Songklanakarin J. Sci. Technol.
39 (5), 565-576, Sep - Oct. 2017

http://www.sjst.psu.ac.th

Original Article

Adaptive neighbor synthetic minority oversampling technique under 1NN outcast handling

Wacharasak Siriseriwan* and Krung Sinapiromsaran

Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Pathum Wan, Bangkok, 10300 Thailand

Received: 25 May 2015; Revised: 13 May 2016; Accepted: 25 July 2016

Abstract

SMOTE is an effective oversampling technique for a class imbalance problem due to its simplicity and relatively high recall value. One drawback of SMOTE is a requirement of the number of nearest neighbors as a key parameter to synthesize instances. This paper introduces a new adaptive algorithm called *Adaptive neighbor Synthetic Minority Oversampling Technique* (ANS) to dynamically adapt the number of neighbors needed for oversampling around different minority regions. This technique also defines a minority outcast as a minority instance having no minority class neighbors. Minority outcasts are neglected by most oversampling techniques but instead, an additional outcast handling method is proposed for the performance improvement via a 1-nearest neighbor model. Based on our experiments in UCI and PROMISE datasets, generated datasets from this technique have improved the accuracy performance of a classification, and the improvement can be verified statistically by the Wilcoxon signed-rank test.

Keywords: class imbalance problem, oversampling, SMOTE, adaptive neighbors approach, minority outcast

1. Introduction

Class imbalance problem is a problem in classification dealing with an imbalanced dataset, the dataset whose amount of instances in one target class is far less than ones in another class. As the class with lesser instances is a target class (positive class), instances in the target class are called either minority instances or positive instances. This problem usually appears in practice and is discovered in various situations such as diagnosis of rare medical conditions (Kousarrizi *et al.*, 2012). The dataset in this problem has a significant characteristic; instances in a positive class are in the minority. Since most classifier algorithms aim to maximize the accuracy performance of the classification, the positive class appears to these algorithms as a less significant class and the accuracy on predicting positive instances, which is a real objective of the problem, would be neglected. It requires extra treatments to maximize the accuracy on predicting positive instances exclusively.

Many researches have introduced effective and practical strategies (He & Garcia, 2009) for improving the prediction rate on a positive class. An approach widely used and studied is a sampling technique during the data-preprocessing process on an imbalance dataset. It transforms the imbalanced dataset into a well-balanced distributed dataset later used to train the classifier. This approach is favorable due to its portability as a researcher is not restricted to any specific classifiers. The simplest idea in this approach is to duplicate existing positive instances until equaling the number of negatives. However, it forces a classifier to learn very specific instances not their general properties and often leads to an overfitting issue. Therefore, the idea of synthesizing positive instances surrounding positive instances is suggested and more widely applied.

This paper concentrates on improving and modifying the so called *'Synthetic Minority Oversampling Technique'*

* Corresponding author.
Email address: wacharasak.s@gmail.com;
wacharasak.s@sci.kmutnb.ac.th;

or SMOTE (Chawla *et al.*, 2002) to overcome some of its drawbacks. One of them is the criterion for choosing a value of parameter K. Researchers normally have a difficult time identifying the appropriate parameter K for a particular dataset. Varying this K for various datasets sometimes does not give the desired result. Most related publications about SMOTE including the original SMOTE paper suggest using K as 5 based on their repeated experiments. However, this number may yield an unsatisfying result for some datasets as shown in the Figure 1.

From the Figure 1, if synthetic instances are generated between the positive instance p and its positive neighbor q, then more positive synthetic instances are generated in the negative region. This circumstance usually happens when the group of positive instances is sparse. A robust idea is to provide different numbers of neighbors for each positive instance according to its density. This would vary a possible location of synthetic instances generated inside the dense area of positives and avoid generating synthetic instances in the sparse area of positive instances.

Moreover, this paper also handles how to deal with a positive instance surrounded by negative instances in order to utilize every positive instance for the accuracy improvement. This positive instance is identified as a minority outcast. SMOTE uses this single positive instance with its positive nearest neighbor to synthesize more positive instances. Figure 1 shows the outcast p and its neighbor q. These two positive instances create a synthetic point s_1. It shows that s_1 is created inside the negative region. Trying to connect this outcast to a group of positive instances misleads a classifier to learn a spurious characteristic.

The next section will explain several oversampling techniques and describe the motivation of this work. Then, the proposed technique is introduced in Section 3. The empirical experiments on UCI and PROMISE datasets and the analysis of our results are shown in Section 4. Finally, the conclusions are drawn in Section 5.

2. Background

Among sampling techniques for class imbalance problems, one significant technique that is widely used and referred is SMOTE (Chawla *et al.*, 2002). It is an oversampling technique that assumes the existence of similarities between positive instances and generates synthetic instances according to these similarities. In SMOTE, it starts with finding K-positive nearest neighbors of each positive instance p, then randomly selecting one of them as np to form a line segment. Along this line segment, a synthetic positive instance p' computed from $p' = p + \text{gap} \times (np - p)$ where $\text{gap} \in [0, 1]$ is added into the dataset. The process continues on other positive instances and repeats until the number of positive instances and negative instances are nearly equal. Caused by these synthesized positive instances, a decision region created during the classification process becomes denser and more expanded. This effect leads some tree-based

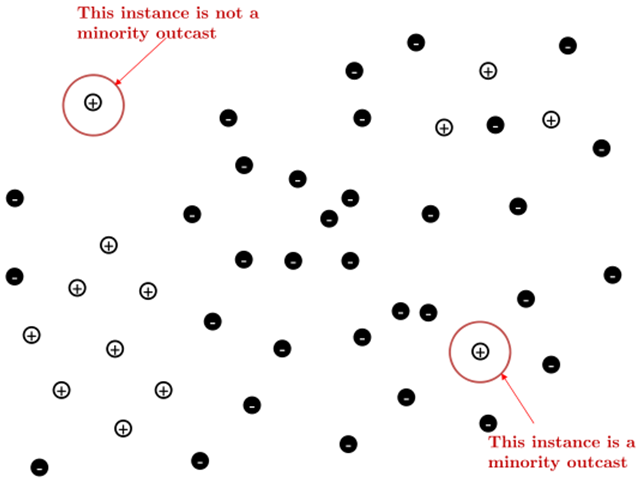

Figure 1. Illustration on the definition of minority outcast.

classifiers to recognize more instances as positive. Japkowicz (2000) shows that SMOTE gives a higher true positive rate which eventually leads to the superior recall value than other techniques. However, it also causes many negative instances to be misclassified. This increases a false positive error and decreases the accuracy of overall classification.

Despite several defective traits of SMOTE, there are many variances of SMOTE providing alternatives of generating synthetic positive instances in order to either further improve the recall value or balance precision and recall Some significant SMOTE variances are introduced here.

'*Adaptive Synthetic Sampling Approach for Imbalanced Learning*" (ADASYN) was introduced by He (2008). ADASYN tries to generate more synthetic instances on the region with less positive instances than one with more positive instances to increase the recognition of positive. This algorithm uses the number of negative neighbors in C nearest neighbors of each positive instance to form a distribution function. The distribution function determines how many synthetic instances are generated from that positive instance. Their paper claimed that the recall value is improved from SMOTE. However, ADASYN further expands the region of positive instances leading to the increasing value of recall and higher false positive rate. The increasing false positive rate may cause lower F-measure and accuracy values which could be critical for some class imbalance problems.

Another adaptation of SMOTE, Borderline-SMOTE (Han *et al.*, 2003) uses the number of negative neighbors in a different approach from ADASYN. This technique names a positive instance, whose C nearest neighbors are all negative as "NOISE" and exclude it from generating instances since it is inside the negative region. It also defines a positive instance whose number of positive neighbors is high as "SAFE" and also exclude it from generating instances since its surrounding region is guaranteed to be positive. Therefore, Borderline-SMOTE uses only "DANGER", positive

instances, which are neither "SAFE" nor "NOISE", in the synthetic generation process. Borderline-SMOTE provides a less recall value than SMOTE but it has a higher precision value in return.

Borderline-SMOTE inspires another variation of SMOTE, '*Safe-Level Synthetic Minority Oversampling Technique*' or Safe-Level SMOTE (Bunkhumpornpat *et al.*, 2009). Safe-Level SMOTE defines a new parameter safe-level for each positive instance. The parameter is calculated from the number of positive instances in its C nearest neighbors. It is used to determine which positive instances should be used to create synthetic instances and the interval of a possible location of a synthetic instance. The ratio of safe-level of paired positive instances alters and shortens the interval that forms a line segment. For positive instances with nonzero safe-level values, a synthetic process is performed with this altered interval. As a result, it avoids placing a synthetic instance close to the positive instance with a lower safe-level. With this concept, Safe-level SMOTE indirectly acknowledges the existence of negative instances located around each positive instance which is different from SMOTE. This approach leads to higher precision and F-measure values. However, since some positive instances with all negative neighbors are excluded, it causes a relatively lower recall value compared to SMOTE or ADASYN.

Another oversampling technique, density-based synthetic minority over-sampling technique or DBSMOTE (Bunkhumpornpat *et al.*, 2012), suggests clustering positive instances into groups using density-based clustering and provides a technique to generate synthetic instances inside each cluster despite that each cluster might not form the convex set. This algorithm applies DBSCAN (Ester, 1996) to cluster positive instances. Then, a synthetic instance is created on the shortest path from each positive instance to the pseudo-centroid of its cluster. This leads the resulting synthetic dataset to be dense around the core of a group of the original positive instances. DBSMOTE is reported to have a good performance on F-measure and AUC values in some experimental settings.

These oversampling techniques incorporate some concepts from SMOTE in order to further improve its performance on a class imbalance problem. However, there are still major drawbacks of SMOTE which are not addressed. These motivated us to tackle drawbacks concepts in this paper which are addressed next.

3. Motivation

Oversampling techniques, especially ones which adopted a synthetic generating idea from SMOTE, are widely accepted as the effective approach for a class imbalance problem. However, our study finds some flaws in the original concept which are not covered by other related works. These flaws are brought up in this section.

The first one is how appropriate is the parameter K, the number of positive neighbors that can be chosen to pair and synthesize new positive instances. Depending with the location of K^{th} nearest neighbor of each positive instance, synthetic instances generated from the instance are no further away from it than that neighbor. The problem arises when the region of these positive instances is too sparse so their nearest neighbors are mostly surrounded by negative instances. After generating synthetic instances from these neighbors, the resulting balanced dataset could contain the conflicting region with original negative instances and newly added positive instances. On the other hand, if the region of positive instances is very dense, the low K could limit the number of neighbors. The distribution of synthetic instances may not be spread uniformly since they have to stay on the line between each positive instance. Our idea is to identify the density of each area of positive instances and use it to choose the appropriated value of K separately.

The second one is utilizing a minority outcast. These outcasts can be found when original positive instances are distributed too sparse inside the vast amount of negative instances. They are considered as one type of the noise of positive instances. However, the positive noises which are outliers of the entire dataset are not minority outcasts, since there is no conflict if synthetic instances are generated between these noises and other positive instances, see Figure 2.

These minority outcasts are excluded from generating synthetic instances by various synthetic oversampling techniques to avoid the case that synthetic instances are generated inside a negative region. Due to the relatively low number of positive instances in an imbalanced dataset, an original positive instance may be crucial on classifying a positive class. Therefore, there should be a treatment for these minority outcasts in order to improve the accuracy performance of predicting positive instances. Two remedies of these two flaws are combined into our proposed method in the next section.

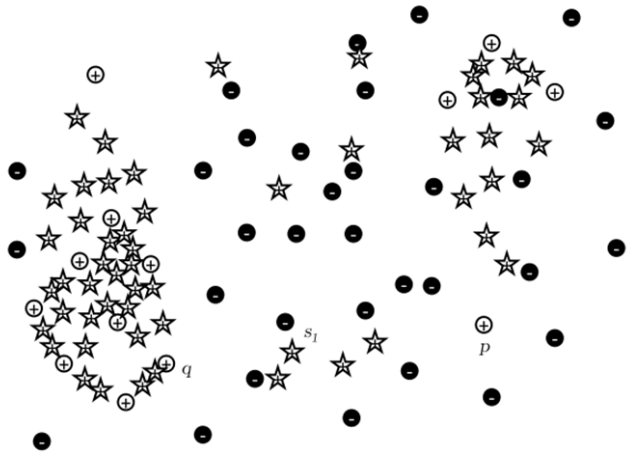

Figure 2. Synthetic dataset generated by SMOTE algorithm with K=5.

4. Proposed Method

Adaptive neighbor Synthetic Minority Oversampling TEchnique under 1NN outcast handling or ANS is introduced based on two objectives. The first objective is to override the decision on a single value of K from a user using K_i for each positive instance p_i. K_i is the number of possible positive neighbors that is chosen to pair up with a positive instance p_i in order to create synthetic instances along the line segment of between that pair. Chawla (2002) uses the value of K as 5 for SMOTE. This setting is also used by other SMOTE-related oversampling techniques as their default preference. However, it could not be verified whether a single value of K as 5 is actually optimal without performing multiple experiments. To avoid this preference or exhaustedly repeated experimental run, adaptive neighbor process in ANS will automatically configures the value of K_i for a positive instance p_i based on the density of surrounding positive instances. The second objective is to exclusively deal with minority outcasts in order to preserve their significance without generating synthetic instance. The minority outcast handling process in ANS will give an alternative way which provides an acceptable accuracy performance.

The first step of ANS is to exclude minority outcasts from the dataset. To identify which positive instance is a minority outcast, C-nearest neighbor algorithm is performed on each positive instance. The positive instances which all of C nearest neighbors are negative are identified as minority outcasts and separated from the set of positive instances while other positive instances are used for generating synthetic instances via SMOTE algorithm. Then, following the process of SMOTE, each positive instance requires at least one positive neighbor to form a line segment that generates a synthetic instance. The maximum distance value between pairs of two closest positive neighbors is chosen as the radius in order to guarantee at least one neighbor for each positive instance p_i. Therefore, every positive instance contains at least one positive nearest neighbor under this radius. After the radius is found, the number of positive nearest neighbor of each positive instance p_i under this radius is counted and defined as K_i for each p_i. Then, SMOTE is performed and each positive instance contains different number of nearest neighbors it can generate synthetic instances with. This process and the minority outcast extracting process are shown in algorithm 1. With different number of K_i, the location of each synthetic instance becomes more scatter inside the dense area of original positive instances and does not form the skeleton-like line as appearing in SMOTE. Moreover, fewer synthetic instances are generated among the region of negative instances since the original instances which locate away from others will not try to generate synthetic instances with neighbors that place too far from them. The ideal resulting synthetic dataset is shown in the Figure 3.

Minority outcasts which are identified and removed in the first step are utilized in the minority outcast handling process first introduced by Siriseriwan and Sinapiromsaran (2016). The additional procedure is to include minority outcasts into a set of negative instances as a sub-dataset and build a 1-nearest neighbor model. This additional 1-nearest neighbor model will provides a small positive region around each outcast. If any unknown instances fall into this region, they will be classified as positive regardless of the result from the trained classifier. Only the 1-nearest neighbor is chosen here due to the definition of outcast which stated that all of its neighbors are negative. Therefore, if two or more neighbors are used for classifying an unknown instance, these additional neighbors will be negative. Then, the unknown instance will never be classified as positive as intended. The detail of this process is shown in algorithm 2.

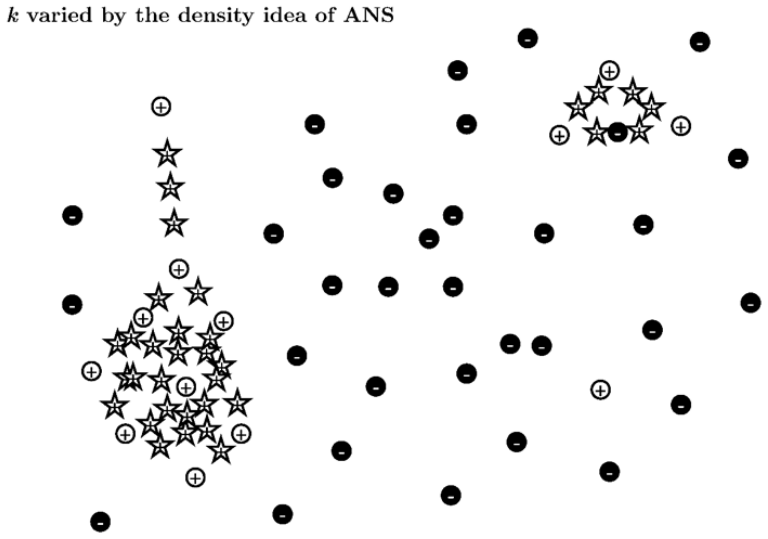

Figure 3. Synthetic dataset generated by ANS.

Algorithm 1: Adaptive neighbor SMOTE algorithm
Data: Numerical-attribute binary class dataset D containing a set of positive instances P and a set of negative instances N and the value of C.
Result: Nearly balanced dataset which is a combination of D and a set of synthetic instances S, a set of minority outcast OC.

1. Initialization $t = 1$;
2. $(Pused, OC, E)$ = OutcastExtraction(D, P, C)
3. Define $\varepsilon = \max E$
4. For p_i in $Pused$ do
5. Let $Np_i = \{p_j \text{ in } Pused | \, d(p_i, p_j) \leq \varepsilon\}$
6. End for
7. While t < the roundup value of $|N|/|Pused|$ do
8. For p_i $Pused$ do
9. Randomly select np_i from Np_i
10. gap = a random number between 0 and 1
11. $p' = p_i + gap \times (np_i - p_i)$
12. Add p' into S.
13. End for
14. $t = t + 1$
15. End while.
16. **Function OutcastExtraction(D, P, C)**
17. Define C_max as the roundup value of $0.25*|D|$
18. Perform C_max-nearest neighbor of P in D
19. Mark the first positive nearest neighbor of p_i in P as fp_i
20. Determine the number of negative neighbors of p_i with smaller radius than $d(fp_i, p_i)$ as out_border_i
21. For $c = 1, \ldots, C\_max$
22. For p_i in P do
23. If $out\_border_i \geq c$ then p_i is the outcast for this c. End for.
24. Count the number of outcast in this c as n_oc_c
25. If $|n\_oc_c - n\_oc_{c-1}| = 0$, set $C = c$. End for.
26. Let $OC = \{p_i \text{ in } P | \, out\_border_i \geq C\}$
27. Let $Pused = \{p_i \text{ in } P | \, out\_border_i < C\}$
28. Keep the distance between p_i in $Pused$ and its nearest positive neighbor as ε_i in E_{Pused}
29. Return $\{Pused, OC, E_{Pused}\}$

Algorithm 2: Minority outcast handling algorithm
Data: Numerical-attribute binary class dataset D containing a set of negative instances N, a set of minority outcast instances OC and a set of unknown instances U.
Result: Vector of assigned class CL $(cl_1, cl_2, \ldots, cl_i)$ of U.

1. OutcastHandling(D, N, OC, U)
2. For u_i in U do
3. Calculate the distance from u_i to every instance in a set N and OC
4. Let u^* be argmin $\{d(u_i, x) | \, x \text{ in } N \text{ or } x \text{ in } OC\}$
5. If u^* in OC then
6. $cl_i = +$
7. End if
8. Otherwise, $cl_i = -$
9. Return CL

Algorithm 1 or Adaptive neighbor process is operated during the synthetic instance generation process to pass them to the classifier. Algorithm 2 or minority outcast handling process is a post-classification process. The result of classification will be concluded after classified data pass through both classifier and minority outcast handling process.

Parameter C in the outcast extraction process

Since the value of K for each positive instance is automatically assigned, ANS does not need to set K. However, C-nearest neighbor process which determines which positive instance is a minority outcast requires the configuration of parameter C. To identify a positive instance p as an outcast, all of C nearest neighbors of p has to be negative. It is obvious to see that the larger the value of C is, the less number of outcasts is. We expect that there should be a number of outcasts to improve the performance but the number of outcasts should not be too much since the minority outcast handling model may overwhelm the actual classification model. Our work decides that the value of C should be the lowest C that the number of outcasts is steady. To reduce time, this C can be identified with only single nearest neighbor run on a training set as shown in algorithm 1. Since the value of C depends on each training dataset not the user preference, ANS becomes a parameter-free algorithm.

To confirm the effectiveness of this method, we conducted experimental evaluation over real-world datasets with five standard classifiers. The next section provides the details of our experimental settings and the result analysis.

5. Experimental Setting and Result Analysis

To compare the effectiveness of ANS and other oversampling techniques, the experiments are conducted on nine datasets from UCI repository (Lichman, 2013); ecoli, glass, letter recognition, haberman, LandSat(satimage), segmentation, yeast, optdigits, and vehicle, and five datasets from PROMISE repository (Menzies, 2012); cm1, jm1, kc1, kc2, and pc1. All datasets are pre-processed into a binary classification problem by selecting one class as the intended positive class and the rest as the negative class. The number of instances, attributes, positive instances and percentage of positive instances are presented in Table 1.

There are five classifiers using in the experiment; decision tree (C4.5), naïve Bayes classifier, multilayer perceptron, support vector machine with the linear square kernel and K-nearest neighbor (with $K = 3$). These classifiers are well-known classification algorithms which are often included in various data mining tools. The performance is evaluated through 5-fold cross-validation scheme in R programming environment. The average F-measure and area under ROC curve (AUC) values of each method, dataset, and classifier from validation are presented in Table 2 and 3. For ANS, both ANS1, which stands for *Adaptive neighbor SMOTE without*

Table 1. Description of datasets used in the experiments.

Name	Instances	Attributes	Positive instances	% of positive instances
cm1	498	21	49	10.91
Ecoli	336	8	20	5.95
Glass	214	11	76	35.51
Haberman	306	4	81	26.47
jm1	10,880	21	2,103	23.96
kc1	2,109	21	326	18.28
kc2	522	21	107	25.78
Letter (H)	20,000	17	734	3.67
Optdigits (0)	5,620	64	554	10.94
pc1	1,109	21	77	7.46
Satimage (4)	6,435	37	626	9.73
Segment (WIN)	2,310	20	330	14.29
vehicle	846	18	218	34.71
Yeast (ME3)	1,484	9	163	10.98

minority outcast handling and ANS2, which stands for *Adaptive neighbor SMOTE with minority outcast handling* are represented. The highest value in each case of classifier and dataset is highlighted in the bold-face type.

The number of datasets in each classifier that each oversampling technique provides the best F-measure value is represented as a stacked bar chart in Figure 4. From the chart, Adaptive neighbor SMOTE without minority outcast handling (ANS1) has 27 cases of different datasets and classifiers which it provides the best F-measure value over other oversampling techniques (excluding ANS2). This number is almost twice than DBSMOTE (15) which has the second most number. For ANS with minority outcast handling (ANS2), the number of cases has increased to 35 which equals to a half of the total number of cases in this experiment. ANS1 and ANS2 still achieve the largest number of cases they provide the best AUC values against other five oversampling techniques. Shown in the stacked bar charts in Figure 5, ANS1 has 36 cases and ANS2 has 44 cases which both are more than half of the total number of cases.

The Wilcoxon signed-rank test is applied per suggestion by Demsar (2006) in order to investigate whether the difference of F-measure and AUC values caused by ANS with minority outcast handling (ANS2) against other over-sampling techniques is significant. The difference between each pair of samples is ranked from the smallest absolute value to the largest absolute value and its sign are collected. The ranks of positive sign difference and the ranks of negative sign difference are summed separately; then the smaller value of these two sums becomes the t-score of the test. This t-score is compared with the critical t-score with a significance level of 0.05 for 14 samples which equals to 22. The result from tests ANS2 against other oversampling techniques in each classifier is shown in the Table 4. The ones whose calculated t-score is lower is highlighted with bold-face type. From Table 4, the positive difference between ANS2 and other oversampling techniques is verified as significant in most classifiers. ANS2 overcomes SMOTE and

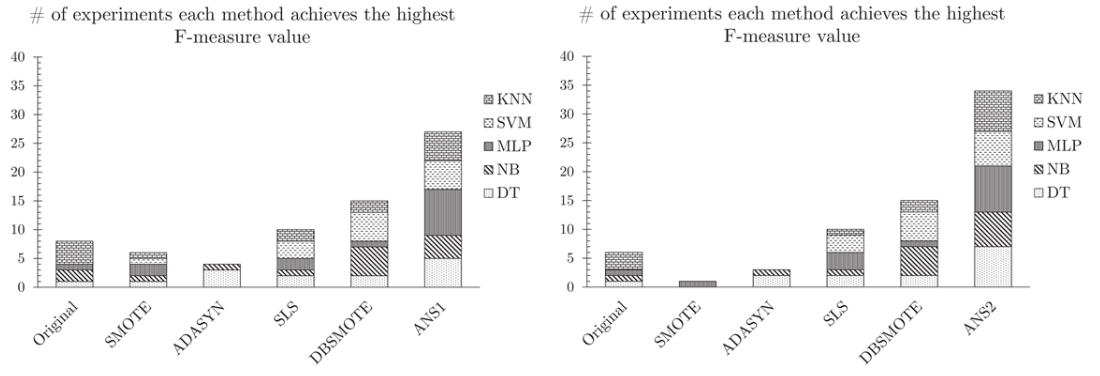

Figure 4. Number of datasets which each oversampling techniques achieve the best F-measure; ANS without outcast handling (ANS1) vs. others (left) and ANS with outcast handling (ANS2) vs. others (right).

Table 2. Performance in the F-measure of the original imbalanced dataset, and generated datasets from SMOTE, ADASYN, Safe-level SMOTE (SLS), DBSMOTE, ANS without minority outcast handling (ANS1) and ANS with minority outcast handling (ANS2).

Dataset	method	Decision Tree	naïve Bayes	Multilayer Perceptron	Support vector machine	K-nearest neighbor
cm1	Original	0.5497±0.1050	0.4967±0.0696	0.3886±0.1248	0.8088±0.1260	0.6771±0.0693
	SMOTE	0.6249±0.1050	0.5798±0.0453	0.6875±0.0563	0.7923±0.0693	0.7610±0.0638
	ADASYN	0.6000±0.0652	0.5727±0.0469	0.5577±0.1058	0.7558±0.0581	0.7339±0.0529
	SLS	0.6504±0.0590	0.5773±0.0360	0.6982±0.0480	0.8065±0.0698	0.7834±0.0596
	DBSMOTE	0.6387±0.0694	0.5772±0.0344	0.7138±0.0634	**0.8273±0.0730**	**0.7929±0.0621**
	ANS1	**0.6971±0.0720**	0.5781±0.0426	0.7195±0.0547	0.8083±0.0710	0.7806±0.0568
	ANS2	0.6952±0.0729	**0.5926±0.0424**	**0.7208±0.0555**	0.8006±0.0708	0.7795±0.0561
ecoli	Original	0.1026±0.0308	0.2474±0.0182	0.0000±0.0000	0.0000±0.0000	0.1647±0.0132
	SMOTE	0.2594±0.0189	0.2545±0.0173	0.3682±0.0283	0.3139±0.0169	0.2931±0.0204
	ADASYN	0.2600±0.0287	0.2580±0.0248	0.3639±0.0237	0.3197±0.0121	0.2974±0.0160
	SLS	0.1897±0.0565	0.2340±0.0228	0.3737±0.0156	0.3078±0.0251	0.2529±0.0232
	DBSMOTE	0.2303±0.0523	0.2360±0.0168	0.3174±0.0255	0.2605±0.0243	0.2784±0.0292
	ANS1	0.2428±0.0136	0.2693±0.0197	0.3832±0.0148	0.3515±0.0055	0.3047±0.0195
	ANS2	**0.2710±0.0175**	**0.3017±0.0304**	**0.3971±0.0230**	**0.3684±0.0215**	**0.3190±0.0163**
glass	Original	0.5810±0.0920	0.5438±0.0263	0.0000±0.0000	0.7048±0.0192	0.7881±0.0407
	SMOTE	0.6521±0.0686	0.6163±0.0173	0.7779±0.0211	0.8033±0.0230	0.8025±0.0181
	ADASYN	0.6764±0.0520	0.5131±0.0179	0.7467±0.0247	0.7443±0.0281	0.8025±0.0205
	SLS	0.7431±0.0190	0.8044±0.0234	0.8123±0.0172	0.8211±0.0183	0.8429±0.0244
	DBSMOTE	0.6633±0.0603	0.6888±0.0126	0.7916±0.0190	0.8211±0.0168	0.8714±0.0473
	ANS1	0.7754±0.0456	**0.8222±0.0191**	**0.7947±0.0126**	**0.8306±0.0099**	**0.8635±0.0055**
	ANS2	**0.8269±0.0435**	0.8222±0.0364	0.7947±0.0201	0.8306±0.0137	0.8635±0.0269
haberman	Original	0.6687±0.0396	0.6140±0.0061	0.2132±0.0471	0.6400±0.0156	0.6988±0.0285
	SMOTE	0.7057±0.0175	0.6145±0.0051	0.5894±0.0085	0.6284±0.0060	0.7184±0.0212
	ADASYN	0.6607±0.0041	**0.6298±0.0087**	0.5548±0.0340	0.6682±0.0239	0.7164±0.0204
	SLS	**0.7069±0.0249**	0.6209±0.0051	0.5704±0.0329	**0.6782±0.0230**	0.7316±0.0104
	DBSMOTE	0.6851±0.0154	0.5899±0.0121	0.5958±0.0172	0.6724±0.0077	0.7041±0.0219
	ANS1	0.6701±0.0242	0.6135±0.0109	0.6132±0.0143	0.6487±0.0096	0.7230±0.0129
	ANS2	0.6853±0.0250	0.6230±0.0144	**0.6197±0.0150**	0.6608±0.0120	**0.7351±0.0155**
jm1	Original	0.2144±0.0620	0.2918±0.0053	0.0467±0.0259	0.2354±0.0420	0.3391±0.0283
	SMOTE	0.4903±0.0197	0.4182±0.0150	0.4655±0.0164	0.4445±0.0151	0.3863±0.0263
	ADASYN	0.4841±0.0262	0.4192±0.0227	0.4741±0.0188	0.4467±0.0157	0.3601±0.0369
	SLS	0.4464±0.0384	0.4276±0.0137	0.4864±0.0274	0.4393±0.0198	0.3758±0.0237
	DBSMOTE	0.4999±0.0331	0.3912±0.0197	0.4882±0.0166	0.4063±0.0372	0.3776±0.0076
	ANS1	0.5038±0.0184	0.4388±0.0233	**0.4928±0.0360**	0.4878±0.0181	0.4078±0.0223
	ANS2	**0.5209±0.0156**	**0.4592±0.0289**	0.4926±0.0388	**0.4977±0.0305**	**0.4204±0.0161**
kc1	Original	0.2455±0.0141	0.2864±0.0012	0.1787±0.0111	0.1742±0.0029	0.3470±0.0037
	SMOTE	0.3798±0.0053	0.3024±0.0008	0.4069±0.0084	0.4320±0.0013	0.4065±0.0047
	ADASYN	0.3711±0.0045	**0.3294±0.0018**	0.4148±0.0086	0.4330±0.0014	0.4011±0.0048
	SLS	**0.4167±0.0078**	0.2926±0.0018	0.4170±0.0047	**0.4399±0.0013**	0.4032±0.0017
	DBSMOTE	0.3564±0.0122	0.3324±0.0053	0.3861±0.0067	0.3913±0.0019	0.3881±0.0022
	ANS1	0.3587±0.0182	0.3091±0.0011	0.4278±0.0061	0.4333±0.0010	**0.4276±0.0035**
	ANS2	0.3600±0.0184	0.3132±0.0029	**0.4279±0.0067**	0.4336±0.0010	0.4263±0.0042
kc2	Original	0.3182±0.0170	0.3992±0.0032	0.3193±0.0209	0.2443±0.0126	0.4022±0.0201
	SMOTE	0.4406±0.0099	0.4163±0.0051	0.4272±0.0059	0.4121±0.0038	0.4511±0.0078
	ADASYN	**0.4511±0.0162**	0.4325±0.0048	0.4232±0.0079	0.4059±0.0044	0.4472±0.0127
	SLS	0.4457±0.0089	0.4083±0.0021	**0.4400±0.0048**	0.4474±0.0086	0.4521±0.0104
	DBSMOTE	0.3838±0.0187	**0.4451±0.0072**	0.4148±0.0084	0.4056±0.0051	0.4358±0.0096
	ANS1	0.4342±0.0090	0.4169±0.0026	0.4400±0.0067	0.4385±0.0015	**0.4813±0.0051**
	ANS2	0.4479±0.0135	0.4323±0.0102	0.4324±0.0073	**0.4479±0.0057**	0.4672±0.0031

Table 2. Continued

Dataset	method	Decision Tree	naïve Bayes	Multilayer Perceptron	Support vector machine	K-nearest neighbor
letter	Original	0.5020±0.0113	0.5073±0.0092	0.5085±0.0135	0.4680±0.0051	0.5491±0.0235
	SMOTE	0.5049±0.0354	0.5472±0.0163	0.5434±0.0151	0.5741±0.0027	0.5351±0.0174
	ADASYN	0.5628±0.0176	0.5472±0.0130	0.5598±0.0142	0.5694±0.0097	0.5421±0.0202
	SLS	0.5394±0.0128	0.5511±0.0170	0.5816±0.0148	0.5927±0.0051	**0.5626±0.0188**
	DBSMOTE	0.5514±0.0109	**0.5575±0.0053**	0.5688±0.0107	0.5848±0.0094	0.5540±0.0110
	ANS1	**0.5832±0.0230**	0.5409±0.0111	0.5918±0.0079	**0.5984±0.0050**	0.5618±0.0108
	ANS2	0.5790±0.0233	0.5386±0.0114	**0.5938±0.0052**	0.5925±0.0064	0.5563±0.0079
optdigits	Original	**0.7771±0.0169**	**0.3431±0.0024**	0.4192±0.0102	0.4864±0.0044	**0.8961±0.0069**
	SMOTE	0.7642±0.0049	0.1821±0.0006	0.5685±0.0172	0.7662±0.0036	0.8227±0.0032
	ADASYN	0.7618±0.0091	0.1258±0.0009	0.4912±0.0175	0.7858±0.0036	0.8140±0.0034
	SLS	0.7453±0.0099	0.1829±0.0005	**0.5858±0.0260**	0.7876±0.0069	0.8561±0.0069
	DBSMOTE	0.7312±0.0063	0.2144±0.0024	0.5820±0.0388	**0.7877±0.0042**	0.8508±0.0034
	ANS1	0.7605±0.0040	0.1880±0.0009	0.5814±0.0264	0.7643±0.0024	0.8033±0.0061
	ANS2	0.7660±0.0044	0.1920±0.0015	0.5831±0.0263	0.7650±0.0020	0.8046±0.0064
pc1	Original	0.9667±0.0049	0.8767±0.0024	**0.9878±0.0014**	0.9936±0.0006	0.9982±0.0000
	SMOTE	0.9687±0.0063	0.9484±0.0067	0.9835±0.0020	0.9945±0.0009	0.9973±0.0004
	ADASYN	0.9632±0.0044	0.5210±0.0111	0.9809±0.0021	0.9945±0.0006	0.9982±0.0007
	SLS	0.9694±0.0045	0.9486±0.0043	0.9869±0.0010	0.9945±0.0006	0.9973±0.0004
	DBSMOTE	**0.9765±0.0040**	.9551±0.0036	0.9870±0.0013	0.9955±0.0005	0.9982±0.0004
	ANS1	0.9756±0.0024	0**0.9592±0.0021**	0.9866±0.0023	**0.9964±0.0017**	**0.9991±0.0006**
	ANS2	0.9756±0.0024	**0.9592±0.0021**	0.9866±0.0023	**0.9964±0.0017**	**0.9991±0.0006**
satimage	Original	0.3438±0.0511	0.2893±0.0095	0.0500±0.0081	0.1292±0.0139	0.3113±0.0194
	SMOTE	0.4063±0.0245	0.2508±0.0097	0.3001±0.0164	0.3175±0.0081	0.3814±0.0197
	ADASYN	0.3583±0.0224	0.2477±0.0108	0.2929±0.0139	0.2990±0.0088	0.3674±0.0233
	SLS	0.3993±0.0271	0.2601±0.0147	0.2920±0.0111	0.3172±0.0190	0.3809±0.0174
	DBSMOTE	0.3332±0.0428	0.2352±0.0102	0.2518±0.0192	0.2410±0.0167	0.3391±0.0397
	ANS1	0.3959±0.0184	0.2662±0.0087	0.2893±0.0117	0.3172±0.0149	0.3711±0.0073
	ANS2	**0.4174±0.0207**	**0.3198±0.0184**	**0.3109±0.0129**	**0.3386±0.0142**	**0.3892±0.0070**
segment	Original	0.5447±0.0128	0.4862±0.0014	0.5357±0.0510	0.5517±0.0044	**0.6895±0.0027**
	SMOTE	0.5662±0.0096	0.4839±0.0017	0.5693±0.0103	0.5939±0.0040	0.6179±0.0020
	ADASYN	0.5653±0.0080	0.4204±0.0022	0.5187±0.0186	0.5398±0.0027	0.6008±0.0028
	SLS	0.5741±0.0068	0.4957±0.0019	0.5946±0.0110	0.6192±0.0012	0.6390±0.0027
	DBSMOTE	0.5574±0.0140	**0.5721±0.0027**	**0.6002±0.0096**	**0.6391±0.0020**	0.6436±0.0047
	ANS1	0.5768±0.0077	0.5088±0.0050	0.5908±0.0132	0.6203±0.0050	0.6024±0.0058
	ANS2	**0.5875±0.0089**	0.5153±0.0049	0.5940±0.0121	0.6244±0.0030	0.6034±0.0053
vehicle	Original	0.8649±0.0085	0.4944±0.0054	0.6931±0.0099	0.1532±0.0182	0.8839±0.0089
	SMOTE	0.8898±0.0083	0.4938±0.0027	0.7999±0.0071	0.5842±0.0007	0.8780±0.0018
	ADASYN	0.8749±0.0073	0.4699±0.0102	0.7925±0.0073	0.6035±0.0042	0.8712±0.0011
	SLS	0.8855±0.0052	0.4937±0.0033	0.8047±0.0147	0.5898±0.0013	0.8749±0.0107
	DBSMOTE	**0.8957±0.0114**	**0.5004±0.0022**	0.7371±0.0158	**0.6127±0.0084**	0.8821±0.0089
	ANS1	0.8853±0.0102	0.4901±0.0020	0.8083±0.0120	0.5964±0.0017	0.8830±0.0052
	ANS2	0.8844±0.0109	0.4988±0.0039	**0.8122±0.0122**	0.5970±0.0012	**0.8847±0.0062**
yeast	Original	0.9199±0.0065	0.5823±0.0093	0.8947±0.0081	0.9504±0.0024	**0.9233±0.0130**
	SMOTE	0.9225±0.0051	0.6200±0.0081	**0.9039±0.0067**	0.9463±0.0032	0.8937±0.0111
	ADASYN	**0.9383±0.0030**	0.6258±0.0101	0.8956±0.0133	0.9554±0.0024	0.8897±0.0061
	SLS	0.9319±0.0045	**0.6280±0.0092**	0.8965±0.0085	0.9505±0.0046	0.8957±0.0184
	DBSMOTE	0.9056±0.0070	0.5674±0.0087	0.9005±0.0095	**0.9593±0.0030**	0.8961±0.0090
	ANS1	0.9252±0.0049	0.6194±0.0079	0.9016±0.0186	0.9503±0.0026	0.8788±0.0115
	ANS2	0.9314±0.0053	0.6227±0.0081	0.8998±0.0185	0.9530±0.0025	0.8788±0.0122

Table 3. Performance in the AUC value of the original imbalanced dataset, and generated datasets from SMOTE, ADASYN, Safe-level SMOTE, DBSMOTE, ANS without minority outcast handling (ANS1) and ANS with minority outcast handling (ANS2).

Dataset	method	Decision Tree	naïve Bayes	Multilayer Perceptron	Support vector machine	K-nearest neighbor
cm1	Original	0.9008±0.0362	0.7851±0.0670	0.5842±0.0288	0.9449±0.0290	0.9007±0.0296
	SMOTE	0.9201±0.0250	0.7852±0.0665	0.9241±0.0363	0.9488±0.0193	0.9037±0.0287
	ADASYN	0.9118±0.0323	**0.8195±0.0466**	0.7603±0.0977	0.9453±0.0185	0.8979±0.0311
	SLS	0.9140±0.0216	0.7818±0.0678	0.9263±0.0309	0.9466±0.0189	0.8961±0.0318
	DBSMOTE	0.9144±0.0265	0.7817±0.0648	0.9222±0.0281	0.9438±0.0209	0.9075±0.0294
	ANS1	**0.9223±0.0239**	0.8124±0.0682	0.9239±0.0271	**0.9547±0.0191**	0.9069±0.0295
	ANS2	0.9156±0.0249	0.8149±0.0687	**0.9273±0.0301**	0.9544±0.0205	**0.9138±0.0299**
ecoli	Original	0.5227±0.0178	0.7377±0.0180	0.5000±0.0000	0.5000±0.0000	0.6312±0.0170
	SMOTE	0.6842±0.0109	0.7437±0.0242	**0.7844±0.0188**	0.7569±0.0170	0.6881±0.0210
	ADASYN	**0.6916±0.0122**	0.7441±0.0334	0.7818±0.0176	0.7597±0.0166	0.6940±0.0144
	SLS	0.6657±0.0367	0.7323±0.0061	0.7711±0.0192	0.7572±0.0174	0.6433±0.0160
	DBSMOTE	0.6622±0.0406	0.6039±0.0267	0.7491±0.0131	0.7202±0.0149	0.6656±0.0202
	ANS1	0.6523±0.0181	0.7472±0.0117	0.7707±0.0118	0.7713±0.0140	0.7289±0.0217
	ANS2	0.6761±0.0141	**0.7748±0.0200**	0.7788±0.0154	**0.7839±0.0139**	**0.7342±0.0206**
glass	Original	0.8651±0.0413	0.9819±0.0018	0.5000±0.0000	0.8960±0.0015	0.9250±0.0115
	SMOTE	0.8803±0.0298	0.9931±0.0028	0.9905±0.0020	0.9889±0.0025	0.9175±0.0114
	ADASYN	0.8537±0.0468	0.9774±0.0025	0.9889±0.0020	0.9857±0.0018	0.9175±0.0011
	SLS	0.9147±0.0053	0.9965±0.0025	0.9897±0.0025	0.9905±0.0028	0.9230±0.0006
	DBSMOTE	0.9271±0.0074	0.9914±0.0032	0.9897±0.0024	0.9897±0.0025	0.9198±0.0015
	ANS1	0.9183±0.0020	**0.9966±0.0006**	**0.9929±0.0034**	**0.9929±0.0029**	**0.9449±0.0145**
	ANS2	**0.9437±0.0111**	0.9936±0.0035	0.9929±0.0043	0.9929±0.0057	0.9437±0.0161
haberman	Original	0.7447±0.0432	0.7001±0.0121	0.5868±0.0249	**0.8247±0.0073**	0.8177±0.0154
	SMOTE	0.7911±0.0154	0.6947±0.0220	0.6494±0.0090	0.8050±0.0092	0.8345±0.0245
	ADASYN	0.7716±0.0091	0.6751±0.0179	0.6335±0.0176	0.8198±0.0066	0.8387±0.0120
	SLS	0.7841±0.0329	0.7001±0.0248	0.6515±0.0135	0.8075±0.0058	0.8315±0.0198
	DBSMOTE	0.7737±0.0087	0.6772±0.0061	**0.6886±0.0073**	0.8046±0.0048	0.8264±0.0165
	ANS1	**0.7944±0.0290**	0.7006±0.0195	0.6529±0.0122	0.8132±0.0103	0.8378±0.0146
	ANS2	0.7912±0.0350	**0.7330±0.0271**	0.6666±0.0140	0.8167±0.0141	**0.8407±0.0160**
jm1	Original	0.5412±0.0195	0.6219±0.0189	0.5528±0.0482	0.6790±0.0171	0.6090±0.0234
	SMOTE	0.6596±0.0180	0.6318±0.0110	0.6822±0.0070	0.6561±0.0171	0.6032±0.0172
	ADASYN	**0.6647±0.0120**	0.6251±0.0121	0.6912±0.0077	0.6737±0.0128	0.6257±0.0296
	SLS	0.6380±0.0316	**0.6541±0.0125**	0.6828±0.0066	0.6535±0.0112	0.5979±0.0174
	DBSMOTE	0.6382±0.0105	0.6050±0.0186	0.6859±0.0105	0.6649±0.0171	0.6233±0.0130
	ANS1	0.6561±0.0285	0.6279±0.0180	0.6971±0.0137	0.7029±0.0124	0.6370±0.0165
	ANS2	0.6510±0.0144	0.6428±0.0066	**0.7069±0.0165**	**0.7137±0.0244**	**0.6446±0.0235**
kc1	Original	0.6685±0.0117	0.6892±0.0003	0.7084±0.0003	0.6207±0.0022	0.6597±0.0029
	SMOTE	0.6838±0.0084	0.6907±0.0003	0.7124±0.0005	0.7152±0.0003	0.6688±0.0043
	ADASYN	0.6850±0.0018	0.6928±0.0006	0.7100±0.0003	0.7123±0.0005	0.6627±0.0042
	SLS	0.6881±0.0046	0.6915±0.0003	**0.7147±0.0008**	**0.7184±0.0002**	0.6665±0.0034
	DBSMOTE	0.6888±0.0023	0.6934±0.0011	0.6652±0.0041	0.6740±0.0024	0.6600±0.0019
	ANS1	0.6988±0.0035	**0.6939±0.0004**	0.7113±0.0010	0.7124±0.0009	**0.6934±0.0024**
	ANS2	**0.7003±0.0040**	0.6929±0.0015	0.7071±0.0018	0.7089±0.0013	0.6908±0.0028
kc2	Original	0.6820±0.0234	0.7924±0.0004	0.7985±0.0012	0.6678±0.0069	0.7042±0.0079
	SMOTE	0.7567±0.0083	0.7960±0.0004	0.7972±0.0028	0.7819±0.0031	0.7328±0.0076
	ADASYN	0.7543±0.0054	**0.7980±0.0010**	0.7989±0.0034	0.7701±0.0032	0.7275±0.0132
	SLS	0.7652±0.0078	0.7905±0.0007	0.7998±0.0020	0.7847±0.0030	0.7168±0.0109
	DBSMOTE	0.7566±0.0075	0.7841±0.0031	0.7658±0.0041	0.7577±0.0038	0.7314±0.0104
	ANS1	**0.7774±0.0043**	0.7934±0.0016	0.8016±0.0016	**0.7851±0.0023**	**0.7730±0.0049**
	ANS2	0.7646±0.0053	0.7827±0.0045	**0.8027±0.0053**	0.7840±0.0041	0.7710±0.0042

Table 3. Continued

Dataset	method	Decision Tree	naïve Bayes	Multilayer Perceptron	Support vector machine	K-nearest neighbor
letter	Original	0.7331±0.0206	0.8287±0.0036	0.8416±0.0019	0.7737±0.0114	0.7718±0.0131
	SMOTE	0.7718±0.0175	0.8296±0.0020	0.8421±0.0042	0.8171±0.0040	0.7708±0.0136
	ADASYN	0.7682±0.0325	**0.8314±0.0029**	0.8425±0.0049	0.7995±0.0066	0.7638±0.0138
	SLS	0.7823±0.0149	0.8276±0.0027	**0.8445±0.0046**	0.8185±0.0082	0.7644±0.0145
	DBSMOTE	0.7952±0.0157	0.8292±0.0032	0.8370±0.0065	0.8026±0.0099	0.7844±0.0117
	ANS1	0.8021±0.0190	0.8291±0.0024	0.8403±0.0015	**0.8274±0.0045**	**0.8043±0.0099**
	ANS2	**0.8149±0.0217**	0.8274±0.0066	0.8371±0.0074	0.8219±0.0058	0.8019±0.0111
optdigits	Original	0.9677±0.0032	0.8644±0.0005	0.8252±0.0231	0.9870±0.0003	0.9789±0.0026
	SMOTE	0.9681±0.0025	0.8604±0.0005	**0.9791±0.0034**	0.9949±0.0003	0.9865±0.0019
	ADASYN	0.9684±0.0008	0.6981±0.0017	0.9690±0.0040	**0.9953±0.0004**	0.9854±0.0015
	SLS	0.9618±0.0019	0.8595±0.0005	0.9790±0.0017	0.9927±0.0012	0.9822±0.0026
	DBSMOTE	0.9568±0.0024	0.8277±0.0009	0.9596±0.0069	0.9888±0.0003	0.9801±0.0020
	ANS1	0.9664±0.0017	0.8557±0.0005	0.9762±0.0025	0.9935±0.0003	**0.9928±0.0015**
	ANS2	**0.9689±0.0019**	**0.8663±0.0014**	0.9775±0.0023	0.9948±0.0003	**0.9928±0.0015**
pc1	Original	0.9885±0.0012	0.9934±0.0006	0.9995±0.0002	0.999981±0.0000	0.9988±0.0005
	SMOTE	0.9929±0.0015	0.9958±0.0006	0.9994±0.0001	0.999983±0.0000	0.9989±0.0004
	ADASYN	0.9923±0.0014	0.9207±0.0029	0.9994±0.0001	0.999989±0.0000	0.9989±0.0004
	SLS	0.9943±0.0007	0.9957±0.0006	0.9995±0.0001	0.999986±0.0000	0.9990±0.0000
	DBSMOTE	0.9936±0.0004	0.9941±0.0006	0.9995±0.0001	**0.999994±0.0000**	0.9988±0.0005
	ANS1	**0.9948±0.0010**	**0.9965±0.0004**	**0.9996±0.0001**	0.999992±0.0000	**0.9992±0.0004**
	ANS2	**0.9948±0.0010**	**0.9965±0.0004**	**0.9996±0.0001**	0.999992±0.0000	**0.9992±0.0004**
satimage	Original	0.6478±0.0312	0.6991±0.0064	0.5929±0.0117	0.6572±0.0304	0.7441±0.0153
	SMOTE	0.8247±0.0224	0.7004±0.0100	0.8345±0.0079	0.8279±0.0050	0.8031±0.0192
	ADASYN	0.8202±0.0128	0.6643±0.0060	0.8281±0.0124	0.8107±0.0043	0.7976±0.0152
	SLS	0.7957±0.0166	0.6808±0.0165	0.8241±0.0053	0.8177±0.0082	0.7492±0.0131
	DBSMOTE	0.7490±0.0431	0.6995±0.0087	0.7638±0.0243	0.7733±0.0086	0.7667±0.0172
	ANS1	0.8248±0.0101	0.6765±0.0120	0.8261±0.0070	0.8214±0.0095	0.8288±0.0140
	ANS2	**0.8487±0.0122**	**0.7301±0.0059**	**0.8511±0.0040**	**0.8304±0.0047**	**0.8349±0.0157**
segment	Original	0.8672±0.0049	0.9197±0.0004	0.9310±0.0028	0.9353±0.0005	0.9130±0.0028
	SMOTE	0.9146±0.0036	0.9192±0.0004	0.9383±0.0019	0.9520±0.0006	0.9290±0.0030
	ADASYN	0.9133±0.0012	0.9091±0.0002	0.9291±0.0095	0.9467±0.0006	0.9277±0.0025
	SLS	0.9128±0.0036	0.9195±0.0005	0.9350±0.0035	0.9502±0.0009	0.9140±0.0034
	DBSMOTE	0.9051±0.0014	0.9164±0.0005	0.9230±0.0026	0.9456±0.0010	0.9205±0.0051
	ANS1	0.9135±0.0043	0.9196±0.0004	0.9388±0.0031	0.9510±0.0013	0.9309±0.0021
	ANS2	**0.9175±0.0042**	**0.9238±0.0009**	**0.9407±0.0026**	**0.9529±0.0008**	**0.9316±0.0019**
vehicle	Original	0.9737±0.0043	0.8361±0.0012	0.9553±0.0026	0.9052±0.0545	0.9702±0.0032
	SMOTE	0.9830±0.0032	0.8347±0.0018	0.9777±0.0010	0.9111±0.0011	0.9753±0.0034
	ADASYN	0.9797±0.0023	0.8205±0.0015	0.9773±0.0021	0.8721±0.0022	0.9758±0.0017
	SLS	0.9828±0.0020	0.8348±0.0020	0.9776±0.0013	0.9149±0.0014	0.9701±0.0034
	DBSMOTE	**0.9842±0.0018**	0.8294±0.0029	0.9657±0.0040	0.9225±0.0030	0.9727±0.0041
	ANS1	0.9825±0.0011	0.8372±0.0010	0.9773±0.0007	0.9242±0.0018	0.9766±0.0015
	ANS2	0.9822±0.0015	**0.8498±0.0052**	**0.9784±0.0011**	**0.9283±0.0022**	**0.9767±0.0015**
yeast	Original	**0.9796±0.0038**	**0.8604±0.0046**	**0.9858±0.0016**	0.9956±0.0005	**0.9868±0.0026**
	SMOTE	0.9697±0.0052	0.8506±0.0057	0.9795±0.0014	0.9955±0.0009	0.9784±0.0030
	ADASYN	0.9753±0.0026	0.8264±0.0021	0.9773±0.0016	0.9951±0.0008	0.9791±0.0023
	SLS	0.9700±0.0022	0.8516±0.0069	0.9808±0.0010	0.9954±0.0008	0.9807±0.0031
	DBSMOTE	0.9682±0.0025	0.7729±0.0069	0.9822±0.0015	**0.9958±0.0005**	0.9800±0.0024
	ANS1	0.9750±0.0040	0.8450±0.0042	0.9797±0.0006	0.9949±0.0006	0.9807±0.0024
	ANS2	0.9788±0.0042	0.8535±0.0079	0.9782±0.0005	0.9933±0.0011	0.9800±0.0027

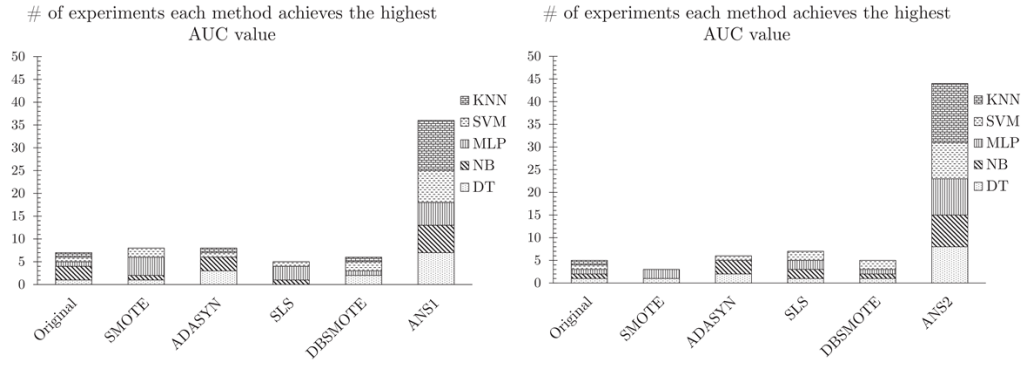

Figure 5. Number of datasets which each oversampling techniques achieve the best AUC; ANS without outcast handling (ANS1) vs. others (left) and ANS with outcast handling (ANS2) vs. others (right).

Table 4. t-score of each test between the F-value of ANS2 against ones of other oversampling techniques in each classifier. For the number of sample equals to 14, the T-critical value is 22.

Classifier	ANS2 vs. SMOTE	ANS2 vs. ADASYN	ANS2 vs. SLS	ANS2 vs. DBSMOTE
Decision Tree	**min{86, 19} = 19**	**min{95, 10} = 10**	**min{83, 22} = 22**	**min{99, 6} = 6**
Na¨ýve Bayes	**min{101, 4} = 4**	**min{90, 15} = 15**	**min{96, 9} = 9**	min{73, 32} = 32
Multilayer Perceptron	**min{103, 2} = 2**	**min{105, 0} = 0**	**min{83, 22} = 22**	**min{96, 9} = 9**
Support vector machine	**min{104, 1} = 1**	**min{87, 18} = 18**	min{69, 36} = 36	min{68, 37} = 37
K nearest neighbor	**min{88, 17} = 17**	**min{98, 7} = 7**	min{66, 39} = 39	min{67, 38} = 38

Table 5. t-score of each test between the AUC value of ANS2 against ones of other oversampling techniques in each classifier. For the number of sample equals to 14, the T-critical value is 22.

Classifier	ANS2 vs. SMOTE	ANS2 vs. ADASYN	ANS2 vs. SLS	ANS2 vs. DBSMOTE
Decision Tree	min{80, 25} = 25	**min{87, 18} = 18**	**min{100, 5} = 5**	**min{102, 3} = 3**
Na¨ýve Bayes	**min{93, 12} = 12**	**min{95, 10} = 10**	min{82, 23} = 23	**min{99, 6} = 6**
Multilayer Perceptron	min{70, 35} = 35	**min{88, 17} = 17**	min{77, 28} = 28	**min{91, 14} = 14**
Support vector machine	**min{88, 17} = 17**	**min{91, 14} = 14**	**min{90, 15} = 15**	**min{102, 3} = 3**
K nearest neighbor	**min{105, 0} = 0**	**min{105, 0} = 0**	**min{103, 2} = 2**	**min{105, 0} = 0**

ADASYN significantly in every classifier and defeats statistically SLS and DBSMOTE in the decision tree and multilayer perceptron and insignificant for rest of the classifiers.

Moreover, the Wilcoxon signed-rank test is also performed on the average AUC values. The result from these tests is shown on Table 5. The same analysis is achieved in most tests. There are only differences between ANS2 vs. SMOTE on decision tree and multilayer perceptron and ANS2 vs. SLS on naive Bayes classifier and multilayer perceptron which are not statistically significant.

6. Conclusions

This work introduces a new variation of SMOTE called *Adaptive neighbor Synthetic Minority Oversampling Technique under 1NN outcast handling* (ANS). It eliminates the parameter *K* of SMOTE for a dataset and assigns different number of neighbors for each positive instance. Simultaneously, this technique extracts minority outcasts out of the training data and uses them to build an exclusive 1NN model. Every parameter for this technique is automatically set within the algorithm making it become parameter-free. The effectiveness of this technique is shown by comparing with other oversampling techniques in a number of datasets and classifiers. We found that ANS has the highest number of cases it provides the best F-measure and AUC values. Wilcoxon sign-rank tests are applied to verify that ANS is statistically better than other techniques. However, the exact condition to determine which kind of dataset ANS performs well is still inconclusive. Based on experimental results and their statistical test, ANS is worth to use for remedy the imbalance of dataset if the good F-measure or AUC value is preferred.

Acknowledgements

The authors wish to thank the Development and Promotion of Science and Technology Talents Project (DPST) to provide scholarships for entire study time in undergraduate and graduate program. We thank our colleagues from Applied Mathematics and Computational Science program, Chulalongkorn University for their supports and advice, and we thank anonymous reviewers for their valuable suggestions.

References

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Density-based synthetic minority oversampling technique. *Applied Intelligence, 36*, 664-684.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-SMOTE: Safe-level-synthetic minority oversampling technique for handling the class imbalanced problem. *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining 2009*, 475-482.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research, 16*, 321-357.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1-30. Retrieved from http://dl.acm.org/citation.cfm?id=1248547.1248548

Ester, M., Kriegel, H., Jorg, S., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *AAAI Press*, 226-231.

Han, H., Wang, W., & Mao, B. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Proceedings of the 2005 International Conference on Advances in Intelligent Computing 1*, 878-887.

He, H., Bai, Y., Garcia, E., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, 1322-1328.

He, H., & Garcia, E. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263-1284.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. *Proceedings of the 2000 International Conference on Artificial Intelligence,* 111-117.

Lichman, M. (2013). UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml

Menzies, T., Caglayan, B., He, Z., Kocaguneli, E., Krall, J., Peters, F., & Turhan, B. (2012). The promise repository of empirical software engineering data. Retrieved from http://promisedata.googlecode.com

Nazari, K. M. R., Seiti, F., & Teshnehlab, M. (2012). An experimental comparative study on thyroid disease diagnosis based on feature subset selection and classification. *International Journal of Electrical and Computer Sciences, 12*(1), 13-19.

Siriseriwan, W., & Sinapiromsaran, K. The effective redistribution for imbalance dataset: Relocating safe-level SMOTE with minority outcast handling. *Chiang Mai Journal of Science, 43*(1), 234-246.