Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance

Shiven Sharma*, Colin Bellinger†, Bartosz Krawczyk‡, Osmar Zaiane† and Nathalie Japkowicz¶
*Weather Telematics Inc., Ottawa, Ontario, Canada
Email: ssharma@weathertelematics.com
†Alberta Machine Intelligence Institute, Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
Email: {cbelling, zaiane}@ualberta.ca
‡Department of Computer Science, Virginia Commonwealth University
Richmond, Virginia, USA
Email: bkrawczyk@vcu.edu
¶Department of Computer Science, American University
Washington, D.C, USA
Email: japkowicz@american.edu

***Abstract*—The class imbalance problem is a pervasive issue in many real-world domains. Oversampling methods that inflate the rare class by generating synthetic data are amongst the most popular techniques for resolving class imbalance. However, they concentrate on the characteristics of the minority class and use them to guide the oversampling process. By completely overlooking the majority class, they lose a global view on the classification problem and, while alleviating the class imbalance, may negatively impact learnability by generating borderline or overlapping instances. This becomes even more critical when facing extreme class imbalance, where the minority class is strongly underrepresented and on its own does not contain enough information to conduct the oversampling process. We propose a novel method for synthetic oversampling that uses the rich information inherent in the majority class to synthesize minority class data. This is done by generating synthetic data that is at the same Mahalanbois distance from the majority class as the known minority instances. We evaluate over 26 benchmark datasets, and show that our method offers a distinct performance improvement over the existing state-of-the-art in oversampling techniques.**

***Keywords*-Class imbalance, synthetic oversampling, classification**

I. Introduction

Rare events are often associated with high risks and little knowledge regarding the source, or the form that they will eventually take. Exemplary domains include fault detection [25], disease classification [20], software failures [8] and customer churn prediction [9]. In these domains and many others, there is one class that has a significantly larger prior probability than the other; this is known as the *class imbalance* problem. Such situations are known to pose challenging classification problems that can lead to a significant degradation in the performance of binary classifiers [5], [15].

Given the frequency and importance of the class imbalance problem in real-world domains, many approaches to increasing the robustness of binary classifiers to imbalance have been studied and proposed. Synthetic oversampling has received a large portion of the research focus in recent years; it has been shown to be effective in mitigating the impact of class imbalance [4], [10], [11] when the imbalance ratios are not extreme (*i.e.*, typically under 1:100). However, many real-life applications exhibit extreme imbalance, which can be both relative, where ratios are very high (over 1:1000), and/or absolute, when the number of minority class instances available for training are very low. For example, in the domain of gamma-ray anomaly detection [21], the datasets for learning have approximately 25,000 benign signatures, but less than 10 anomalies. In fraud detection domains, fraudulent data is typically very rare; Wei *et. al* [26] note that for online banking fraud detection, there are only 5 fraudulent examples in a dataset of over 300,000 transactions. The domain of software defect prediction can exhibit extreme absolute imbalance; a publicly available dataset by NASA has over 1500 non-defective samples, but only 16 defective samples [22]. In such cases, the few instances that we have are usually too important to be ignored, and they should be utilized in the most effective manner for classifier induction. Unfortunately, given their scarcity, there is not enough information to use them as a catalyst for synthesizing additional training instances; existing methods that do, such as SMOTE, can harm performance in these situations [17].

In this work, we ask the question: is there an effective methodology for synthetic oversampling the minority class in domains that exhibit extreme imbalance? Our research demonstrates that the answer is yes, and the key to it is to *utilize the distributional information in the majority class* for generating synthetic minority data. While some variants of SMOTE utilize the majority class training instances to guide the generative process for post-hoc cleaning, such as Borderline SMOTE [14] or SMOTE with the removal of Tomek links and nearest neighbour editing [23], the oversampling process is

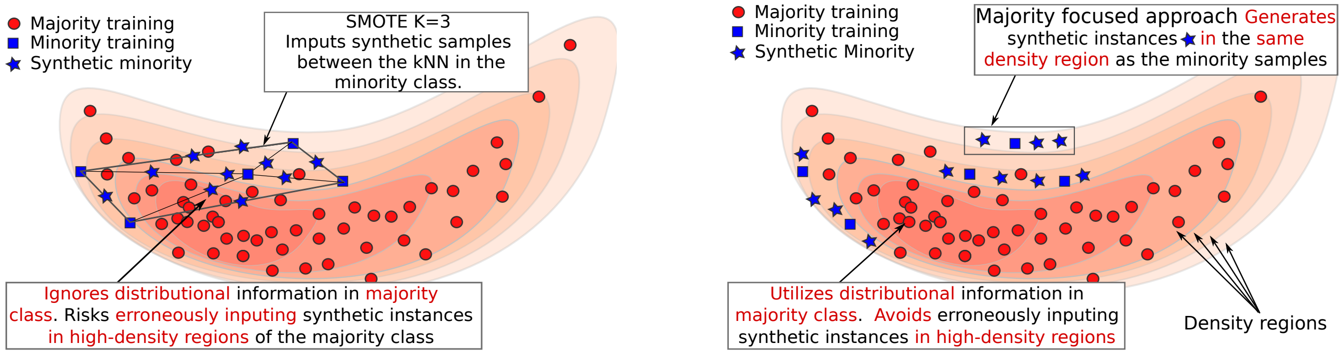
Majority training
Minority training
Synthetic minority
SMOTE K=3
Imputs synthetic samples
between the kNN in the
minority class.
Ignores distributional information in majority
class. Risks erroneously inputing synthetic instances
in high-density regions of the majority class
Majority training
Minority training
Synthetic Minority
Majority focused approach Generates
synthetic instances in the same
density region as the minority samples
Utilizes distributional information in
majority class. Avoids erroneously inputing
synthetic instances in high-density regions
Density regions

Fig. 1. Illustration of the generation of synthetic samples within the convex-hull using SMOTE (left), and along the density contours of the majority class using SWIM (Sampling WIth the Majority), the proposed majority-focused approach (right).

still primarily dependent on the minority class data. In extreme imbalance, the few samples of the minority class instances offer minimal distributional information, and what they offer may be misleading due to rarity, sparsity and noise. This can cause traditional oversampling methods to generate erroneous synthetic training instances that may harm the classifier.

We propose a majority-focused strategy to synthetically inflate the rare class under the moniker of SWIM (Sampling WIth the Majority). SWIM is intuitive, easy to implement, and efficient over domains exhibiting extreme imbalance. It is founded on the intuition that *a*) the synthetic minority instances should be generated in regions of the data space that have similar densities with respect to the majority class as the real minority instances, and *b*) that they should be generated in regions that neighbour the real minority instances. Specifically, instead of asking: *given the minority class data, where should new minority class instances be generated*, we ask: *given the majority class data and the relative position of the minority class instances, where should new minority class instances be generated*. Our method does not require any knowledge regarding the distribution of the minority class; the existing minority samples are simply indicators of the position of the minority population with respect to the majority class. Thus, we constrain the generation of synthetic minority samples by directing the generation in regions strategically positioned with respect to the majority class.

In general terms, this can be summarized with the following steps:

1) Estimate the PDF $\hat{p}_+()$ of the large majority class X_+
2) Synthesize a minority instances x'_- from a random minority instance x_- as $x'_- = x_- + r$, such that r shifts x_- to a neighbouring region of the data space where $\hat{p}_+(x'_-) = \hat{p}_+(x_-)$.

This is to say that x'_- and x_- are in neighbouring regions of the data space with the same probability densities with respect to the majority class. This is contrasted with SMOTE [11] and its derivatives [13], the standard methods for synthetic minority oversampling, in Fig. 1 for a generic case.

We formalize SWIM using a Mahalanobis distance (MD)-based approach [19]. The MD of each given minority class instance corresponds with a hyperelliptical density contour around the majority class, and we inflate the minority class by generating synthetic samples along these contours. This serves to overcome two well-known limitations of SMOTE-based methods by generation of synthetic samples outside of the convex-hull formed by the minority class instances, and prevents them from being generated in higher probability areas of the majority class. Using the MD to model the majority class involves the implicit assumption about the Gaussian nature of the data. While in practice data will not always strictly satisfy the Gaussian assumption, the MD has been shown to work well on a wide variety of outlier detection and classification problems in data mining and machine learning. This, along with our results, suggests that the method is robust in spite of its underlying assumption[1].

We empirically validate SWIM with respect to SMOTE, and its derivatives that aim to remove and/or avoid generating harmful synthetic instances, on 26 benchmark datasets that have been selected to exhibit both extreme relative (high imbalance ratios) and absolute (small number of minority training samples) imbalance. Our results show that our method has a large advantage on domains with extreme absolute and relative imbalance. In this space, it outperforms the existing state-of-the-art methods for synthetic oversampling.

We summarize the contributions of our paper as follows:

- We emphasize that taking the distribution of the majority class instances into account may significantly benefit the oversampling process.
- We develop a Mahalanobis distance-based implementation of SWIM for minority synthetic oversampling that explicitly uses the majority class in the generative process.
- We demonstrate that using the proposed approach enables the generation of beneficial synthetic instances for extreme levels of imbalance.
- We compare the performance of SWIM to the state-of-the-art methods in synthetic minority oversampling on highly imbalanced benchmark datasets.

[1] We discuss this and methods to manage complex non-Gaussian data in the Section V-B.

II. Related Work

In this work, we focus on supervised binary classification problems over highly imbalanced domains. The process of binary classification utilizes a training set $X_{n\times m} \in \mathbb{R}$ and corresponding labels $Y_n \in \{0,1\}$. The objective is to induce a function, $f(x_i) \rightarrow y_i$, that maps the training instances $x_i \in X$ to their corresponding class labels $y_i \in Y$. This problem is made more challenging in imbalanced domains where there are far few examples of the minority class X_{min}, $y = 1$ than in the majority class, X_{maj} $y = 0$. This has been shown to cause the induced classifier $f(\cdot)$ to become biased towards the larger class, thus leading to poor performance [15].

Two paradigms exist for dealing with imbalanced classification problems. When the minority class is rare or unavailable, one-class classification is applied. However, binary learning quickly becomes advantageous as the number of instances increases [5]. This has motivated research into extending the life span of binary classifiers over increasingly imbalanced domains using both sampling and cost-based approaches [12], [24]. In this paper, we focus on sampling approaches.

The most basic re-sampling strategies are Random Under-Sampling (RUS), and Random OverSampling (ROS). These balance class distributions in the training set by randomly discarding instances of the majority class, and/or randomly replicating instances of the minority class. These strategies, however, suffer from the loss of information and the risk of overfitting, respectively.

To avoid these shortcomings, and to expand the regions of the data space occupied by the minority training instances, the Synthetic Minority Oversampling TEchnique was proposed (SMOTE) [11]. It produces a balanced training set by interpolating synthetic instances between nearest neighbours in the set of minority class instances in the training set. This procedure relies entirely on the minority class training instances; the outcome is that the resulting synthetic data is situated within the convex-hull formed by the minority class. Furthermore, by ignoring the majority instances, SMOTE may actually increase overlapping between classes. Thus, in cases of extreme imbalance, the synthesized set has the risk of harming the performance.

The success of SMOTE, along with the recognition of its limitations, has spawned a large number of variants [13]. The main focus of these has been to delete (clean) instances generated by SMOTE that are deemed to be harmful to the induction of a classifier, and direct the spread of the synthetic instances into regions of the data space that will correct the classification bias. These more recent methods have incorporated the majority class by using the Euclidean distances to the *k*-nearest neighbours and/or calculating the density/class distribution in the local neighbourhood. This relegates the majority class information to a post-hoc cleaning process [3], such as the removal of Tomek links and nearest neighbour editing [23], or to guiding the generation process based on a local perspective of the data around the minority class instances, rather than a global perspective. This is the case with Adaptive Synthetic Oversampling (ADASYN), borderline SMOTE, and Majority Weighted Minority Oversampling Technique [2], [14], [16]; the only majority class information used is that which is present within the local neighbourhood of the generated sample. In these methods, the distribution of the minority class remains the key component of the generative process. Consequently, an insufficient number of minority samples will negatively impact the generative process.

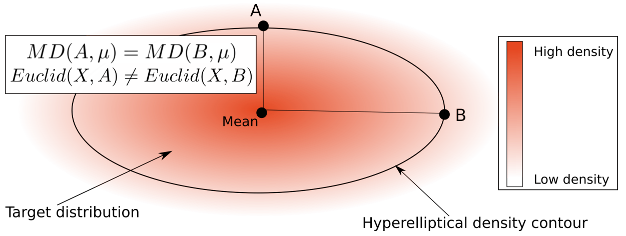

Fig. 2. Illustration of the Mahalanobis distance between two points **A** and **B** from the mean. Both points have the same Mahalanobis distance, but different Euclidean distances from the mean.

In addition to the SMOTE-based methods that rely on the Euclidean distance to the *k*-nearest neighbours, Abdi *et. al* [1] proposed the use of the MD for synthetic minority oversampling. The fundamental distinction with our method is that they do not utilize the majority class information. Rather, they generate synthetic samples using the MD calculated on the small, and potentially error prone, minority class training set; new samples are generated at the same MD as a reference minority point from the minority class mean. Therefore, this method is susceptible to failure due to the limitations of the dearth of minority class data in the training set, as the estimated mean and covariance matrix would be unrepresentative of the latent minority distribution.

At their core, all current state-of-the-art oversampling methods still rely on the representativeness of the minority class instances to produce a beneficial synthetic set. Alternatively, our method does not make any assumptions regarding what the minority class represents, except where existing samples are positioned with respect to the majority class. The information for generating synthetic samples comes from the populous majority class, and thus, our method is effective for classification problems in which the minority class is rare, a situation that is both common and of great importance [17].

III. Method

In this section, we describe and discuss our proposed implementation of SWIM. We begin by providing a brief overview of the Mahalanobis distance, followed by a description of our algorithm for oversampling, as well as a discussion on the efficacy of our method.

A. Mahanobis Distances

The MD provides a calculation of the distance between the mean of the distribution, and a query point, that accounts for the density along the path. Thus, two points have the same MD from the mean if they lie on the same hyperelliptical density contour. This is contrasted with Euclidean distance in Fig. 2.

The calculation of the MD involves knowing the mean μ and the covariance matrix Σ of the distribution. In practice, however, the parameters are estimated as $\overline{\mu}$ and $\overline{\Sigma}$ on a sample population. Larger, more representative sets, such as is typical in the majority class training data, produce better estimates of these parameters. Once the parameters are estimated, the MD of an instance x from the mean μ is calculated as:

$$MD(x, \overline{\mu}) = (x - \overline{\mu})^T \overline{\Sigma}^{-1} (x - \overline{\mu})$$

B. Algorithm

The algorithm is based on the intuition that *Synthetic minority instances should be generated in similar density regions (contours) relative to the majority class as the real minority instances*. This is because, given only a handful of minority class instances, there is not going to be enough distributional information to determine where synthetic instances should be generated. However, if we look at how this data is distributed in relation to the vast majority class, synthetic data can be generated such that it is similarly distributed with respect to the majority class. In the absence of more minority class data, this relative distributional information is the most beneficial knowledge we have for generating synthetic data.

We now explicitly detail the various steps for oversampling with SWIM. Assuming the parameters of the MD have been estimated on the majority class A, the steps to generate a new sample, s_{new}, using a parent sample, x, of the minority class, B, is outlined below:

Step 1 **Centre the majority and minority classes**: Centring the data simplifies the calculation of the distances; this will be evident in the fourth step, when we generate a new sample point. Let $\mu_{\mathbf{a}}$ be the feature mean vector of the majority class A. We centre the majority class to have $\vec{0}$ mean, and then centre the minority class with mean vector of the majority class:

$$\begin{aligned} A_c &= A - \mu_{\mathbf{a}} \\ B_c &= B - \mu_{\mathbf{a}} \end{aligned} \tag{1}$$

Step 2 **Whiten the minority class**: Let $\mathbf{\Sigma}$ denote the covariance matrix of A_c, and $\mathbf{\Sigma}^{-1}$ denote its inverse. $\mathbf{\Sigma}^{-\frac{1}{2}}$ is the square root of $\mathbf{\Sigma}^{-1}$. We whiten the centred minority class as:

$$B_w = B_c \mathbf{\Sigma}^{-\frac{1}{2}} \tag{2}$$

The MD is equivalent to the Euclidean distance in the whitenend space of a distribution. Thus, by whitening, we simplify the calculations for generating synthetic data (as will be evident in Step 4).

Step 3 **Find feature bounds**: These are used to bound the spread of the synthetic samples. For each feature f in B_w, we find its mean μ_f and standard deviation σ_f. We then calculate an upper and lower bound on its value, u_f and l_f, as follows:

$$\begin{aligned} u_f &= \mu_f + \alpha\sigma_f \\ l_f &= \mu_f - \alpha\sigma_f \end{aligned} \tag{3}$$

$\alpha \in \mathbb{R}$ controls the number of standard deviations we want the bounds to be. Therefore, larger α values cause a greater amount of spread along the corresponding density contour.

Step 4 **Generate new samples**: For each feature f, we generate a random number between u_f and l_f. Thus, we obtain a sample point, s, where each feature s_f is $l_f \leq s_f \leq u_f$. This process can be repeated t times, where t is the desired number of artificial instances to be generated based on the reference datum x. For each minority reference datum in the whitened space, we generate samples that are at the same Euclidean distance from the mean of the majority class[2]. Since we centred the data, this implies that the new sample will have the same Euclidean norm as the minority datum. Therefore, we transform s as:

$$s_{norm} = s\frac{\|x\|_2}{\|s\|_2} \tag{4}$$

Step 5 **Scale sample back to original space**: s_{norm} exists in the whitened space of the minority class, with the same Euclidean distance from the mean vector $\vec{0}$ as x in the whitened space. We now have to transform the point back into the original space. This is done as:

$$s_{new} = (\mathbf{\Sigma}^{-\frac{1}{2}})^{-1} s_{norm}, \tag{5}$$

where the synthetic sample s_{new} will be in the same density contour as its parent minority instances x.

As the method involves the computation of matrix inverses, if there are linearly dependent columns, the calculations will fail. To handle this case, we check the rank r of the majority class A. If $r < d$, where d is the dimensionality of A, then we calculate the QR-decomposition of A. The non-zero values of the resulting upper-triangular matrix correspond to the linearly independent columns of A. Using the steps outlined above, we can then oversample and classify the data in the sub-space defined by the features represented by these columns.

C. SWIM versus SMOTE

We now examine the generative processes of SWIM and the SMOTE-family[3] of algorithms in extreme imbalance. Specifically, we examine where artificial data is synthesized by each method; in order for a robust binary classifier to be induced, data must be synthesized in areas corresponding with the real minority class distribution.

The key differences between the methods are that SMOTE and all its variants rely on the nearest neighbours in the

[2] This takes advantage of the whitening done in Step 2, as instead of dealing with the Mahalanobis distance, we can use the Euclidean distance.

[3] While our analysis in this section is focused on SMOTE, the insights apply to all neighbourhood-based sampling procedures. These insights are empirically validated in the experiments conducted in this paper.

minority class to determine where the synthetic instances should be generated, whereas SWIM uses the distribution of the majority class and the relative positions of the minority instances. The results of these fundamental differences can be striking in cases of extreme imbalance. For this demonstration, we created a complex, highly imbalanced artificial dataset with three minority training instances and 300 majority training instances. In order to produce representative results, we created a balanced test set with 300 instances in each class. The demonstration is presented in Figure 3; the figure shows the results of synthetic oversampling using SWIM (top left), and SMOTE (top right) on an extremely imbalanced toy domain. For completeness, we also show the result of the binary classifier without synthetic oversampling (bottom). The majority class training instances are shown as red squares with black outlines, and the corresponding test instances are shown as red circles. The minority class training instances are shown as blue squares with white outlines, and the corresponding test instances are shown as blue circles. In the top two plots where synthetic oversampling was applied prior to training the binary classifier, the synthetic instances are shown as blue squares with black outlines.

The instances synthesized by SMOTE are limited to a small area formed by the convex-hull of the minority training data. Alternatively, using SWIM causes the synthetic instances to be spread along the density contours corresponding to the MDs of the minority data from the majority class. The result can be seen in terms of the decisions surfaces (represented by the shading in the plots) induced by the three classifiers. Using the information in the majority class, our method produces samples that lead to a more representative decision surface, and thus, better classification performance.

Each of the visualized binary support vector classifiers were tested on 300 minority and majority class instances. Because this is an artificial classification problem that has been crafted to demonstrate the competing methods, we have access to a balanced test set. Given the balanced test set, we can confidently compare the methods using accuracy. The baseline classifier achieved a low accuracy of 0.69. Adding the instances generated by SMOTE to the training set improved the performance to 0.86. Alternatively, adding the instances generated via SWIM improved the accuracy to 0.95.

IV. Experimental Framework

To evaluate the proposed system, we compare the performance of binary classifiers induced on data re-sampled with state-of-the-art re-sampling methods and SWIM.

Data sets: Table I lists the 26 benchmark datasets[4] utilized in our evaluation. These were selected because they have high imbalance ratios ($> 1 : 100$) at extreme absolute imbalance levels (less than 10 minority training samples), as well as a wide variety of dimensionalities and sizes so as to reflect the real-world. We randomly down-sample the minority class in the training sets to simulate different levels of extreme imbalance. Specifically, we test at three different levels, with minority training set sizes of 4, 7 and 10.

Classification: Our experiments involve binary classifiers and sampling methods. The binary classifiers used are Naïve Bayes (NB), Nearest Neighbour (IBK), Decision Trees (J48), Multilayer Perception (MLP) and Support Vector Machines (SVM). With respect to the sampling methods, we employ random oversampling and undersampling, SMOTE [11] ($K = 3, 5, 7$), Borderline SMOTE (B1,B2) [14], SMOTE with one-sided selection with Tomek links [18] and adaptive synthetic sampling [16]. We use the *sklearn* package in Python for our experiments; all classifiers are run with their default settings. Data is normalized before training and testing.

Evaluation: The geometric mean of the true positive (majority class) and true negative (minority class) rate is used to evaluate classifier performance [18]. It is given by $g - mean = \sqrt{TPR \times TNR}$, where TPR is the true positive rate, and TNR is the true negative rate. Because each class is treated separately, it is immune to imbalance. Evaluation is done by randomly dividing the data into equal training and testing portions, and then randomly removing minority class training instances to achieve the desired level of absolute imbalance. This process is randomly repeated 30 times to ensure accurate estimations of the g-mean given the potential for large variances caused by the small minority training sets.

V. Results

We begin by examining the primary question we consider in this paper: at extreme imbalance, how does our proposed method, SWIM, compare to existing state-of-the art re-sampling methods. Table II lists the g-means obtained over the various datasets by our method, SWIM, and the best performing re-sampling method (ALT), and the baseline classifier (Baseline), when no sampling is performed. We are interested in comparing SWIM to best alternative re-sampling method, and thus report the average g-mean for the best performing combination of classifier and SWIM, along with the average g-means of the best performing combination of classifier and alternative re-sampling.

These results demonstrate the superiority of SWIM over the best alternatives for extreme imbalance. In particular, SWIM outperforms the competing methods on 23 of the 26 datasets.

In addition, we evaluate the cases of imbalance involving 7 and 10 minority training instances; while they represent extreme absolute imbalance over all datasets, they have marginally less extreme imbalance ratios. Specifically, less than half the datasets have imbalance ratios greater than 1:100 at these minority training set sizes.

Figure 4 depicts the relative performances over the three size categories. We plot the difference in performance (g-mean) between SWIM and the best alternative re-sampling method. The datasets are sorted in order of an increasing performance advantage for SWIM at size 4 in the training set. For each size, we make the following observations:

[4] http://archive.ics.uci.edu/ml/index.php
http://homepage.tudelft.nl/n9d04/occ/index.html

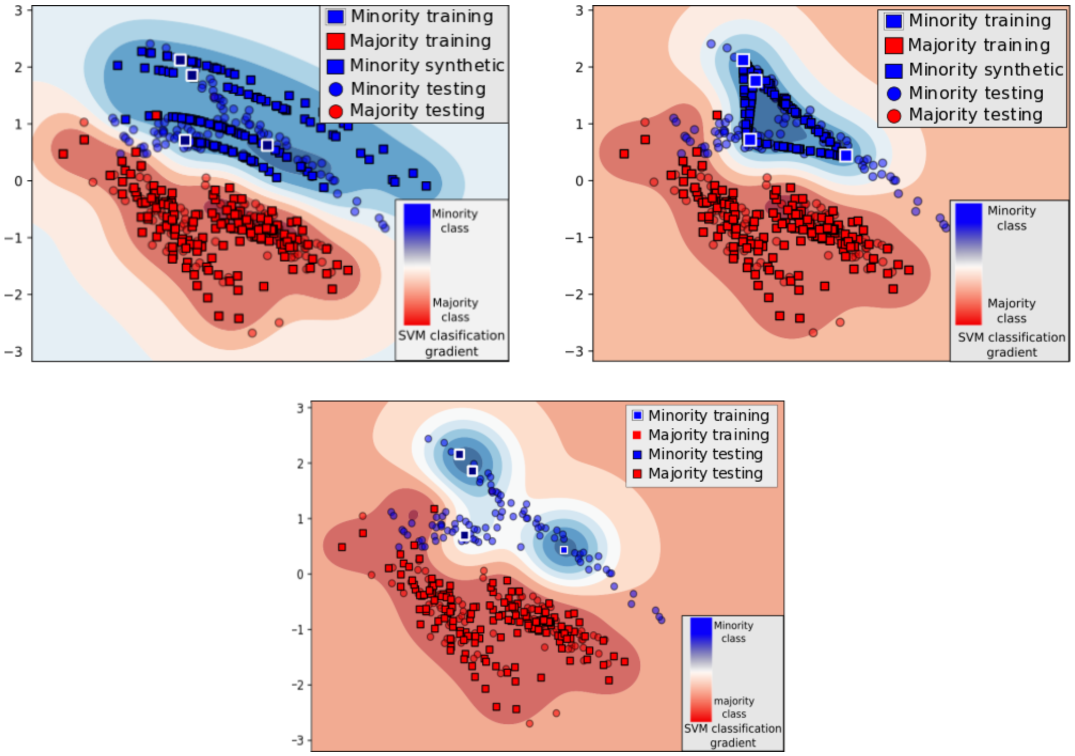

Fig. 3. Illustration of classifiers produced by using SWIM (top left) and SMOTE (top right). For reference, the bottom plot shows the binary support vector classifier induced over the imbalanced training set

- **Size 10**: Alternative sampling methods have a higher g-mean on 14 out of 26 datasets. However, most of the relative advantages are small; 10 of these are less than 0.05.
- **Size 7**: At this size, SWIM gains a significant advantage, as it is superior on 17 out of 26 datasets. Five of these offer improvements that are greater than 0.05.
- **Size 4**: At this extreme case, SWIM is very prominent. In particular, it outperforms alternatives on 23 of the 26 datasets. On 6 datasets, the difference in performance is over 0.1, whereas its over 0.05 on 14 datasets.

A. *Statistical Significance*

We use the Bayesian signed test [6] to evaluate the results presented in the previous section. The Bayesian signed test is alternative to the frequentist sign and signed-rank test, which is based on the Dirichlet process [7]. It enables the comparison of two classification methods over multiple datasets. In this case, we are comparing the use of SWIM for synthetic oversampling to the best alternative method over each of the datasets reported in Table I. The comparison is performed for training sets with 10, 7 and 4 minority instances.

Using the Bayesian method enables us to ask questions about posterior probabilities, which we cannot answer using null hypothesis tests. These include questions such as: is method A better than B? Based on the experiments, how probable is that A is better? How high is the probability that A is better by more than 1%? Indeed, these are the questions that we are actually interested in when comparing classifiers in data mining.

Based on the assumption of the Dirichlet process, the posterior probability for the Bayesian signed test is calculated as a mixture of Diracs deltas centred on the observation.

Figure 5 presents the three posterior plots of the Bayesian signed test for the comparison of SWIM to the best alternative method. The posteriors are calculated with the prior parameter of the Dirichlet as $s = 0.5$ and $z_0 = 0$ as suggested by the authors in [7]. The posterior plots report the samples from the posteriors (cloud of points), the simplex (the large orange triangle) and three regions. The region on the bottom left indicates the case where it is more probable that SWIM is better than the best alternative, the bottom right indicates the opposite, and the top region indicates that it is more probable that neither method is better. To summarize the plots, the more points that are close to one edge of the triangle, the better, statistically, the method associated with the region is. The closer the points are to the base of triangle, the bigger the statistical difference is.

The plots in Figure 5 validate that the relative probability of SWIM performing better than the best alternative re-sampling method increases with the decrease in the number of minority training samples. In the left most plot, corresponding with size 10, the point cloud is roughly centered in the triangle indicating that methods are approximately equally likely to perform better. The point clouds in the plots for size 7 and 4 shift down and to the left. This indicates that SWIM becomes

TABLE I

List of Datasets, the dimensions (Dims), majority class size (Maj. size), and imbalance ratios of the datasets. R4, R7, and R10 are the imbalance ratios when 4, 7, and 10 minority training instances are used.

Dataset	Name	Dim.	Maj. Size	R4	R7	R10
D1	Abalone 9-18	8	689	1:173	1:99	1:69
D2	Ada Agnostic	48	3430	1:858	1:490	1:343
D3	Alphabets	15	3077	1:770	1:440	1:308
D4	Analcat Data DMFT	7	642	1:161	1:92	1:65
D5	Diabetes	8	500	1:125	1:72	1:50
D6	Forest Cover	54	2970	1:743	1:425	1:297
D7	KDD Synthetic Control	61	500	1:125	1:72	1:50
D8	Mfeat Karhunen	64	1800	1:450	1:258	1:180
D9	Delft pump AR	160	531	1:133	1:76	1:54
D10	Spambase spam	57	2788	1:697	1:399	1:279
D11	Waveform 0	21	600	1:150	1:86	1:60
D12	Page Blocks	10	4913	1:1229	1:702	1:492
D13	PC4	37	1280	1:320	1:183	1:128
D14	Piechart	37	644	1:161	1:92	1:65
D15	Pima Indians	8	500	1:125	1:72	1:50
D16	Pizza Cutter	37	609	1:153	1:87	1:61
D17	Ring Norm	20	3736	1:934	1:534	1:374
D18	Thoracic Surgery	37	400	1:100	1:58	1:40
D19	Vehicle 0	18	647	1:162	1:93	1:65
D20	Vehicle 1	18	629	1:158	1:90	1:63
D21	Vehicle 2	18	628	1:157	1:90	1:63
D22	Vehicle 3	18	634	1:159	1:91	1:64
D23	Vowel 10	13	898	1:225	1:129	1:90
D24	Wine Quality Red 4	11	1546	1:387	1:221	1:155
D25	Wine Quality White 3 vs 7	11	880	1:220	1:126	1:88
D26	Wisconsin	9	444	1:111	1:64	1:45

TABLE II

G-mean obtained by using the best combination of classifiers with our proposed sampling method (SWIM), alternative re-sampling methods (ALT), and without sampling (Baseline), in the extreme case of 4 minority class training instances.

Dataset	Baseline	ALT	SWIM	Dataset	Baseline	ALT	SWIM
D1	0.481	0.612	**0.723**	D14	0.455	0.516	**0.576**
D2	0.451	0.445	**0.539**	D15	0.276	0.479	**0.509**
D3	0.27	0.451	**0.620**	D16	0.468	0.506	**0.552**
D4	0.279	0.440	**0.440**	D17	0.442	0.614	**0.799**
D5	0.259	0.509	**0.580**	D18	0.414	0.428	**0.453**
D6	0.561	**0.554**	0.550	D19	0.534	0.758	**0.814**
D7	0.958	0.965	**0.996**	D20	0.450	0.549	**0.560**
D8	0.274	**0.933**	0.899	D21	0.541	0.739	**0.791**
D9	0.569	0.872	**0.903**	D22	0.402	0.505	**0.569**
D10	0.440	0.550	**0.685**	D23	0.724	0.738	**0.812**
D11	0.301	**0.701**	0.688	D24	0.224	0.502	**0.535**
D12	0.647	0.679	**0.793**	D25	0.451	0.572	**0.730**
D13	0.572	0.559	**0.611**	D26	0.874	0.956	**0.958**

increasingly more likely to be the best method with high significance. For size 4, nearly all of the points fall inside SWIMs region; therefore, it is almost always better than the best alternative.

B. Discussion

Figure 4 highlights two distinct categories of datasets. The first group includes a set of 3 datasets on which an alternative re-sampling is best at each level of absolute imbalance. The second group includes the datasets on which SWIM was superior on at least the most extreme level of imbalance. By examining the PCA plots of the datasets in each group we are able to see the great advantage of the majority-focus strategy, as well as the situations where the alternative re-sampling methods remain strong even in extreme absolute imbalance.

Figure 6 shows the PCA plots for two example datasets from the group of three on which the alternative re-sampling were always better. The plot on the left shows a situation in which there is little overlap between the two classes, and the minority class is uni-modal. The results is that the convex-hull formed by the minority training points (off-white stars) only covers regions representative of the minority class, and does not spread into higher density regions of the majority class. This leads to good performance using SMOTE-based methods. The plot on the right has significant overlap, but the minority class remains a cohesive unimodal group, which enables SMOTE-based methods to populate an area representative of the minority class.

In both cases, the minority distribution is relatively compact, and uni-modal, and therefore, regardless of the extent of

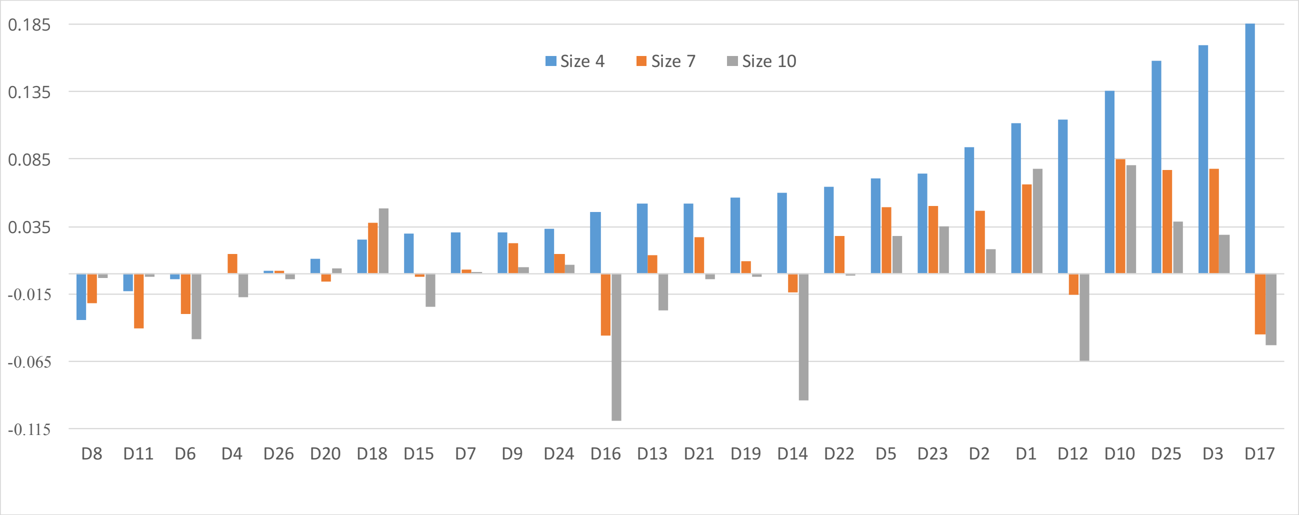

Fig. 4. Relative differences in g-mean between SWIM and ALT (best performing alternative re-sampling method) for each dataset $d_i \in D = \{d_1, d_2...\}$, where the differences is calculated as $diff(SWIM(d_i) - ALT(d_i))$. The colours of the bars correspond to training with 4, 7 and 10 minority training instances.

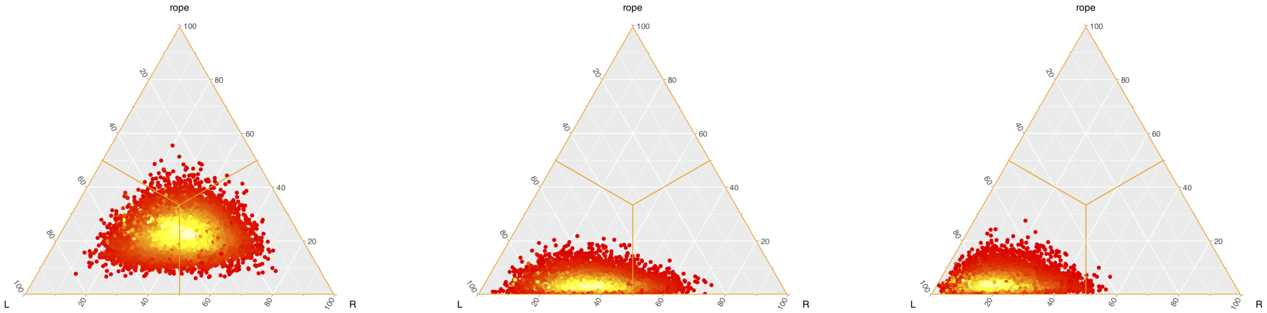

Fig. 5. Posteriors for SWIM (L) vs. the best alternative (R) on the data sets with size 10, 7 and 4 minority class training sets for the Bayesian sign-rank test. Higher concentration of points on one of the sides of the triangle shows that a given method has a higher probability of being statistically significantly better.

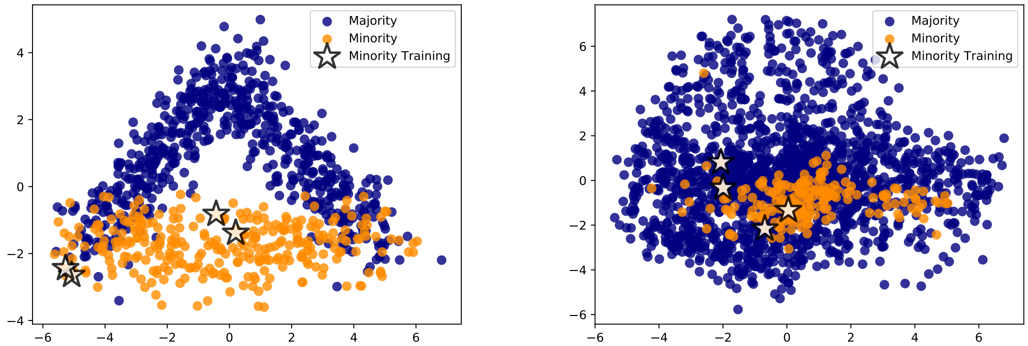

Fig. 6. PCA plots of example datasets for when alternative re-sampling is always better. The plot on the left is for D11, and the plot on the right is for D8.

imbalance, algorithms based on SMOTE are able to populate in the representative regions of the minority class. Using the equi-density neighbourhood approach, SWIM, has the ability to generate data outside the compact minority space regions, thereby negatively impacting classification.

Figure 7 shows example PCA plots of datasets from the group in which SWIM is better. In this cases, the minority class exists as a cluster of points, but also includes a large number of points that are spread away from the cluster. In such cases, SMOTE is easily mislead by the spread. This can cause it to synthesize instances in directions that overlap with higher density regions of the majority class and/or in directions that do not help shift the decision boundary. Alternatively, our majority-focused approach benefits from using the relative position of the minority training instances in the majority class to spread the synthetic instances into neighbouring regions of similar densities. In the case of the plot on the left, for example, SMOTE will only fill the small triangle formed by the minority class training instances, whereas SWIM will spread this synthetic instances further down into th distribution of the minority class, represented by the orange dots. This leads to better coverage of the minority space, and thus, better generalization on the part of the induced classifier.

Thus, sampling methods, such as SMOTE, that primarily rely on the minority class are less impacted by extreme imbalance if the distributions are relatively compact with respect to the majority class. However, when there is multi-modality and spread, they are severely impacted by extreme imbalance, whereas SWIM is robust, as it primarily utilizes the majority class distribution to guide the sampling process.

We now look at the important task of setting the single free parameters, α, in our method, and K for SMOTE[5]. Figure 8 presents a comparison of the variability of the choice of α for SWIM, and k for SMOTE across the datasets considered. The left pie chart shows the proportion of datasets for which setting $\alpha = \{1, 1.5, 2\}$ produced the best result for SWIM, and the right pie chart shows the proportion of datasets for which setting $K = \{3, 5, 7\}$ produced the best results for SMOTE. For SWIM, $\alpha = 2$ produced the best performance over most of the datasets; we found this to be the case across all levels of absolute imbalance, and thus we recommend this as an initial setting. For SMOTE, the choice of K is less clear cut, with both 5 and 7 being viable options depending on the dataset.

Finally, as we noted in the introduction, the Mahalanobis distance implicitly includes a Gaussian assumption about the distribution of the data. Regardless of the fact that many domains do not conform to the Gaussian assumption, methods such as naïve Bayes classifiers and the Mahalanobis distance for outlier detection have shown to be very successful in machine learning and data mining applications. In addition, we have shown SWIM to be superior over a wide variety of domains with various levels of complexity. Although it did not occur here, one can imagine a situation arising in which the majority class is composed of a complex, non-parametric distribution with multiple cluster in the data. In the case that SWIM fails to a achieve sufficient performance on such datasets, the majority class can be pre-processed into a set of k simplified clusters prior to synthetic oversampling; a similar approach has been successfully applied to one-class classification in Sharma *et. al* [21]. In this case, we propose that SWIM be applied to each cluster of the majority class, which the Mahalanobis distance will be better able to represent, separately in order to generate synthetic instances.

VI. Concluding Remarks

We present a method for synthetic oversampling, SWIM (Sampling WIth the Majority) in domains exhibiting extreme imbalance that utilizes the rich information offered by the majority class. Using the distribution of the majority class in the generation process enables us to synthesize in regions of the minority space that would otherwise be inaccessible. This is an essential feature in cases of extreme imbalance, as it enables the induced classifier to both decreases its bias and increases its generalization over the minority class. Alternatively, classical methods of re-sampling fail in such domains as they do not sufficiently take advantage of the information in the majority class; by using only the minority class, a very limited space is explored and sampled. We demonstrate the efficacy of our method over 26 benchmark datasets which include a wide variety of real-world properties. Our results show that SWIM has a significant advantage when the relative and absolute imbalance is very high.

Synthetic oversampling by explicitly using the majority class data opens the doors for many interesting research areas. Using the Mahalanobis distance (MD) has some key advantages, such as mathematical and computational efficiency, ease of sampling, and interpretability. Currently, in the synthesization process, we use all known minority class instances for synthesizing artificial data. However, it may be more prudent to select a beneficial subset of instances, or assign weights to instances based on the regions of the data space in which they lie. Alternative strategies for generating synthetic data in the whitened space are also an avenue for future research. Instance generation is also vital for other domains like incremental and online learning [27], and the application of our method in these areas will be explored. Finally, one-class classifiers are the other alternative employed for handling extremely imbalanced datasets. We will extend our work to compare and evaluate our proposed method against state-of-the art in one-class classifiers.

The core idea of SWIM, as outlined in Sections 1 and 3, is that we generate minority samples that have the same relative probability density with respect to the majority class as the known minority points. In the approach presented in this paper, we use the MD to this end. However, any appropriate density estimation method can be used to harness this insight and generate samples. The discovery of other sampling algorithms under this framework is an exciting area of future research.

[5] The parameter that controls the amount of samples to generate is the same for both methods. Specifically, we generate enough samples to make both the majority and minority class sizes equal.

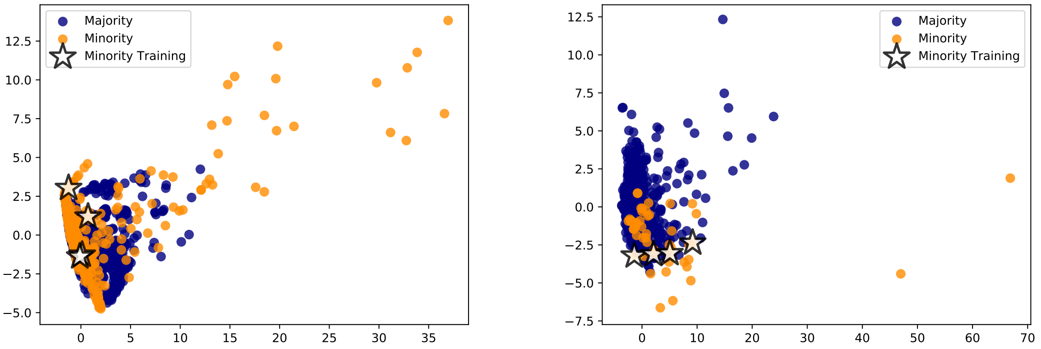

Fig. 7. PCA plots of exemplar datasets for when Mahalanobis synthetic oversampling is better. The plot on the top is for D12, and at the bottom is for D16.

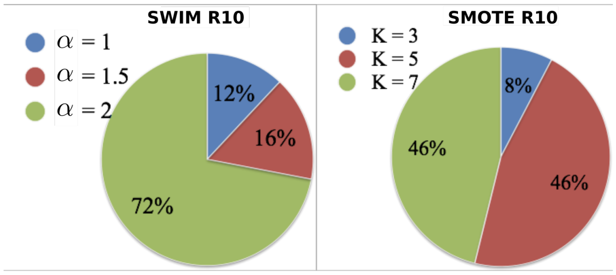

Fig. 8. The number of datasets on which a specific free parameter choice was best for SWIM (left) and SMOTE (right), when 10 minority class instances are available for training.

References

[1] Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling techniques. IEEE transactions on Knowledge and Data Engineering 28(1), 238–251 (2016)

[2] Barua, S., Islam, M.M., Yao, X., Murase, K.: Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on Knowledge and Data Engineering 26(2), 405–425 (2014)

[3] Batista, G., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets 6(1), 20 (2004)

[4] Bellinger, C., Drummond, C., Japkowicz, N.: Manifold-based synthetic oversampling with manifold conformance estimation. Machine Learning 107(3), 605–637 (2018)

[5] Bellinger, C., Sharma, S., Japkowicz, N.: One-class versus binary classification: Which and when? In: Machine Learning and Applications, 11th International Conference on. vol. 2, pp. 102–106. IEEE (2012)

[6] Benavoli, A., Corani, G., Demsar, J., Zaffalon, M.: Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. Journal of Machine Learning Research 18, 77:1–77:36 (2017)

[7] Benavoli, A., Corani, G., Demšar, J., Zaffalon, M.: Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. The Journal of Machine Learning Research 18(1), 2653–2688 (2017)

[8] Bennin, K.E., Keung, J., Phannachitta, P., Monden, A., Mensah, S.: Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Transactions on Software Engineering (2017)

[9] Burez, J., Van den Poel, D.: Handling class imbalance in customer churn prediction. Expert Systems with Applications 36(3), 4626–4636 (2009)

[10] Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: Smoteboost: Improving prediction of the minority class in boosting. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 107–119. Springer (2003)

[11] Chawla, N., Bowyer, K., Hall, L., W.P., K.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)

[12] Elkan, C.: The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)

[13] Fernández, A., Garcia, S., Herrera, F., Chawla, N.V.: Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research 61, 863–905 (2018)

[14] Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new oversampling method in imbalanced data sets learning. In: International Conference on Intelligent Computing. pp. 878–887 (2005)

[15] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)

[16] He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (3), 1322–1328 (2008)

[17] Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence 5(4), 221–232 (2016)

[18] Kubat, M., Matwin, S.: Addressing the curse of imbalanced data sets: One-sided sampling. In: Proceedings of the Fourteenth International Conference on Machine Learning. pp. 179–186 (1997)

[19] Mahalanobis, P.C.: On the generalized distance in statistics. National Institute of Science of India (1936)

[20] Miri Rostami, S., Ahmadzadeh, M.: Extracting predictor variables to construct breast cancer survivability model with class imbalance problem. Journal of AI and Data Mining (2017)

[21] Sharma, S., Somayaji, A., Japkowicz, N.: Learning over subconcepts: Strategies for 1-class classification. Computational Intelligence 34(2), 440–467 (2018)

[22] Siers, M.J., Islam, M.Z.: Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem. Information Systems 51, 62–71 (2015)

[23] Tomek, I.: Modifications of CNN. IEEE Trans. System, Man, Cybernetics 6(11), 769–772 (1976)

[24] Wang, H., Gao, Y., Shi, Y., Wang, H.: A fast distributed classification algorithm for large-scale imbalanced data. In: IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. pp. 1251–1256 (2016)

[25] Wang, S., Minku, L.L., Yao, X.: Resampling-based ensemble methods for online class imbalance learning. IEEE Transactions on Knowledge and Data Engineering 27(5), 1356–1368 (2015)

[26] Wei, W., Li, J., Cao, L., Ou, Y., Chen, J.: Effective detection of sophisticated online banking fraud on extremely imbalanced data. World Wide Web 16(4), 449–475 (2013)

[27] Zhu, Y., Ting, K.M., Zhou, Z.: New class adaptation via instance generation in one-pass class incremental learning. In: 2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017. pp. 1207–1212 (2017)