Scientific Journal of Informatics
Vol. 10, No. 1, Feb 2023
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Simulation Study of Imbalanced Classification on High-Dimensional Gene Expression Data

Masithoh Yessi Rochayani[1*], Umu Sa'adah[2], Ani Budi Astuti[3]
[1]Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Diponegoro, Indonesia
[2]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Brawijaya, Indonesia
[3]Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Brawijaya, Indonesia

Abstract.
Purpose: Classification of gene expression is useful for studying a disease. However, it faces two obstacles: an imbalanced class and a high dimension. The motivation of this study is to examine the effectiveness of undersampling before feature selection on high-dimensional data with imbalanced classes.
Methods: Least Absolute Shrinkage and Selection Operator (Lasso), which can select features, can handle high-dimensional data modeling. Random undersampling (RUS) can be used to deal with imbalanced classes. The Classification and Decision Tree (CART) algorithm is used to construct a classification model because it can produce an interpretable model. Thirty simulated datasets with varying imbalance ratios are used to test the proposed approaches, which are Lasso-CART and RUS-Lasso-CART. The simulated data are generated from parameters of real gene expression data.
Results: The results of the simulation study show that when the minority class accounts for more than 25% of the observation size, the Lasso-CART method is appropriate. Meanwhile, the method RUS-Lasso-CART is effective when the minority class size is at least 20 observations.
Novelty: The novelty of this simulation study is the use of the RUS-Lasso-CART hybrid method to address the classification problem of high-dimensional gene expression data with imbalanced classes.

Keywords: High-dimensional data, Imbalanced class, Undersampling, Feature selection, Gene expression data
Received December 2022 / **Revised** January 2023 / **Accepted** February 2023

INTRODUCTION

High-dimensional data modeling has become the center of attention in the last few decades. The emergence of high-dimensional data has presented challenges for data mining. High-dimensional data is characterized by more features than observations. An example of high-dimensional data is gene expression data generated from microarray experiments. Microarray data has tens of thousands of feature genes, but far fewer objects of observation. Classification is an analytical technique that is widely applied to microarray studies which aim to study differences in gene expression in various types of diseases. However, classification will be faced with challenges when dealing with high-dimensional data.

Reducing the dimensions at the preprocessing stage is an option for modeling high-dimensional data. There are two methods for reducing dimensionality: feature selection and feature extraction [1]. Feature selection reduces dimensions by eliminating irrelevant features. While feature extraction transforms all features at a lower dimension.

For gene expression data, feature selection is more applicable because not all DNA mutations in genes cause cancer [2], and the new features resulting from feature extraction are different from the original gene features, making them difficult to interpret [3].

*Masithoh Yessi Rochayani.
Email addresses: yessirochayani@lecturer.undip.ac.id (Rochayani), u.saadah@ub.ac.id (Sa'adah), ani_budi@ub.ac.id (Astuti)
DOI: 10.15294/sji.v10i1.40589

Overview of feature selection

Feature selection methods can be categorized into three types, namely the filter, wrapper, and embedded methods [4]–[7]. Filter methods, such as chi-square and information gain, have low computational costs contrary to the wrapper and the embedded methods, so the running time is fast. However, the filter method does not interact with the classifier and ignores the correlation between features. Wrapper methods, such as Forward Selection and Backward Elimination, involve interacting with the model to select features, as well as considering the dependencies among features. However, the wrapper method has a long commutation time especially if there are myriad features. Embedded methods, such as the Least Absolute Shrinkage and Selection Operator (Lasso) [8], perform feature selection through statistical learning procedures. This method is more computationally efficient than the wrapper method, involves interaction with the model, and is less prone to overfitting. However, when the features are few, the embedded method is problematic for selecting [9].

Various studies have applied feature selection methods, using filters, wrappers, or embedded in microarray data [10]–[15]. Research [10] used four feature selection methods (both filter and wrapper), i.e. information gain, gain ratio, reliefF, and correlation for gene selection in five datasets, i.e. Lung Cancer, DLBCL, Colon Cancer, Prostate Cancer, and Leukemia. The data that has been selected for features is then modeled using Support Vector Machine (SVM). The results of the study show that the hybrid information gain and SVM provide the best accuracy for the five datasets.

Research [11] proposed a novel gene selection technique. The proposed method is divided into two stages. In the first stage, the dataset is clustered using k-means clustering. Then, the filtering technique, i.e. Signal-to-Noise Ratio (SNR) score, is used to rank each gene in every cluster. The top-scored genes from each cluster are collected and a new feature subset is generated. In the second stage, the new feature subset is used as input to the Particle Swarm Optimization (PSO) and the optimized feature subset is produced. After the two-stage gene selection, then K-Nearest Neighbor (KNN), SVM, and Probabilistic Neural Network (PNN) are applied. The results show that the addition of PSO for gene selection provides better accuracy than without the use of PSO.

Research [12] proposed an embedded method which is the development of Lasso and adaptive penalized logistic regression (APLR) for gene selection and classification of high-dimensional cancer data. The method is called correlation-based penalized logistic regression (CBPLR). The proposed method has a significant impact on penalized logistic regression by selecting the number of genes with a high Area Under the Curve (AUC) and low misclassification rate. Thus, the proposed method could be used for gene selection in other research on high-dimensional cancer classification.

Problem with imbalanced class

So far, only a few studies have considered class balance in high-dimensional data classification modeling [16]. The results of the investigation [17] show that feature selection is not sufficient in bioinformatics when faced with class imbalance. Several previous studies that have examined the class imbalance problem in classification modeling on high-dimensional data are Blagus and Lusa [18]. They used Synthetic Minority Oversampling Technique (SMOTE) at the preprocessing stage for the classification of imbalanced high-dimensional data. However, applying oversampling for high-dimensional gene expression data will cause the data size to be much bigger and take more time to execute [19]. Therefore, high-dimensional data with an imbalanced class should be processed using the undersampling technique.

Research [20] and [21] applied undersampling before feature selection. To test the proposed method, those studies used several high-dimensional data, several methods for feature selection, and several classification methods. The results of research [20] and [21] showed that the application of undersampling before feature selection in most of the high-dimensional data was imbalanced, giving higher accuracy than feature selection without undersampling.

Research [22] used the hybrid RUS and Lasso methods to find biomarkers in gene expression data that have imbalanced classes. In that study, two data were used, i.e. breast tumor and ovarian tumor. In breast tumor data, the biomarkers obtained are in line with biological theory. However, in ovarian cancer data, some of the genes obtained have never been studied for their effect on ovarian cancer.

Problem with model interpretation
The purpose of classification modeling is not only to get high accuracy. Another goal is to find new knowledge from the classification results. Classification methods such as SVM are reliable to obtain high accuracy, but the models obtained cannot be interpreted [23].

A decision tree is a classification method that creates prediction models based on tree structures. Classification and Regression Tree (CART) is one of the popular decision tree algorithms [24]. This algorithm can be used to model various data patterns because no assumptions are needed. This approach has the benefit of making it simple to interpret the obtained model. So that it can be used to discover new knowledge [25]. Research [26] and [27] classify breast cancer data using the hybrid method of Lasso and CART. The results can be interpreted and in line with biological theory.

The purpose of this study is to examine the effectiveness of using undersampling before feature selection on high-dimensional data with imbalanced classes. The method used is the RUS-Lasso-CART hybrid method [22], [26]. For this purpose, we used 30 simulated datasets generated from real gene expression data parameters. The simulated data is set to have different imbalance ratios.

THEORETICAL REVIEW

Logistic Regression

Let $i = 1, 2 \ldots, n$, n denotes n observations, $j = 1, 2.., p$ denotes p independent variables, with β_0 denoting the constant. The logistic regression model

$$\pi(x) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_i\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_i\right)} = logit^{-1}\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_i\right) \quad (1)$$

where logit is the inverse of the logit transformation, with regression coefficients $\beta = (\beta_0, \ldots, \beta_p)$. The regression coefficients are determined by maximizing the likelihood function [28], [29]. Maximizing the likelihood function is equivalent to maximizing the log-likelihood function since the natural logarithm is a monotonic increasing function. Therefore, we can work with the simpler log-likelihood instead of the original likelihood. The log-likelihood function is

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \log\left(\pi(x_i)\right) + (1 - y_i) \log\left(1 - \pi(x_i)\right)\right] \quad (2)$$

The logistic regression coefficient is obtained by solving the optimization problem expressed by the equation

$$\hat{\boldsymbol{\beta}} = \arg\min\{-l(\boldsymbol{\beta})\} \quad (3)$$

where $-l(\beta_j)$ is the negative log-likelihood function.

Least Absolute Shrinkage and Selection Operator (Lasso)

Lasso is one of the regularization methods that can be used for variable selection. Lasso works by adding constraints L_1, i.e. $\sum_{j=1}^{p} |\beta_j|$. The addition of the constraint to the objective function of the logistic regression is expressed by

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{Lasso}} = \arg\min\{-l(\boldsymbol{\beta})\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq t \quad (4)$$

With the Lagrange multiplier, the optimization problem with constraints as in equation (4) can be turned into an optimization problem without constraints

$$\hat{\boldsymbol{\beta}} = \arg\min\left\{-l\left(\beta_j\right) + \lambda \sum_{j=1}^{p} |\beta_j|\right\} \quad (5)$$

where $\lambda > 0$.

Classification and Regression Tree (CART)

The CART method builds a decision tree by binary-splitting. It begins with splitting the root node which holds the entire training set. Let D be the training set. The following are the steps for the CART algorithm *[25]*:

1) Determine the Gini index of D using (6)

$$Gini\ (D) = 1 - \sum_{i=1}^{m} p_i^2 \quad (6)$$

where p_i is the proportion of class C_i in node D.

2) Select a potential split point. For continuous-scaled variables, the observed values of the j^{th} feature, i.e., $\{x_{ij} | \ i = 1, \dots, n\}$, are arranged in ascending order. Then, the midpoint of two consecutive values becomes a potential point. As an illustration, the midpoint between $x_{(c)j}$ and $x_{(c+1)j}$ is $u_{cj} = \frac{x_{(c)j} + x_{(c+1)j}}{2}$.

3) *For each* u_{cj}, calculate the Gini index using (7).

$$Gini_{u_{cj}}(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (7)$$

where D_1 and D_2 are training sets' subsets following u_{jc}'s split.

4) Apply formula (8) to determine the goodness of split $\Delta\ Gini(u_{cj})$.

$$\Delta\ Gini(u_{cj}) = Gini\ (D) - Gini_{u_{cj}}(D) \quad (8)$$

5) For $j = 1, \dots, k$, repeat steps 2 through 4 then identify the highest $\Delta\ Gini(u_{cj})$. The candidate with the highest $\Delta\ Gini(u_{cj})$ is used as the node.

6) Repeat steps 1 through 5 to split the child nodes until the nodes containing homogeneous data are attained.

PROPOSED METHODS

Simulation design

Simulation studies are essential to understand the behavior of statistical methods [30]. In this study, we use 30 simulated datasets, generated based on real gene expression data parameters. The real data is high-dimensional data with 1,545 observations in the form of tumor tissue samples and 10,936 genes (i.e. OVA_Breast data downloaded from openml.org). However, the simulated data were generated with a smaller size, i.e. 1,000 features (we name them $X_1, X_2, \dots, X_{1000}$). The strategy used to get real data parameters is to use 1,000 features of the real data to get the mean vector $\boldsymbol{\mu} = \boldsymbol{\mu}_{real_{(1,000\times1)}}$, variance-covariance matrix $\boldsymbol{\Sigma}_{real_{(1,000\times1,000)}}$, and vector of coefficients $\boldsymbol{\beta}_{real_{(1,000\times1)}}$. To produce high-dimensional simulated data, the size of the observation is set less than the number of features, $n = 200$ and $n = 500$ as in the study [31]. Thus, the simulated dataset has sizes $(n; p) = (200; 1{,}000)$ and $(n; p) = (500; 1{,}000)$ where n represents the number of observations and p represents the number of features.

The steps for generating simulated data are described as follows.

1) Generating a feature matrix $\boldsymbol{X}_{(n\times1,000)}$ ($n = 200$ or $n = 500$) from a multivariate normal distribution with $\boldsymbol{\mu} = \boldsymbol{\mu}_{real_{(1,000\times1)}}$, and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{real_{(1,000\times1,000)}}$.
2) Using $\boldsymbol{\beta}_{real_{(1,000\times1)}}$ as the vector of coefficients of the simulated data.
3) Generating a binary response variable $\boldsymbol{Y}$ uses the inverse logit function:

$$\mathrm{Y_i} = \frac{1}{1 + \exp\left(-\sum_{j=1}^{p} X_{ij}\beta_j\right)} \quad (10)$$

To obtain a different imbalance ratio, when generating a response variable, observations with a probability value of more than h ($h \in [0.1, 0.5]$) are labeled "Positive", and observations with a probability value of less than h are labeled "Negative".

Steps of modeling

In this study, the undersampling method used is Random Undersampling (RUS). The regularization method used to select features is the Least Absolute Shrinkage and Selection Operator (Lasso). Meanwhile, for decision tree modeling, the Classification and Regression Tree (CART) algorithm is used. Data analysis used a hybrid method without undersampling, namely Lasso-CART, and a hybrid method with undersampling, namely RUS-Lasso-CART. The steps of data analysis using Lasso-CART and RUS-Lasso-CART are described as follows.

1) Split the original data into the training set and the test set randomly with a ratio of 80:20.
2) Undersampling with RUS on the training set to get a balanced training set. For the Lasso-CART method, skip this step and go to step 3.
3) Normalize the training set using Z-score normalization.
4) Perform feature selection with Lasso on the normalized training set. In this process, the tuning parameter (λ) is updated up to 100 iterations by decreasing its value at the ratio of

$$r = \sqrt[M]{\frac{\lambda_M}{\lambda_0}} \tag{9}$$

where $m = 1, 2, \ldots, M$ stands for the iteration index ($M = 100$) and the glmnet package defines $\lambda_M = 0.01\ \lambda_0$ [26].

5) Determine the optimum λ using k-fold cross-validation (k-fold CV). The feature set selected at the optimum is called F*.
6) Modeling the decision tree with the CART algorithm on the original training set with F* features until the maximum tree is obtained.
7) Pruning the tree according to the 1 standard error rule (1-SE rule).
8) Evaluate the model based on the Area Under the Curve (AUC) value.

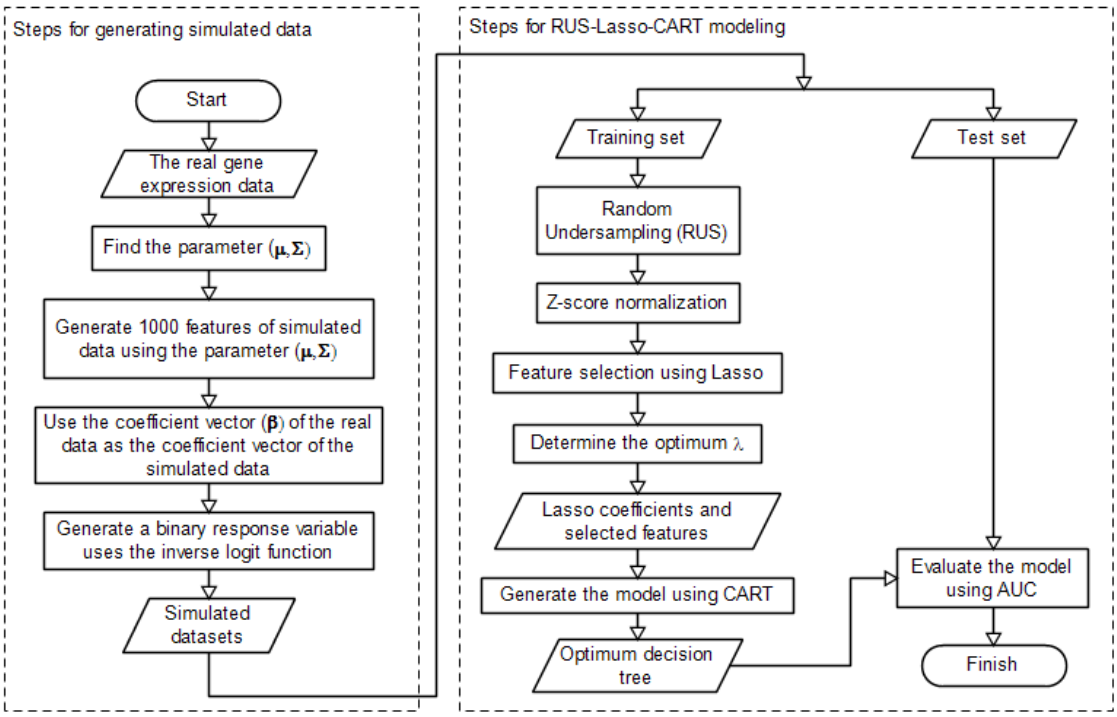

Figure 1. Flowchart of RUS-Lasso-CART modeling

RESULTS AND DISCUSSIONS

Results of feature selection

Feature selection using Lasso was performed on the original training set (without undersampling) and the undersampled training set. The number of non-zero coefficients of Lasso depends on the tuning parameter used. To determine the optimum tuning parameter (λ optimum), k-fold cross-validation is used. In this study, k = 10, while for Lasso and k = 5 for RUS-Lasso because the size of the observations is smaller after undersampling. The optimum tuning parameter is the one that produces the smallest average binomial

deviance from the cross-validation [32]. Table 1 presents the optimum λ and the number of selected features (non-zero coefficient) in 10 simulated datasets.

Table 1. The number of non-zero coefficients at the optimum λ

Number of observations on the training set	i-th dataset	Imbalanced ratio	Number of selected features of Lasso	Number of non-selected features of RUS-Lasso
160	1	48.00%	54	52
	2	20.625%	14	20
	3	13.75%	30	5
	4	8.125%	16	0
400	5	31.50%	26	43
	6	17.25%	63	26
	7	6.00%	29	6
	8	5.75%	39	13
	9	4.75%	15	0
	10	4.00%	24	12

In Table 1, Lasso produces features with more non-zero coefficients at optimum λ. While RUS-Lasso tends to give fewer non-zero coefficients and sometimes no feature is selected, as in the 4th and the 9th datasets. This condition occurs when the size of the observations is very small after undersampling. In the 4th dataset, there are only 26 observations (13 in the positive class and 13 in the negative class) after RUS. When the 5-fold CV is used, each fold contains only 5 to 6 observations. Based on the 5-fold CV results, the best results are obtained at the initial tuning parameter (λ_0), because it has the smallest binomial deviance. In fact, all the coefficients are zero at λ_0. So that no feature is selected. The k-fold CV method cannot be used when the size of the observations is too small, even though the k has been reduced too. In the 9th dataset, 38 observations were obtained after using RUS. The k-fold CV results show that λ which produces the smallest average binomial deviance is λ_0, so all coefficients are zero. Thus, the RUS-Lasso method is unable to retain features with non-zero coefficients when the dataset has a very small minority class size (i.e. less than 20 observations).

Decision tree modeling

After the selected features are obtained from Lasso and RUS-Lasso, the next step is to construct a decision tree model. The results of the simulated data modeling show that when the minority class size is not too small, which is more than 25% observation in the training set, the Lasso-CART and RUS-Lasso-CART provide the same model. This means that undersampling does not affect the modeling results when the two classes are balanced. One of the modeling results can be seen in Figure 2 which presents a decision tree of the dataset that has a minority class size of 31.50 %.

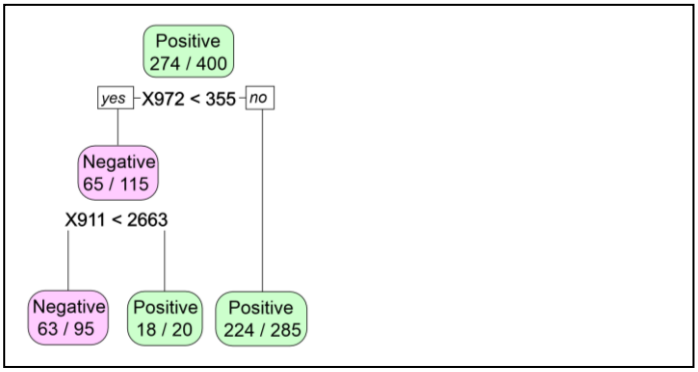

Figure 2. Decision tree of the simulated data with a minority class size of 31.50%. (left: Lasso-CART; right: RUS-Lasso-CART)

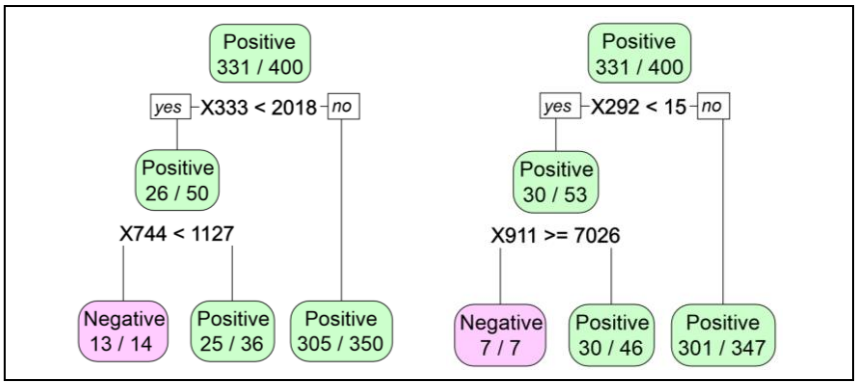

Figure 3. Decision tree of the simulated data with a minority class size of 17.25%. (left: Lasso-CART; right: RUS-Lasso-CART)

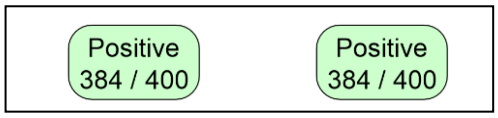

Figure 4. Decision Tree of the simulated data with a minority class size of 4%. (left: Lasso-CART; right: RUS-Lasso-CART)

The ineffectiveness of undersampling in balanced classes has been shown by [33]. According to the study, undersampling is effective when the two classes have a medium imbalance ratio. However, [33] provided no information about the size of the minority class when undersampling can work effectively. Based on the simulation results in the present study, undersampling is effective for datasets with minority class sizes smaller than 25%. Figure 3 presents a model of a dataset that has a minority class size of 17.25%.

The results of the simulation study also show that when the two classes are very unequal, which means that the size of the minority class is very small, two possibilities might occur. First, Lasso leaves no features at optimum λ as the 4th and 9th simulated datasets (see Table 1). Second, CART is unable to split the root node. As seen in Figure 4, CART could not split the root node and construct a decision tree because the root node is already homogeneous. Figure 4 is the decision tree model of the 10th dataset which the minority class size is only 4%.

Model evaluation

After obtaining a decision tree from the simulated data, an evaluation of the model is carried out to find out which of the Lasso-CART and RUS-Lasso-CART give better results. The AUC is used to measure the feasibility of the model because this metric is robust for imbalanced classes [34]. Figure 5 presents the AUC of the test sets from the simulated data with n of the training set is 160.

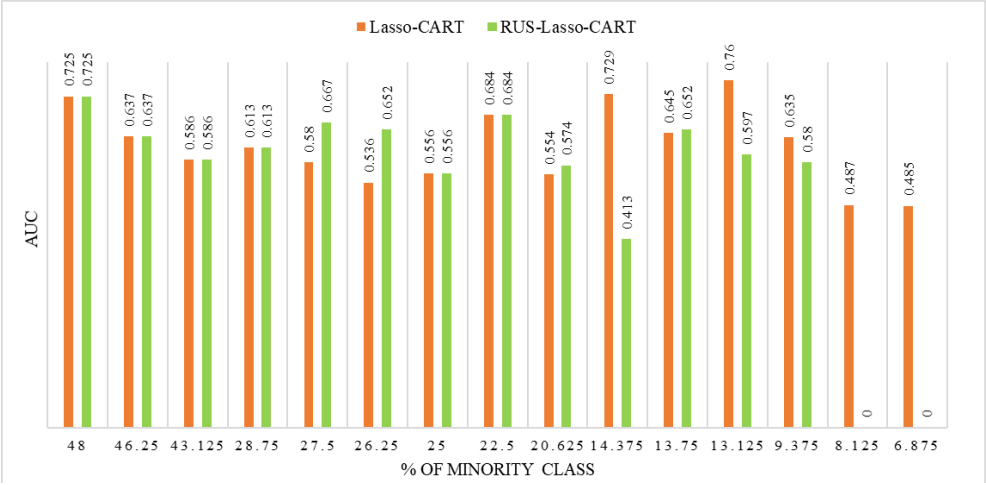

Figure 5. AUC value from the dataset with 160 observations in the training set

Based on Figure 5, when the minority class size is above 28.75%, the Lasso-CART and RUS-Lasso-CART methods give the same AUC value because the obtained models are the same. When the minority class size is around 27% to 20%, the RUS-Lasso-CART gives an AUC value that tends to be higher than the AUC value of the Lasso-CART. However, when the minority class size is very small, i.e. less than 20%, RUS-Lasso-CART gives a lower AUC value compared to Lasso-CART and even gives a zero AUC value because the model cannot be generated.

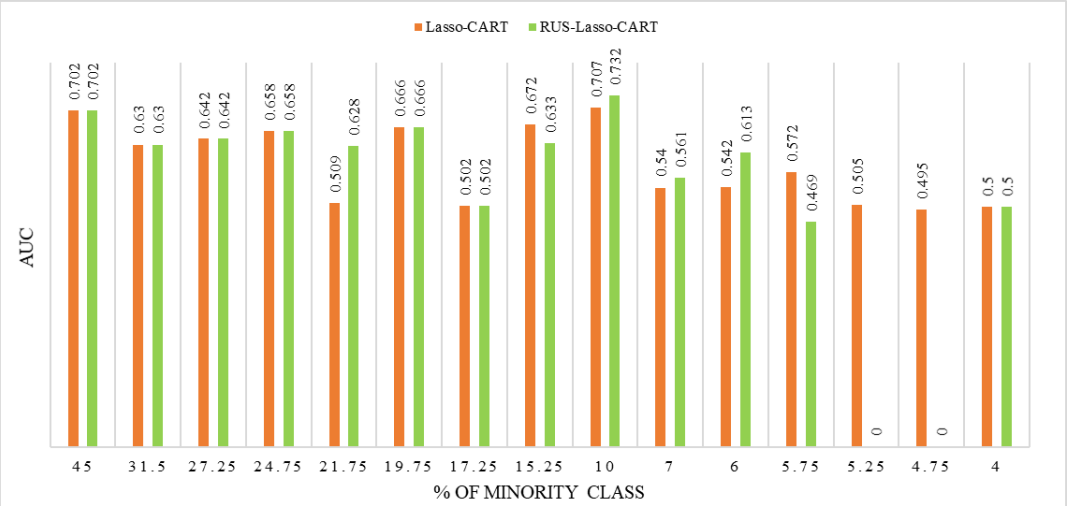

Figure 6. AUC value from the dataset with 400 observations in the training set

The same result is also shown by the simulated dataset with 400 observations. In Figure 6, the same AUC values are obtained from the Lasso-CART and RUS-Lasso-CART when the minority class size is above 24.75%. When the minority class size is around 21% to 6%, RUS-Lasso-CART tends to give better results than Lasso-CART. The RUS-Lasso-CART method will give poor results or cannot produce a decision tree model when the minority class has a size smaller than 6%.

Based on the results of the simulation study, it can be concluded that undersampling is effective at medium unbalance ratios, as stated by [33]. When the minority class is more than 25% of the observations size, undersampling does not affect the modeling results. And when the two classes are very imbalanced with the size of the observations in the minority class of fewer than 20 observations, the model cannot be obtained.

CONCLUSION

In a balanced condition, (i.e. the minority class size is more than 25%), the same model is obtained from Lasso-CART and RUS-Lasso-CART. In other words, undersampling does not affect on the modeling results. At medium imbalance ratios (i.e. the minority class size is between 5% and 25%), using the undersampling method before feature selection can give a higher AUC value. Thus, for high-dimensional data with medium unbalance ratios, RUS-Lasso-CART is more appropriate to use. When the minority class size is very small (i.e. less than 20 observations), two possibilities might occur i.e. Lasso leaves no features at optimum λ, or CART is unable to find the root node split.

The Lasso-CART and RUS-Lasso-CART methods proposed in this study cannot work when the minority class size is too small. For further research, it can be focused on handling the problem of classifying high-dimensional data with very small minority class sizes.

In this study, k-fold cross-validation was used to select the optimum tuning parameter (λ). The k-fold cross-validation method divides the data randomly into k-folds which allows different optimum λ to be obtained for different randomization results. Other criteria such as Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), or modified BIC can be used for choosing the optimum tuning parameters instead of the k-fold cross-validation.

REFERENCES

[1] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 56–70, 2020, doi: 10.38094/jastt1224.

[2] Q. Jiang and M. Jin, "Feature Selection for Breast Cancer Classification by Integrating Somatic Mutation and Gene Expression," *Front. Genet.*, vol. 12, no. February, pp. 1–12, 2021, doi: 10.3389/fgene.2021.629946.

[3] L. Wang, Y. Wang, and Q. Chang, "Feature selection methods for big data bioinformatics: A survey from the search perspective," *Methods*, vol. 111, no. August 2016, pp. 21–31, 2016, doi: 10.1016/j.ymeth.2016.08.014.

[4] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19. pp. 2507–2517, Oct. 2007. doi: 10.1093/bioinformatics/btm344.

[5] J. Jumanto, M. F. Mardiansyah, R. Pratama, M. F. Al Hakim, and B. Rawat, "Optimization of breast cancer classification using feature selection on neural network," *J. Soft Comput. Explor.*, vol. 3, no. 2, pp. 105–110, 2022, doi: 10.52465/joscex.v3i2.78.

[6] P. Pampouktsi *et al.*, "Techniques of Applied Machine Learning Being Utilized for the Purpose of Selecting and Placing Human Resources within the Public Sector," *J. Inf. Syst. Explor. Res.*, vol. 1, no. 1, pp. 1–16, 2023, [Online]. Available: https://shmpublisher.com/index.php/joiser/article/view/91

[7] M. R. Wijaya, R. Saptono, and A. Doewes, "The Effect of Best First and Spreadsubsample on Selection of a Feature Wrapper With Na**[illegible]**ve Bayes Classifier for The Classification of the Ratio of Inpatients," *Sci. J. Informatics*, vol. 3, no. 2, pp. 139–148, Nov. 2016, doi: 10.15294/sji.v3i2.7910.

[8] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[9] S. Biswas, M. Bordoloi, and B. Purkayastha, "Review on Feature Selection and Classification using Neuro-Fuzzy Approaches," *Int. J. Appl. Evol. Comput.*, vol. 7, no. 4, pp. 28–44, Oct. 2016, doi: 10.4018/ijaec.2016100102.

[10] L. Gao, M. Ye, X. Lu, and D. Huang, "Hybrid Method Based on Information Gain and Support Vector Machine for Gene Selection in Cancer Classification," *Genomics, Proteomics Bioinforma.*, vol. 15, no. 6, pp. 389–395, 2017, doi: 10.1016/j.gpb.2017.08.002.

[11] B. Sahu and D. Mishra, "A Novel Feature Selection Algorithm using Particle Swarm Optimization for Cancer Microarray Data," in *International Conference on Modeling Optimization and Computing (ICMOC-2012)*, 2012, no. 38, pp. 27–31. doi: 10.1016/j.proeng.2012.06.005.

[12] Z. Y. Algamal and M. H. Lee, "Penalized Logistic Regression with the Adaptive LASSO for Gene Selection in High-Dimensional Cancer Classification," *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9326–9332, 2015, doi: 10.1016/j.eswa.2015.08.016.

[13] C. Kang, Y. Huo, L. Xin, B. Tian, and B. Yu, "Feature selection and tumor classification for microarray data using relaxed Lasso and generalized multi-class support vector machine," *J. Theor. Biol.*, vol. 463, pp. 77–91, 2018, doi: 10.1016/j.jtbi.2018.12.010.

[14] H. Cai, P. Ruan, M. Ng, and T. Akutsu, "Feature weight estimation for gene selection: A local hyperlinear learning approach," *BMC Bioinformatics*, vol. 15, p. 70, Mar. 2014, doi: 10.1186/1471-2105-15-70.

[15] S. Liu *et al.*, "Feature selection of gene expression data for Cancer classification using double RBF-kernels," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–14, 2018, doi: 10.1186/s12859-018-2400-2.

[16] B. Pes and G. Lai, "Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study," *PeerJ Comput. Sci.*, vol. 7, 2021, doi: 10.7717/PEERJ-CS.832.

[17] A. A. Shanab and T. M. Khoshgoftaar, "Is gene selection enough for imbalanced bioinformatics data?," *Proc. - 2018 IEEE 19th Int. Conf. Inf. Reuse Integr. Data Sci. IRI 2018*, pp. 346–355, 2018, doi: 10.1109/IRI.2018.00059.

[18] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," 2013.

[19] P. Kaur and A. Gosain, "Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise," 2018. doi: 10.1007/978-981-10-6602-3_3.

[20] A. A. Shanab, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse, "Comparison of approaches to alleviate problems with high-dimensional and class-imbalanced data," *Proc. 2011 IEEE Int. Conf. Inf. Reuse Integr. IRI 2011*, pp. 234–239, 2011, doi: 10.1109/IRI.2011.6009552.

[21] H. Yin and K. Gai, "An Empirical Study on Preprocessing High-dimensional Class-imbalanced Data for Classification," in *IEEE 17th International Conference on High Performance Computing and Communications*, 2015, pp. 1314–1319. doi: 10.1109/HPCC-CSS-ICESS.2015.205.

[22] M. Y. Rochayani, U. Sa, and A. B. Astuti, "Finding Biomarkers from a High-Dimensional Imbalanced Dataset Using the Hybrid Method of Random Undersampling and Lasso," *ComTech Comput. Math. Eng. Appl.*, vol. 11, no. 2, pp. 75–81, 2020, doi: 10.21512/comtech.v11i2.6452.

[23] G. Rätsch, S. Sonnenburg, and C. Schäfer, "Learning Interpretable SVMs for Biological Sequence Classification," *BMC Bioinformatics*, vol. 7, no. 1, p. S9, 2006, doi: 10.1186/1471-2105-7-S1-S9.

[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman and Hall, 1984.

[25] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*. Waltham: Morgan Kaufmann, 2012.

[26] M. Y. Rochayani, U. Sa'adah, and A. B. Astuti, "Two-stage Gene Selection and Classification for a High-Dimensional Microarray Data," *J. Online Inform.*, vol. 5, no. 1, pp. 9–18, 2020, doi: 10.15575/join.v5i1.569.

[27] U. Sa'adah, M. Y. Rochayani, and A. B. Astuti, "Knowledge discovery from gene expression dataset using bagging lasso decision tree," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 2, pp. 1151–1159, 2021, doi: 10.11591/ijeecs.

[28] S. Devika, L. Jeyaseelan, and G. Sebastian, "Analysis of sparse data in logistic regression in medical research: A newer approach.," *J. Postgrad. Med.*, vol. 62, no. 1, pp. 26–31, 2016, doi: 10.4103/0022-3859.173193.

[29] S. Doerken, M. Avalos, E. Lagarde, and M. Schumacher, "Penalized logistic regression with low prevalence exposures beyond high dimensional settings," *PLoS One*, vol. 14, no. 5, pp. 1–14, 2019, doi: 10.1371/journal.pone.0217057.

[30] T. P. Morris, I. R. White, and M. J. Crowther, "Using simulation studies to evaluate statistical methods," *Stat. Med.*, vol. 38, no. 11, pp. 2074–2102, 2019, doi: 10.1002/sim.8086.

[31] G. Romeo and M. Thoresen, "Model selection in high-dimensional noisy data: a simulation study," *J. Stat. Comput. Simul.*, vol. 89, no. 11, pp. 2031–2050, 2019, doi: 10.1080/00949655.2019.1607345.

[32] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall, 2015. doi: 10.1201/b18401.

[33] A. Dal Pozzolo, O. Caelen, and G. Bontempi, "When is Undersampling Effective in Unbalanced Classification Tasks?," in *Machine Learning and Knowledge Discovery in Databases*, A. Appice, P. P. Rodrigues, V. Santos Costa, C. Soares, J. Gama, and A. Jorge, Eds. Cham: Springer International Publishing, 2015, pp. 200–215. doi: 10.1007/978-3-319-23528-8_13.

[34] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo, "Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data," *2019 Int. Conf. Comput. Control. Informatics its Appl. Emerg. Trends Big Data Artif. Intell. IC3INA 2019*, pp. 14–18, 2019, doi: 10.1109/IC3INA48034.2019.8949568.