Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

A Comparative Performance of SMOTE, ADASYN and Random Oversampling in Machine Learning Models on Prostate Cancer Dataset

Aditya Herdiansyah Putra [1]*, Abu Salam [2]**
* Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang
111202113948@mhs.dinus.ac.id [1], abu.salam@dsn.dinus.ac.id [2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Imbalance Class;
Oversampling;
Classification;
Machine Learning;
Prostate Cancer.

ABSTRACT

Class imbalance in medical datasets, including prostate cancer, can affect the performance of machine learning models in detecting minority cases. This study compares three oversampling techniques-SMOTE, ADASYN, and Random Oversampling-in addressing data imbalance in prostate cancer classification. The techniques are applied to Random Forest (RF), Decision Tree (DT), and LightGBM (LGBM) models which are evaluated using accuracy, precision, recall, F1-score, and ROC AUC. To improve the reliability of the evaluation, this study applied K-Fold Cross Validation, which helps reduce the risk of overfitting and ensures more stable and accurate accuracy results. The results show that the effectiveness of oversampling techniques depends on the algorithm used. Random Oversampling gave the best results in Random Forest with accuracy 0.85, recall 0.888, precision 0.873, F1-score 0.879, and ROC-AUC 0.838. The Decision Tree model showed the highest performance with SMOTE, resulting in accuracy 0.80, recall 0.838, precision 0.843, F1-score 0.839, and ROC-AUC 0.788. Meanwhile, the LGBM model obtained the highest improvement with ADASYN, achieving accuracy 0.89, recall 0.919, precision 0.913, F1-score 0.913, and ROC-AUC 0.879. In conclusion, there is no superior oversampling method for all models, as the effectiveness of resampling depends on the algorithm as well as the prioritized evaluation metrics.

I. INTRODUCTION

Machine learning optimization in detecting prostate cancer is very important in the medical world because it can increase the chances of successful treatment and extend the life expectancy of patients. Machine learning has gotten to be an progressively imperative apparatus in helping the conclusion of medical data-based diseases, including prostate cancer. With its ability to detect patterns that are difficult for humans to identify, machine learning can provide more accurate predictions in supporting medical decisions [1]. However, when applied to medical classification, one of the main challenges often faced is class imbalance in the dataset used [2].

Class imbalance is a condition where the number of samples in one class is much larger than the other classes. In the case of prostate cancer datasets, there are often more patients with malignant tumors than patients diagnosed with benign tumors. This imbalance can cause machine learning models to be more accurate in classifying the majority class (malignant tumors) but less sensitive in detecting the minority class (benign tumors)[3]. Therefore, a special approach is needed to overcome data imbalance so that the model can provide more precise predictions.

One of the commonly used methods to handle class imbalance is the data resampling technique. Resampling is a technique in data processing used to change the number of samples in a dataset. One of the resampling methods is oversampling [4]. Basically, oversampling can be done by increasing the sample of minority classes, either by creating a synthetic sample or by duplicating an existing sample. [5]. There are several studies that have discussed the importance of oversampling techniques in handling class imbalance. For example, research by Rahmi et al. showed that SMOTE is superior to Random Oversampling in detecting cervical cancer in Indonesia based on a higher AUC value using the Naïve Bayes model [6]. However, research conducted by M. Khushi et al. proved that over-sampling techniques with Random

e-ISSN: 2548-6861

Oversampling (ROS) and Random Forest proved to be the most effective in improving lung cancer prediction on unbalanced datasets [2]. In addition, there is also a mention that the Adaptive Synthetic (ADASYN) technique has a satisfactory performance in the case of lung cancer with several models. This research was conducted by Assegie at al. who proved the technique on several models such as Logistic Regression, SVM, and Random Forest [7]. From some of these studies, resampling techniques with SMOTE, Random Oversampling and ADASYN methods can have an influence in optimizing machine learning performance in several case studies.

From the great results of the three resampling techniques in several previous studies, this study will compare the performance of the three techniques by adding the three resampling techniques in several machine learning models to be tested in the case of prostate cancer detection. This aims to analyze the performance results of each resampling technique and make comparisons to get the most optimal oversampling technique to be used in prostate cancer detection cases.

II. METHOD

In this research, several experiments will be conducted to test three oversampling techniques namely SMOTE, Random Oversampling and ADASYN on several machine learning methods. This research flow begins with data collection from Kaggle. Then the data is preprocessed through label encoding and normalization stages before being divided into training and test data. Next, three resampling techniques are applied, namely SMOTE, Random Oversampling, and ADASYN, to handle class imbalance in the dataset. The resampled data was then used to train various machine learning models, including Random Forest, Decision Tree and LightGBM. After training, the models were evaluated to assess their performance before the research process was completed. The research flow is depicted in figure 1.

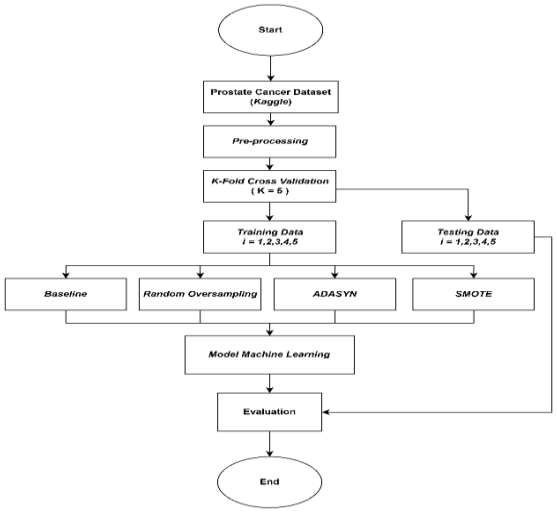

Figure 1. Research Scheme

A. Data Collection

This study uses the public dataset Prostate_Cancer available on Kaggle. This dataset consists of 100 patient data with a diagnosis of prostate cancer. Each data has 8 numerical variables as features, which include radius, texture, perimeter, area, smoothness, compactness, symmetry, and fractal dimension. In addition, this dataset has one categorical variable as a label that classifies the data into two categories, namely Malignant (M) for malignant prostate cancer and Benign (B) for benign prostate cancer. The visualization of the dataset is depicted in table 1.

TABLE I
PROSTATE CANCER DATASET PARAMETERS

Name	Data Type	Value Range
Radius	Int64	9 – 25
Texture	Int64	11 – 27
Perimeter	Int64	52 - 172
Area	Int64	202 - 1878
Smoothness	Float64	0.07 - 0.143
Compactness	Float64	0.038 - 0.345
Symmetry	Float64	0.135 - 0.304
Fractal_Dimention	Float64	0.053 - 0.097
Diagnosis_Result	Object	M, B

This dataset is used as the basis in building a classification model to detect prostate cancer more accurately. This dataset has an imbalance of data with the number of malignant being 62 and the number of malignant being 38. The visualization of the imbalance data is shown in Figure 2.

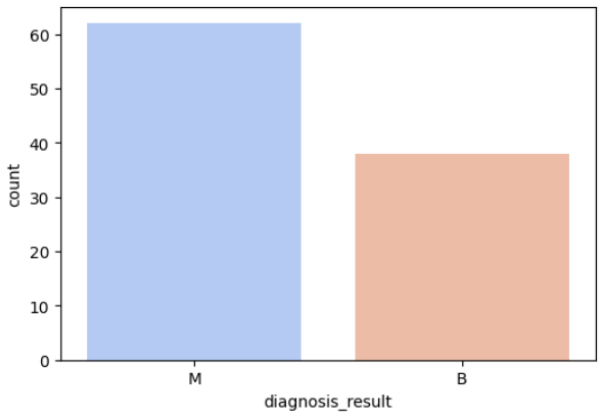

Figure 2. Class Distribution on Dataset

B. Preprocessing

In the preprocessing stage, the data will be subjected to a label encoding process. Label Encoding is a technique in machine learning that transforms text-based categorical data into a numerical format. This process is done by assigning a unique numerical value to each category or label in a

categorical variable, so that the data can be more easily processed by the model [8]. At this stage, the category on the malignant label (M) is converted to 1 while the belign (B) is converted to 0. Then the data is normalized on each feature, namely radius, texture, perimeter, area, smoothness, compactness, symmetry, and fractal dimension to ensure uniform data scale and avoid dominance of certain features in the model training process. At this stage, normalization uses the StandardScaler technique. Standard Scaler is a preprocessing method that standardizes features by removing the mean and scaling the unit variance for each sample [9]. This technique is used to prevent the dominance of features with large values so that the model training process is more optimal.

C. K-Fold Cross Validation

K-Fold Cross Validation is a validation method in machine learning used to assess model performance by dividing the dataset into K parts (folds). In this process, the dataset is separated into two main parts: training data and test data. The training data is used to teach the model to recognize patterns in the data, while the test data serves to measure the extent to which the model can generalize to new data that has never been seen before. [10], [11]. In this dataset, the division is done with a ratio of 80% for training data and 20% for test data. With this split data ratio, the model has enough data to learn as well as can be tested with a balanced proportion [11]. Each section is alternately used as test data, while the rest is used as training data. This process is repeated K times so that each section is used as test data once. This technique helps reduce the risk of overfitting in model evaluation and ensures more stable and accurate accuracy results as the model is tested with various subsets of data [12]. The following is the formula for Kfold Cross Validation.

$$CV = \frac{1}{K}\sum_{i=1}^{K} Acc_i$$

K : Number of folds
Acc_i : Model accuracy
CV : Average accuracy of all folds

D. Oversampling

In this study, an oversampling technique was performed to overcome the class imbalance in the prostate cancer dataset. The three oversampling methods used are SMOTE (Synthetic Minority Over-sampling Technique), Random Oversampling, and ADASYN (Adaptive Synthetic Sampling Approach).

1. SMOTE (Synthetic Minority Over-sampling Technique)
   SMOTE is an oversampling technique that generates new synthetic data to increase the number of samples in the minority class. This approach utilizes k-nearest neighbor to create synthetic samples in the feature space based on a certain percentage of minority classes [13], [14]. First, SMOTE identifies the minority classes in the dataset, then randomly selects samples from those classes. After that, the algorithm searches for k-nearest neighbors of the selected sample using a distance metric such as Euclidean distance. From this, synthetic data is generated by interpolating between the selected data point and one of its neighbors using the following formula:

$$X_{syn} = X_i + (X_{knn} - X_i) \times \delta$$

X_{syn} : Synthetic data generated
X_i : Original sample data from the minority class
X_{knn} : One of the nearest neighbors of X_i
δ : Random value between 0 and 1

   This process is repeated until the number of minority class samples reaches a better level of balance with the majority class. Thus, SMOTE helps address the problem of imbalanced data without simply duplicating the original data, thereby reducing the risk of overfitting and improving the performance of machine learning models [14].
2. Random Oversampling
   Random Oversampling is done by adding random copies of the minority class data until it reaches a balance with the majority class [15], [16]. This technique works by randomly selecting samples from the minority class and duplicating them so that the data distribution becomes more balanced. Mathematically, in determining the number of new samples for the minority class after oversampling, it is calculated by the following formula.

$$N_{result} = N_{minor} + k \times (N_{major} - N_{minor})$$

   N_{minor} = Initial sample size of the minority class
   N_{major} = Initial sample size of the majority class
   K = Oversampling factor ($0 \leq k \geq 1$)
   N_{result} = The number of minority samples after oversampling.
3. ADASYN (Adaptive Synthetic Sampling Approach)
   ADASYN is a method like SMOTE that works by adjusting the number of synthetic samples generated based on the complexity of the data [17]. If a minority sample is in a low-density region or close to the majority class boundary, then more synthetic samples will be generated to improve the representation of the minority class. Conversely, if a minority sample is already in a high-density region, fewer synthetic samples will be generated.

e-ISSN: 2548-6861

E. Model Machine Learning

Machine learning is a data-driven approach that allows computers to recognize patterns from previous data and predict without explicit programming instructions. In this study, machine learning is used to build a classification model to detect prostate cancer based on numerical features available in the dataset. Evaluation of classification performance is done by applying three main algorithms, namely Random Forest, Decision Tree, and LightGBM.

Random Forest is the general principle of a random ensemble consisting of Decision trees in which this model divides classes in a binary manner repeatedly until reaching the final result [18]. Meanwhile, Decision Tree is a classification method that divides data into branches based on certain features that are used to make decisions hierarchically [19]. LightGBM is based on Gradient Boosting Decision Tree (GBDT), optimizing training speed and memory efficiency through Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) techniques without compromising accuracy [20]. With this approach, the research aims to determine the best algorithm to detect prostate cancer more efficiently and accurately.

F. Evaluation

The last stage in this research is evaluation. Evaluation in machine learning is the process of measuring the performance of a model using certain metrics to determine how well the model can make predictions. The evaluation metrics used in this research are as follows:

1. Accuracy: The value resulting from how often the model makes correct predictions overall [21]. The accuracy value is obtained by calculating with the following formula.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Prediction}$$

2. Precision: A value resulting from how many of the positive predictions are actually positive [22]. High precision means that the model rarely misclassifies negatives as positives.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

3. Recall: A value that measures how much positive data was correctly classified [23]. This value is used when positive data but predicted as negative should be minimized.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

4. F1-Score: the harmonic means of precision and recall. This is useful when the dataset is not balanced, as it considers the balance between precision and recall [24]. The F1-Score value can be obtained through the following formula:

$$F1 - Score = 2 \times \frac{True\ Positive}{True\ Positive + False\ Negative}$$

5. ROC AUC: The metrics used to survey the model can separate between positive and negative classes [25]. ROC (Receiver Operating Characteristic) can be a curve that describes the relationship between True Positive Rate (TPR) and False Positive Rate (FPR) on different sides of the choice. AUC (Area Under the Curve) measures the area under the ROC curve, with values between and 1. An AUC close to 1 indicates that the event has good classification execution. An AUC around 0.5 implies that the event is no better than a random guess. An AUC close to 1 indicates that the event is very poor at recognizing classes.

III. RESULT AND DISCUSSION

This section will show the test results of three oversampling methods namely SMOTE, ADASYN, and Random Oversampling applied to several Machine Learning models. This aims to find the oversampling method that has the most optimal performance in Machine Learning on Prostate Cancer Dataset. In each test result, the value obtained is the average of each iteration on K-Fold Cross Validation, which has been done 5 times, to ensure more stable and accurate results.

A. Test Results Based on Accuracy on the Model

Based on the results shown in Table 2, each model shows a different response to the resampling method. In the Random Forest (RF) model, the Random Oversampling technique produced the highest accuracy of 0.85 outperforming other resampling methods as well as the baseline. The Decision Tree (DT) model obtained the highest accuracy improvement with SMOTE reaching 0.80 while the ADASYN method decreased the model accuracy to 0.71. Meanwhile, the LightGBM (LGBM) model showed the most significant improvement in accuracy with ADASYN reaching 0.89 which is higher than all other resampling techniques.

TABLE 2
MODEL TESTING ACCURACY RESULTS

Model	Baseline	SMOTE	ADASYN	Random Oversampling
RF	0,84	0,82	0,83	**0,85**
DT	0,77	**0,80**	0,71	0,77
LGBM	0,81	0,80	**0,89**	0,85

B. Testing Results Based on Precision on the Model

Based on the test results shown in Table 3, each oversampling method has a varying impact on the precision value depending on the model used. In the Random Forest (RF) model, the Random Oversampling technique gives the highest precision of 0.873, slightly better than ADASYN 0.871 and baseline 0.860, while SMOTE 0.858 shows a small decrease. The Decision Tree (DT) model had the highest precision at Baseline of 0.850 but experienced a slight decrease after applying SMOTE at 0.843 and dropped further with ADASYN at 0.786. Nevertheless, the Random Oversampling method managed to increase the precision of the DT model to 0.826, although it has not exceeded the baseline. Meanwhile, the LightGBM (LGBM) model showed the most significant increase in precision with ADASYN of 0.913, which is the highest value compared to other oversampling methods. The precision value in this model also increases with Random Oversampling, which is 0.864. However, SMOTE which gets a value of 0.824 only provides a slight change compared to the baseline of 0.825.

TABLE 3
MODEL TESTING PRECISION RESULTS

Model	Baseline	SMOTE	ADASYN	Random Oversampling
RF	0,860	0,858	0,871	**0,873**
DT	**0,850**	0,843	0,786	0,826
LGBM	0,825	0,824	**0,913**	0,864

C. Test Results Based on Recall on the Model

Based on the test results in Table 4, the Random Forest (RF) model achieved the highest recall in the Baseline with a value of 0.888, which is the same as the results using Random Oversampling. Meanwhile, the SMOTE method resulted in a slight decrease in recall to 0.857, followed by ADASYN which recorded a value of 0.855. In the Decision Tree (DT) model, the SMOTE technique showed an increase in recall compared to the baseline with a value of 0.838, while ADASYN decreased to 0.743. Even so, the Random Oversampling method was still able to increase recall to 0.808, although it did not surpass the results obtained with SMOTE. Meanwhile, the LightGBM (LGBM) model recorded the best recall performance with ADASYN, reaching 0.919, which is the highest value compared to other oversampling methods. Recall in this model also increased with the application of Random Oversampling, reaching 0.905. On the other hand, the SMOTE method produced a recall of 0.871, slightly lower than the baseline of 0.885.

TABLE 4
MODEL TESTING RECALL RESULTS

Model	Baseline	SMOTE	ADASYN	Random Oversampling
RF	**0,888**	0,857	0,855	**0,888**
DT	0,775	**0,838**	0,743	0,808
LGBM	0,885	0,871	**0,919**	0,905

D. Testing Results Based on F1-Score on the Model

In Table 5, each machine learning model shows the highest F1-Score performance with different oversampling methods. The Random Forest (RF) model achieves the highest F1-Score of 0.879 when using Random Oversampling, indicating that this method can improve the balance between precision and recall in the model. In the Decision Tree (DT) model, the SMOTE method provided the best results with an F1-Score of 0.839, which was superior to the other methods. This shows that SMOTE can improve the classification of minority classes in the Decision Tree, improving the balance in detecting positive samples without increasing false positives too much. Meanwhile, the LightGBM (LGBM) model achieved the highest F1-Score of 0.913 with ADASYN, making it the most effective oversampling method for this model. ADASYN successfully improved the model's performance in better detecting positive samples, strengthening the model's generalizability to unbalanced data.

TABLE 5
MODEL TESTING F1-SCORE RESULTS

Model	Baseline	SMOTE	ADASYN	Random Oversampling
RF	0,872	0,855	0,862	**0,879**
DT	0,806	**0,839**	0,759	0,813
LGBM	0,852	0,844	**0,913**	0,882

E. Testing Results Based on ROC-AUC on the Model

Based on the test results using ROC AUC, each model shows the best performance with different oversampling methods. In Table 6, the Random Forest (RF) model achieves the highest ROC AUC of 0.838 with Random Oversampling, indicating that this method can improve the model's ability to distinguish between positive and negative classes better. In the Decision Tree (DT) model, the SMOTE method provides the best results with an ROC AUC of 0.788, indicating that this technique is more effective in improving model performance than other resampling methods. Meanwhile, the LightGBM (LGBM) model obtained the highest ROC AUC of 0.879 when using ADASYN, making it the most optimal oversampling method to improve the model's ability to better classify the data.

e-ISSN: 2548-6861

TABLE 5
MODEL TESTING ROC-AUC RESULTS

Model	Baseline	SMOTE	ADASYN	Random Oversampling
RF	0,824	0,809	0,822	**0,838**
DT	0,768	**0,788**	0,703	0,759
LGBM	0,784	0,776	**0,879**	0,831

F. Model Evaluation Metrics

This section presents a visualization of the test results to see the impact of applying oversampling techniques on model performance. The graphs show a comparison of the results before and after oversampling is applied to the unbalanced data. With this visualization, we can observe the pattern of changes and improvements that occur after adjusting the data distribution. The visualization of the test results is presented in the figure below.

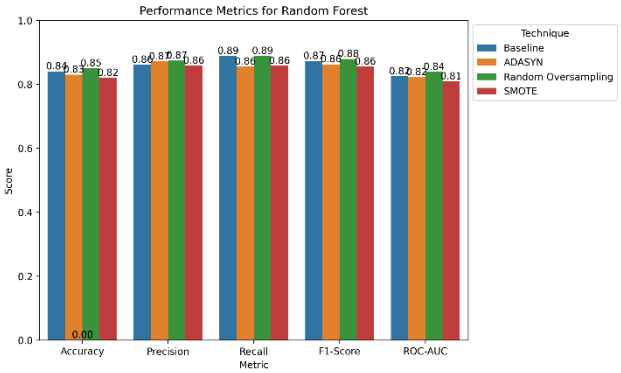

Figure 3. Evaluation Metrix for Random Forest

Based on Figure 3, Random Oversampling (RO) performs better than ADASYN and SMOTE in the Random Forest model. RO has higher Accuracy, indicating the model maintains overall accuracy. RO's Recall and F1-Score were superior, indicating better ability to recognize minority classes without sacrificing Precision. RO's ROC-AUC is also higher, indicating the model is better at distinguishing between positive and negative classes. Meanwhile, SMOTE decreased Accuracy and ROC-AUC, potentially leading to overfitting.

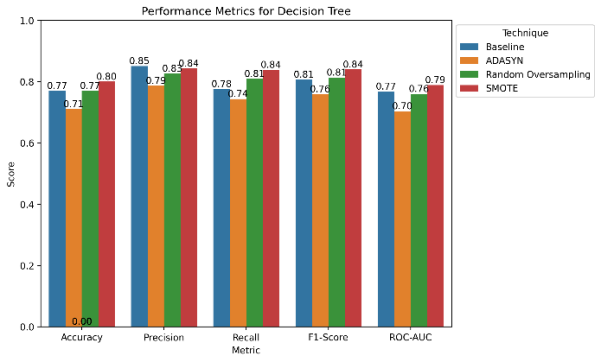

Figure 4. Evaluation Matrix for Decision Tree

From the results shown in Figure 4, SMOTE provides the best overall performance in almost all metrics, especially in Precision 0.84, Recall 0.84, and F1-Score 0.84. The Random Oversampling technique also shows improvement over the Baseline but is still below SMOTE. Meanwhile, ADASYN seems to provide a smaller improvement, even having a lower Accuracy value than the other techniques.

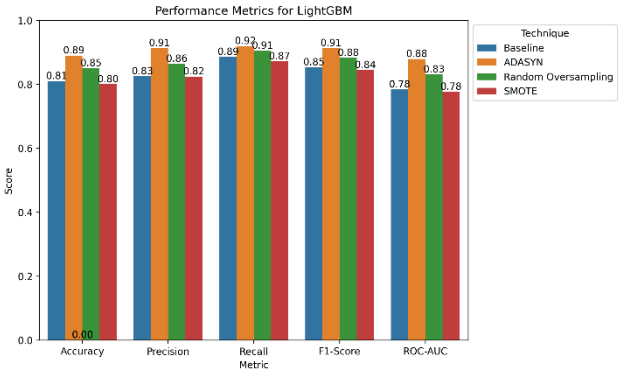

Figure 5. Evaluation Matrix for LightGBM

From the metrics shown in Figure 5, the ADASYN technique has the best overall performance. ADASYN achieved the highest values of Accuracy 0.89, Precision 0.91, Recall 0.92, F1-Score 0.91, and ROC-AUC 0.88. This technique shows a good balance between Precision and Recall, resulting in the highest F1-Score. Meanwhile, Baseline had the lowest performance in most metrics, while SMOTE and Random Oversampling performed reasonably well but did not exceed ADASYN.

IV. CONCLUSION

The optimal oversampling method depends on the characteristics of the Machine Learning model and the metric evaluation results. Random Forest (RF) works best with Random Oversampling as it improves class balance without compromising model performance. This technique resulted in an Accuracy 0.85, Precision 0.873, Recall 0.888, F1-Score 0.879 and ROC AUC 0.838. Meanwhile, Decision Tree (DT)

showed the best results with SMOTE, which helps to balance the classes by creating synthetic samples. With this method, DT achieved an Accuracy 0.80, Precision 0.843, Recall 0.838, F1-Score 0.839, and ROC AUC 0.788 thus being able to better recognize data patterns. LightGBM (LGBM) achieved the highest performance with ADASYN, which is more adaptive in fitting synthetic samples to the original data distribution. This technique improved the model's sensitivity to minority classes, as shown by Accuracy 0.89, Precision 0.913, Recall 0.919, F1-Score 0.913, and ROC AUC 0.879.

In conclusion, Random Oversampling is suitable for RF because it keeps the performance stable, SMOTE is effective for DT because it balances the classes, and ADASYN is the best choice for LGBM because of its ability to fit the data more flexibly. For future research, it is recommended to combine oversampling and undersampling techniques to optimize the handling of data imbalance. In addition, testing on more complex models, such as advanced deep learning or ensemble learning, can provide greater insight into the effectiveness of resampling methods. That way, further studies can help improve the accuracy and generalizability of the model in detecting prostate cancer or other diseases with similar dataset characteristics.

REFERENCES

[1] D. Kusuma Ningrum and A. Maytsa Ismawardi, "EFEKTIVITAS ALGORITMA KECERDASAN BUATAN DALAM IMPLEMENTASI KESEHATAN MENTAL : SYSTEMATIC LITERATURE REVIEW," 2025.

[2] M. Khushi *et al.*, "A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data," *IEEE Access*, vol. 9, pp. 109960–109975, 2021, doi: 10.1109/ACCESS.2021.3102399.

[3] A. Muzakir, A. Desiani, and A. Amran, "Klasifikasi Penyakit Kanker Prostat Menggunakan Algoritma Naïve Bayes dan K-Nearest Neighbor," *Komputika : Jurnal Sistem Komputer*, vol. 12, no. 1, pp. 73–79, May 2023, doi: 10.34010/komputika.v12i1.9629.

[4] F. Gurcan and A. Soylu, "Learning from Imbalanced Data: Integration of Advanced Resampling Techniques and Machine Learning Models for Enhanced Cancer Diagnosis and Prognosis," *Cancers (Basel)*, vol. 16, no. 19, Oct. 2024, doi: 10.3390/cancers16193417.

[5] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, Institute of Electrical and Electronics Engineers Inc., Apr. 2020, pp. 243–248. doi: 10.1109/ICICS49469.2020.239556.

[6] N. S. Rahmi, N. W. S. Wardhani, M. B. Mitakda, R. S. Fauztina, and I. Salsabila, "SMOTE Classification and Random Oversampling Naive Bayes in Imbalanced Data : (Case Study of Early Detection of Cervical Cancer in Indonesia)," in *Proceedings of the 2022 IEEE 7th International Conference on Information Technology and Digital Applications, ICITDA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICITDA55840.2022.9971421.

[7] T. A. Assegie, A. O. Salau, K. Sampath, R. Govindarajan, S. Murugan, and B. Lakshmi, "Evaluation of Adaptive Synthetic Resampling Technique for Imbalanced Breast Cancer Identification," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 1000–1007. doi: 10.1016/j.procs.2024.04.095.

[8] C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi," *Technologia : Jurnal Ilmiah*, vol. 15, no. 1, p. 93, Jan. 2024, doi: 10.31602/tji.v15i1.13457.

[9] T. Zulhaq Jasman, E. Hasmin, C. Susanto, and W. Musu, "Perbandingan Logistic Regression, Random Forest, dan Perceptron pada Klasifikasi Pasien Gagal Jantung," *CSRID Journal*, vol. 14, no. 3, pp. 271–286, 2022, doi: 10.22303/csrid.14.3.2022.271-286.

[10] R. Oktafiani, A. Hermawan, and D. Avianto, "Pengaruh Komposisi Split data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning," *Jurnal Sains dan Informatika*, pp. 19–28, Jun. 2023, doi: 10.34128/jsi.v9i1.622.

[11] Baiq Nurul Azmi, Arief Hermawan, and Donny Avianto, "Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 4, no. 4, pp. 281–290, Feb. 2023, doi: 10.35746/jtim.v4i4.298.

[12] W. Wijiyanto, A. I. Pradana, S. Sopingi, and V. Atina, "Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa," *Jurnal Algoritma*, vol. 21, no. 1, May 2024, doi: 10.33364/algoritma/v.21-1.1618.

[13] Ridwan, E. Heni Hermaliani, and M. Ernawati, "Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada Klasifikasi Ujaran Kebencian," Jan. 2024. [Online]. Available: http://jurnal.bsi.ac.id/index.php/co-science

[14] M. Persada Pulungan, A. Purnomo, and A. Kurniasih, "PENERAPAN SMOTE UNTUK MENGATASI IMBALANCE CLASS DALAM KLASIFIKASI KEPRIBADIAN MBTI MENGGUNAKAN NAIVE

e-ISSN: 2548-6861

BAYES," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, Sep. 2024, doi: 10.25126/jtiik.2024117989.

[15] S. Diantika, "PENERAPAN TEKNIK RANDOM OVERSAMPLING UNTUK MENGATASI IMBALANCE CLASS DALAM KLASIFIKASI WEBSITE PHISHING MENGGUNAKAN ALGORITMA LIGHTGBM," 2023.

[16] R. Aryanti, T. Misriati, and R. Hidayat, "Klasifikasi Risiko Kesehatan Ibu Hamil Menggunakan Random Oversampling Untuk Mengatasi Ketidakseimbangan Data," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 3, no. 5, pp. 409–416, 2023, [Online]. Available: https://djournals.com/klik

[17] I. Dey and V. Pratap, "A Comparative Study of SMOTE, Borderline-SMOTE, and ADASYN Oversampling Techniques using Different Classifiers," in *Proceedings - 2023 3rd International Conference on Smart Data Intelligence, ICSMDI 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 294–302. doi: 10.1109/ICSMDI57622.2023.00060.

[18] N. Wuryani, S. Agustiani, I. Komputer, and N. Mandiri, "Random Forest Classifier untuk Deteksi Penderita COVID-19 berbasis Citra CT Scan," *Jurnal Teknik Komputer AMIK BSI*, vol. 7, no. 2, 2021, doi: 10.31294/jtk.v4i2.

[19] R. N. Ramadhon, A. Ogi, A. P. Agung, R. Putra, S. S. Febrihartina, and U. Firdaus, "Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank," 2024.

[20] E. Febriantoro, E. Setyati, and J. Santoso, "PEMODELAN PREDIKSI KUANTITAS PENJUALAN MAINAN MENGGUNAKAN LightGBM," *SMARTICS Journal*, vol. 9, no. 1, pp. 7–13, Apr. 2023, doi: 10.21067/smartics.v9i1.8279.

[21] H. Mahmud Nawawi, A. Baitul Hikmah, A. Mustopa, and G. Wijaya, "Model Klasifikasi Machine Learning untuk Prediksi Ketepatan Penempatan Karir," *Jurnal SAINTEKOM*, vol. 14, no. 1, pp. 13–25, Mar. 2024, doi: 10.33020/saintekom.v14i1.512.

[22] A. Alim Murtopo, M. Aditdya, P. Septiana Ananda, and G. Gunawan, "PENERAPAN COMPUTER VISION UNTUK MENDETEKSI KELENGKAPAN ATRIBUT SISWA MENGGUNAKAN METODE CNN," vol. 11, no. 2, 2024.

[23] E. Ramadanti, D. A. Dinathi, C. Sri, K. Aditya, and R. Chandranegara, "Diabetes Disease Detection Classification Using Light Gradient Boosting (LightGBM) With Hyperparameter Tuning," *Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 2, 2024, doi: 10.33395/v8i2.13530.

[24] A. Candra, Moh. Erkamim, M. Muharrom, and E. Prayitno, "Klasifikasi Stunting Pada Balita Berdasarkan Status Gizi Menggunakan Pendekatan Support Vector Machine (SVM)," *Jurnal Ilmiah FIFO*, vol. 16, no. 2, p. 171, Nov. 2024, doi: 10.22441/fifo.2024.v16i2.007.

[25] C. Prakoso and A. Hermawan, "KLIK: Kajian Ilmiah Informatika dan Komputer Perbandingan Model Machine Learning dalam Analisis Sentimen Ulasan Pengunjung Keraton Yogyakarta pada Google Maps," *Media Online*, vol. 4, no. 3, pp. 1292–1302, 2023, doi: 10.30865/klik.v4i3.1419.