

Oversampling Methods for Handling Imbalance Data in Binary Classification

Theodorus Riston[1], Sandi Nurhibatulloh Suherman[1], Yonnatan Yonnatan[1], Fajar Indrayatna[1], Anindya Apriliyanti Pravitasari[1], Eka Novita Sari[2], and Tutut Herawan[3,4(✉)]

[1] Department of Statistics, Universitas Padjadjaran, Jl. Ir. Soekarno KM. 21, Jatinangor, Sumedang, West Java 45363, Indonesia
{theodorus19001,sandi19002,yonnatan19001}@mail.unpad.ac.id, {fajar.indrayatna,anindya.apriliyanti}@unpad.ac.id

[2] AMCS Research Center, Jalan Griya Taman Asri, Sleman, Yogyakarta, Indonesia
eka@amcs.co

[3] Sekolah Tinggi Pariwisata Ambarrukmo Yogyakarta, Jl. Ringroad Timur No. 52, Bantul, Daerah Istimewa Yogyakarta 55198, Indonesia
tutut@um.edu.my

[4] Institute for Big Data Analytics and Artificial Intelligence, UiTM Shah Alam, 40450 Shah Alam, Selangor Darul Ehsan, Malaysia

Abstract. Data preparation occupies the majority of data science, about 60–80%. The process of data preparation can produce an accurate output of information to be used in decision making. That is why, in the context of data science, it is so critical. However, in reality, data does not always come in a predefined distribution with parameters, and it can even arrive with an imbalance. Imbalanced data generates a lot of problems, especially in classification. This study employs several oversampling methods in machine learning, i.e., Random Oversampling (ROS), Adaptive Synthetic Sampling (ADASYN), Synthetic Minority Over-sampling Technique (SMOTE), and Borderline-SMOTE (B-SMOTE), to handle imbalanced data in binary classification with Naïve Bayes and Support Vector Machine (SVM). The five methods will be run in the same experimental design and discussed in search of the best and most accurate model for the datasets. The evaluation was assessed based on the confusion matrices with precision, recall, and F1-score calculated for comparison. The AUC and ROC curve is also provided to evaluate the performance of each method via figures. The proposed work reveals that SVM with B-SMOTE has better classification performance, especially in datasets with high similarity characteristics between the minority and majority classes.

Keywords: Imbalance Data · Oversampling · SMOTE · Naïve Bayes · SVM

1 Introduction

One of the most common problems in data preparation is imbalanced binary classification. A dataset is considered to be imbalanced if one of the target classes has a much smaller number of instances in comparison to the other classes, hence we name the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14105, pp. 3–23, 2023.
https://doi.org/10.1007/978-3-031-37108-0_1

larger class as the majority class, and the smaller class as the minority class. We need to generate a model that can correctly classify new data that belongs to the minority class, because, in most cases, the minority class is often the intended class to be predicted. Having imbalanced data leads to inaccurate classification results, because imbalanced data leads to a bias towards the majority class during the classification process. One way to handle the imbalanced data is to resample the original dataset, using either under-sampling for the majority class or over-sampling for the minority class until the two classes are almost balanced.

In the under-sampling approach used in [1], several instances of the majority class are randomly removed in order to get a new dataset that is balanced between majority and minority classes. However, this method causes inaccurate results, because it might eliminate important information needed in modeling. The over-sampling approach assigns new instances to the minority class with the aim of balancing the new dataset.

There are several methods used in the over-sampling technique. First, Random Over-Sampling (ROS), which will randomly select the instances of the minority class to be replicated, which results in a balanced class. However, the disadvantage of the ROS method is that it can cause overfitting by randomly replicating instances of the minority class [2]. Chawla *et al*. then proposed the Synthetic Minority Over-Sampling Technique (SMOTE) as another method of handling imbalanced data [3]. There are also many other synthetic over-sampling techniques, such as Borderline-SMOTE [4] and Adaptive Synthetic Sampling (ADASYN) [5]. In this study, the four techniques are employed to handle the imbalance data from binary labels. To discover the most effective approach, several unbalanced open datasets were used in this study. The Support Vector Machine (SVM) and Naive Bayes (NB) Classifier methods are utilized to carry out the classification, since these two well-known methods produce effective results for binary classification. The SVM classifier [6, 7] has gained popularity in various classification tasks due to its improved performance in binary classification scenarios [8]. Given a data set, SVM aims to find a discriminant hyperplane that preserves the optimal edges of boundary instances, called support vectors. Therefore, SVM focuses on increasing the generalizability of the training data. In addition to the SVM, the NB classification is a popular and widely used classification [9]. The Bayesian classifier assigns the most likely class to a given instance, described by its feature vector. Although independence is generally a bad assumption, in practice, NB often competes well with more sophisticated classifiers [10].

There are many over-sampling methods that require research to study and determine which technique is better to use in handling imbalanced data. Therefore, we will test oversampling techniques to overcome several imbalanced datasets, then compare the classification results of each method to determine which method is the best in handling imbalanced data. In evaluating the classification with concentration in the minority class, precision and recall are more effective [11]. In addition, F-value [11] integrating recall and precision is used instead of recall and precision. Its value is large when both recall and precision are large. Moreover, in evaluating the classification, we can use Receiver Operating Characteristic (ROC) Curve, a standard technique that can summarize classifier performance over a range of tradeoffs between True Positive Rate and False Positive Rate [12].

The remainder of the paper is organized as follows. In Sect. 2, more comparable studies are provided and discussed in order to lead to the research question. Section 3 details the methodology, including the tools used to gather tweets and comments, as well as to support the method employed in this research. For Sect. 4, the findings are presented and discussed, and in Sect. 5, we conclude this research and highlight future research.

2 Related Works

Data sampling techniques are used to deal with imbalance issues as they can improve the quality and robustness of the learning algorithm. In its development, there are many techniques to handle imbalanced data, ROS, SMOTE, BSMOTE and ADASYN. However, we need to know which technique works best for binary data so that we can create new high-quality datasets. Jian *et al.* [13] has investigated by comparing two methods of oversampling, Random Oversampling (ROS) and SMOTE, to handle imbalanced data. The research uses 19 datasets from various sources, including engineering, finance, information, and medical sciences. This research uses the SVM as its classification method. The result of this research was that SMOTE is better than ROS, which was concluded by comparing the recall and area under curve (AUC) values of the two methods. Similarly, Pereira and Saraiva [14] performed a comparative analysis on the handling of imbalanced data. SVM is one of the classification methods used in this analysis, and the imbalance data processing methods included was SMOTE and ROS. In this analysis, the ROS method was found to be not as good as SMOTE because the F-Value and AUC values in ROS were smaller than in SMOTE.

On the other hand, Khemakhem *et al.* [15], also conducted research related to handling imbalance data using ROS and SMOTE for classification using SVM. The results of the credit risk assessment for the imbalance data set with this method show that the ROS method is better than the SMOTE method with the AUC measurement. Additionally, Hordri *et al.* [16] conducted research on handling imbalance data on credit card fraud. One of the classification methods used is NB, along with methods of handling imbalanced data such as ROS and SMOTE. The results of the study with a training data ratio of 30:70, concluded that the SMOTE method was better than ROS in Naive Bayes classification when using a precision measurement. However, it is inversely proportional to the use of the F-Value and AUC measurement.

Oreški and Oreški [17] conducted a research on 30 different datasets to compare SVM and Naive Bayes classification methods to handle imbalanced data with SMOTE. The average AUC of various datasets in the research using the SVM classification method before handling imbalance data is 0.6087 and after handling imbalance data increases the AUC to 0.7889. Meanwhile, with NB, there was a decrease in AUC after handling imbalance data, from 0.7928 to 0.7596. The research has shown that in the domain of class imbalance datasets, SMOTE resampling technique has statistically significant positive influence on the performance of SVM classifiers, measured by the AUC measure.

In 2020, Lu *et al.* [18] conducted research related to telecom fraud using ADASYN. The results of this research, handling imbalance data using ADASYN, Recall and F1 values with SVM classification is better than using Naive Bayes classification. Then, Gosain and Sardana [19] conducted follow-up research comparing SMOTE, BSMOTE and ADASYN. On 6 datasets, the SMOTE method outperforms BSMOTE and ADASYN using the Precision and F-Value measurement scales with Naive Bayes classification. Meanwhile, with SVM classification, SMOTE method is better than BSMOTE and ADASYN evaluated using the Precision, F-Value, and AUC.

Table 1. Related Work of Binary Classification with Imbalanced Data

References	Data	Model	Evaluation	
			Type	%
Jian *et al.* [13]	19 datasets from various sources, including engineering, finance, and medical sciences	SVM-ROS	Recall (Average)	54.73
		SVM-SMOTE	Recall (Average)	55.45
		SVM-ROS	AUC	89.69
		SVM-SMOTE	AUC	91.18
Pereira and Saraiva [14]	Electricity consumption data from normal consumers and electricity thieves	SVM-No Balance	AUC	56.14
		SVM-ROS	AUC	59.69
		SVM-SMOTE	AUC	63.42
		SVM-No Balance	F-Value	18.42
		SVM-ROS	F-Value	20.56
		SVM-SMOTE	F-Value	22.81
Khemakhem *et al.* [15]	Tunisian credit risk assessment	SVM-No Balance	AUC	87.9
		SVM-ROS	AUC	97.4
		SVM-SMOTE	AUC	96.1
Hordri *et al.* [16]	Credit card fraud detection	NB-ROS	Precision	93.27
		NB-SMOTE	Precision	93.28
		NB-ROS	F-Value	88.81
		NB-SMOTE	F-Value	88.02
		NB-ROS	AUC	91.07
		NB-SMOTE	AUC	90.38
Oreški and Oreški [17]	30 different datasets from KEEL repository	SVM-No Balance	AUC (Average)	60.87
		SVM-SMOTE	AUC (Average)	78.89
		NB-No Balance	AUC (Average)	79.28

(*continued*)

Table 1. (*continued*)

References	Data	Model	Evaluation	
			Type	%
		NB-SMOTE	AUC (Average)	75.96
Lu *et al.* [18]	Telecom fraud Identification	SVM-ADASYN	Recall	66.30
		NB-ADASYN	Recall	57.50
		SVM-ADASYN	F-Value	66.70
		NB-ADASYN	F-Value	60.80
Gosain and Sardana [19]	Pima India Diabetes, Breast Cancer Wisconsin, Statlog (Heart) Dataset, Ionosphere, Spam Base Dataset, and German	NB	Precision (Average)	74.33
		NB-SMOTE	Precision (Average)	84.78
		NB-BSMOTE	Precision (Average)	79.07
		NB-ADASYN	Precision (Average)	81.48
		NB	F-Value (Average)	76.10
		NB-SMOTE	F-Value (Average)	80.95
		NB-BSMOTE	F-Value (Average)	77.73
		NB-ADASYN	F-Value (Average)	76.55
		NB	AUC (Average)	90.17
		NB-SMOTE	AUC (Average)	89.17
		NB-BSMOTE	AUC (Average)	89.75
		NB-ADASYN	AUC (Average)	87.98
		SVM	Precision (Average)	85.30
		SVM-SMOTE	Precision (Average)	91.70
		SVM-BSMOTE	Precision (Average)	88.62
		SVM-ADASYN	Precision (Average)	89.82
		SVM	F-Value (Average)	79.18

(*continued*)

Table 1. (*continued*)

References	Data	Model	Evaluation	
			Type	%
		SVM-SMOTE	F-Value (Average)	81.65
		SVM-BSMOTE	F-Value (Average)	79.95
		SVM-ADASYN	F-Value (Average)	78.03
		SVM	AUC (Average)	83.85
		SVM-SMOTE	AUC (Average)	83.92
		SVM-BSMOTE	AUC (Average)	83.48
		SVM-ADASYN	AUC (Average)	81.83
P. Pereira *et al.* [29]	8 datasets from a major automotive assembly company related with a distinct type of steering wheel angle sensor	None	AUC (Average)	57.24
		GC	AUC (Average)	67.31
		SMOTE	AUC (Average)	60.12
		RU	AUC (Average)	60.98
		TL	AUC (Average)	57.32

In 2021, Pereira *et al.* [29] conducted research comparing several balancing techniques with some Machine Learning methods and has found that while Gaussian Copula is the best oversampling technique combined with Random Forest (RF) base learner, SMOTE has shown competitive results on individual basis on several datasets, and is aggregately better results compared to no use of balancing techniques at all, evaluated with AUC. Table 1 summarizes the state-of-the-art techniques presented by many researchers for dealing with imbalance data using SVM and Naive Bayes classification.

3 Material and Proposed Method

3.1 Data

The data used for this research is public data taken from UCI and kaggle sources, taking 8 datasets that have different numbers of samples and characteristics (See Table 2). Also, heterogeneous data means that it contains mixed data types; categorical and numeric features.

3.2 Data Preprocessing

3.2.1 Data Standardization

Normalize all continuous feature values to the same scale in order to avoid having different scales of feature values. All feature values are normalized to the range [0, 1] using the Min-Max Scalar standardization procedure. The equation for the Min-Max

Table 2. Summary Description of Datasets

Source	Number of Samples	Number of Features	Labels
Attrition [27]	1,470	34	{Yes, No}
Flare 1 [25]	323	12	{0, 1}
Flare 2 [25]	1,066	12	{0, 1}
Haberman [25]	306	3	{1, 2}
Mammography [26]	111,183	6	{−1, 1}
Telcochurn [28]	7,043	19	{Yes, No}
Wine-Red [25]	1,590	11	{1, 2}
Wine-White [25]	4,898	11	{1, 2}

Scalar is as follows:

$$u_i^{'} = \frac{u_i - \min(u)}{\max(u) - \min(u)}, \tag{1}$$

where u_i is original data and $u_i^{'}$ is standardized data with $i = 1, \ldots, n$. Moreover $min(u)$ and $max(u)$ are the feature's minimum and maximum values, respectively [20].

3.2.2 Data Split

Data split stage which data is prepared before it is processed. At this stage, the qualities are chosen first. The attributes with 80% missing value and redundant information were not used in the study. The data is translated to a numerical format without losing its real meaning, so the software could read the data.

3.3 Methodology for Binary Classification

3.3.1 Naïve Bayes

Naïve Bayes (NB) classification is a supervised learning approach that employs Bayes' theorem as a framework for categorizing observations into one of a predefined set of classes based on predictor variables' information. NB classifiers estimate the conditional probability that an observation belongs to a certain class based on the values of the predictor variables, assuming that the predictor variables are class-conditionally independent, and hence ignore predictor variable covariance.

The NB classifier is a subset of Bayesian networks. The class variable is at the root of a two-level structure used by NB Classifiers, with all characteristics in the next level. From one class feature to another, directed arcs link them. This structure expresses a strong independence assumption, implying that all characteristics are independent of the class value. Despite the fact that the conditional independence requirement is impractical, the NB classifier works well. According to Bayes theorm, the posterior probability that

an observation Y has class index k given the values of predictor variables are described as:

$$\hat{P}\big(Y = k|X_1, \ldots, X_p\big) = \frac{\pi(Y = k)\prod_{j=1}^{P} P(X_j|Y = k)}{\sum_{k=1}^{K} \pi(Y = k)\prod_{i=1}^{P} P(X_j|Y = k)}. \tag{2}$$

The prior probability in Eq. (2) that the class index is k is $P(Y = k)$. The process calculates a distinct Bayes distribution for each class for each predictor X and observations are allocated to the class with the highest posterior probability given the predictor values. For binary classification the class index k is set to 2.

3.3.2 Support Vector Machine

The Support Vector Machine (SVM) is a supervised learning approach for regression and clarifying issues that uses linear combinations of kernel basis functions to create predictions. SVM employs the structural risk minimization (SRM) concept, which aims to reduce the upper limit of generalization error rather than empirical error. The kernel is used to translate the input data into a high-dimensional space where the data will be segregated linearly in classification issues. The kernel is used to discover the best hyperplane that minimizes the distance between two data sets in the regression issue.

In N-dimensional space, SVM is used to discover the optimum hyperplane. The optimal hyperplane on the Support Vector Machine, which divides the two data sets included in the N-dimensional space, is depicted in Fig. 1. Support Vectors are vectors that are close to the ideal hyperplane (SVs). The correctness of the SVM model is determined by the kernel parameters used, since these parameters have a substantial influence on the kernel method's performance. The number of parameters is determined by the distance between several data sets.

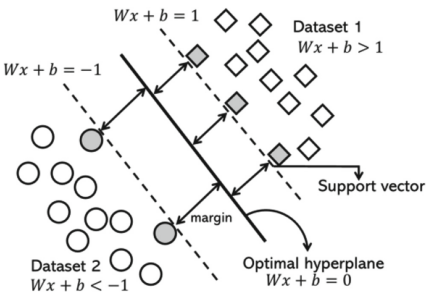

Fig. 1. SVM Hyperplane [7]

Find that $\{x_i, y_i\}_{i=1,\ldots,n}$, input vector $x_i \in \Re^d$ denoted as $y_i \in \Re$. Objective model SVM to find linear function as follows:

$$f(x) = \langle w, \phi_i(x)\rangle + b, \tag{3}$$

with w and b are weight vector and constants that have been estimated from data, while ϕ_i is a non-linear function. So, regression problem can defined as follows:

$$R(C) = \frac{C}{n}\sum_{i=1}^{n} L_\varepsilon(f(x_i), y_i) + \frac{1}{2}\|w\|^2, \tag{4}$$

which $L_\varepsilon(f(x_i), y_i)$ denoted as loss function intensive as in Eq. (5).

$$L_\varepsilon(f(x_i), y_i) = \begin{cases} |f(x) - y| - \varepsilon; for |f(x) - y| \geq \varepsilon, \\ 0; for\ everything\ else. \end{cases} \tag{5}$$

Variable slack ξ_i and ξ_i^* barrier is done by

$$\begin{aligned} &Minimize: \\ &R\left(w, \xi_i^*\right) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), \\ &Subject\ to: \\ &\begin{cases} y_i - \langle w, x_i\rangle - b \leq \varepsilon + \xi_i^*, \\ \langle w, x_i\rangle + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases} \end{aligned} \tag{6}$$

where C is a regularized constant greater than 0 to perform balancing between the training error and flatness model which C represents penalty for prediction error that greater than ε. ξ_i *and* ξ_i^* slack extension from actual values to boundary values of ε. Optimization from Eq. (6) can be converted to Lagrange multipliers quadratic problem:

$$f(x) = \sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right)K(x_i, x) + b, \tag{7}$$

where α_i dan α_i^* in Eq. (7) are *Lagrange multipliers*:

$$\sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right) = 0, \tag{8}$$

which satisfies $0 \leq \alpha_i \leq C; 0 \leq \alpha_i^* \leq C; i = 1, \ldots, n$.

The $K()$ in Eq. (7) is a Keren function with value that contain product from 2 vector x_i and x_j inside feature $\phi(x_i)$ and $\phi\left(x_j\right)$, such $K\left(x_i, x_j\right) = \phi(x_i).\phi\left(x_j\right)$.

3.4 Methodology for Handling Imbalanced Data

3.4.1 Random Over-Sampling

The Random Over-Sampling approach works by repeating certain cases and randomly adding some observations in the minority class. The main disadvantage of ROS is that it might cause overfitting due to information replications [21] and increased processing time [22]. The majority and minority classes are represented by the red and purple bars in Fig. 2. The Random Oversampling procedure, as shown in the diagram, selects samples from the minority class and duplicates them to ensure that the classes are evenly represented.

Random Over-sampling depends on a direct application of the distributional hypothesis and is based on the premise that related documents (such as minority-class texts) may be predicted to include semantically comparable terms.

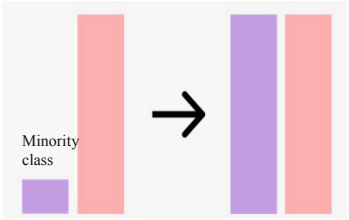

Fig. 2. Data Resampling Using Random Over-Sampling [24]

3.4.2 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Techniques is an oversampling approach that works by generating false observations between neighboring minority observations. The benefit of SMOTE over the ROS technique is that it eliminates the problem of machine learning classifiers overfitting [22]. By finding a straight line between existent locations, SMOTE generates random synthetic points. The feature vector and its closest neighbor in a minority class are then identified, and the additional data points are multiplied by a random value between 0 and 1. These synthetic data points are added to the training data collection that will be used to train the models. However, two key limitations of this method are that it is unsuccessful with high-dimensional data, and while generating synthetic data, SMOTE does not hunt for neighboring instances from other classes, which might lead to more noisy data owing to class overlapping. The SMOTE method is shown in Fig. 3, with pink squares representing the minority (positive) class and blue circles indicating the majority class. SMOTE developed synthetic positive samples in the minority class, which are shown as orange triangles between pink squares. The original positive samples and manufactured samples together make up the minority group. This will bring the majority and minority classes closer together.

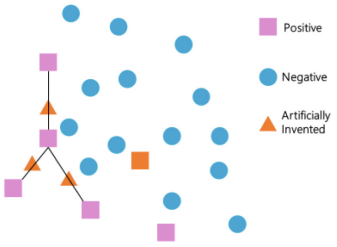

Fig. 3. SMOTE Synthetic Data [3]

To find SMOTE algorithm, choose k and N to represent the number of closest neighbors and synthetic observations, respectively. The algorithm is given as follows;

1. Let x_i, $i = 1, 2, \ldots, n_s$, denote the observations belonging to the minority class and let A denote the set of all x_i, such that $A \ni x_i$. For every x_i;

2. Calculate the Euclidean distance between x_i and all other elements of A to obtain the k-nearest neighbors of x_i;
3. Let S_{ik} denote the set of the k-nearest neighbors of x_i;
4. Randomly sample N synthetic observations denoted x_{ij}, $(j = 1, 2, \ldots, N)$ drom S_{ik} with replacement;
5. Let λ denote a number in the range [0,1]. For a given $x_{ij,}$ draw a λ uniformly and then generate a synthetic observations by the formula $x_k = x_i + \lambda(x_i + x_{ij})$;
6. Execute Step 5 for every x_{ij};
7. Stop Algorithm.

3.4.3 Borderline-SMOTE

One of the SMOTE variations is Borderline-SMOTE. Rather than just constructing false observations between neighboring observations in the minority, it focuses on samples closer to the decision border [4]. Borderline-SMOTE like standard SMOTE oversamples observations in the minority class and its immediate neighbors. BorderlineSMOTE has two different parameter variations: borderline-1 and borderline-2. Based on the number of closest neighbors each sample has from the majority class samples samples in the minority class are initially categorized into three sets: danger noise and safe.

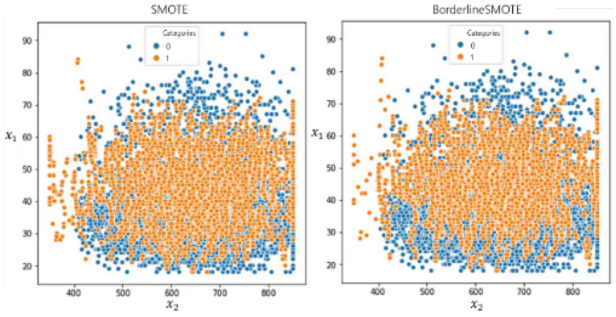

Fig. 4. B-SMOTE Case data [4]

The difference between oversampling data with SMOTE and Borderline-SMOTE is seen in the Fig. 4. Although it may seem to be comparable, there are distinctions in how the synthetic data is generated. B-SMOTE divided the data into 3 classes; "*D*", "*S*", and "*N*", which are danger, safe, and noise, respectively. Then it loops the algorithm below until all the data points are classified;

a Iterates over all instances in minority class $x_i \, \varepsilon \, S_{min}$, using K-NN algorithm on x_i and gets k neighbors of x_i;
b Calculates the number of sample belonging to majority class in the k nearest neighbors of x_i;
c Sample assignation as follows: $\frac{k}{2} \leq |S_i \cap S_{max}| < k$;
d x_i will be assigned to D;
e The sample x_i is to be judged as a noice instance and its divided into N and S set.

3.4.4 Adaptive Synthetic (ADASYN)

Another synthetic data oversampling strategy is the Adaptive Synthetic (ADASYN) sampling approach, which was initially proposed in [23], with the goal of achieving data class balance by adaptively producing data samples in the minority class depending on their distribution with a given balance level. There are two goals in this algorithm. For starters, the algorithm can figure out how many samples are needed for each sample in the minority class. Second, ADASYN pushes machine learning algorithms to recognize or learn the difficult-to-learn examples. The ADASYN method improves classification performance by minimizing the bias induced in the original unbalanced data sets, as seen in Fig. 5. Furthermore, when the equilibrium level is improved by ADASYN, it shows a trend for error reduction.

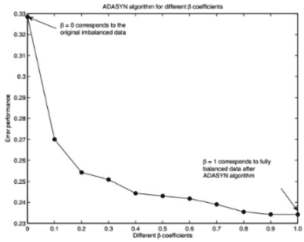

Fig. 5. ADASYN Algorithm for Data Balancing [23]

Choose k and β which denote the number of nearest neighbors and the desired levek if class balance after generating the synthetic data, then;

a. Let n_i denote the number of observation of the majority class and let n_s denote the number of observations of the minority class. Calculate $G = n_l - n_s.\beta$;
b. Let x_i, $i = 1, \ldots, n_s$, denote the observations belonging to the minority class and let A denote the set of all xi, such that A $\ni x_i$. For every x_i;
c. Calculate the Euclidean distance between x_i and all other elements of A to obtain the k-nearest neighbors of x_i;
d. Let S_{ik} denote the set of the k-nearest neighbors of x_i;
e. Define Δ_i as the number of the observations in the k-nearest neighbors region of x_i that belong to the majority class. Calculate the ratio r_i defines as $r_i = \frac{\Delta_i}{k}, i = 1, .., n_s$;
f. Normalize r_i according to $\hat{r} = \frac{r_i}{\sum r_i}$, so that probability equal 1;
g. Calculate $g_i = r_i.G$ which is the number of synthetic observations that need to be generated for each x_i;
h. Randomly sample g_i synthetic observations denoted x_{ij} from S_{ik} with replacement;
i. Let λ denote a number in the range [0,1]. For a given x_{ij}, generate a synthetic observations according to $x_k = x_i + \lambda\left(x_i - x_{ij}\right)$, where λ is uniformly drawn for each x_k;
j Stop Algorithm.

3.4.5 Receiver Operator Characteristics (ROC) and Precision-Recall (PR) Curve

A classifier identifies instances as positive or negative in a binary decision issue. A confusion matrix, also known as a contingency table, may be used to illustrate the

classifier's choice. There are four categories in the confusion matrix: True positives (TP) are positive instances that have been appropriately categorized as such. False positives (FP) are negative cases that have been mistakenly categorized as positive. True negatives (TN) are negatives that have been appropriately identified as such. Finally, false negatives (FN) are positive cases that have been mislabeled as negative. ROC and PR curves are often used to assess a machine learning algorithm's performance on a given dataset. There are a specified number of good and negative instances in each dataset. We demonstrate that the ROC and PR worlds have a strong link.

In ROC space, the aim is to be in the upper-left-hand corner, and the ROC curves on the left of Fig. 6 seem to be quite near to ideal. The aim in PR space is to be in the upper-right-hand corner, and the PR right curves illustrate that there is still a lot of potential for growth.

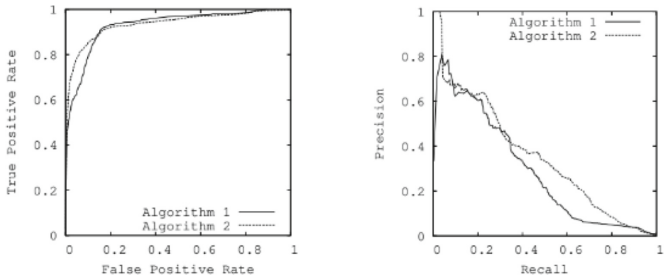

Fig. 6. Comparison of ROC and PR [12]

In this study, the F1-score and accuracy are employed to assess system performance. Precision and recall are required to compute F1-Score. Precision and recall may be determined as shown in Table 3 below.

Table 3. Confusion Matrix for Binary Classed Problems.

		Actual Class	
		1	0
Prediction	1	*TP*	*FP*
	0	*FN*	*TN*

The following Equation are the formula for Overall Accuracy (OA), precision, recall, F-measure, respectively

$$OA = \frac{TP + TN}{TP + FN + FP + TN}, \tag{9}$$

$$Precision = \frac{TP}{TP + FN}, \tag{10}$$

$$Recall = \frac{TP}{TP + FN}, \tag{11}$$

$$F - Measure = \frac{\left(1 + \theta^2\right).Recall.Precision}{\theta^2.Recall + Precision}, \tag{12}$$

where θ is a coefficient to adjust the importance relative of precision and recall.

4 Results and Discussion

4.1 Results

In our experiments, five performance measures are used; precision, recall, F-value, and the area under curve for both ROC and Precision-Recall curves. All of the calculations were made and concluded using R software. The measures were used to evaluate performance of four over-sampling techniques; Random Over Sampling (ROS), Adaptive Synthetic (ADASYN), Synthetic Minority Oversampling Technique (SMOTE), and Borderline-Synthetic Minority Oversampling Technique (B-SMOTE). The value of θ used for F-value is set to 1 and the value of k in all oversampling techniques are set to 5. The classifiers used as classifiers in the experiment are NB and SVM. Eight datasets were used from UCI Machine Learning Database and Kaggle as defined in Table 4 below. Prior to classifying and performing oversampling on the dataset, Principal Component Analysis (PCA) were used to transform the categorical data into numeric-float data in order for it to be oversampled using ADASYN, SMOTE, and B-SMOTE. PCA reduces the number of variables/dimensions of a dataset, while preserving as much information as possible. A percentage split of "80:20" were then used for testing and validation of data. Eighty percent (80%) of the data is for training and twenty percent (20%) is for testing. For the Flare 1 and Flare 2 datasets, we choose the class label "0" as minority and regard the others (>0) as class label "1" being the majority, as we only study two-class problem in this paper.

Table 4. Description of Datasets Used in the Experiments.

Name	Instances	Attributes	Positive	Negative	%Minority
Attrition	1470	35	1233	237	16.12
Flare 1	323	10	265	58	17.96
Flare 2	1066	10	865	201	18.86
Haberman	306	3	225	81	26.47
Mammography	11183	6	10923	260	2.32
Telcochurn	7043	20	5174	1869	26.54
Wine-Red	1599	12	1382	217	13.57
Wine-White	4898	12	3838	1060	21.64

Upon choosing the dataset for this experiment, we adopted an imbalance threshold of 40% while considering them, which states if the number of minority instances is less than 40% of the total instances in the dataset, it is considered as an imbalanced dataset, and therefore chosen for our experiment and evaluation. Table 4 shows the description of the datasets and that all of them fulfills the 40% threshold. Tables 5, 6, and 7 shows the results of actual Precision, Recall, and F-value, on the datasets using both SVM and NB for classification, and by four different methods of oversampling (ROS, ADASYN, SMOTE, and B-SMOTE) to tackle the imbalance on the data. Table 8 and 9 shows the result of the area under curve for the Receiver Operating Characteristic (ROC) Curve and the Precision-Recall (PR) Curve using the same methods of classification and oversampling to tackle the imbalanced datasets. Individually, for B-SMOTE, three stars (***) means that it has the highest value compared to other methods and the actual data without oversampling; two stars (**) means it has the highest value compared to other methods but not the actual data without oversampling; one star (*) means it is nor the lowest or the highest value; and no stars () means it is the lowest value. As for the mean of each method, **bold** denotes a significantly higher or lower value than actual mean, with an α of 5%, tested with two dependent samples t-test in Rstudio.

Figures 7, 8, 9 and 10 shows the curves for the ROC Curve and the PR Curve using SVM and NB before and after applying B-SMOTE to handle imbalanced data.

Table 5. Results in Terms of Precision Measure in the Experiments on Datasets.

Precision (SVM)						Precision (NB)					
Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE	Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE
Attrition	0.8493	1.0000	0.6534	0.6803	0.8000*	Attrition	0.8511	1.0000	0.5906	0.6069	0.6065*
Flare 1	0.9608	1.0000	1.0000	0.9804	1.0000***	Flare 1	0.9167	1.0000	0.7541	0.8214	0.8364*
Flare 2	0.9062	0.9620	0.9620	0.9105	0.9755***	Flare 2	0.8930	0.6866	0.6866	0.7321	0.7171**
Haberman	0.8000	0.6792	0.6944	0.6833	0.7381***	Haberman	0.8036	0.6061	0.5000	0.6557	0.6923**
Mammography	0.9887	0.9059	0.8873	0.9270	0.9949***	Mammography	0.9934	0.8349	0.7915	0.8875	0.9141**
Telcochurn	0.8213	0.7763	0.8468	0.8185	0.8307*	Telcochurn	0.9102	0.8036	0.7648	0.8324	0.7966*
Wine-Red	0.8987	0.9303	0.9077	0.8968	0.9237*	Wine-Red	0.9380	0.7719	0.8403	0.8419	0.8387*
Wine-White	0.8326	0.8681	0.8681	0.8636	0.8601*	Wine-White	0.9175	0.7595	0.7727	0.7877	0.7987*
Mean	0.8822	0.8902	0.8525	0.8376	0.8904	**Mean**	0.9029	0.8078	**0.7126**	**0.7670**	**0.7751**

From Table 5, Borderline-SMOTE (B-SMOTE) has shown to be the most effective method for SVM classifier, because it has the highest point of precision (number of positive class predictions that actually belong to the positive class) in 50% of the datasets, and scored higher points than the actual method without oversampling on 37.5% of the datasets. It can be concluded that B-SMOTE is a good method of oversampling for imbalance data that are classifying true positives using an SVM model. For the NB classifier however, B-SMOTE performs excellent on 40% of the datasets, but still couldn't beat the actual precision when oversampling is not used, but still the best method if oversampling is used.

However, the performance of B-SMOTE while being evaluated using Recall on Table 6 is contradictive. When SVM is used, B-SMOTE shows better results only for two out of eight datasets. And when NB classifier is used, B-SMOTE perform better

Table 6. Results in Terms of Recall Measure in the Experiments on Datasets.

Recall (SVM)						Recall (NB)					
Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE	Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE
Attrition	1.0000	1.0000	0.6777	0.7562	0.6833*	Attrition	0.9677	1.0000	0.7273	0.7273	0.7000
Flare 1	0.9800	0.9787	1.0000	0.9615	1.0000***	Flare 1	0.8800	0.9787	0.8846	0.8846	0.9200***
Flare 2	0.9721	0.8352	0.8362	0.9719	0.8983*	Flare 2	0.9330	0.8187	0.8187	0.9213	0.8305*
Haberman	0.9167	0.7826	0.5814	0.9318	0.6889*	Haberman	0.9375	0.8696	0.7674	0.9091	0.8000*
Mammography	0.9986	0.9240	0.9461	0.9623	0.9726**	Mammography	0.9616	0.8841	0.7471	0.8395	0.7775*
Telcochurn	0.9267	0.7017	0.8032	0.8513	0.8070*	Telcochurn	0.8456	0.7042	0.7160	0.7483	0.7034
Wine-Red	0.9821	0.7937	0.8520	0.8309	0.8364*	Wine-Red	0.8643	0.7098	0.7220	0.7243	0.7564**
Wine-White	0.9710	0.7700	0.7332	0.7902	0.7575*	Wine-White	0.7335	0.6307	0.6295	0.6308	0.6291
Mean	0.9684	**0.8482**	**0.8037**	**0.8820**	**0.8305**	**Mean**	0.8904	0.8245	**0.7516**	0.7982	**0.7646**

at two out of eight datasets, and perform worst at three out of eight datasets. By these results, we conclude that B-SMOTE is not the best to measure Recall (number of positive class predictions made out of all positive examples in the dataset) when using SVM classifier, and is the worst method when NB classifier is used. And by the F-measure from Table 7, B-SMOTE shows that it is not the best method either when using SVM or NB classification.

Table 7. Results in Terms of F-value Measure in the Experiments on Datasets.

F-value (SVM)						F-value (NB)					
Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE	Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE
Attrition	0.9185	1.0000	0.6653	0.7162	0.7371*	Attrition	0.9057	1.0000	0.6519	0.6617	0.6499
Flare 1	0.9703	0.9892	1.0000	0.9709	1.0000**	Flare 1	0.8980	0.9892	0.8142	0.8519	0.8762*
Flare 2	0.9380	0.8941	0.8941	0.9402	0.9353*	Flare 2	0.9126	0.7469	0.7469	0.8159	0.7696*
Haberman	0.8544	0.7273	0.6329	0.7885	0.7126*	Haberman	0.8654	0.7143	0.6055	0.7619	0.7423*
Mammography	0.9936	0.9148	0.9158	0.9443	0.9836*	Mammography	0.9772	0.8588	0.7687	0.8628	0.8403*
Telcochurn	0.8401	0.7371	0.8244	0.8346	0.8187*	Telcochurn	0.8912	0.6266	0.7396	0.7882	0.7471*
Wine-Red	0.9386	0.8566	0.8790	0.8626	0.8779*	Wine-Red	0.8996	0.7395	0.7767	0.7787	0.7954**
Wine-White	0.8965	0.8162	0.7949	0.8252	0.8056*	Wine-White	0.8152	0.6892	0.6938	0.7006	0.7038**
Mean	0.9188	0.8669	0.8258	0.8478	0.8589	**Mean**	0.8956	0.7956	**0.7247**	**0.7777**	**0.7656**

The presence of data balancing does not appear to have any significant effect on the classification results when comparing the average precision and F-value of the SVM classifier. While there are several significant values for recall in the SVM method and precision, recall, and F-value in NB, However, those metric values are less than the actual imbalanced data; hence, this is not a worthwhile outcome.

When measuring using curves like ROC or PR, we need to take a look at the value of area under curve (AUC). The Receiver Operator Characteristic (ROC) plots the True Positive Rate (TPR) against False Positive Rate (FPR) to measure the ability of a classifier to distinguish between classes, and the higher the AUC (the closer it is to one) the better the performance at distinguishing between positive and negative classes. The PR Curve summarizes the trade-off between the TPR and Positive Predictive Value (PPV) using different probability threshold, and the higher the AUC (the closer it is to one) the better.

Table 8. Results Score on Area Under Curve for ROC in the Experiments on Datasets.

Area Under ROC Curve (SVM)						Area Under ROC Curve (NB)					
Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE	Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE
Attrition	0.4252	0.5000	0.4908	0.4721	0.5360***	Attrition	0.4422	0.5000	0.4427	0.4504	0.4619*
Flare 1	0.4923	0.5047	0.5000	0.5051	0.5000*	Flare 1	0.5154	0.5047	0.4579	0.4798	0.4755*
Flare 2	0.4695	0.5347	0.5347	0.4820	0.5206*	Flare 2	0.4812	0.4494	0.4494	0.4311	0.4588*
Haberman	0.4426	0.4611	0.5380	0.3961	0.5174*	Haberman	0.4344	0.3889	0.3750	0.3896	0.4593***
Mammography	0.4951	0.4951	0.4833	0.4902	0.5056***	Mammography	0.5156	0.4855	0.5142	0.5140	0.5375***
Telcochurn	0.4901	0.5232	0.5133	0.4882	0.5077*	Telcochurn	0.4298	0.5531	0.5164	0.5297	0.5314*
Wine-Red	0.4875	0.5374	0.5156	0.5186	0.5242*	Wine-Red	0.4938	0.5205	0.5356	0.5354	0.5251*
Wine-White	0.4357	0.5280	0.5402	0.5228	0.5309*	Wine-White	0.5776	0.5420	0.5479	0.5534	0.5550**
Mean	0.4673	**0.5105**	**0.5145**	0.4844	**0.5178**	**Mean**	0.4863	0.4930	0.4799	0.4854	0.5006

Table 9. Results Score on Area Under Precision-Recall in the Experiments on Datasets.

Area Under Precision-Recall Curve (SVM)						Area Under Precision-Recall Curve (NB)					
Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE	Name	Actual	ROS	ADASYN	SMOTE	B-SMOTE
Attrition	0.4270	0.5000	0.4937	0.4807	0.5248***	Attrition	0.4397	0.5000	0.4607	0.4657	0.4738*
Flare 1	0.4927	0.5031	0.5000	0.5036	0.5000*	Flare 1	0.5146	0.5031	0.4713	0.4857	0.4832*
Flare 2	0.4670	0.5246	0.5246	0.4872	0.5145*	Flare 2	0.4796	0.4642	0.4642	0.4511	0.4710**
Haberman	0.4450	0.4728	0.5256	0.4248	0.5123*	Haberman	0.4378	0.4238	0.4177	0.4204	0.4713***
Mammography	0.4905	0.4966	0.4884	0.4931	0.5039***	Mammography	0.5295	0.4900	0.5099	0.5098	0.5260**
Telcochurn	0.4511	0.5158	0.5094	0.4911	0.5055*	Telcochurn	0.4861	0.5362	0.5116	0.5224	0.5225*
Wine-Red	0.4917	0.5261	0.5108	0.5130	0.5169*	Wine-Red	0.4958	0.5143	0.5248	0.5246	0.5176*
Wine-White	0.4401	0.5193	0.5282	0.5163	0.5218*	Wine-White	0.5731	0.5290	0.5336	0.5382	0.5387**
Mean	0.4631	**0.5073**	**0.5101**	**0.4887**	**0.5125**	**Mean**	0.4945	0.4951	0.4867	0.4897	0.5005

In Table 8 and Table 9, we can see that, in general, data balancing on SVM has a significant effect. Different from the metric results in Tables 5, 6 and 7, the AUC indicates that balanced data has better values than imbalanced data. NB has the same results, although the difference in value is not statistically significant. When viewed individually, we conclude that B-SMOTE is the best-performing method to produce higher AUC scores when both the NB and SVM classification techniques are used.

Figures 7 and 8 shows the results of ROC and PR curves for Attrition dataset using BSMOTE as a method of oversampling to handle imbalanced data. These curves show an increase in the areas under the curve, while also decreasing the elbow angles of the curves. We can see before BSMOTE, the ROC curves for Attrition dataset were in quite steep and sharp elbow shapes, and after BSMOTE they became duller angles. This meant that BSMOTE has increased Sensitivity, FPR, Precision, and Recall in a more consistent manner also, decreasing the rate of error. While AUC is increased over several instances of observation and methods, further research is needed because an AUC of 0.5 or close to 0.5 is quite common for a balanced dataset.

Lastly, Fig. 10 show the results of ROC and PR curves for Mammography dataset using BSMOTE as method of oversampling to handle imbalanced data. These curves show an increase in both the AUC and also a decrease in the angles of the curve. Same as the results for Attrition dataset, this meant that BSMOTE has successfully increased

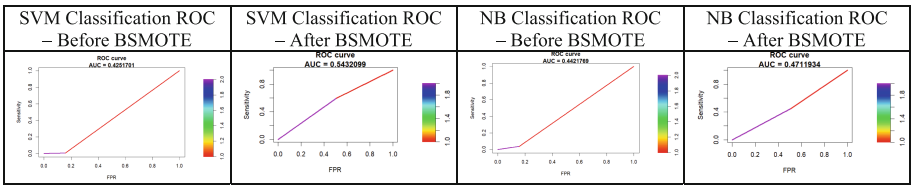

Fig. 7. Results of ROC curve in the experiment of Employee Attrition (Attrition) Dataset using BSMOTE as method of handling imbalanced dataset.

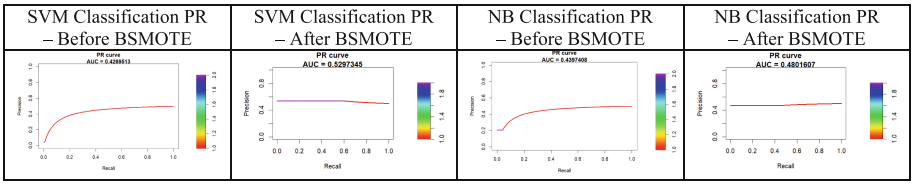

Fig. 8. Results of PR curve in the experiment of Employee Attrition (Attrition) Dataset using BSMOTE as method of handling imbalanced dataset.

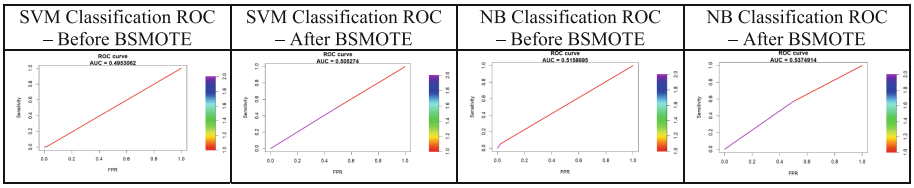

Fig. 9. Results of ROC curve in the experiment of Mammography Dataset using BSMOTE as method of handling imbalanced dataset.

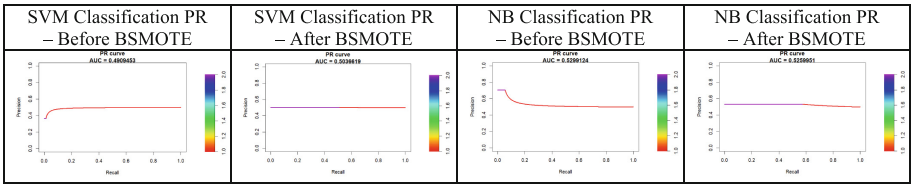

Fig. 10. Results of PR curve in the experiment of Mammography Dataset using BSMOTE as method of handling imbalanced dataset.

these statistics, while maintaining, or even increasing the consistency of these curves, decreasing the rate of error.

4.2 Discussion

The results show that, in general, balancing data does not always result in a good classification, but when you examine the findings from each of the balancing techniques utilized separately, B-smote clearly demonstrates its class. Even though B-SMOTE might not

be the best at each and every measure of evaluation, it has shown great performance in the evaluations. It even performed to be the best method in a few datasets. This is due to the nature of the Borderline-SMOTE method.

According to Han *et al.* [4], B-SMOTE is a method that expands the original distribution of the dataset inwards from the borders. B-SMOTE first see the real distribution of the datasets and concludes the minority, then it calculates which of the data points are borderline minority data points, and it calculates and makes new data points by pointing inwards from these borderline minority data points. Therefore, if the distribution is definitive, then B-SMOTE will only add inside these borders and doesn't increase the number of noise or the majority class. This makes B-SMOTE a more precise method of oversampling because it does not affect the majority or the other classes, and it might even increase the precision, if the dataset supports it. However, if we look at the F-value and the recall, B-SMOTE performed the worst on some of the datasets. This is due to the distribution of the datasets. These datasets are not synthetic or randomized/generated datasets. The borders of each class in these datasets are blurry, and sometimes it might even mix with other classes. By nature, they have no set or definitive shapes for B-SMOTE to use in order to determine the borders. This makes B-SMOTE harder to predict and make points that does not intersect with other classes, making the data even more blurry, and reducing the possibility of the model correctly predicting the class and increasing the number of misclassified points and false negatives, reducing the score of recall. In short, it can be said that Borderline-SMOTE is a precise and good method of oversampling to tackle imbalanced dataset, if the dataset has a good shape, or at least a definitive border between the majority class and the minority class, in order to minimize the number of misclassified synthetic data points later in the calculations.'

5 Conclusion and Future Research

Imbalanced Data has been a problem for a long time, and has recently got more attention, especially from data scientists and data miners. There are many techniques one can use to tackle this problem. However, some of the traditional techniques are still unsatisfactory. Borderline-SMOTE presents a new modern technique that might be used to handle the class imbalance problem. The experiments show that the performance of B-SMOTE, evaluated by precision, recall, and F-value are better than ROS, ADASYN, and SMOTE at some datasets, when either SVM or NB classifiers are applied. This is due to the fact that B-SMOTE carefully samples the dataset and expands it inside the borders of the class for it to decrease the misclassification of the data. We can conclude in individual examination, that it does increase the precision, recall, f-value, and area under curves of ROC and PR, for some of the datasets, and it may improve prediction on those datasets. However, the experiments also shows that while it may be more precise when used on correct datasets, it may increase misclassification when use on datasets with blurry borders or intersecting classes. In handling imbalanced data with no set shapes or borders, there needs to be a new method of classification, that may increase the accuracy and area under curve, in a way that the B-SMOTE cannot do, due to the fact that the datasets are mixed.

Future research might also use B-SMOTE and modify the parameters in order to increase the performance by reducing the radius of the borders, the distance between

synthetic data, etc. B-SMOTE might not be the best at handling imbalanced data, but it has shown great performance and may be one of the better alternatives to tackle an imbalanced dataset.

Acknowledgement. The authors thanks to the Department of Statistics and Research Center for Artificial Intelligence and Big Data, Universitas Padjadjaran which supports this research. The work of Tutut Herawan and Eka Novita Sari is supported by AMCS Research Center.

References

1. Bach, M., Werner, A., Żywiec, J., Pluskiewicz, W.: The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis. Inf. Sci. **384**, 174–190 (2017)
2. Bekkar, M., Alitouche, D., Akrouf, T., Alitouche, T.A.: Imbalance data learning approaches review. Data Min. Knowl. **3**(4), 15–33 (2013)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority oversampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
4. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
5. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks, pp. 1322–1328 (2008)
6. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (2000). https://doi.org/10.1007/978-1-4757-3264-1
7. Schlkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge (2001)
8. Begg, R., Palaniswami, M., Owen, B.: Support vector machines for automated gait classification. IEEE Trans. Biomed. Eng. **52**(5), 828–838 (2005)
9. Kotsiantis, S.B., Pintelas, P.E.: Mixture of expert agents for handling imbalanced data sets. Ann. Math. Comput. Teleinformat. **1**(1), 46–55 (2003)
10. Rish, I.: An empirical study of the Naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence (IJCAI), pp. 41–46 (2001)
11. Buckland, M., Gey, F.: The relationship between recall and precision. J. Am. Soc. Inf. Sci. **45**(1), 12–19 (1994)
12. Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. **30**(6), 1145–1159 (1997)
13. Jian, C., Gao, J., Ao, Y.: A new sampling method for classifying imbalanced data based on support vector machine ensemble. Neurocomputing **193**, 115–122 (2016)
14. Pereira, J., Saraiva, F.: A comparative analysis of unbalanced data handling techniques for machine learning algorithms to electricity theft detection. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE Press (2020)
15. Khemakhem, S., Said, F.B., Boujelbene, Y.: Credit risk assessment for unbalanced datasets based on data mining, artificial neural network and support vector machines. J. Model. Manag. **13**(4), 932–951 (2018)
16. Hordri, N.F., Yuhaniz, S.S., Azmi, N.F.M., Shamsuddin, S.M.: Handling class imbalance in credit card fraud using resampling methods. Int. J. Adv. Comput. Sci. Appl. **9**(11), 390–396 (2018)

17. Oreški, G., Oreški, S.: An experimental comparison of classification algorithm performances for highly imbalanced datasets. In: 25th Central European Conference on Information and Intelligent Systems, Faculty of Organization and Informatics Varazdin, pp. 1–4 (2014)
18. Lu, C., Lin, S., Liu, X., Shi, H.: Telecom fraud identification based on ADASYN and random forest. In: 2020 5th International Conference on Computer and Communication Systems (ICCCS), pp. 447–452. IEEE Press (2020)
19. Gosain, A., Sardana, S.: Handling class imbalance problem using oversampling techniques: a review. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 79–85. IEEE Press (2017)
20. Azeem, M., Usman, M., Fong, A.C.M.: A churn prediction model for prepaid customers in telecom using fuzzy classifiers. Telecommun. Syst. **66**(4), 603–614 (2017). https://doi.org/10.1007/s11235-017-0310-7
21. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newslett. **6**(1), 20–29 (2004)
22. Hartati, E.P., Bijaksana, M.A., et al.: Handling imbalance data in churn prediction using combined SMOTE and RUS with bagging method. J. Phys. Conf. Ser. **971**(1), 012007 (2018)
23. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (IEEE 2008), pp. 1322–1328 (2008)
24. Lahera, G.: Unbalanced datasets what to do about them medium (2018). https://medium.com/strands-tech-corner/unbalanced-datasetswhat-to-do-144e0552d9cd
25. UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets.php. Accessed 2 June 2022
26. Mammography Dataset. OpenML. https://www.openml.org/search?type=data&sort=runs&id=310&status=active. Accessed 2 June 2022
27. IBM HR Analytics Employee Attrition & Performance. Kaggle. https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset. Accessed 2 June 2022
28. Telco customer churn: IBM dataset. Kaggle. https://www.kaggle.com/datasets/yeanzc/telco-customer-churn-ibm-dataset. Accessed 22 June 2022
29. Pereira, P.J., Pereira, A., Cortez, P., Pilastri, A.: A comparison of machine learning methods for extremely unbalanced industrial quality data. In: Marreiros, G., Melo, F.S., Lau, N., Lopes Cardoso, H., Reis, L.P. (eds.) EPIA 2021. LNCS (LNAI), vol. 12981, pp. 561–572. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86230-5_44