (0.091)	0.6715 (0.060)	0.6714 (0.094)	0.6696 (0.064)	0.6698 (0.092)	0.6800 (0.059)	0.6803 (0.092)
ENET	0.7138 (0.061)	0.7116 (0.085)	0.6733 (0.057)	0.6722 (0.093)	0.6733 (0.052)	0.6726 (0.090)	0.6998 (0.061)	0.6992 (0.095)
SCAD	0.7114 (0.054)	0.7098 (0.083)	0.6735 (0.056)	0.6732 (0.090)	0.6670 (0.058)	0.6669 (0.091)	0.6894 (0.059)	0.6901 (0.091)
MCP	0.6880 (0.010)	0.6870 (0.082)	0.6673 (0.057)	0.6663 (0.089)	0.6647 (0.059)	0.6639 (0.092)	0.6866 (0.057)	0.6874 (0.089)
AB	0.6991 (0.064)	0.6958 (0.087)	0.6673 (0.054)	0.6634 (0.086)	0.6605 (0.058)	0.6583 (0.094)	0.6929 (0.050)	0.6897 (0.083)
RF	0.6975 (0.056)	0.6933 (0.089)	0.6729 (0.045)	0.6691 (0.078)	0.6738 (0.054)	0.6703 (0.090)	0.6942 (0.055)	0.6902 (0.088)
LR	0.7001 (0.065)	0.6987 (0.089)	0.6761 (0.058)	0.6662 (0.097)	0.6770 (0.059)	0.6769 (0.094)	0.7008 (0.058)	0.7000 (0.086)
SVM	0.7026 (0.058)	0.7014 (0.086)	0.6422 (0.059)	0.6430 (0.099)	0.6459 (0.066)	0.6477 (0.105)	0.6668 (0.058)	0.6658 (0.092)

For assessing the importance of bootstrapping and FS screening of the proposed framework, we measured the performance of ERRM without FS screening. The results in Table 3 shows the results of ensembles method with and without bootstrapping procedure. We see that having the bootstrapping approach which is random sampling with replacement is a better approach in the ensembles.

Table 3. Comparison of proposed ERRM with and without bootstrapping.

	Bootstrapping	**Accuracy (SD)**	**Gmean (SD)**
ERRM without feature selection (FS) screening	Yes	0.7129 (0.053)	0.7093 (0.091)
ERRM without FS screening	No	0.6947 (0.057)	0.6944 (0.089)

The performance of the regularized regression models used in the proposed ensembles algorithm is tested with the FS screening method and without the FS screening method. In the former approach, the regularized regression models were built and tested using the proposed RLFS screening method with the selected amount of significant features, whereas in the latter approach, the regularized models used all the features for building the model. The performances of the penalized models with the FS screening showed better accuracies and Gmean than without FS screening, as reported in Table 4.

Table 4. Comparison of regularized regression models used in the ERRM with and without FS screening.

	FS Screening	**Accuracy (SD)**	**Gmean (SD)**
LASSO	Yes	0.7073 (0.064)	0.7058 (0.087)
	No	0.6740 (0.061)	0.6752 (0.125)
ALASSO	Yes	0.6878 (0.065)	0.6869 (0.091)
	No	0.6740 (0.061)	0.6752 (0.125)
ENET	Yes	0.7138 (0.061)	0.7116 (0.085)
	No	0.6740 (0.061)	0.6752 (0.125)
SCAD	Yes	0.7114 (0.054)	0.7098 (0.083)
	No	0.6740 (0.061)	0.6752 (0.125)
MCP	Yes	0.6880 (0.010)	0.6870 (0.082)
	No	0.6740 (0.061)	0.6752 (0.125)

4. Discussion

We investigated the performance of the proposed combination of ERRM with the RLFS method using simulation studies and a real data application. The RLFS method ranks the features by employing the lasso method with a resampling approach and the *b*-SIS criteria to set the threshold for selecting the optimal number of features, and these features are applied on the ERRM classifier, which uses bootstrapping and rank aggregation to select the best performing model across the bootstrapped samples and helps in attaining the best prediction accuracy in a high dimensional setting. The ensemble framework ERRM was built using five different regularized regression models. The regularized regression models are known for having the best performances in terms of variable selection and prediction accuracy on gene expression data.

To show the performance of our proposed framework, we used three different simulation scenarios with low, medium, and high correlation structures that matched the gene expression data. To further illustrate our point, we also used SMK-CAN-187 data. Figure 2 shows the boxplots of the average number of true important features, where the RLFS shows higher detection power than the other FS methods such as IG, Chi2, and MRMR. From the results of both simulation studies and experimental data, we showed that all the individual classifiers with the RLFS method performed much better compared to the IG, Chi2, and MRMR. We also observed that all the individual classifiers showed much instability with the other three FS methods. This means that the individual classifiers do not work well with more noise and less true important variables in the model. The SVM and ENET classifiers with all the FS methods performed a little better among all the classifiers. However, the performance was relatively still low in comparison to the proposed ERRM classifier with every FS method. The tree-based ensemble methods RF and AB with RLFS also attained good accuracies but were not the best compared to the ERRM classifier.

The proposed ERRM method was assessed with the FS screening and without the FS screening step along with the bootstrapping option. The ERRM with FS screening and bootstrapping approach works better than ERRM without the FS screening and bootstrapping technique. Also, the results from Table 3 show that the ensemble with bootstrapping is a better approach to both the filtered and unfiltered data. On comparing the performance of the individual regularized regression models used in the ensembles, the individual models with the proposed RLFS screening step showed comparatively better accuracy in comparison to the individual regularized regression models without the FS screening. This means that using the reduced number of significant features with RLFS is a better approach instead of using all the features from the data.

The importance of FS method was not addressed in any of the ensemble approaches [37–39], and the classification accuracies achieved by the corresponding proposed methods were much closer to the accuracies attained by existing approaches. In this paper, we compared the various combination of FS methods with different classifiers. The ERRM showed better overall performance not only with the RLFS but also with the other FS methods compared in this study. This means that the ERRM is robust and works much better on the highly correlated gene expression data. The rule of thumb fpr attaining the best prediction accuracy is that more the true important variables, better the prediction accuracy. Henceforth, from the results of simulation and experimental data, we see that the proposed combination of RLFS-ERRM is better compared to the other existing combinations of FS and classifiers, as seen in the Tables 1 and 2. The proposed ERRM classifier showed the best performance across all the FS methods, with the highest performance achieved with the RLFS method. The proposed RLFS method attained a higher number of significant features compared to other FS methods. However, the drawback is that with the increase in the correlation structure, there is a decreasing performance in selecting the significant features, as shown in Figure 2. The ensembles algorithms are known to be computationally expensive [39] because of the tree-based nature. However, in our proposed framework, before the ensembles of ERRM, we apply FS methods to remove the irrelevant features and keep significant features. This filtering step not only helps with improving prediction accuracy but also with overcoming the drawback of computational time required, as the number of features processed becomes lower.

5. Conclusions

In this paper, we proposed a combination of the ensembles of regularized regression models (ERRM) with resampling-based lasso feature selection (RLFS) for attaining better prediction accuracies in high dimensional data. We conducted extensive simulation studies where we showed the better performance of RLFS in detecting the significant features than other competitive FS methods. The ensemble classifier ERRM also showed better average prediction accuracy with the RLFS, IG, Chi2, and MRMR compared to other classifiers with these FS methods. We also saw an improved performance in the ensemble method when used with bootstrapping. On comparing the performances of individual regularized regression models, all the models showed an increase in their accuracies with the FS screening approach. In both the simulation study and the experimental data SMK-CAN-187, the better performance was achieved by the proposed combination of RLFS and ERRM compared to all other combinations of FS and classifiers. The minor drawback in the proposed framework is that, in the case of highly correlated data, there is smaller number of significant features selected with all the FS methods. As future work, we plan to focus on improving the detecting power of true important genes with the new FS method.

Author Contributions: Conceptualization, S.K. and A.R.P.; formal analysis and investigation, A.R.P.; Writing—Original draft preparation, A.R.P.; supervision and reviewing, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We would like to thank the editors and three anonymous reviewers for their insightful comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FS	feature Selection
RLFS	resampling-based lasso feature selection
ERRM	ensemble regularized regression models
IG	information gain
Chi2	chi-square
MRMR	minimum redundancy maximum relevance
ALASSO	adaptive lasso
AB	adaptive boosting
RF	random forests
LR	logistic regression
SVM	support vector machines
SD	standard deviation

Appendix A

The data are generated based on a high correlation data structure with ρ = 0.8. The performance of the proposed combination of RLFS-ERRM is relatively better than the other combinations of the FS methods and classifiers. The results for simulation scenario S3 are shown in Figure A1. The average accuracies and Gmeans for all the FS and classifiers are noted in Table A1. The SVM and ENET classifiers with all the FS methods showed a little better performance among all individual classifiers. However, the accuracies and Gmeans attained by the proposed ensemble classifier ERRM with the FS methods RLFS, IG, and Chi2 were relatively better compared to the individual classifiers with FS methods. The best performance was achieved by the proposed RLFS-ERRM combination with an accuracy of 0.9586 and Gmean of 0.9596. The second-best performing combination was MRMR-SVM. The lowest performance in terms of accuracy and the Gmean was shown by Chi2-MCP. The MCP classifier has the lowest accuracy with all the FS methods. This explains why the MCP does not perform well when the predictor variables are highly correlated.

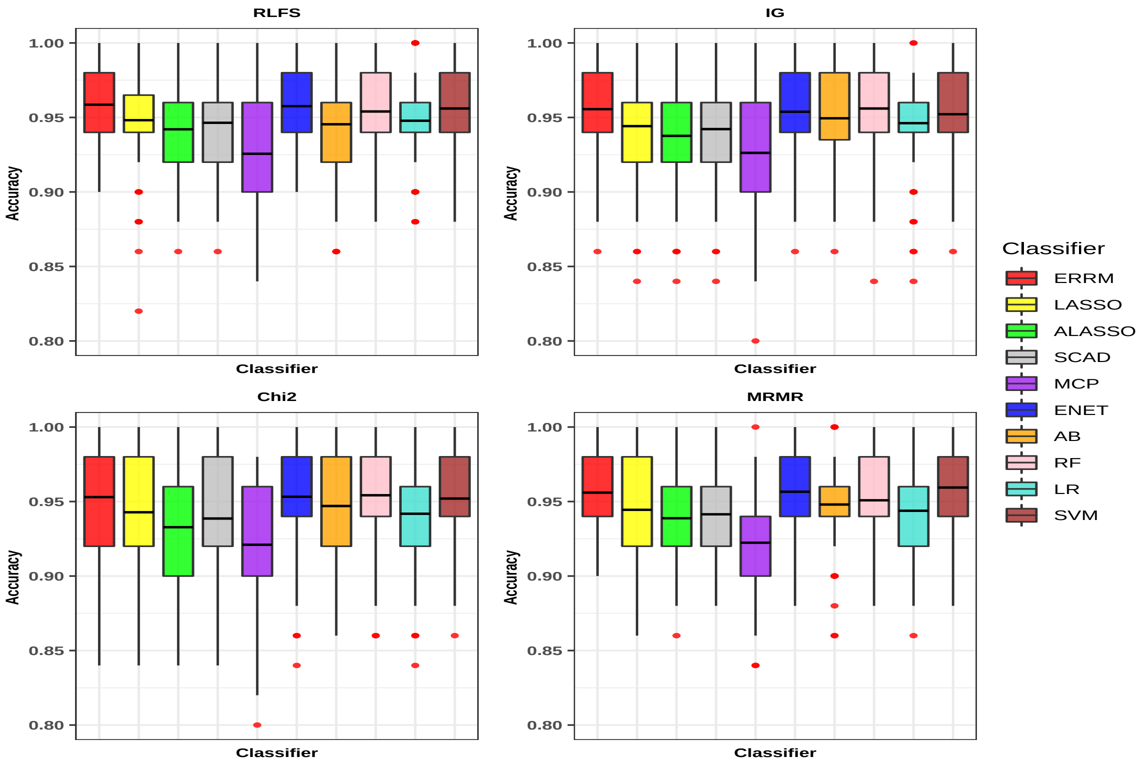

Figure A1. Boxplot showing the accuracies of Classifiers with FS methods in simulation scenario: S3 (Correlation = 0.8).

Table A1. Average values taken over 100 iterations in simulation scenario: S3 (High correlation: 0.8).

Classifier	Proposed RLFS		IG		Chi2		MRMR	
	Accuracy (SD)	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**	**Accuracy (SD)**	**Gmean (SD)**
Proposed ERRM	0.9586 (0.025)	0.9596 (0.039)	0.9556 (0.027)	0.9565 (0.041)	0.9530 (0.034)	0.9544 (0.045)	0.9560 (0.024)	0.9558 (0.037)
LASSO	0.9482 (0.033)	0.9493 (0.050)	0.9442 (.030)	0.9194 (0.045)	0.9428 (0.037)	0.9447 (0.051)	0.9444 (0.032)	0.9442 (0.042)
ALASSO	0.9420 (0.031)	0.9425 (0.051)	0.9376 (0.030)	0.9379 (0.045)	0.9328 (0.041)	0.942 (0.056)	0.9388 (0.033)	0.9389 (0.047)
ENET	0.9576 (0.025)	0.9587 (0.039)	0.9538 (0.029)	0.9546 (0.042)	0.9532 (0.034)	0.9546 (0.045)	0.9566 (0.024)	0.9562 (0.036)
SCAD	0.9464 (0.031)	0.9475 (0.049)	0.9422 (0.030)	0.9428 (0.045)	0.9386 (0.043)	0.9401 (0.055)	0.9414 (0.031)	0.9408 (0.043)
MCP	0.9256 (0.040)	0.9270 (0.062)	0.9262 (0.038)	0.9269 (0.055)	0.9210 (0.041)	0.9221 (0.058)	0.9224 (0.034)	0.9223 (0.048)
AB	0.9454 (0.032)	0.9469 (0.047)	0.9494 (0.030)	0.9501 (0.044)	0.9470 (0.034)	0.9482 (0.046)	0.9480 (0.029)	0.9481 (0.040)
RF	0.9540 (0.030)	0.9557 (0.043)	0.9560 (0.029)	0.9565 (0.043)	0.9542 (0.032)	0.9556 (0.044)	0.9508 (0.027)	0.9510 (0.039)
LR	0.9478 (0.029)	0.9482 (0.045)	0.9462 (0.030)	0.9469 (0.044)	0.9418 (0.038)	0.9432 (0.050)	0.9438 (0.028)	0.9437 (0.041)
SVM	0.9560 (0.027)	0.9568 (0.041)	0.9522 (0.030)	0.9527 (0.043)	0.9520 (0.031)	0.9526 (0.042)	0.9594 (0.026)	0.9587 (0.037)

References

1. Tariq, H.; Eldridge, E.; Welch, I. An efficient approach for feature construction of high-dimensional microarray data by random projections. *PLoS ONE* **2018**, *13*, e0196385. [CrossRef] [PubMed]
2. Bhola, A.; Singh, S. Gene Selection Using High Dimensional Gene Expression Data: An Appraisal. *Curr. Bioinform.* **2018**, *13*, 225–233. [CrossRef]
3. Dai, J.J.; Lieu, L.H.; Rocke, D.M. Dimension reduction for classification with gene expression microarray data. *Stat. Appl. Genet. Mol. Biol.* **2006**, *5*, 6. [CrossRef] [PubMed]
4. Lu, J.; Kerns, R.T.; Peddada, S.D.; Bushel, P.R. Principal component analysis-based filtering improves detection for Affymetrix gene expression arrays. *Nucleic Acids Res.* **2011**, *39*, e86. [CrossRef]
5. Bourgon, R.; Gentleman, R.; Huber, W. Reply to Talloen et al.: Independent filtering is a generic approach that needs domain specific adaptation. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, E175–E175. [CrossRef]
6. Bourgon, R.; Gentleman, R.; Huber, W. Independent filtering increases detection power for high-throughput experiments. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 9546–9551. [CrossRef]
7. Ramsköld, D.; Wang, E.T.; Burge, C.B.; Sandberg, R. An Abundance of Ubiquitously Expressed Genes Revealed by Tissue Transcriptome Sequence Data. *PLoS Comput. Biol.* **2009**, *5*, e1000598. [CrossRef]
8. Li, L.; Kabesch, M.; Bouzigon, E.; Demenais, F.; Farrall, M.; Moffatt, M.; Lin, X.; Liang, L. Using eQTL weights to improve power for genome-wide association studies: A genetic study of childhood asthma. *Front. Genet.* **2013**, *4*, 103. [CrossRef]
9. Calle, M.L.; Urrea, V.; Vellalta, G.; Malats, N.; Steen, K.V. Improving strategies for detecting genetic patterns of disease susceptibility in association studies. *Stat. Med.* **2008**, *27*, 6532–6546. [CrossRef]
10. Bock, C. Analysing and interpreting DNA methylation data. *Nat. Rev. Genet.* **2012**, *13*, 705–719. [CrossRef]
11. Sun, H.; Wang, S. Penalized logistic regression for high-dimensional DNA methylation data with case-control studies. *Bioinformatics* **2012**, *28*, 1368–1375. [CrossRef]
12. Kim, S.; Halabi, S. High Dimensional Variable Selection with Error Control. *BioMed Res. Int.* **2016**, *2016*, 8209453. [CrossRef] [PubMed]
13. Kim, S.; Kim, J.M. Two-Stage Classification with SIS Using a New Filter Ranking Method in High Throughput Data. *Mathematics* **2019**, *7*, 493. [CrossRef]
14. Fan, J.; Lv, J. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc. Ser. B* **2008**, *70*, 849–911. [CrossRef] [PubMed]
15. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993.
16. Okeh, U.; Oyeka, I. Estimating the Fisher's Scoring Matrix Formula from Logistic Model. *Am. J. Theor. Appl. Stat.* **2013**, *2013*, 221–227.
17. Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
18. Peng, H.; Long, F.; Ding, C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef]
19. Ditzler, G.; Morrison, J.C.; Lan, Y.; Rosen, G.L. Fizzy: Feature subset selection for metagenomics. *BMC Bioinform.* **2015**, *16*, 358. [CrossRef]
20. Su, C.T.; Yang, C.H. Feature selection for the SVM: An application to hypertension diagnosis. *Expert Syst. Appl.* **2008**, *34*, 754–763. [CrossRef]
21. Kursa, M.B.; Rudnicki, W.R. Feature Selection with the Boruta Package. 2010.
22. Urbanowicz, R.J.; Meeker, M.; Cava, W.L.; Olson, R.S.; Moore, J.H. Relief-based feature selection: Introduction and review. *J. Biomed. Inform.* **2017**, *85*, 189–203. [CrossRef] [PubMed]
23. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
24. Freund, Y. An Adaptive Version of the Boost by Majority Algorithm. *Mach. Learn.* **2001**, *43*, 293–318. [CrossRef]
25. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Appl.* **1998**, *13*, 18–28. [CrossRef]
26. Statnikov, A.R.; Wang, L.; Aliferis, C.F. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinform.* **2008**, *9*, 319–319. [CrossRef] [PubMed]
27. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **1996**, *58*, 267–288. [CrossRef]
28. Marquardt, D.W.; Snee, R.D. Ridge Regression in Practice. *Am. Stat.* **1975**, *29*, 3–20.

29. Yang, X.G.; Lu, Y. Informative Gene Selection for Microarray Classification via Adaptive Elastic Net with Conditional Mutual Information. *arXiv* **2018**, arXiv:1806.01466.
30. Fan, J.; Li, R. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360. [CrossRef]
31. Zhang, C.H. Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.* **2010**, *38*, 894–942. [CrossRef]
32. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference And Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009.
33. Dietterich, T.G. Ensemble Methods in Machine Learning. In *International Workshop on Multiple Classifier Systems*; Springer: London, UK, 2000; pp. 1–15.
34. Maclin, R.; Opitz, D.W. Popular Ensemble Methods: An Empirical Study. *arXiv* **2011**, arXiv:1106.0257.
35. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140.:1018054314350. [CrossRef]
36. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
37. Datta, S.; Pihur, V.; Datta, S. An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data. *BMC Bioinform.* **2010**, *11*, 427. [CrossRef] [PubMed]
38. Ahn, H.; Moon, H.; Fazzari, M.J.; Lim, N.; Chen, J.J.; Kodell, R.L. Classification by ensembles from random partitions of high-dimensional data. *Comput. Stat. Data Anal.* **2007**, *51*, 6166–6179. [CrossRef]
39. Lim, N.; Ahn, H.; Moon, H.; Chen, J.J. Classification of high-dimensional data with ensemble of logistic regression models. *J. Biopharm. Stat.* **2009**, *20*, 160–171. [CrossRef]
40. R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2008; ISBN 3-900051-07-0.
41. Kursa, M.B. *Praznik: Collection of Information-Based Feature Selection Filters*; R Package Version 5.0.0; R Foundation for Statistical Computing: Vienna, Austria, 2018.
42. Natalia Novoselova, Junxi Wang, F.P.F.K. *Biocomb: Feature Selection and Classification with the Embedded Validation Procedures for Biomedical Data Analysis*; R package version 0.4; R Foundation for Statistical Computing: Vienna, Austria, 2018.
43. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.* **2010**, *33*, 1–22. [CrossRef]
44. Breheny, P.; Huang, J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Ann. Appl. Stat.* **2011**, *5*, 232–253. [CrossRef]
45. Liaw, A.; Wiener, M. Classification and Regression by randomForest. *R News* **2002**, *2*, 18–22.
46. Meyer, D.; Dimitriadou, E.; Hornik, K.; Weingessel, A.; Leisch, F. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*; R Package Version 1.7-1; R Foundation for Statistical Computing: Vienna, Austria, 2019.
47. Pihur, V.; Datta, S.; Datta, S. *RankAggreg: Weighted Rank Aggregation*; R package version 0.6.5; R Foundation for Statistical Computing: Vienna, Austria, 2018.
48. The RLFS-ERRM Resources 2019. Available online: https://sites.google.com/site/abhijeetrpatil01/file-cabinet/blfs-errm-manuscript-files-2019 (accessed on 25 December 2019).
49. Feature Selection Datasets. Available online: http://featureselection.asu.edu/old/datasets.php (accessed on 25 December 2019).
50. Bolón-Canedo, V.; Sánchez-Maroño, N.; Alonso-Betanzos, A.; Benítez, J.M.; Herrera, F. A review of microarray datasets and applied feature selection methods. *Inf. Sci.* **2014**, *282*, 111–135. [CrossRef]
51. Wang, M.; Barbu, A. Are screening methods useful in feature selection? An empirical study. *PloS ONE* **2018**, *14*, e0220842. [CrossRef]
52. Tsai, C.; Chen, J.J. Multivariate analysis of variance test for gene set analysis. *Bioinformatics* **2009**, *25*, 897–903. [CrossRef] [PubMed]
53. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature Selection: A Data Perspective. *ACM Comput. Surv.* **2017**, *50*, 94:1–94:45. [CrossRef]
54. Chen, X.D.; Ishwaran, H. Random forests for genomic data analysis. *Genomics* **2012**, *99*, 323–329. [CrossRef] [PubMed]
55. Bielza, C.; Robles, V.; Larrañaga, P. Regularized logistic regression without a penalty term: An application to cancer classification with microarray data. *Expert Syst. Appl.* **2011**, *38*, 5110–5118. [CrossRef]

56. Liao, J.G.; Chin, K.V. Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *Bioinformatics* **2007**, *23*, 1945–1951. [CrossRef] [PubMed]
57. Zou, H.; Hastie, T. Regularization and Variable Selection via the Elastic Net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320. [CrossRef]
58. Li, Y.; Zhang, Y.; Zhao, S. Gender Classification with Support Vector Machines Based on Non-tensor Pre-wavelets. In Proceedings of the 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 7–10 May 2010; pp. 770–774.