Minority Oversampling Using Sensitivity

Jianjun Zhang
Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering South China University of Technology
Guangzhou, China
jjzhangscut@gmail.com

Ting Wang
Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering South China University of Technology
Guangzhou, China
tingwang@ieee.org

Wing W. Y. Ng*
Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering South China University of Technology
Guangzhou, China
wingng@ieee.org

Witold Pedrycz
Department of Electrical and Computer Engineering University of Alberta
Edmonton, Alberta, Canada
wpedrycz@ualberta.ca

Shuai Zhang
School of Computing Ulster University
Jordanstown, Belfast, United Kingdom
s.zhang@ulster.ac.uk

Chris D. Nugent
School of Computing Ulster University
Jordanstown, Belfast, United Kingdom
cd.nugent@ulster.ac.uk

***Abstract*—The Synthetic Minority Oversampling Technique (SMOTE) is effective to handle imbalance classification problems. However, the random candidate selection of SMOTE may lead to severe overlap between classes and introduce new noise factors. Many variants of SMOTE have been proposed to relieve these problems by generating new examples in safe regions. Most of these methods generate new examples with existing minority examples without considering the negative impact that class imbalance have brought on these examples. In this paper, we handle the imbalance classification using Bayes' decision rule and propose a novel oversampling method, the Minority Oversampling using Sensitivity (MOSS). Candidates for new example generations are selected considering their sensitivity with respect to class imbalance. New examples are then generated by interpolating the candidate and one of its adjacent examples. Experiments on 30 datasets confirm the superiority of the MOSS against one baseline method and seven oversampling methods.**

***Index Terms*—bayes rule, imbalance, oversampling, sensitivity**

I. Introduction

Class imbalance occurs when a class consists of much less examples than the other class [1] and it is prevalent in many real-world applications, for examples anomaly detection [2], electricity pricing [3], and diagnosis of rare diseases [4]. Traditional classifiers are prone to be biased to majority class and minority examples are often misclassified. Misclassification in the minority class is not preferable because in many cases misclassifying minority examples are often more costly than misclassifying majority ones. For example, classifying a fraud transaction as normal one may lead to severe financial consequences.

Over the past decades, many algorithms have been proposed to handle the class imbalance problem [5] [6] [7]. Existing approaches can be roughly categorized into two types: data-level approaches and algorithm-level approaches [1]. In this work, we only focus on the data-level approaches, because they are classifier independent and are more versatile. Note that data-level approaches are often combined with ensemble learning to create more powerful algorithms [4].

Data-level approaches, also called resampling approaches, include oversampling and undersampling methods. Undersampling methods remove data to balance the class distribution, which risk the loss of important concepts [8]. Moreover, when the number of minority examples is small, undersampling would produce an undersized dataset, which may in turn limit classifier performance [9]. Oversampling, on the other hand, may encourage overfitting when observations are merely randomly duplicated [10]. This problem can be avoided by adding genuinely new samples. One possible solution is the Synthetic Minority Oversampling Technique (SMOTE) [11], which exhibits popularity among oversampling methods. To avoid overfitting, new samples are generated along a line

This work was supported partly by the National Natural Science Foundation of China under Grant 61876066, and the Guangdong Province Science and Technology Plan Project (Collaborative Innovation and Platform Environment Construction) 2019A050510006 . Jianjun Zhang was supported by the China Scholarship Council (file No. 201906150060). Support from the Canada Research Chair (CRC) is fully acknowledged.

Corresponding author: Wing W. Y. Ng.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

connecting a candidate minority example and one of its adjacent minority examples to enhance the representation of the minority class. However, the random candidate selection of SMOTE may introduce new challenges. Regions with more minority examples may be inflated further resulting the within-class imbalance. Another problem is that new noisy samples may be introduced when noisy examples already exist in the data [12], which would further complicate the learning problem.

A lot of variants of the SMOTE have been proposed to relieve these problems. Instead of blind oversampling, these methods attempt to generate new examples at specific locations of the feature space. Most of these variants focus on how to determine the locations of oversampling regions. For example, the Borderline-SMOTE [13] generates new examples only near the decision boundary by ignoring noisy and safe examples. Similarly, the Safe-Level-SMOTE [14] generates new examples in regions in which minority examples have a high safe level. The adaptive synthetic sampling approach (ADASYN) [15] generates more new examples near examples that are difficult to learn where the difficulty of learning of a minority example is measured by the proportions of majority examples in its k-nearest neighbors. The majority weighted minority oversampling technique (MWMOTE) [16] assigns different weights to hard-to-learn minority examples and generates new examples using a clustering approach. Then, new examples are generated inside minority clusters. In contrast, the k-influential neighborhood oversampling [17] provides an alternative way to identify noisy examples. The k-influential neighborhood (K-IN) of an example is interpreted as the region in feature space where this sample has direct influence via its k-nearest neighbors and indirect influence via its reverse k-nearest neighbors. Noisy minority examples are those whose K-IN has smaller number of minority examples than a threshold. Noisy samples are removed and the SMOTE is then applied to the minority class.

All methods above utilize a k-nearest neighbors-based method to determine where to generate new examples and the distance metric is Euclidean distance. The noise reduction a priori synthetic oversampling (NRAS) [18], on the other hand, utilizes the difference of conditional probability of belonging to the minority class as the distance metric and removes noises before oversampling. Similar to the NRAS, the Certainty Guided Minority Oversampling (CGMOS) also estimates the conditional probability distribution function to calculate the posterior probability of all examples using Bayes's rule [19]. The posterior probability is compared before and after a new example being added to the dataset. Examples that lead to higher improvement of posterior probability have higher chances to be selected as candidate for the SMOTE.

Apart from k-nearest neighbors-based method, clustering methods are also popular in resampling-based method because they can preserve data distribution [8]. For example, the k-means SMOTE applies oversampling only on safe regions to avoid noise generation where the safe region is discovered by k-means clustering [9]. A high ratio of minority examples in the cluster indicates a safe region.

These methods have shown superior performance against the original SMOTE in many empirical studies, implying that generating new examples in certain important regions help enhance the representation of minority class and thus the classification performance on minority class can be improved. In this work, we attempt to determine the regions of interest using Bayes' decision rule and then propose a novel oversampling method, the Minority Oversampling using Sensitivity (MOSS), for imbalance classification. Based on Bayes' decision rule, the sensitivity to imbalance of each example can be computed. Examples with low sensitivity are noisy and safe examples, while those with high sensitivity are borderline examples located in areas between two classes. In MOSS, examples with higher sensitivity are assigned higher sampling weight so that new examples can be generated in important regions while lowering the risk of introducing new noises.

The original aspects of this work are that we propose the sensitivity of an example to analyze the impact that class imbalance bring on this example, and then propose a novel sensitivity-based oversampling method for imbalance classification problems. The major contributions of this work are as follows.

1) The sensitivity with respect to imbalance of each example can be efficiently computed using Bayes' rule. This provides a systematic way to analyze the impact of imbalance on the dataset.
2) The sensitivity of each example is used as a criterion to guide the candidate selection for oversampling, which can generate informative examples near decision boundary for better classification.

The paper is organized as follows. Section II gives the details of the proposed method. Section III reports the empirical experiments on 30 datasets comparing the proposed method and 7 existing oversampling methods. Section IV concludes the work.

II. Proposed Method

In this section, we first introduce some basic notations. Then we show how to compute the sensitivity and apply oversampling using sensitivity.

A. Definition of Sensitivity

Given a dataset $D = \{(x_j, y_j)|j = 1, 2, \cdots, N\}$ where N is the number of examples, $x_j \in R^m$ is the feature vector of the j^{th} example with m being the number of features, and $y_j \in \{c_+, c_-\}$ is the corresponding label of x_j with c_+ and c_- being the positive and negative class respectively. The cardinality of the positive class is N_+ while that of the negative class is N_- and $N_+ + N_- = N$. In a binary imbalance problem, one usually denote the minority class as positive class and the majority class as negative class.

Then with Bayes' theorem, one can compute the posterior probability of an example using class conditional probability density function and prior probability as follows:

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

$$P(c|x_j) = \frac{p(x_j|c)P(c)}{\sum_c p(x_j|c)P(c)} \tag{1}$$

where $c \in \{c_+, c_-\}$.

The Bayes' optimal decision rule is given:

$$g(x_j) = \begin{cases} c_+, & P(c_+|x_j) > P(c_-|x_j) \\ c_-, & P(c_-|x_j) \geq P(c_+|x_j) \end{cases} \tag{2}$$

In (1), the denominator is the same for each example which can be omitted, so (2) can be reformulated as:

$$g(x_j) = \begin{cases} c_+, & \Gamma_+ > \Gamma_- \\ c_-, & \Gamma_- \geq \Gamma_+ \end{cases} \tag{3}$$

where

$$\Gamma_+ = p(x_j|c_+)P(c_+) \tag{4}$$

$$\Gamma_- = p(x_j|c_-)P(c_-) \tag{5}$$

In imbalance classification problems, usually $P(c_-) \gg P(c_+)$, which makes Γ_- dominate Γ_+ in most cases. This reveals the fact that minority examples are often misclassified in imbalance classification, because error rate is still very low. In order to alleviate this problem, the calibrated decision rule can be applied:

$$g(x_j) = \begin{cases} c_+, & \Gamma'_+ > \Gamma_- \\ c_-, & \Gamma_- \geq \Gamma'_+ \end{cases} \tag{6}$$

where

$$\Gamma'_+ = p(x_j|c_+)P(c_-) \tag{7}$$

In fact, (6) is the decision rule when the two classes are balanced. In this case, the effects of the dominance of $P(y_j = c_-)$ are alleviated. In the imbalanced case, negative class dominates positive class and lots of positive examples are recognized as negative class. Using the calibrated decision rule in (6) one would move the decision boundary towards the negative class to mitigate the negative impacts brought by the class imbalance. Based on this rule, we define the sensitivity of an example to imbalance as follows:

$$S(x_j, y_j) = \frac{\Gamma'_+}{\Gamma'_+ + \Gamma_-} - \frac{\Gamma_+}{\Gamma_+ + \Gamma_-} \tag{8}$$

Note that $\forall j, S(x_j, y_j) \geq 0$ because in an imbalanced dataset one has $P(c_-) \geq P(c_+)$. The sensitivity effectively measures the impacts on each example brought by class imbalance. An example with high sensitivity indicates that the posterior probability decreases dramatically due to class imbalance. Thus, this example is expected to contain more information and learning from it may help alleviate the negative influence brought by class imbalance. An example with low sensitivity, on the other hand, shows its robustness to class imbalance. Class imbalance does not have much influence on this example, indicating that this example is easy to learn. Learning too many of this type of examples may not help the classifier generalize well. As a matter of fact, an example with low sensitivity is either a safe example surrounded by examples from the same class, or a noisy example surrounded by examples from the other class. In both cases, this example is easy to learn and a classifier would produce high confidence on classifying it (although the noisy example would be misclassified). Learning from safe examples may help generalization, but learning from too many noisy examples may adversely hurt the performance. How to properly handle this situation would be one of our important future works. Overall, using the sensitivity information, one can locate the most informative regions where new examples can be generated.

B. Minority Oversampling using Sensitivity

We first give the overall procedure of the proposed method, the Minority Oversampling using Sensitivity (MOSS) and then explain the details.

The MOSS works as follows. Firstly, the sensitivity of each positive example is computed. Secondly, the sampling probability of the positive example is obtained based on the sensitivity information. Thirdly, a set of candidate examples are drawn from the positive class. Then, a random number ranging from 0 to 0.5 is generated for each candidate, and a new example is generated on the line connecting the candidate and one of its nearest neighbors where the distance to the candidate is scaled by the random number. The overall procedure is given in Algorithm 1.

The sensitivity consists of the class conditional probability density function and the prior probability. The prior probability can be estimated by the class ratio in the training set. Some works adopt the class conditionally independent assumption and the assumption that features follow a Gaussian distribution to estimate the class conditional probability density. These assumptions, although often being violated in real-world applications, have been empirically shown to be effective in several oversampling methods, for example CGMOS [19], Random Walk Oversampling [20], and Sampling With Majority [21]. In this work, we follow the similar idea and assume that each feature is class conditionally independent from each other and follows a Gaussian distribution. The class conditional probability density $P(x_j|c)$ is estimated as:

$$P(x_j^{(i)}|c) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} exp(-\frac{(x_j^{(i)} - \mu_{ij})^2}{2\sigma_{ij}^2}) \tag{9}$$

where $x_j^{(i)}$ is the i^{th} feature of example $x_j, i = 1, 2, \cdots, m$, μ_{ij} and σ_{ij}^2 are the mean and variance of $x_j^{(i)}$, respectively. Multivariate normal distribution or non-parametric kernel density estimation can be used to estimate the probability density and may be more accurate than the proposed method. But in this work, we are using the sensitivity of each example to compute a sampling probability for each example, absolutely accurate density is not needed and the proposed method is more efficient compared to these two methods. Having computed the prior and density, the sensitivity can now be determined using (8).

Based on the sensitivity, the sampling probability of the positive examples to determine the chances of being selected as the candidate is computed. As discussed in subsection II-A,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

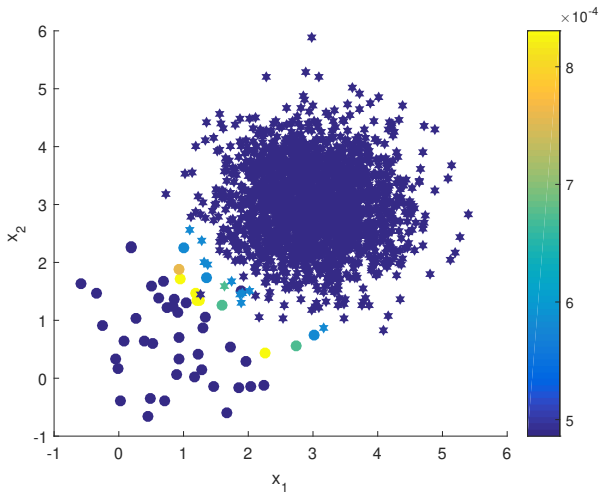

Fig. 1. Sampling probability distribution

examples with high sensitivities are more informative and therefore higher probabilities should be assigned to them. The sampling probability is computed as follows:

$$W(x_j, y_j) = \frac{S(x_j, y_j) + 1}{\sum_{j=1}^{N_+} S(x_j, y_j) + N_+} \tag{10}$$

Fig. 1 illustrates the sampling probability of each example in a toy dataset. The dataset contains 2050 examples, where 50 from the positive class and 2000 from the negative class. Examples are generated using two different normal distributions. The positive examples are generated from the distribution $N(\mu_+, \Sigma_+)$, where $\mu_+ = [1, 1]^\top$ and $\Sigma_+ = \left[\begin{smallmatrix} 0.5 & 0 \\ 0 & 0.5 \end{smallmatrix}\right]$. The negative examples are generated from the distribution $N(\mu_-, \Sigma_-)$, where $\mu_- = [3, 3]^\top$ and $\Sigma_- = \left[\begin{smallmatrix} 0.5 & 0 \\ 0 & 0.5 \end{smallmatrix}\right]$. It can be observed from Fig. 1 that examples located near the boundary of the two classes have higher sampling probabilities, while those located far away from the boundary have lower probabilities. Examples near the boundary are considered to contain more information of the distribution of two classes.

With the sampling probability, $N_- - N_+$ positive examples are drawn with replacement from the positive class as the candidates. A new example will be generated from each candidate so that a balanced dataset would be created with each class having N_- (i.e., the number of negative examples) examples. The new examples are generated on the line connecting the candidate and an adjacent example randomly chosen from its k-NN. Specifically, a new example x' is generated using the following expression:

$$x' = x_j + r * (x_k - x_j) \tag{11}$$

where x_j is the candidate, $x_k \in KNN(x_j)$ and r is a random number drawn from a uniform distribution $U(0, 0.5)$. The difference between the candidate and its adjacent example is scaled by a random number which is less than 0.5, so that the generated example is located closer to the candidate. This avoids generating new examples in the overlapping areas between the two classes, which may further complicate the learning problem.

Lots of SMOTE variants apply the k-NN to determine the regions of interest, for examples Borderline-SMOTE [13] and MWMOTE [16]. The major difference between the MOSS and these methods is that the MOSS takes the impact of class imbalance into account and alters the posterior probability accordingly, while these SMOTE variants only make use of the information of imbalanced dataset.

Algorithm 1 Minority Oversampling using Sensitivity

Input:

D, training dataset of size N containing N_+ positive examples and $N - N_+$ negative examples;
k, number of nearest neighbors to be used in k-NN;

Output:

Balanced dataset D'

1: Initilize $D' = \phi$
2: Compute the sensitivity S of each positive example using (8).
3: Compute the sampling probability W of each positive example using (10).
4: Draw $N - 2 * N_+$ positive examples from positive class with replacement based on W and denote them as M.
5: **for** each example $x_i \in M$ **do**
6: Draw a random number r from $U(0, 0.5)$.
7: Generate a new example x' using (11).
8: $D' = D' \bigcup x'$
9: **end for**
10: $D' = D' \bigcup D$

C. Time Complexity

The time complexity of the MOSS involves computing the sensitivity and sampling probability, and generating new examples by finding k-NN of each positive example. Time complexity for computing sensitivity and sampling probability are both $O(N)$. Finding k-NN of all positive examples requires time complexity of $O((N_- - N_+)Nmk) = O(N^2mk)$ because $N_- \approx N$ given $N_- \gg N_+$. As such, the overall time complexity is $O(N^2mk + N + N) = O(N^2mk)$.

III. Experimental Studies

In this section, we perform experiments on 30 imbalance datasets accessed from the KEEL [22] dataset repository to validate the effectiveness of the proposed method. The characteristics of these datasets are shown in Table I including the name, number of features, number of examples, class distribution, and imbalance ratio (IR). The IR is defined as $\frac{N_-}{N_+}$. From Table I, one can note that the IR varies from 2.46 to 100.14 which can evaluate the effects of imbalance ratio on performances of different methods. A thirty-time five-fold cross validation is used to evaluate the performance of each method. In each repetition of cross validation, the resampling method is performed on the training set and tested on the remaining validation set. The cross validation process

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

is repeated for thirty times to record the average performance. The performance metric used in this work is the Area under ROC curve (AUC), which is commonly used in imbalance classification.

Seven existing oversampling methods are used as the reference, which are the ROS, SMOTE [11], both versions of Borderline-SMOTE (BSMOTE1, BSMOTE2) [13], MWMOTE [16], safe-level SMOTE (SLSMOTE) [14], and k-means SMOTE (KSMOTE) [9]. Default parameters are used as suggested in the original papers proposing these methods. The parameter k used in the MOSS is set to 5 as the same as in the SMOTE variants. A baseline method (NONE) without any treatment to the imbalance problem is also used as the baseline. The classification and regression tree (CART) is used as the classifier in this work.

TABLE I
CHARACTERISTICS OF 30 DATASETS. #F MEANS THE NUMBER OF FEATURES, #E MEANS THE NUMBER OF EXAMPLES, CD MEANS THE CLASS DISTRIBUTION, AND IR MEANS THE IMBALANCE RATIO.

NAME	#F	#E	CD	IR
yeast1	10	1484	429/1055	2.46
vehicle2	18	846	218/628	2.88
SPECT_F	44	267	55/212	3.85
segment0	23	2308	329/1979	6.02
glass6	9	214	29/185	6.38
yeast3	10	1484	163/1321	8.1
page_blocks0	10	5472	559/4913	8.79
ecoli-0-3-4_vs_5	7	200	20/180	9
yeast-2_vs_4	8	514	51/463	9.08
ecoli-0-6-7_vs_3-5	7	222	22/200	9.09
yeast-0-2-5-6_vs_3-7-8-9	10	1004	99/905	9.14
ecoli-0-4-6_vs_5	6	203	20/183	9.15
CM1	23	498	49/449	9.16
ecoli-0-1_vs_2-3-5	7	244	24/220	9.17
ecoli-0-3-4-6_vs_5	7	205	20/185	9.25
ecoli-0-3-4-7_vs_5-6	7	257	25/232	9.28
PC1	21	1109	77/1032	13.4
glass4	9	214	13/201	15.46
ecoli4	7	336	20/316	15.8
page-blocks-1-3_vs_4	10	472	28/444	15.86
glass-0-1-6_vs_5	9	184	9/175	19.44
flare-F	11	1066	43/1023	23.79
yeast6	10	1484	35/1449	41.4
winequality-red-8_vs_6-7	11	855	18/837	46.5
kddcup-land_vs_portsweep	40	1061	21/1040	49.52
winequality-white-3-9_vs_5	11	1482	25/1457	58.28
winequality-red-3_vs_5	11	691	10/681	68.1
kddcup-buffer_overflow_vs_back	31	2233	30/2203	73.43
kr-vs-k-zero_vs_fifteen	6	2193	27/2166	80.22
kddcup-rootkit-imap_vs_back	47	2225	22/2203	100.14

A. Results and Analysis

Table II reports the average AUC values over 30 runs of different methods on different datasets. We omit the standard deviations for succinct presentation of results. In Table II, the best performance on each dataset is highlighted using bold face. The last but one row in Table II shows the mean value of each method over 30 datasets, and the last row shows the mean rank of each method over 30 datasets where the lower rank the better. As can be seen from Table II, the proposed MOSS produces a mean value of 87.07, which is 3.72% higher than the second highest value (83.95 yielded by the BSMOTE1). In addition, the MOSS achieved the best results in 19 out of 30 datasets. Compared with other methods, the MOSS outperforms the NONE, ROS, SMOTE, BSMOTE1, BSMOTE2, MWMOTE, SLSMOTE, and KSMOTE in 29, 24, 29, 28, 28, 29, 28, and 29 out of 30 datasets, respectively, which confirms the superior performance of the MOSS to the compared methods.

The MOSS yields the lowest mean rank, i.e., the best rank among all methods. Interestingly, the more advanced oversampling methods (i.e., BSMOTE1, BSMOTE2, MWMOTE, SLSMOTE, KSMOTE) do not show significant improvement against the original SMOTE and all of them yield worse mean rank than the SMOTE. These methods attempt to generate new examples in safe regions via different mechanisms, but they all ignore the impact on the datasets brought by the class imbalance, which may be one of the reasons why they yield worse results than the MOSS.

B. Non-parametric Statistical Analysis

To show the differences among different methods, a one-sided Wilcoxon signed-ranks test [23] with significance level of 0.05 is employed as suggested in [24]. If a p-value computed by the test is lower than the significance level (0.05), it indicates that the performances between compared methods are significantly different. The Friedman's test with a post-hoc Hochberg's test [25] will be also applied to compare the proposed method with other methods over multiple datasets. Similarly, a p-value lower than 0.05 indicates the significant statistical difference.

Table III reports the results obtained by the Wilcoxon test for the proposed method. Because the MOSS yields the best average results among all methods, a significant difference means that the MOSS significantly outperforms the compared method. As can be seen from Table III, all p-values computed by the test are much lower than 0.05. This means the MOSS significantly outperforms all compared methods with 95% confidence.

Table IV reports the results obtained by the Friedman's test and the post-hoc Hochberg's test. The MOSS yields the lowest mean rank so it is used as the control method. The p-value computed by the Friedman's test is 0, so there are significant differences among the performances of different methods. Then the Hochberg's test is performed to find out which pair of methods produce significantly different results. The $p_{Hochberg}$s in Table IV are all lower than 0.05, which means the MOSS significantly outperforms all compared methods over 30 datasets.

IV. CONCLUSIONS AND FUTURE WORKS

In this work, we propose a novel oversampling method, the Minority Oversampling using Sensitivity (MOSS), for imbalance classification problems. The MOSS selects the candidate minority examples for oversampling by considering the sensitivity to the class imbalance, i.e., the impact of class imbalance on each minority example. The example that is more

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE II

AVERAGE AUC VALUES (IN PERCENTAGE) OVER 30 RUNS. THE MOSS PRODUCES THE BEST AVERAGE AUC RESULTS IN 19 OUT OF 30 DATASETS. WE OMIT THE STANDARD DEVIATIONS FOR SUCCINCT PRESENTATION OF RESULTS.

NAME	NONE	ROS	SMOTE	BSMOTE1	BSMOTE2	MWMOTE	SLSMOTE	KSMOTE	MOSS
yeast1	79.28	**84.07**	80.97	79.08	79.31	80.52	80.02	79.65	81.85
vehicle2	96.84	**97.93**	95.78	96.94	97.1	96.11	96.22	96.46	97.12
SPECT_F	**98.5**	96.47	94.65	94.12	94.64	92.19	97.53	**98.5**	98.23
segment0	83.97	82.58	84.33	85.4	83.61	82.46	86.68	85.1	**86.76**
glass6	84.59	81.31	85.17	81.8	78.42	84.99	85.5	83.23	**86.55**
yeast3	84.78	83.42	**87.75**	85.62	85.53	85.38	84.69	85.32	86.86
page_blocks0	85.35	85.04	87.63	88.49	87.68	87.18	88.86	84.69	**89.04**
ecoli-034_vs_5	84.02	85.3	86.85	89.2	88.16	85.04	88.95	87.46	**89.23**
yeast-2_vs_4	83.83	82.46	84.69	83.73	78.97	84.49	**85.4**	83.14	85.17
ecoli-067_vs_35	84.41	84.11	86.63	85.47	85.25	83.81	**87.92**	85.51	86.88
yeast-0256_vs_3789	81.41	79.81	82.43	81.55	79.47	83.02	82.43	81.78	**83.44**
ecoli-046_vs_5	88.27	86.6	86.95	86.5	85.63	83.79	88.28	84.12	**89.78**
CM1	81.14	81.75	73.13	73.7	71.5	67.53	75.26	79.28	**81.29**
ecoli-01_vs_235	61.48	64.48	65.18	65.92	**69.28**	61.29	61.28	61.48	67.09
ecoli-0346_vs_5	90.69	85.18	85.27	87.06	87.93	84.83	90.69	87.64	**91.09**
ecoli-0347_vs_56	88.79	82.43	90.22	90.22	88.83	90.22	88.79	88.08	**91.39**
PC1	90.02	89.17	90.23	**91.25**	88.73	89.38	90.62	88.62	90.81
glass4	60.25	58.25	81.74	78.84	77.89	56.59	60.25	60.25	**98.67**
ecoli4	55.36	56.1	82.76	91	83.82	55.54	55.36	55.36	**96.4**
page-blocks-13_vs_4	65.92	63.16	59.71	57.62	52.79	57.54	65.92	65.75	**66.06**
glass-016_vs_5	91	83.37	91	91	**100**	91	91	91	**100**
flare-F	76.08	50	71.8	74.17	70.97	70.06	60.29	76.26	**76.81**
yeast6	91.45	90.95	90.35	87.54	89.53	90.97	91.45	91.62	**92.06**
winequality-red-8_vs_67	74.28	72.97	74.9	75.51	74.63	74.71	73.82	74.21	**76.12**
kddcup-land_vs_portsweep	89.35	88.95	91.53	90.74	89.87	90.75	89.35	89.88	**92.58**
winequality-white-39_vs_5	90.86	94.32	94.35	95.9	**96.14**	93.13	92.78	91.74	94.45
winequality-red-3_vs_5	98.86	98.62	99.05	98.84	98.81	99.06	98.76	98.9	**99.29**
kddcup-buffer_overflow_vs_back	72.31	**73.42**	72.22	72.33	71.27	73.55	70.94	72.83	72.86
kr-vs-k-zero_vs_fifteen	71.31	**74.1**	72.08	71.38	70.02	73.37	70.29	71.36	73.73
kddcup-rootkit-imap_vs_back	87.49	**91.73**	87.89	87.42	87.85	89.5	88.37	87.12	90.63
Mean Value	82.4	80.93	83.91	83.95	83.12	81.27	82.26	82.21	**87.07**
Mean Rank	5.67	6.13	4.48	4.78	5.85	5.82	5.03	5.68	**1.55**

TABLE III

RESULTS OBTAINED BY THE WILCOXON TEST FOR ALGORITHM MOSS

Compared Methods	P-value
NONE	9.314E-9
ROS	5.974E-6
SMOTE	4.656E-8
BSMOTE1	3.856E-7
BSMOTE2	1.382E-6
MWMOTE	1.8626E-8
SLSMOTE	4.712E-7
KSMOTE	5.588E-9

TABLE IV

RESULTS OBTAINED BY THE FRIEDMAN'S TEST. THE P-VALUE COMPUTED BY THE FRIEDMAN'S TEST IS 0, INDICATING THE EXISTENCE OF SIGNIFICANT DIFFERENCE AMONG ALL METHODS. THE $p_{Hochberg}$ SHOWS THE P-VALUE COMPUTED BY THE POST-HOC HOCHBERG'S TEST.

Algorithm	$p_{Hochberg}$
ROS	0
BSMOTE2	0
MWMOTE	0
KSMOTE	0
NONE	0
SLSMOTE	0.000003
BSMOTE1	0.00001
SMOTE	0.000033

sensitive to the class imbalance is assigned higher probability so that it has higher chance to be selected as the candidate. In this way, new examples are generated based on more informative examples. Experiments on 30 imbalance datasets are carried out to empirically demonstrate the effectiveness of the MOSS, which shows that the MOSS yields significantly better results than 8 methods (1 baseline method without any preprocessing procedure and 7 oversampling methods). In conclusion, the MOSS is effective in handling imbalance classification problems compared with existing oversampling methods.

The MOSS is only evaluated against oversampling methods, other types of approaches like cost-sensitive methods and algorithm-level methods can be applied to better evaluate the performance of the MOSS. In addition, oversampling is considered in this work, while some works claim that under-sampling is preferred to handle imbalance problems in certain situations [26]. It would be interesting to extend the MOSS to undersampling setting. Combining the idea of sensitivity and ensemble learning is also one of our important future works. For comprehensive analysis of the proposed idea, we would also like to evaluate the method on several real-world applications, for example fraud detection and diagnosis of rare diseases, in terms of AUC as well as other metrics including

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

F-measure, G-mean, and Precision-Recall Curve.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[2] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys*, vol. 49, no. 2, 2016, article 31.

[3] W. W. Y. Ng, J. Zhang, C. S. Lai, W. Pedrycz, L. L. Lai, and X. Wang, "Cost-sensitive weighting and imbalance-reversed bagging for streaming imbalanced and concept drifting in electricity pricing classification," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1588–1597, March 2019.

[4] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, July 2012.

[5] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, "Dual autoencoders features for imbalance classification problem." *Pattern Recognition*, vol. 60, pp. 875–889, 2016.

[6] C. S. Lai, Y. Tao, F. Xu, W. W. Ng, Y. Jia, H. Yuan, C. Huang, L. L. Lai, Z. Xu, and G. Locatelli, "A robust correlation analysis framework for imbalanced and dichotomous data with uncertainty," *Information Sciences*, vol. 470, pp. 58–77, 2019.

[7] J. Zhang, T. Wang, W. W. Y. Ng, S. Zhang, and C. D. Nugent, "Undersampling near decision boundary for imbalance problems," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, July 2019, pp. 1–8.

[8] W. W. Y. Ng, J. Hu, D. S. Yeung, S. Yin, and F. Roli, "Diversified sensitivity-based undersampling for imbalance classification problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2402–2412, Nov 2015.

[9] G. Douzas and F. Bacao, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Information Sciences*, vol. 465, pp. 1–20, 2018.

[10] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?" *DMIN*, pp. 35–41, 2007.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[12] J. Zhang and W. Ng, "Stochastic sensitivity measure-based noise filtering and oversampling method for imbalanced classification problems," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 403–408.

[13] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," *international conference on intelligent computing*, pp. 878–887, 2005.

[14] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *PAKDD '09 Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2009, pp. 475–482.

[15] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[16] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote–majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.

[17] R. F. de Morais and G. C. Vasconcelos, "Boosting the performance of over-sampling algorithms through under-sampling the minority class," *Neurocomputing*, 2019.

[18] W. A. Rivera, "Noise reduction a priori synthetic over-sampling for class imbalanced data sets," *Information Sciences*, vol. 408, pp. 146–161, 2017.

[19] X. Zhang, D. Ma, L. Gan, S. Jiang, and G. Agam, "Cgmos: Certainty guided minority oversampling," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1623–1631.

[20] H. Zhang and M. Li, "Rwo-sampling: A random walk over-sampling approach to imbalanced data classification," *Information Fusion*, vol. 20, pp. 99 – 116, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253514000025

[21] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz, "Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance," in *2018 IEEE International Conference on Data Mining (ICDM)*, Nov 2018, pp. 447–456.

[22] I. Triguero, S. González, J. M. Moyano, S. García, J. Alcalá-Fdez, J. Luengo, A. Fernández, M. J. del Jesús, L. Sánchez, and F. Herrera, "Keel 3.0: An open source software for multi-stage analysis in data mining," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 1238–1249, 2017.

[23] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 196–202, 1945.

[24] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[25] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[26] A. D. Pozzolo, O. Caelen, and G. Bontempi, "When is undersampling effective in unbalanced classification tasks," *European Conference on Machine Learning*, vol. 9284, pp. 200–215, 2015.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE