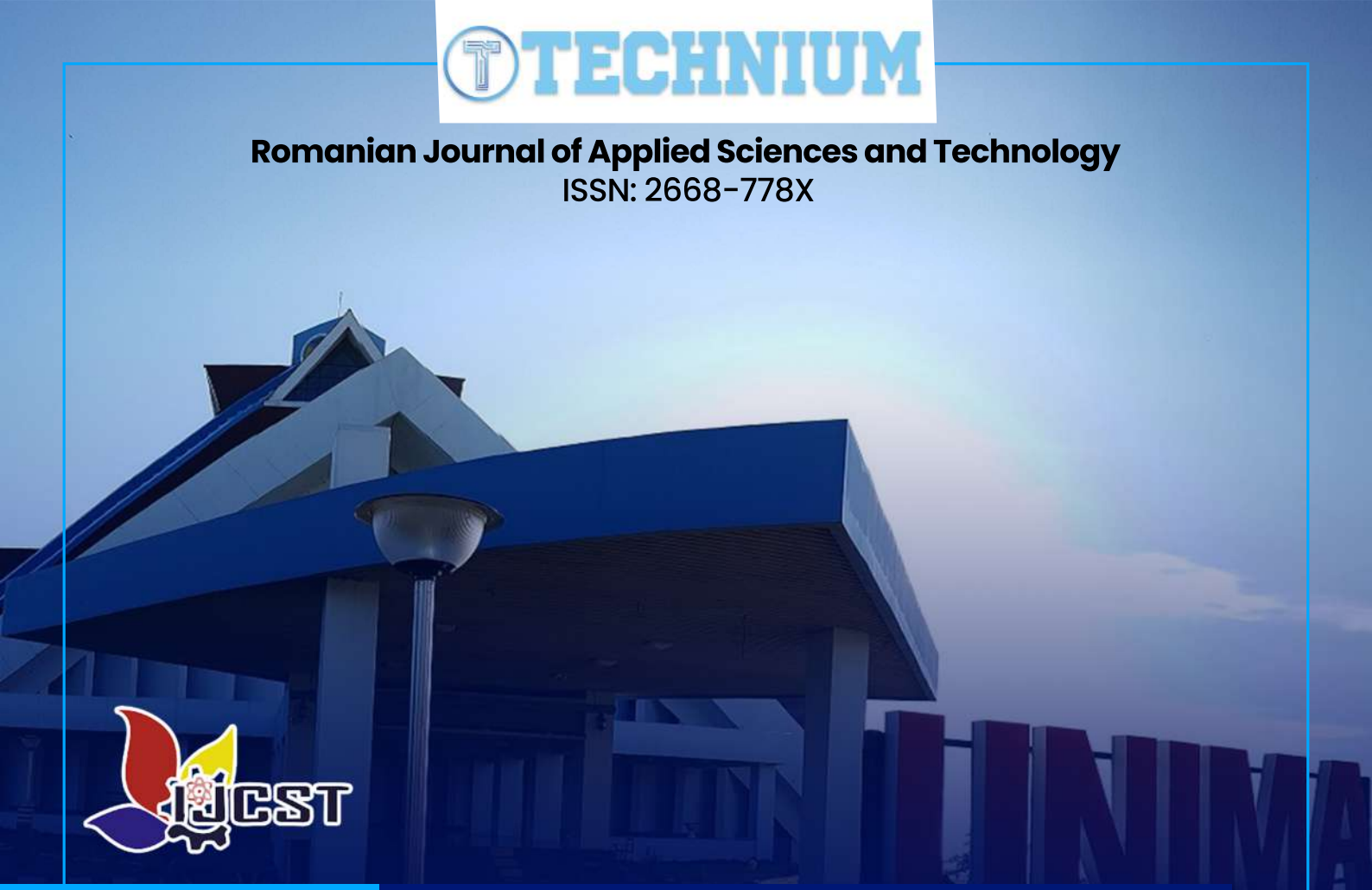
TECHNIUM

Romanian Journal of Applied Sciences and Technology
ISSN: 2668-778X



vol. 16 / 2023

The 7th International Conference on Science Technology

organized by
Faculty of Social Science and
Law Universitas Negeri Manado and
Consortium of International Conference
on Science and Technology

The Innovation Breakthrough in Digital and Disruptive Era

TECHNIUM

Technium: Romanian Journal of Applied Sciences and Technology
Vol. 16, 273-279, October, 2023
ISSN: 2668-778X
www.techniumscience.com

Lung Cancer Classification Using Random Oversampling and Gradient Boosted Decision Tree

Wahyudi Setiawan[1*], *Yoga* Dwitya Pramudita[2], *Mulaab*[3]

[1]Department of Information Systems, University of Trunojoyo Madura, Bangkalan, East Java 69162
[2,3]Department of Informatics, University of Trunojoyo Madura, Bangkalan, East Java 69162

Abstract. Lung cancer has the highest number of sufferers in men, especially in Indonesia. An unhealthy lifestyle, smoking, and pollution also aggravate the patient's condition. In this study, a diagnosis was made of patients with suspected lung cancer. For an experiment, the data from public datasets, "Cancer Patient," "Survey Lung Cancer," and "Cancer_Data." The research phase includes exploratory data analysis (EDA), pre-processing, and classification. EDA aims to know data types, missing values, correlations between attributes, and outliers. Pre-processing consists of data cleaning and data discretization. In the next process, we use randomized oversampling to overcome imbalanced data. The final step was classification using Gradient Boosted Decision Tree (GBDT). The experiment scenario uses imbalanced and balanced data. For the testing scenario, the variation in learning rate and the number of trees were used with Randomized Search Tuning. The distribution of training and testing data uses 5-fold cross-validation. The result shows that using balanced data between classes is better than imbalanced data. In addition, we also classify the dataset with the k-nearest neighbor and support vector machine. The GBDT produces better performance for two datasets.

1 Introduction

Lung cancer is found on the inside or outside of the lungs. Patients with lung cancer are higher in men than women. The percentage of male vs. people living with female cancer is 3:1. Most patients are aged 40 years and over. Data from the Global Cancer Observatory shows that more than 2 million people will have lung cancer in 2020. This fact is comparable to 11.4% of world cancer patients. According to data from the Cancer world bank, lung cancer cases are equivalent to 12.22 per 100 thousand people. Hungary has the highest number of people with lung cancer, 56.7 per 100 thousand people [1]–[4].

In 2020, Indonesia had 34,383 new cases of lung cancer, or around 8.8% of the total cancer cases. Indonesia was the third highest lung cancer sufferer after breast and cervical cancer. A study at a cancer center hospital in Indonesia showed that the peak age of lung cancer prevalence for men was 65 years, while for women, it was 50 years. The number of lung cancer cases increases at 45 years [4]–[6].

Research on lung cancer includes using machine learning. The initial stage is to understand the attributes that affect lung cancer, such as age, gender, air pollution, alcohol use, dust allergies, and other features. Data can be obtained primarily or using public data. Research on lung cancer classification includes using public data with 309 records, 15 attributes, and two classes. Classification uses Random Forest and Naïve Bayes Classifier (NBC) algorithms. The percentage of training is 80% testing is 20%. The test results show that the random forest has an accuracy of 98.38%, precision of 98.28%, and recall of 100%. In comparison, NBC has 95% accuracy, 100% precision, and 94.74% recall [7]. Subsequent research uses the same dataset with a pre-processing imbalance of Synthetic Minority Oversampling Technique (SMOTE) data. Data uses 10-fold Cross-Validation. The various machine learning models use as classifiers. The best result shows an accuracy of 97.1% in the Rotation Forest [8]. Subsequent studies use the same data as the Support Vector Machine and k-nearest neighbor classifications. The test results show that the support vector machine has an accuracy of up to 95.18%, 88% precision, and 82.5% recall [9]. Research with various machine models was also carried out using SVM, NB, DT, and logistic regression. The lung cancer dataset is from UCI machine learning, namely BRATS, OASIS, and NBTR. The test results obtained an accuracy of 99.2%, 87.87%, 90%, and 66.7%, respectively [10]. Other studies using a dataset with ten attributes, 178 records, and two classes classify using SVM, k-NN, NB, and LR. Classification results show an accuracy of 99.37%, 98.73%, 96.18%, and 98.09% [11]. Experiments have been carried out only limited to 2 classes. This study also classifies low, medium, and high lung cancer levels.

2 Method

This study consists of several stages of research methodology. Data Collection performs public data retrieval from kaggle. Exploratory data analysis performs data understanding and data preparation. Pre-processing performs Data Discretization to match data

Technium: Romanian Journal of Applied Sciences and Technology
Vol. 16, 273-279, October, 2023
ISSN: 2668-778X
www.techniumscience.com

types forimbalanced data between classes using Random Oversampling. Then perform Data Splitting between training and testing using 5-fold cross-validation at the end of the classification with the Gradient Boost Decision Tree. The system block diagram is shown in Figure 1.

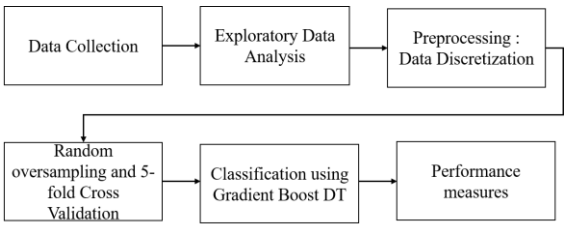


Fig. 1. Block diagram of lung cancer classification using random oversampling and GBDT.

2.1 Data Collection

This study uses public lung cancer data from kaggle. Data usage can be downloaded as shown in Table 1.

Table 1. The dataset used for testing

Dataset	Link
Survey Lung Cancer	https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer
CancerPatient	https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link
Cancer_Data	https://www.kaggle.com/datasets/erdemtaha/cancer-data

The first dataset consists of 16 features, including labels. Features consist of gender, age, smoking, yellow_fingers, anxiety, peer_pressure, chronic diseases, fatigue, allergy, wheezing, coughing, alcohol, shortness of breath, swallowing difficulty, chest pain, and the lung cancer label "yes" or "no." The dataset has 309 records, with class "yes" totaling 270, while type "no" is 39 [12].

The second dataset has 26 features such as index, patient id, age, gender, air pollution exposure, alcohol use, dust allergy, occupational hazards, genetic risk, chronic lung disease, balanced diet, obesity, smoking, passive smoker, chest pain, coughing of blood, fatigue, weight loss, shortness of breath, wheezing, swallowing difficulty, clubbing of fingernails, snoring, and label levels namely "Low," "Medium," and "High." All features are of categorical type except age, which is of numeric type. The total data number of 1,000 consists of 365 high-risk lung cancer, 332 medium, and 303 low-risk lung cancer[13].

The third dataset has 31 features, including a diagnosis label consisting of two classes, "Benign" and "Malignant." The data consists of 596 records. The benign class has 357 records, while the malignant type has 212 records [14]. The first and second datasets contain features that cause lung cancer levels. The third dataset contains texture features that include size characteristics of cancer categorized as benign or malignant.

2.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an initial test phase that aims to identify patterns, obtain anomalies, perform hypothesis testing, and check assumptions [15], [16]. The EDA components in this study are shown in Table 2.

Table 2. EDA Components

No	EDA components	Objective
1	Data description	Do a description of each attribute
2	Missing value	Gets a null value
3	Data duplication	Get the number of duplicated data
4	Outlier	Detects outlier data
5	The amount of data after dropping duplicates	Get unique data information per class
6	Jumlah data after balanced	Knowing the amount of balanced data per class

2.3 Pre-processing using Data Discretization

Data discretization is changing numeric-valued attributes into nominal (categorical). Data Discretization can be done by Binning, Regression, and Outlier analysis. This study uses the binning algorithm, which has two stages, namely [17]:

1. Use the equal width (or distance) binning approach.

$$W = \frac{(IN_{max} - IN_{min})}{c} \quad (1)$$

W = equal width binning
IN_{max} = the highest value in the column
IN_{min} = the lowest value in the column
c = coefficient (1,2,3,…, data-1)

2. Perform smoothing by bin boundary.

$$boundary = IN_{min} + (i \times W) \quad (2)$$

i = range/category data (1,2,…,k)

2.4 Random oversampling and cross-validation

Data between classes occurs imbalance. Classification with imbalanced data can cause learning in the minority class to be less successful. Therefore this study uses a randomized oversampling technique. Minority class duplicated randomized data. The final target is to equalize the number of minority classes with the majority class [18], [19]. If the data is balanced, then split the data into two parts: training and testing data. Data splitting uses 5-fold cross-validation [20], [21].

TECHNIUM

Technium: Romanian Journal of Applied Sciences and Technology
Vol. 16, 273-279, October, 2023
ISSN: 2668-778X
www.techniumscience.com

2.5 Classification using Gradient Boosting

This study uses the Gradient Boosting Decision Tree (GBDT). GBDT is a regression method, but it can be used for classification by changing the loss function model. The classification uses the logarithmic loss function. The regression uses squared errors; t GBDT algorithm for classification has the following stages [22]–[24]:

STEP 1. Initialize the model

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{3}$$

F_0 = initial constant prediction; L = loss function (cross-entropy loss)

$$L = -(y_i \times \log(p) + (1 - y_i) \times \log(1 - p)) \tag{4}$$

y_i = classification target; p = predicted probability of class n; argmin means searching for γ (gamma) that minimizes $\Sigma L(y_i, c)$.

STEP 2. For $m = 1$ *to* M:

Do as many iterations as time. M is the number of trees created, and small shows the index of each tree.

2.1 Calculate the residuals

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad for\ i = 1, \dots, n \tag{5}$$

r_{im} = residuals, obtained from the derivative loss function by taking into account previous predictions F_{m-1} and multiplying it by -1

a. Train regression tree

$$r_{jm} \quad for\ j = 1, \dots, j_m \tag{6}$$

j = node (i.e leave) pada tree; m = tree index; J = number of leaves

2.3 Count γ_{jm}

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \quad for\ j = 1, \dots, j_m \tag{7}$$

Next, get γ_{jm}, which performs a minimize loss function on each terminal node. $\sum_{x_i \in R_{jm}} L$ means aggregating the loss on all the x_i that belong to the node R_{jm}

2.4 Update the model

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm1\,(x \in R_{jm})} \tag{8}$$

F_m = model update; in = learning rate;

2.6 Performance System

The performance system calculates the accuracy, precision, recall, and f1-score. The quality of the classification can be tested using the Confusion Matrix (CM), which compares the classification results of the system with the data labels. In evaluating multiclass classification, there are three commonly used matrices: accuracy, precision, recall, and f1-score. Accuracy calculates the ratio of correct predictions to the total evaluated data. Precision calculates the level of accuracy between the requested data and the answers or results provided by the system. Recall is used to count the amount of data that is classified correctly with the total data that should be in that class.

Meanwhile, the f1-score compares weighted precision and recall [25], [26]. The following is an example of a confusion matrix for multiclass classification. Figure 2 is an example of the Confusion Matrix for a multiclass category.

Predicted

Actual	A	B	C
A	TN	FP	TN
B	FN	TP	FN
C	TN	FP	TN

Fig 2. Multiclass Confusion Matrix

3 Result and discussion

The initial stage is to conduct exploratory data analysis (EDA). The components of the EDA that have been carried out are data description, missing values, duplicate data, statistical data, and outliers. Table 3 shows the results of data exploration from each dataset.

Table 3. EDA Components

EDA components	Dataset		
	"CancerPatient"	"Survey Lung Cancer"	"Cancer_Data "
Data description	There is one attribute of numeric, while the other attributes are categorical.	The "Gender" attribute is of a type object, so it needs to be changed categorically.	All features are numeric.
Missing value	0	0	0
Duplicated Data	848	33	0
Outlier	no	no	no
Data after drop duplicate	high 53 medium 52 low 47	yes 238 no 38	benign 357 malignant 212
Data after balanced	High 53, medium 53, low 53	Yes 238 No 238	benign 357 malignant 357

In the "Cancer patient" dataset, there is a very high duplication of 1,000 data; there are 848 identical records. Then cleaning duplicated data until 152 unique data remained with high 53, medium 52, and low 47 levels. The data is then processed directly with Data Discretization pre-processing. The value of the 'age' attribute is changed from numeric data to categorical. Feature 'age' is divided into three categories: young, middle-aged, and old. The next step is oversampling the classes with a smaller number so each class has 53 records. The total data becomes 159.

For the 'Survey Lung Cancer' dataset, there are 33 of 309 duplicate data. Then do the cleaning of the same data so that the unique data is left for class 'yes' 238

Technium: Romanian Journal of Applied Sciences and Technology
Vol. 16, 273-279, October, 2023
ISSN: 2668-778X
www.techniumscience.com

and 'no' 38. The following process is oversampling, which causes the total data to be 476. The dataset 'Cancer_Data' has no data duplication actual data for benign 357, malignant 212. After oversampling, the actual data becomes 714.

Furthermore, the GBDT classification steps require learning rate variables and n_estimator (number of stress). The learning rate values include 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, and 1. The n-estimator values use 50, 75, 100, 150, 200, and 300. The default depth of the tree is 30. For splitting data, there is 5-fold cross-validation. The experimental results for each dataset are shown in Tables 4(a) until 6(b).

Table 4a. Performance measures before randomizing oversampling for 'Cancer_Patient.'

Class	Prec.	Rec.	f1-score	Support
High	1	1	1	13
Low	1	0.92	0.96	12
Medium	0.93	1	0.96	13
Accuracy			**0.97**	38
Macro avg.	0.98	0.97	0.97	38
Weighted avg.	0.98	0.97	0.97	38

Table 4b. Performance measures after randomized oversampling for 'Cancer_Patient'

Class	Prec.	Rec.	f1-score	Support
High	1	1	1	13
Low	1	0.92	0.96	13
Medium	0.93	1	0.97	14
Accuracy			**0.97**	40
Macro avg.	0.98	0.97	0.98	40
Weighted avg.	0.98	0.97	0.97	40

Support= size of testing data; macro avg. = performa between classes; weighted avg = performa between class with weight

Table 4(a) shows the performance measure for the "Cancer Patient" dataset: Before randomized oversampling, the data consists of 3 classes showing an accuracy of up to 0.97 from 38 testing data. While precision, recall, and f1-score have the same value, namely 0.97. The best experimental results are obtained when the n_estimator is 50, and the learning rate is 1. The macro average is the average of each performance measure, while the weighted average is intended for classes with imbalanced data.

Table 4(b) for after randomized oversampling, the data consists of three classes showing an accuracy of up to 0.97 out of 40 testing data. Precision, recall, and f1-score have values of 0.98, 0.97, and 0.98 for the macro average f1-score and 0.97 weighted average f1-score. The best experimental results are obtained when the n_estimator is 150, and the learning rate is 0.25.

The difference in the results of the performance measure in the first test scenario is not too significant. The data in each class are relatively balanced; only one test data differs for each category. Therefore the results are the same between before and after randomized oversampling.

Table 5a. Performance measures before randomizing oversampling for 'Survey Lung Cancer

	Prec.	Rec.	f1-score	Support
No	0.80	0.40	0.53	10
Yes	0.91	0.98	0.94	59
Accuracy			**0.90**	69
Macro avg.	0.85	0.69	0.74	69
Weighted avg.	0.89	0.90	0.88	69

Table 5b. Performance measures after randomized oversampling for 'Survey Lung Cancer'

Class	After randomized oversampling			
	Prec.	Rec.	f1-score	Support
No	0.9	1	0.94	60
Yes	1	0.88	0.94	59
Accuracy			0.94	119
Macro avg.	0.95	0.94	0.94	119
Weighted avg.	0.95	0.94	0.94	119

Table 5(a). Shows the performance measure for the "Lung Cancer Survey" dataset: Before randomized oversampling, the data consists of 2 classes showing an accuracy of up to 0.90 from 69 testing data. While precision, recall, and f1-score have values of 0.85, 0.69, and 0.74. The best experimental results are obtained when the n_estimator is 75, and the learning rate is 0.05.

Table 5(b) shows performance after randomized oversampling. Experimental results using oversampling show an accuracy of up to 0.94 of 119 testing data. While precision, recall, and f1-score have values of 0.95, 0.94, and 0.94. The best experimental results are obtained when the n_estimator is 300, and the learning rate is 0.75. The experimental results show that randomized oversampling shows an increase in performance. At the time before oversampling, the amount of testing data in the "no" class is far less than in the "yes" class. Of course, this affects the performance of the minimum class. The recall value and f1-score in the "no" class were 0.4 and 0.53, respectively. Compare with the results of after oversampling recall and f1-score 1.00 and 0.94, respectively.

Table 6a. Performance measures before randomizing oversampling for 'Cancer Data'

	Prec.	Rec.	f1-score	Support
Benign	0.95	1	0.97	90
Malignant	1	0.91	0.95	53
Accuracy			0.97	143
Macro avg.	0.97	0.95	0.97	143
Weighted avg.	0.97	0.97	0.97	143

Table 6b. Performance measures after randomized oversampling for 'Cancer Data'

Class	After randomized oversampling			
	Prec.	Rec.	f1-score	Support
Benign	0.99	1	0.99	90
Malignant	1	0.99	0.99	89
Accuracy			0.99	179

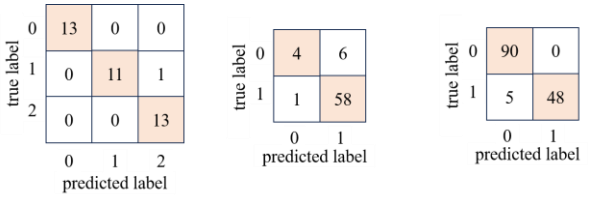
Technium: Romanian Journal of Applied Sciences and Technology
Vol. 16, 273-279, October, 2023
ISSN: 2668-778X
www.techniumscience.com

Macro avg.	0.99	0.99	0.99	179
Weighted avg.	0.99	0.99	0.99	179

Table 6(a) shows the performance measure for the "Cancer_Data" dataset. The data consists of 2 classes before oversampling aaaan accuracy of 0.97 out of 143 testing data. Precision value 0.97, recall 0.965, f1-score 0.97. Best experiment when n_estimator 300, learning rate 1.

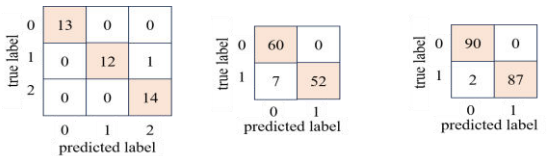
Meanwhile, Table 6(b) performance of oversampling shows an accuracy of up to 0.99 out of 179 testing data. While precision, recall, and f1-score have a value of 0.99. The best experimental results are obtained when the n_estimator is 100, and the learning rate is 0.5.

Furthermore, to find out in more detail the success of testing in each class, Figure 3 shows the confusion matrix of each dataset.



(a) (b) (c)

Fig. 3. Confusion matrix before oversampling
(a) CancerPatient (b) Survey Lung Cancer (c) Cancer_Data



(a) (b) (c)

Fig. 4. Confusion matrix after oversampling
(a) CancerPatient (b) Survey Lung Cancer (c) Cancer_Data

Figure 3(a) shows that there is an error in the "Cancer Patient" data testing dataset in classifying the "medium" class. The data should be classified as a "medium" class, but the prediction results show a "low" category. Figure 3(b) shows that there is an error in the data testing dataset "Survey Lung Cancer" in classifying class "No." In addition, there are six errors in the class classification "Yes." Figure 3(c) shows that there are five data testing errors for the "Cancer_Data" dataset in classifying the "malignant" class. Data should be classified as a "malignant" class, but the prediction results are "benign."

Figure 4(a) shows the same results as Figure 3(a). Figure 4(b) shows that there are seven errors in the data testing dataset "Survey Lung Cancer" in classifying class "No." The data should be classified as a "No" class, but the prediction results show a "Yes" class. Figure 4(c) shows that there are two data testing errors for the "Cancer_Data" dataset in classifying the "malignant" type. Tables 4, 5, and 6, as well as Figures 3 and 4 (a), (b), and (c), show that using data with randomized oversampling can produce better performance than without doing it.

Furthermore, to determine the reliability of the GBDT method, the next step is to compare it with other machine learning methods, namely k-nearest neighbor (k-NN) and Support Vector Machine (SVM). Table 7 shows the experimental results of comparing machine learning methods using the same dataset.

Table 7a. Comparison of Performance using Dataset 'Cancer Patient'

Method	Acc.	Proc.	Rec.
kNN	0.80	0.79	0.80
SVM	0.93	0.93	0.93
GBDT	0.97	0.97	0.97

Table 7b. Comparison of Performance using dataset 'Survey Ling Cancer'

Method	Acc.	Proc.	Rec.
kNN	0.90	0.92	0.90
SVM	0.99	0.99	0.99
GBDT	0.94	0.95	0.94

Table 7c. Comparison of Performance using dataset 'Cancer Data'

Method	Acc.	Proc.	Rec.
kNN	0.95	0.96	0.95
SVM	0.75	0.84	0.75
GBDT	0.99	0.99	0.99

Table 7(a) until 7(c) shows that the "Cancer patient" and "Cancer_Data" datasets perform better than the other two methods. However, SVM performs best in the "Lung Cancer Survey" dataset. The experiment proves there is no justification for the best method, depending on the data used. However, it can be ascertained from three experiments with different datasets that the GBDT method is superior on average compared to other methods.

4 Conclusion

In this study, classification of lung cancer data has been made using the Gradient Boosting Decision Tree. Three datasets have been experimented with to get good system performance. The experimental results show that using Randomized Oversampling can improve system performance. In addition, the GDBT classification method can be used as an alternative to k-NN and SVM. You can use a dataset with more records and multiclass in future research. Furthermore, you can compare the use of the GDBT and its variants, such as the Light Gradient Boost Machine (LGBM) and eXtreme Gradient Boosting (XGBoost).

Acknowledgment

This work was supported by the second year of the National Competitive Research Program & Assignment 2023. National Competitive Basic Research scheme from the Indonesian Ministry of Education, Culture, Research, and Technology.

References

[1] E. Goodarzi, R. Beiranvand, H. Naemi, S. Rahimi Pordanjani, and Z. Khazaei,

"Geographical Distribution Incidence and Mortality of Breast Cancer and Its Relationship With the Human Development Index (HDI): an Ecology Study in 2018," *World Cancer Res. J.*, vol. 7, pp. 1–11, 2020, doi: 10.32113/wcrj_20201_1468.

[2] L. Wheless, J. Brashears, and A. J. Alberg, "Epidemiology of lung cancer," *Lung Cancer Imaging*, pp. 1–15, 2013, doi: 10.1007/978-1-60761-620-7_1.

[3] H. Sung *et al.*, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA. Cancer J. Clin.*, vol. 71, no. 3, pp. 209–249, 2021, doi: 10.3322/caac.21660.

[4] S. V. S. Deo, J. Sharma, and S. Kumar, "GLOBOCAN 2020 Report on Global Cancer Burden: Challenges and Opportunities for Surgical Oncologists," *Ann. Surg. Oncol.*, vol. 29, no. 11, pp. 6497–6500, 2022, doi:10.1245/s10434-022-12151-6.

[5] F. Ramadhaniah, D. Khairina, D. T. Sinulingga, E. Suzanna, and A. M. Jayusman, "Distribution of Lung Cancer Patients in Dharmais Cancer Hospital Year 2008-2012, "*J. Respirologi Indonesia.*, vol. 39, no. 1, pp. 31–36, 2019, doi: 10.36497/jri.v39i1.1.

[6] D. K. Sanie, A. D. Susanto, and F. Harahap, "Respiratory Disorders and Lung Physiology in Scavengers in Bantar Gebang, Bekasi, "*J. Respirologi Indonesia.*, flight. 39, no. 2, pp. 70–78, 2019.

[7] L. Sari, A. Romadloni, and R. Listyaningrum, "Application of Data Mining in Lung Cancer Prediction Analysis Using Random Forest Algorithm, "*Infotekmen*, vol. 14, no. 01, p. 155–162, 2023, doi: 10.35970/infotekmesin.v14i1.1751.

[8] E. Dritsas and M. Trigka, "Lung Cancer Risk Prediction with Machine Learning Models," *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, doi: 10.3390/bdcc6040139.

[9] S.I. Maiyanti*et al.*, "Comparison of Classification of Lung Cancer Using Support Vector Machine and K-Nearest Neighbor," vol. 18, no. 1, pp. 54–62, 2023.

[10] V. Prakash and P. Smitha Vas, "Survey on Lung Cancer Detection Techniques," *2020 Int. Conf. Comput. Perform. Eval. ComPE 2020*, no. June, pp. 800–803, 2020, doi: 10.1109/ComPE49325.2020.9200019.

[11] Z. Karhan and T. Tunc, "Lung cancer detection and classification with DGMM-RBCNN technique," *Neural Comput. Appl.*, vol. 33, no. 22, p. 15601–15617, 2021, doi: 10.1007/s00521-021-06182-5.

[12] M. A. Bhat, "Lung Cancer Does Smoking Cause Lung Cancer," *Kaggle*, 2021. https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer (accessed May 08, 2023).

[13] "Lung Cancer Prediction Air Pollution, Alcohol, Smoking & Risk of Lung Cancer," *Kaggle*, 2022. https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link%0D%0A (accessed May 08, 2023).

[14] E. Taha, "Cancer Data Benign and malignant cancer data," *Kaggle*, 2023. https://www.kaggle.com/datasets/erdemtaha/cancer-data (accessed May 08, 2023).

[15] "Exploratory Data Analysis (EDA)," E. Camizuli and E. J. Carranza.*Encycl. Archaeol. Sci.*, no. 3, pp. 1–7, 2018, doi: 10.1002/9781119188230.saseas0271.

[16] T. Milo and A. Somech, "Automating Exploratory Data Analysis via Machine Learning: An Overview," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 2617–2622, 2020, doi: 10.1145/3318464.3383126.

[17] D. Dey, "Binning or Discretization," *Geeks for geeks*, 2022. https://www.geeksforgeeks.org/ml-binning-or-discretization/.

[18] A. Chkifa and M. Dolbeault, "Randomized least-squares with minimal oversampling and interpolation in general spaces," pp. 1–17, 2023, [Online]. Available: http://arxiv.org/abs/2306.07435.

[19] T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," *Inf.*, vol. 14, no. 1, 2023, doi: 10.3390/info14010054.

[20] T. R. Mahesh *et al.*, "AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease," *Account Intel. Neuroscientists*, vol 2022, 2022, doi: 10.1155/2022/9005278.

[21] M. A. Khan *et al.*, "Geopolymer Concrete Compressive Strength via Artificial Neural Network, Adaptive Neuro-Fuzzy Interface System, and Gene Expression Programming With K-Fold Cross Validation," *Front. Mater.*, vol. 8, no. May, pp. 1–19, 2021, doi: 10.3389/fmats.2021.621163.

[22] R. Sun, G. Wang, W. Zhang, L. T. Hsu, and W. Y. Ochieng, "A gradient boosting decision tree based GPS signal reception classification algorithm," *Appl. Soft Comput. J.*, vol. 86, no. xxxx, p. 105942, 2020, doi: 10.1016/j.asoc.2019.105942.

[23] J. G. Ghatkar, R. K. Singh, and P. Shanmugam, "Classification of algal bloom species from remote sensing data using an extreme gradient boosted decision tree model," *Int. J. Remote Sens.*, vol. 40, no. 24, p. 9412–9438, 2019, doi: 10.1080/01431161.2019.1633696.

[24] K. Zhou, J. Zhang, Y. Ren, Z. Huang, and L. Zhao, "A gradient boosting decision tree algorithm combining synthetic minority oversampling technique for lithology identification," *Geophysics*, vol. 85, no. 4, pp. WA147–WA158, 2020, doi: 10.1190/geo2019-0429.1.

[25] M. Koklu and I. A. Ozkan, "Multiclass

classification of dry beans using computer vision and machine learning techniques," *Comput. Electron. Agric.*, vol. 174, no. May, p. 105507, 2020, doi:10.1016/j.compag.2020.105507.

[26] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label Classifier Performance Evaluation with Confusion Matrix," pp. 01–14, 2020, doi: 10.5121/csit.2020.100801.