Improving Imbalanced Dataset Classification Using Oversampling and Gradient Boosting

1st Nurheri Cahyana
Informatics Engineering
Universitas Pembangunan Nasional "Veteran" Yogyakarta
Daerah Istimewa Yogyakarta, Indonesia
nur.hericahyana@upnyk.ac.id

2st Siti Khomsah
Informatics Engineering
Institut Teknologi Telkom
Purwokerto, Jawa Tengah, Indonesia
siti@ittelkom-pwt.ac.id

3st Agus Sasmito Aribowo
Informatics Engineering
Universitas Pembangunan Nasional "Veteran" Yogyakarta
Daerah Istimewa Yogyakarta, Indonesia
sasmito.skom@upnyk.ac.id

***Abstract*—Imbalanced data classification is challenging task for various datasets in the real world. One of technique to enlarge the sample in minority class is oversampling to fix size as majority class. This research aims to test SMOTE, Borderline-SMOTE, and ADASYN to handle dataset imbalance and to observe its impact toward classification accuracy. Gradient Boosting applied as a classifier and seven datasets are used in this research. Accuracy, recall, precision, F1-Score, AUC were also implemented to measure classifier performance. Experiments showed that oversampling technic increase accuracy from 2% to 11% for the dataset Mammography, Liver Disorders, Diabetes (Pima Indian), Indian Liver, Habberman, and Immunotherapy. Borderline-SMOTE increases higher accuracy compared to other oversampling method. Surprisingly, Breast Cancer Wisconsin has steady accuracy with or without oversampling. Even though, oversampling good for data imbalanced, the sensibility of oversampling algorithm and the nature of dataset must considered.**

Keywords— Imbalanced dataset, Oversampling, Gradient-Boosting-Classifier

I. INTRODUCTION

The challenge of real-world classification using machine learning is finding an accurate classifier. There were many obstacles faced in order to obtain goodclassifier, one of which is imbalanced dataset problem. Health field datasets in leading repositories are generally in binary class and imbalanced. Imbalance can reduce classification accuracy and cause errors in diagnosis.

In binary class, the dataset is classified as imbalance if the representation of two-class is not approximately equal [1]. The significantly less representative class is called as minority-class, and the other is called as the majority-class. Mostly, the majority class biases toward themselves. A classifier addresses the model predictive to the majority class and overlooks minority class. Various techniques have been discussed for handling the imbalanced data. Roughly, several solutions were proposed. They are a data level approach, algorithm approach, cost sensitivity, and ensemble learning[2]. Practically, the data level approach is dealing with re-sampling such as undersampling, oversampling, or a combination of the two. Resampling is usually done in pre-processing [2].

Moreover, resampling minority class is one of impressive dives. Oversampling is a technique to enlarge sample in minority class until its size as majority class. Increasing sample is done by replication the instance of a minority class or adding new instances by a random subset of the minority class. Even though it enhanced accuracy, it is not significant.

Synthetic Minority Oversampling Technique (SMOTE) is another oversampling technic because it promises better accuracy [1]. SMOTE generates new synthetic data among convex between two instances in the minority class. SMOTE have variants, Borderline-SMOTE1, and Borderline-SMOTE2 that proposed by Han *et al*. [3]. Unlike SMOTE, Borderline-SMOTE1 is only over-sampled object from borderline samples. The object in minority class close to majority class is called borderline samples. This method is efficient for oversampling if a half of the nearest neighbor comes from the majority class [4]. Nevertheless, original SMOTE reveals better performance compared to Tomek-Link oversampling in Random Forest classifier[5].

Adaptive Synthetic (ADASYN), a novel approach motivated by SMOTE has discussed. ADASYN has adaptability shift decision boundary to focus on examples that are difficult to learn. ADASYN generate synthetic dataset based on density distribution. Even though ADASYN also take k nearest neighbors in minority class to consider new data like in SMOTE, ADASYN only generates the number instances based on number of examples in k that belong to majority class[6].

Resampling is only part of technic applied to handling imbalanced. The ensemble is another way to enhance classifier performance. Ensemble learning combines several methods to achieve high accuracy. Boosting is one of the methods in an ensemble that can improve accuracy and reduce its variance when applying it to low-dimensional data. Community researcher discusses many classification algorithms based on boosting like AdaBoost, Stochastic Gradient Boosting, and Gradient Boosting [7]. Gradient boosting is unique because it works based on the weak learner, but it produces robust classifier. Like AdaBoost, Gradient Boosting builds many trees from synthetic dataset sequences but it produces large trees than trees in AdaBoost[8].

Research on resampling has been published in two years like combining support vector machine and weighted-SMOTE [9], using heuristic oversampling method based on K-Means and SMOTE[10] dan combining SMOTE and Random Forests for parkinson disease [11]. The results give high accuracy but not compared with condition before use oversampling.

Only a single approach cannot handle the problem of imbalanced data. The synergy between sampling technique and ensemble has stood out [12]. Therefore, this research proposed combination of data level approach and ensemble learning to achieve high-performance classifier. In the preprocessing step, SMOTE, Borderline-SMOTE, and ADASYN will be used. It was followed by applying Gradient Boosting to construct the classifier. The aim of our approach is to gain high-performance classifier.

II. METHOD

This research aim to compare the impact of oversampling toward the accuracy of gradient-boosting. The workflow of this research is figured in Fig. 1. First, gradient-boosting was used to classify the original dataset (without oversampling). Second, three algorithms (SMOTE, ADASYN, SMOTE Borderline) were proposed to oversampling instances in minority-class. Each algorithm detects minority-class then re-sampling these instances until both minority and majority class were balanced. Synthetic-dataset is combination between majority-class instance and new minority-class instance. These oversampling were done at the pre-processing step. Then, gradient-boosting classifies synthetic-dataset were split into training and testing dataset. The training dataset is 75% and the testing dataset is 25%. Finally, there are three output where GB-SMOTE refers to the result of Gradient Boosting with SMOTE, GB-ADASYN refers to the result of Gradient Boosting with ADASYN, and GB-BSMOTE refers to the result of the Gradient Boosting with Borderline–SMOTE.

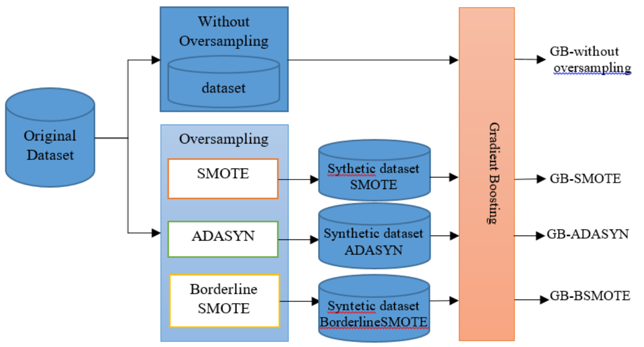


Fig. 1. The workflow of the proposed methodology

A. Dataset

This research used several datasets taken from UCI Learning [13]. Seven datasets are chosen with some considerations such as (i) types of classification (binary), (ii) imbalanced rates, and (iii) size of the dataset. The description of these datasets is shown in Table I, where IR is imbalanced ratio.

TABLE I. DATASETS

No	Name Of Dataset	Instance	Class Instance	IR
1	Liver Disorder	345	Type 1= 145; Type 2= 200	1.4
2	Diabetes(Pima Indian)	768	Not Diabetes=500 Diabetes = 268	1.9
3	Breast Cancer (Winconsin)	699	Benign=458 Malignant=241	1.9
4	Haberman's Survival	306	Survive=225 Died=81	2.7
5	Immunotherapy	90	Type 1=19 Type 2 =71	3.7
6	Indian Liver	583	Type 1=416 Type 2=167	2.49
7	Mammography	961	Benign=516 Malignant=445	1.2

B. Oversampling

Oversampling is one of resampling methods to handle imbalanced data. Community research claims oversampling leverage accuracy. As the name mentioned, oversampling is process to generate new-sample related to the original one by increasing the number of instances. Many ways were proposed by researchers to conduct oversampling. The state of the art of oversampling is Synthetic minority-oversampling technique (SMOTE) proposed by Chawla *et al.* in 2002. One such variance is Borderline-SMOTE. Because the primacy and weakness from SMOTE, ADASYN (Adaptive Synthetic) was proposed by the researcher to create a synthetic dataset.

1) SMOTE (Synthetic Minority-Oversampling Technique)

SMOTE is an oversampling technique that produces synthetic dataset by generating new instances among the line that connect two features. These artificial instances were generated in the following way: (i) taking feature *a* under examination from minority-class and finds k nearest features *b* within the same minority-class; (ii) for each feature of *b* chosen, calculate the difference between feature *a* and *b*, then multiply the difference by a random number between 0 and 1, and add it to feature *b* under examination. These ways generate new instances along convex between two features *a* and *b*. Fig. 2. illustrates how SMOTE work.

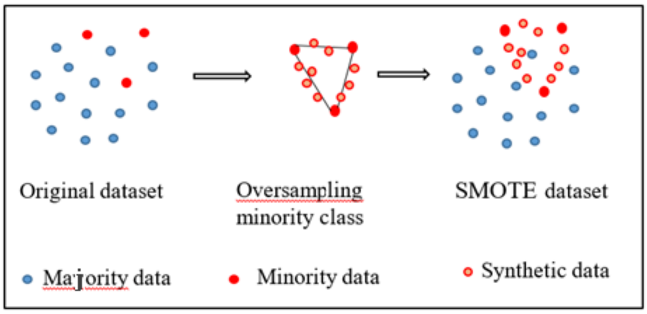


Fig. 2. SMOTE

2) Borderline-SMOTE

Borderline-SMOTE is a variant of SMOTE proposed by Han [3]. The basic concept is how to finds out minority instance in the borderline, generate synthetic instance from them, and added to the original dataset. Steps of Borderline-SMOTE are in the following ways:

(i). Training set is *T*, minority class is *P*, and majority is *M.*

(ii). For every $p_i\,(i=1,2,3,\ldots,pnum)$ in the minority class *P* : finds out m-nearest neighbors from the whole training set *T* (minority and majority).

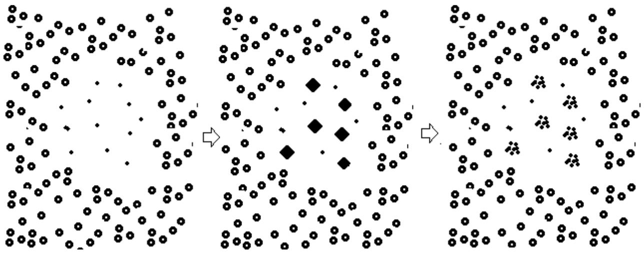
(iii). For each P_i, the *s* nearest neighbors from its *k* nearest neighbors in *P* is selected randomly.

(iv). The system calculates the differences dif_j between P'_i and *s* nearest neighbors from *P* then multiply this difference by a random number r_j between 0 and 1 and the result of multiplication added to P'_i.

New synthetic data is denoted by:

$$Synthetic_j = p'_i + r_j \times dif_j,\ \ j=1,2,\ldots,s$$

Repeat the procedures for each P_i. Fig.3. is an illustration of how works.



(a) (b) (c)

Fig.3. (a) Original dataset; (b) The borderline minority examples (diamond shape). (c) The borderline synthetic minority examples (dots).

3) ADASYN (Adaptive Synthetic)

ADASYN algorithm is an oversampling approach that uses weight distribution for different minority class instances according to their level of difficulty in learning. The steps of ADASYN algorithm are as follow.

(i). Suppose training dataset D with m samples (X_i, Y_i) where i=1..m and X_i are instances in dimensional feature space X and $y_i \in Y = \{1, -1\}$ is the class identity label associated with X_i. Define m_s is the minority class and m_l is the majority class while $m = m_s + m_l$

(ii). Estimate the degree of imbalanced as $d = m_s/m_l$.

(iii). If $d < d_{th}$ (d_{th} is a threshold that tolerated for imbalance ratio) then determine the number of synthetic data instances that need to be generated for the minority class defined by $G = (m_l - m_s) \times \beta$ where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after generation of the synthetic data. β =1 means a fully balanced data set is created after the generalization process.

(iv). For each example $x_i \in$ minority class, find out k nearest neighbors based on the Euclidean distance in n-dimensional space, and calculate the ratio r_i defined as $r_i = \Delta_i / k$, ($i = 1, \ldots, m_s$) where Δ_i is the number of samples in the k nearest neighbors of x_i that belong to the majority class, and $r_i \in [0, 1]$.

(v). Normalize r_i according to :

$$\hat{r_i} = r_i / \sum_{i=1}^{m_s} r_i,$$

so that $\hat{r}_i$ is a density distribution

$$(\sum_i \hat{r_i} = 1)$$

(vi). Calculate the number of synthetic data examples that need to be generated for each minority example X_i with by $g_i = \hat{r_i} \times G$ where G is the total number of synthetic data examples that need to be generated for the minority

(vii). For each minority class data example X_i, generate g_i synthetic data examples.

The fundamental idea of ADASYN algorithm is using a density distribution as a criterion to automatically vote the number of synthetic samples that require to generate for each minority data example.

4) Gradient Boosting

The basic concept of gradient boosting classifier (GBC) is inspired by AdaBoost (Adaptive Boosting). Boosting is a method that adapts weak learner to get a strong learner. In boosting, each new tree corresponds to the modified version of the original data. Boosting is tailored to the sequence of *weak learners* than conducted randomly. In each stage within build trees, the same samples are used but the weight is different in each iteration. At the end of the process, AdaBoost merge the whole tree used weight it's trees.

Here are steps in AdaBoost. Firstly, the training dataset is assigned by equal weight w. For the first, the weight for each record is obtained by dividing it's with total_sampel. Example, if we have eight samples, the weight of each sampel is 1/8. Next, AdaBoost is started by building a very short tree (Stump). The stump is built by considering the best feature, which has a minimum weight. Since the weight is the same for all feature, the first stump consider the first feature, then all stump is built base on other features. The Gini index from all stump is calculated. Therefore, the weight of all node on all stump is evaluated. Then, new weight replaces old weight. In the second stage, AdaBoost builds the new stumps based on an error that the previous stump made. So, this stump did little better than last stumps and little larger. AdaBoost build stumps continuously in this procedure until it fits.

In contrast, Gradient Boosting (GB) starts by making a single leaf, instead of tree or stump. This leaf represent the weight of all the instance. When trying to predict continuous value, the first guess is the average value. Then, GB builds a tree. Like AdaBoost, the next tree is built based on the error made by the previous tree was built. GB tree is usually more giant than a stump. GB algorithm completely defined in [7]. This research use GBC from scikit-learn.org [14] .

C. Evaluation Method

Since accuracy is not appropriate to a skewed dataset, the performance of classification algorithms is pace out by F1-score and AUC (Area Under ROC). These measures consider the value of confusion matrix below.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

F1-score is the average of Precision and Recall, which is the best value at 1 and the worst at 0. The formula for Recall is shown in (1) and Precision in (2), where TP refers to True Positive, FN refers to False Negative, and FP refers to false positive.

$$recall = \frac{TP}{TP+FN} \quad (1)$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

Receiver Operating Characteristics (ROC) curve is figured by the true positive rate on the y-axis and false positive rate on the x-axis. AUC shows the performance of the classifiers. AUC is the arithmetic mean of true positive-rate and true negative-rate. It is also related to the probability that a classifier model will rank a randomly selected positive sample higher than a randomly selected negative sample. The formula for AUC is as shown in (3).

$$TN - Rate = \frac{TN}{TN+FP} \quad (3)$$

III. Result And Discussion

A. Preprocessing

There are two steps of preprocessing in this research, imputation and oversampling. Imputation aim to replace missing value but it is not our concern in this research. Several datasets contain attributes with NAN (null or character). Imputation did in this way: if attributes are null and it is non-binomial then replace it with zero. Another way, if its binomial (1..0) then imputation is done by K-Nearest Neighbor. The clean datasets without empty attributes have resulted by imputation.

This research using the kernel for the oversampling method from Python toolbox [15][16]. Oversampling is done with SMOTE, Borderline SMOTE, and ADASYN. Firstly, each dataset *X* has read then it processed by SMOTE. The results are stored in X_SMOTE. Secondly, re-read and process it with . The results are stored in X_BSMOTE. Finally, re-read dataset and process with Adasyn. The results are stored in X_ADA.

The result of oversampling are balanced dataset. It is shown in Building Gradient Boosting Classifier Preprocessing

Gradient Boosting apply to all balance dataset from the previous process. The result is a Gradient Boosting Classifier (GBC). We analyze GBC performance through comparing those accuracies, recall, precision, and AUC.

. This process show that the number instance after oversampling has an equal size in both classes.

TABLE II. Datasets Before And After Oversampling

No	Name Of Dataset	Class Instance Before Oversampling	Class Instance After Oversampling
1	Liver Disorder	Type 1= 145; Type 2= 200	Type 1=200; Type 2= 200
2	Diabetes(Pima Indian)	Not Diabetes=500 Diabetes = 268	Not Diabetes=500 Diabetes = 500
3	Breast Cancer (Winconsin)	Benign=458 Malignant=241	Benign=458 Malignant=458
4	Haberman's Survival	Survive=225 Died=81	Survive=225 Died=225
5	Immunotherapy	Type 1=19 Type 2 =71	Type 1=71 Type 2 =71
6	Indian Liver	Type 1=416 Type 2=167	Type 1=416 Type 2=416
7	Mammography	Benign=516 Malignant=445	Benign=516 Malignant=516

B. Building Gradient Boosting Classifier Preprocessing

Gradient Boosting apply to all balance dataset from the previous process. The result is a Gradient Boosting Classifier (GBC). We analyze GBC performance through comparing those accuracies, recall, precision, and AUC.

Comparison between SMOTE, Borderline-SMOTE, ADASYN, and those accuracies on each dataset has shows on Fig. 4., Fig. 5., Fig. 6., Fig. 7., Fig. 8., Fig. 9., and Fig. 10.

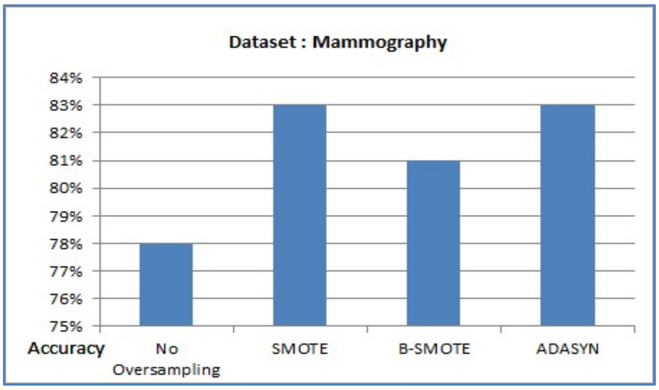


Fig. 4. Mammography

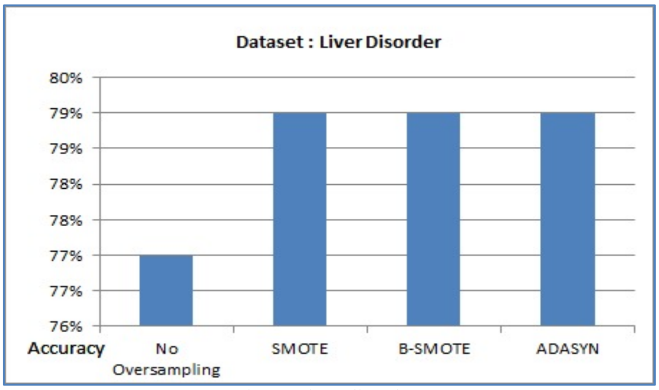


Fig. 5. Liver disorder

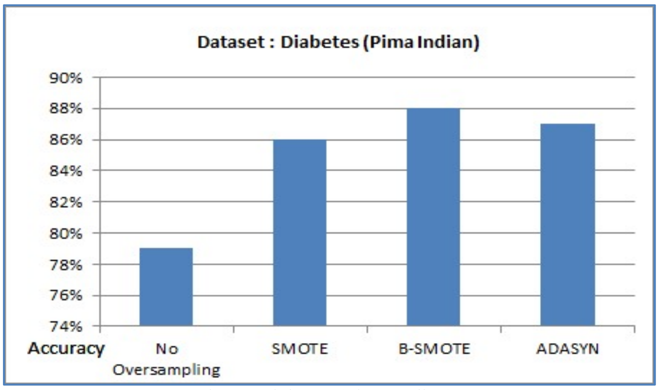


Fig. 6. Diabetes (Pima Indian)

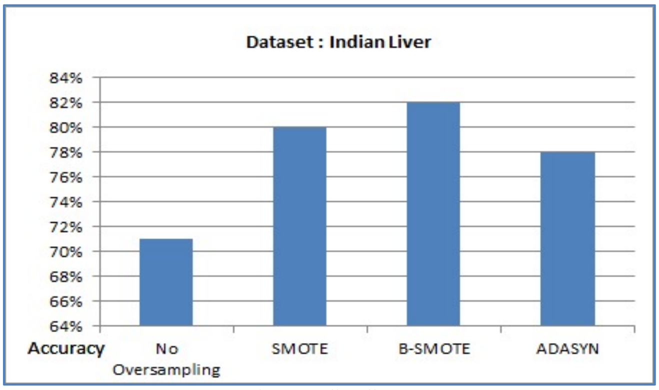


Fig. 7. Indian liver

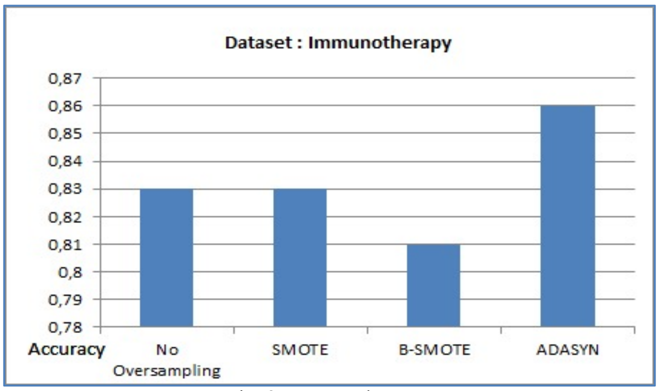


Fig. 8. Immunotherapy

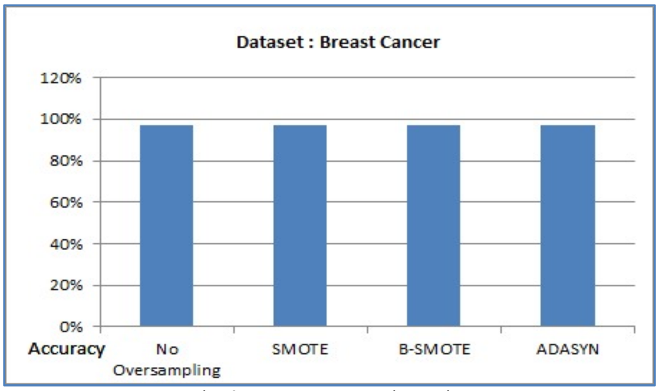


Fig. 9. Breast cancer Wisconsin

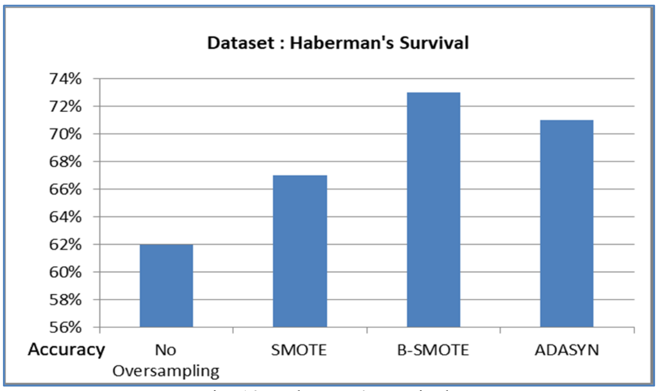


Fig. 10. Haberman's survival

Accuracy of Gradient Boosting Classifier (GBC) with three oversampling process (SMOTE, Borderline-SOTE, and ADASYN) compared with accuracy of GBC without oversampling process. The result of the comparison presented inTable III.

Recall, Precision, F1-Score, and AUC indicate classifier at a reasonable rate. F1-Score shows that GBC is not too over fitting or under fitting. GBC can classify instances in both minority and majority class. Oversampling gives high AUC for all dataset compared to without oversampling. Look at imbalanced ratio (IR), this average is less then 4. The high imbalanced ratio is impressive to observed in another way since it affected to decrease accuracy. Overall, oversize minority samples can boost the classification performance.

TABLE III. ACCURACY, PRECISION, RECALL, AUC-ROC, F1-SCORE

	Dataset	IR	Methods	Accuracy	AUC	Precision	Recall	F1-Score
1	Liver Disorder	1.4	SMOTE	**0.79**	0.878	0.84	0.69	0.76
			ADASYN	**0.79**	0.871	0.82	0.75	0.78
			Borderline SMOTE	**0.79**	0.893	0.81	0.78	0.8
			Without Oversampling	0.77	0.84	0.82	0.66	0.73
2	Diabetes (Pima Indian)	1.9	SMOTE	0.86	0.947	0.88	0.83	0.85
			ADASYN	0.87	0.929	0.91	0.82	0.86
			Borderline SMOTE	**0.88**	0.929	0.91	0.83	0.87
			Without Oversampling	0.79	0.854	0.82	0.88	85
3	Breast Cancer Winconsin	**1.9**	**SMOTE**	**0.97**	0.983	0.99	0.94	0.97
			ADASYN	**0.97**	0.986	0.99	0.94	0.96
			Borderline SMOTE	**0.97**	0.974	0.99	0.94	0.96
			Without Oversampling	**0.97**	0.995	0.97	0.97	0.97
4	Haberman's Survival	2.7	SMOTE	0.67	0.763	0.67	0.64	0.65
			ADASYN	0.71	0.768	0.7	0.69	0.7
			Borderline SMOTE	**0.73**	0.766	0.72	0.71	0.72
			Without Oversampling	0.62	0.65	0.63	0.91	0.74
5	Immunotherapy	3.7	SMOTE	0.83	0.916	0.72	0.93	0.81
			ADASYN	**0.86**	0.948	0.76	0.93	0.84
			Borderline SMOTE	0.81	0.932	0.71	0.86	0.77
			Without Oversampling	0.83	0.658	0.5	0.25	0.33
6	Indian Liver	2.49	SMOTE	0.8	0.892	0.78	0.81	0.8
			ADASYN	0.78	0.883	0.76	0.79	0.78
			Borderline SMOTE	**0.82**	0.882	0.83	0.8	0.82
			Without Oversampling	0.71	0.765	0.71	0.97	0.82
7	Mammography	1.2	**SMOTE**	**0.83**	0.925	0.8	0.87	0.83
			ADASYN	**0.83**	0.911	0.81	0.84	0.82
			Borderline SMOTE	0.81	0.894	0.77	0.85	0.81
			Without Oversampling	0.78	0.874	0.78	0.83	0.8

Fig. 1. and Table III. shows the accuracy for almost all datasets has increased by oversampling. The improvement is from 2% to 11% for the Mammography (accuracy increased by 5% using SMOTE and ADASYN), Liver Disorder (accuracy increased by 2% using thre method), Diabetes (Pima Indian) (accuracy increased by 9% using Borderline-SMOTE), Indian Liver (accuracy increased by 11% usingBorderline-SMOTE), Haberman's Survival (increased by 11% using Borderline-SMOTE), and Immunotherapy (increased by 3% using ADASYN).

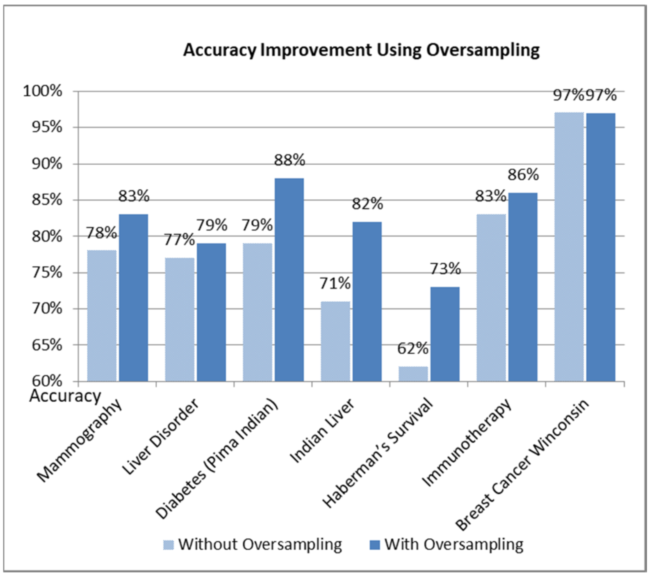


Fig. 11. Comparison accuracy with and without oversampling

Borderline-SMOTE seems increase much accuracy then SMOTE and ADASYN. In other hands, oversampling does not improve accuracy on Breast Cancer Wisconsin.

IV. Conclusion

The Gradient Boosting Classifier is a method based on a decision tree that can classify imbalanced datasets with high accuracy and performance. Combining process with oversampling will increase the accuracy of about 2% to 11%. The results show oversampling increase accuracy around 5%. It means oversampling can still raise up accuracy, although it does not work on dataset of Breast Cancer Wisconsin. It needs investigation on why oversampling not increase accuracy in Breast Cancer Wisconsin dataset.

Solutions for handling the imbalance dataset can be tested with undersampling methods, and another's classifier algorithm. So, it can be known what combination of methods can provide better classification results.

References

[1] Nitesh V. Chawla, K. W. Bowyer, and L. O. Hall, "SMOTE: Synthetic Minority Over-sampling Technique Nitesh," *J. Artif. Intell. Res.*, no. Sept. 28, pp. 321–357, 2002.

[2] H. Ali, M. N. M. Salleh, R. Saedudin, K. Hussain, and M. F. Mushtaq, "Imbalance class problems in data mining: A review," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 3, pp. 1552–1563, 2019.

[3] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," pp. 878–887, 2005.

[4] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline Over-sampling for Imbalanced Data Classification," *Int. J. Knowl. Eng. Soft Data Paradig.*, vol. 3, no. 1, pp. 4–21, 2009.

[5] M. P. Paing, C. Pintavirooj, S. Tungjitkusolmun, S. Choomchuay, and K. Hamamoto, "Comparison of Sampling Methods for Imbalanced Data Classification in Random Forest," *BMEiCON 2018 - 11th Biomed. Eng. Int. Conf.*, pp. 1–5, 2019.

[6] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," no. 3, pp. 1322–1328, 2008.

[7] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," in *Transportation Research Part C: Emerging Technologies*, 2015, vol. 58, pp. 308–324.

[8] R. Blagus and L. Lusa, "Gradient boosting for high-dimensional prediction of rare events," *Comput. Stat. Data Anal.*, vol. 113, pp. 19–37, 2017.

[9] Hartono, O. S. Sitompul, Tulus, and E. B. Nababan, "Biased support vector machine and weighted-SMOTE in handling class imbalance problem," *Int. J. Adv. Intell. Informatics*, vol. 4, no. 1, pp. 21–27, 2018.

[10] G. Douzas, F. Bacao, and F. Last, "Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on K-Means and SMOTE," *Inf. Sci. (Ny).*, vol. 465, pp. 1–20, 2018.

[11] K. Polat, "A Hybrid Approach to Parkinson Disease Classification using speech signal: The combination of SMOTE and Random Forests," *2019 Sci. Meet. Electr. Biomed. Eng. Comput. Sci. EBBT 2019*, pp. 1–3, 2019.

[12] M. Galar, A. Fern, E. Barrenechea, and H. Bustince, "A review of ensembles for the class imbalance problem," *IEEE Trans. Syst. Cybern. C Appl. Rev.*, vol. 42, no. 4, pp. 463–484, 2012.

[13] "UCI Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu/ml/datasets.htm. [Accessed: 01-Oct-2019].

[14] "Gradien Boosting Classifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html. [Accessed: 26-Sep-2019].

[15] "Oversampling." [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html. [Accessed: 25-Sep-2019].

[16] G. Lemaˆ, F. Nogueira, W. S. West, O. Mv, and C. K. Aridas, "Imbalanced-learn : A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," vol. 18, no. 17, pp. 1–5, 2017.