Adaptive Generation-based Approaches of Oversampling using Different Sets of Base and Nearest Neighbor's Instances

Hatem S Y Nabus[1], Aida Ali[2], Shafaatunnur Hassan[3]
Siti Mariyam Shamsuddin[4], Ismail B Mustapha[5]
Department of Computer Science, School of Computing
Faculty of Engineering, University Technology Malaysia
(UTM), Johor, Malaysia

Faisal Saeed[6]
Department of Computing and Data Science
School of Computing and Digital Technology
Birmingham City University
Birmingham, UK

Abstract—**Standard classification algorithms often face a challenge of learning from imbalanced datasets. While several approaches have been employed in addressing this problem, methods that involve oversampling of minority samples remain more widely used in comparison to algorithmic modifications. Most variants of oversampling are derived from Synthetic Minority Oversampling Technique (SMOTE), which involves generation of synthetic minority samples along a point in the feature space between two minority class instances. The main reasons these variants produce different results lies in (1) the samples they use as initial selection / base samples and the nearest neighbors. (2) Variation in how they handle minority noises. Therefore, this paper presented different combinations of base and nearest neighbor's samples which never used before to monitor their effect in comparison to the standard oversampling techniques. Six methods; three combinations of Only Danger Oversampling (ODO) techniques, and three combinations of Danger Noise Oversampling (DNO) techniques are proposed. The ODO's and DNO's methods use different groups of samples as base and nearest neighbors. While the three ODO's methods do not consider the minority noises, the three DNO's include the minority noises in both the base and neighbor samples. The performances of the proposed methods are compared to that of several standard oversampling algorithms. We present experimental results demonstrating a significant improvement in the recall metric.**

Keywords—Class imbalance; nearest neighbors; base samples; initial selection; SMOTE

I. Introduction

One of the most challenging machine learning problems to both the academia and industry in the last couple of decades is one associated with learning from data that is unbalanced [1]. This problem is known to arise in both binary and multiclass classification tasks when data instances from one class, known as the majority class occur more frequently than instances of other classes, known as the minority classes [2]. This obvious disproportion in the distribution of data instances across classes leans the classifier towards significant bias to the majority class which in turn results in the misclassification of instances of other classes [3]. What makes the class imbalance problem more interesting is the fact that the minority class is often the class of interest in most real-life application domain, thus, the cost of misclassifying the minority class is often higher than that of the majority class [4, 5]. For instance, given a machine learning fraud detection system, legitimate transactions occur more often than fraudulent ones, but the cost of misclassifying a fraudulent transaction as legitimate is greater than the opposite. Therefore, approaches to addressing class imbalance problem are aimed at increasing the accuracy and sensitivity of the classifier to the minority class.

The approaches to dealing with class imbalance problem can broadly be grouped into two categories [6]. The first category entails algorithmic creation/modification to improve learning of the minority class samples. The second category of approaches is the most popularly used category, data level methods, which resamples the data distribution to ensure balanced data distribution across the respective classes via oversampling, under-sampling or their hybrid combination.

This paper focuses on oversampling methods that involve the generation of synthetic data samples to augment the minority class. A leading oversampling method that serves as the basis for most of the recent oversampling methods is the Synthetic Minority Over-sampling Technique (SMOTE) algorithm [2]. SMOTE basically generates artificial samples along the length of the line joining neighboring minority class samples.

SMOTE has also inspired several approaches to counter the issue of class imbalance. It is standard benchmark for learning from imbalanced data [7]. Based on SMOTE, several techniques have been proposed in the literature, and these techniques have been categorized according to some properties include: (1) initial selection of instances to be oversampled (technically called base samples), (2) integration with Under-sampling as step in the technique, (3) type of interpolation, (4) operation with dimensionality changes, (5) adaptive generation of synthetic examples, (6) possibility of relabeling and (7) filtering of noisy generated instances.

Each SMOTE-based extension might have different properties from the aforementioned aspects. However, a large number of them use the three common aspects include: initial selection, type of interpolation (the common type is 'range restricted'), and the adaptive generation of the new samples. This study, therefore, focuses on those three properties.

The most common standard technique that utilizes initial selection and the 'range restricted' interpolation aspects is SMOTE_BORDERLINE [8]. This research, thus, started with adopting the same initial selection of instances to be oversampled in SMOTE-BORDERLINE. The common standard technique that uses adaptive generation of synthetic examples is ADASYN [9], and this is also adopted in this study to be used in our proposed techniques. The minority classes have been classified into three different groups namely safe, danger, and noise; according to its level of difficulty [8, 10-12].

Consequently, six new oversampling techniques, namely, ODO1, ODO2, ODO3, DNO1, DNO2, and DNO3 are proposed. For the ODO techniques, only the borderline examples (Danger group) of the minority class are over-sampled, while in case of the DNO techniques, both the minority danger and minority noise examples are oversampled. The main difference between the three ODO methods lies in the criteria for choosing the nearest neighbors (NN) group. In ODO1, the NN is the minority class except the minority noises, while in ODO2; the NN is the same as the base example which are the borderline examples (minority danger). In ODO3, the NN group includes the whole classes except the minority noises.

Similarly, the main difference between the three DNO methods is the criteria for choosing the nearest neighbors. In DNO1, the NN is the minority class, while in DNO2, the NN are the same as the base examples which consist of the Danger and Noise examples. Lastly, in DNO3, the NN group consists of the whole classes (minority and majority).

Table I shows how each of the proposed methods differs from the standard techniques (SMOTE, Borderline1, Borderline2, and ADASYN). Moreover, in this study, three aspects are added for more clarification about the methods and they are: (1) Nearest Neighbor group, (2) 'how to choose from NN group' and (3) 'noise considered?'

Hence, the major contribution of this study includes the implementation of the proposed methods as well as a tabular overview showing the differences between the methods in details and more clarifications, and this includes the initial selection / base samples used, the NN groups, the method of NN selection, type of interpolation, adaptive generation, and the representation of the minority noises (noises considered?) as shown in Table I. The proposed oversampling techniques were experimentally analyzed using four classification algorithms and evaluation metrics across 15 publicly available datasets from Machine Learning Repositories. The performances of the proposed methods are compared to SMOTE, Borderline SMOTE and ADASYN oversampling methods. In addition, statistical analysis was also carried out using Friedman aligned and Holm's tests.

The organization of this article is as follows. An overview of pertinent studies and oversampling methods is provided in Section II while the procedure of the proposed methods is listed in Section III followed by the experimental design in Section IV. The experimental results and conclusion are respectively presented in Sections V and VI.

II. Related Work

Given that this study focuses on oversampling through synthetic data generation which is a data level approach, a short review of related studies is presented here in this regard. References [7, 13, 14] are important articles for an in-depth review of imbalance resolution approaches. The most basic form of oversampling is known as Random Oversampling which involves random sampling of minority class samples with replacement till it matches the size of the majority class samples. A major drawback of this approach is high likelihood of overfitting that results from the exposure of the classifier to the same information.

An oversampling approach that sidesteps the challenges associated with basic random oversampling is SMOTE which involves synthetic data generation along the length of the line joining neighboring minority class samples. SMOTE generates synthetic samples for any minority class including minority noises which also participate as nearest neighbors. However, when the separation between majority and minority class clusters is not clear, noisy samples may be generated [2]. On the other hand, borderline-SMOTE methods [8] intend to prevent producing noisy samples by detecting the boundary instances between the majority and minority classes, which are then utilized to identify useful informative minority class samples. Although both SMOTE-Borderline1 and SMOTE-Borderline 2 do not generate any sample for minority noises, dealing with those noises as nearest neighbors may generate new samples located near the noises or overlap with them. The study in [9] aims to distribute the new synthetic samples according to the level of difficulties by making the most difficult samples have more new samples. However, this approach results in that minority noises will have the big portion of the new synthetic samples.

From the afore-highlighted, it is obvious that the methods vary in how they deal with the base and nearest neighbor's samples. Similarly, some of them give the minority noises the advantage of being more represented in the new samples while others ignore them completely. However, the use of other different groups is still lacking, therefore, using different sample groups of the base and nearest neighbors are needed.

III. Proposed Methods' Procedure

Suppose that the whole training set is X, the minority class is P and the majority class is N, and P={p_1,p_2,…,p_{num}}, N = (n_1,n_2,…,n_{num}) Where p_{num} and n_{num} are the number of minority and majority examples. The detailed procedure of ODO1 explained in Fig. 1.

The difference between ODO1, ODO2, and ODO3 is the NN groups as we mentioned above. Additionally, the difference between ODO's techniques and DNO's techniques is that, in DNO's methods, minority noises are added to both base samples and NN samples as declared in Table I. Further, In situations where the NN is from the majority class, a random value between 0 and 0.5 will be multiplied by the difference between the base example and its nearest negative example as in SMOTE_Borderline2 [8].

TABLE I. DIFFERENCES BETWEEN THE OVERSAMPLING METHODS

Method	**Initial Selection/ Base samples**	**NN group**	**How to choose from neighbors**	**Type of interpolation**	**Adaptive generation**	**Noise considered?**
SMOTE	Any from minority	The 5 NN all minority	Randomly	On the line between base and NN New sample= base + (rand(0,1)*diff)	-	Base ➔yes NN ➔yes
Borderline1	Minority_Danger (3,4)	the 5 NN (minority)	Randomly	New sample= base + (rand(0,1)*diff)	-	Base ➔NO NN ➔yes
Borderline2	Minority_Danger (3,4)	the 5 NN (minority + majority)	Randomly	*(range restricted)* *If NN is minority* New sample= base + (rand(0,1)*diff) *If NN is majority* New sample= base + (rand(0,0.5)*diff)	-	Base ➔No NN ➔yes
ADASYN	Minority (1,2,3,4,5)	the 5 NN all minority	Randomly	Weighted distribution New sample= base + (rand(0,1)*diff)	Weighted distribution	Base ➔yes NN ➔yes
ODO1	Minority_Danger (3,4)	the 5 NN (minority-noise)	Randomly	New sample= base + (rand(0,1)*diff)	Weighted distribution	Base ➔No NN ➔No
ODO2	Minority_Danger (3,4)	the 5 NN (minority Danger)	Randomly	New sample= base + (rand(0,1)*diff)	Weighted distribution	Base ➔No NN ➔No
ODO3	Minority_Danger (3,4)	the 5 NN (minority-noise) + majority	Randomly	*(range restricted)* *If NN is minority* New sample= base + (rand(0,1)*diff) *If NN is majority* New sample= base + (rand(0,0.5)*diff)	Weighted distribution	Base ➔No NN ➔No
DNO1	Minority_Danger and Noise (3,4,5)	the 5 NN (minority)	Randomly	New sample= base + (rand(0,1)*diff)	Weighted distribution	Base ➔Yes NN ➔Yes
DNO2	Minority_Danger and Noise (3,4,5)	the 5 NN (minority Danger and Noise)	Randomly	New sample= base + (rand(0,1)*diff)	Weighted distribution	Base ➔Yes NN ➔Yes
DNO3	Minority_Danger and Noise (3,4,5)	the 5 NN (minority + majority)	Randomly	*(range restricted)* *If NN is minority* New sample= base + (rand(0,1)*diff) *If NN is majority* New sample= base + (rand(0,0.5)*diff)	Weighted distribution	Base ➔Yes NN ➔Yes

IV. EXPERIMENTAL DESIGN

The performance of the proposed methods is evaluated using 15 benchmark imbalanced datasets of varying imbalance rations (IR) from the Machine Learning Repositories (UCI, Kaggle, Keel, Datahub) and this is a common practice in class imbalance learning. Table II shows a summary of the 15 datasets. The performances of the proposed oversampling techniques were evaluated and compared with SMOTE, SMOTE_Borderline1, SMOTE_Borderline2, and ADASYN. Since accuracy has been shown in representative works as an insufficient evaluation metric for imbalanced datasets, Recall, and F1-measure are employed in this study. Additionally, the four classifiers considered for evaluation in this study are Decision Trees (DT) [15], Logistic Regression (LR) [16], RandomForest (RF) [17] and Support Vector Machine (SVM) [18].

For each combination of dataset, classifier and evaluation metric, an aligned ranking score is used to rank each oversampling method including the baseline. In addition to the 10 oversampling algorithms considered in this study, the performance of the classifiers on the original dataset without oversampling is also used as the baseline.

Thus, the best performing method has the biggest ranking score while the smallest ranking score indicates the worst performing method. Additionally, two statistical tests, Friedman aligned ranks and Holm, were also used to further establish the significance of our findings. While the Friedman aligned rank's test recognizes the difference in outcomes obtained from many attempts when the normality assumption may not hold, the Holm's test is a nonparametric t-test used to establish whether a control method outperforms comparative methods.

```
Only Danger Oversampling (ODO1) algorithm:
Step 1. Extract the X_min as the minority samples.
Step 2. Define m_min and m_maj as the number of minority class examples
  and the number of majority class examples, respectively. Therefore,
  $m\_min \leq m\_maj$ and $m\_min + m\_maj = X$.
Step 3. Calculate the degree of class imbalance:
  $d = m\_min/m\_maj$, where $d \in (0, 1]$.
Step 4. Calculate the number of synthetic data examples that need to be
  generated for the minority class:
  $G = (m\_maj - m\_min) \times \beta$
  Where $\beta \in [0, 1]$ is a parameter used to specify the desired balance
  level after generation of the synthetic data. $\beta = 1$ means a fully
  balanced data set is created after the generalization process.
Step 5. Determine the three Minority groups (Noise, Danger, Safe)
Step 6. Now, we find the KNN (K=5) for each example $x_i$ in the danger group
  in the whole training dataset X.
Step 7. calculate the ratio $r_i$ defined as:
  $r_i = \Delta_i/K$, $i = 1, ..., X\_d$. X_d is the number of examples in
  Danger group.
  where $\Delta_i$ is the number of examples in the K nearest neighbors of $x_i$
  that belong to the majority class, therefore $r_i \in [0, 1]$.
Step 8. Normalize $r_i$ according to $r_i^{\wedge} = r_i / \sum_{i=1}^{X\_d} ri$, so that $r_i^{\wedge}$ is a
  density distribution ($\Sigma\, r_i^{\wedge} = 1$)
Step 9. Calculate the number of synthetic data examples that need to be
  generated for each minority_danger example $x_i$:
  $gi = \hat{}r_i \times G$ where G is the total number of synthetic data examples
  that need to be generated for the minority_danger class.
Step 10. Determine the minority group without noises
  X_min_no_noise = (X_min)-(Noise)
Step 11. find the KNN (K=5) for each example $x_i$ in the danger group in the
  X_min_no_noise. In this step, we guarantee that we don't use any
  minority noise as a NN.
Step 12. For each minority_danger class data example $x_i$, generate $g_i$ synthetic
  data examples according to the following steps:

  Do the Loop from 1 to $g_i$:
    (i) Choose a minority data example ($x_{zi}$)
    randomly from the nearest neighbors for data $x_i$.
    (ii) Generate the synthetic data example:
      $s_i = x_i + (x_{zi} - x_i) \times \lambda$
  where $(x_{zi} - x_i)$ is the difference vector in n dimensional spaces,
  and $\lambda$ is a random number: $\lambda \in [0, 1]$.
  End Loop
```

Fig. 1. ODO1 Algorithm.

To evaluate the performance of the classifiers on each dataset and method, a stratified k-fold cross validation experimental setup was applied with $k = 5$. Each oversampling method is performed on only the training portion dataset during k-fold CV and tested on their respective test folds [19]. The presented results represent the means validation performance. When the data you are using to train a machine learning algorithm happens to have the information you are trying to predict that is called Data leakage [20]. Therefore, to prevent leaking the data, the data preparation was performed within cross validation folds.

The hyperparameter tuning of the classifiers was done on the original datasets with no oversampling (baseline) and then the obtained optimal parameters are used when applying the oversampling methods to have fairness with all techniques, while the various oversampling algorithms' hyperparameters were tuned using the default values, except an important parameter in this study that is k nearest neighbor which must be equal to 5 in all oversampling techniques since the proposed methods are built on this number of nearest neighbors. The classifiers and standard oversampling algorithms were implemented using Python modules Scikit-Learn [21] and Imbalanced-Learn [22].

V. Experimental Result and Discussion

At first, in the favor of explaining more about the nature of work of the oversampling standard techniques and the proposed methods, this research visualized their generating of the new samples using a synthetic dataset as you can see in Fig. 2, in addition to the detail description in Table I.

On your imbalanced classification problem, you can choose to use precision or recall. The number of false positive errors will be reduced if precision is maximized, while the number of false negative errors will be reduced if recall is maximized. As a result, precision may be a better fit for classification problems where false positives are a concern. Alternately, recall may be more appropriate on classification problems when false negatives are more important [23]. With dataset such as Breast Cancer, the concern is the recall, therefore, try to reduce the False Negative (FN) as possible as can, while with dataset such as Spam mails dataset, the task will be more focus on precision since it is needed to reduce the False Positive (FP) the most. This study tries to improve the recall without hurting the precision too much.

For each combination of classifier and evaluation metric, the mean rankings of the oversampling approaches over data sets are shown in Table III. The Friedman aligned test is used to statistically confirm the conclusion and the results are shown in Table IV. As a result, the null hypothesis is rejected at a significance level of 0.05., i.e., the oversampling methods do not perform equally in mean rankings for all evaluation metrics.

Table V shows that our proposed method DNO3 is always the first or the second winner with all classifiers when the metric measure is the recall, therefore, DNO3 oversampler is used as a control method in the Holm's test to see if DNO3 result is a significant or not. The adjusted p-values are shown in Table VI.

DNO3 ranked as the best method among all techniques regarding the recall results, and then DNO1 coming as the second. By looking at the differences between the DNO1 and DNO3, the only difference is the NN samples. DNO3 will deal with all classes in the NN whether they are minority or majority class, while DNO1 will only consider the minority class in the NN. This shows the importance of considering both minority and majority classes in the nearest neighbors.

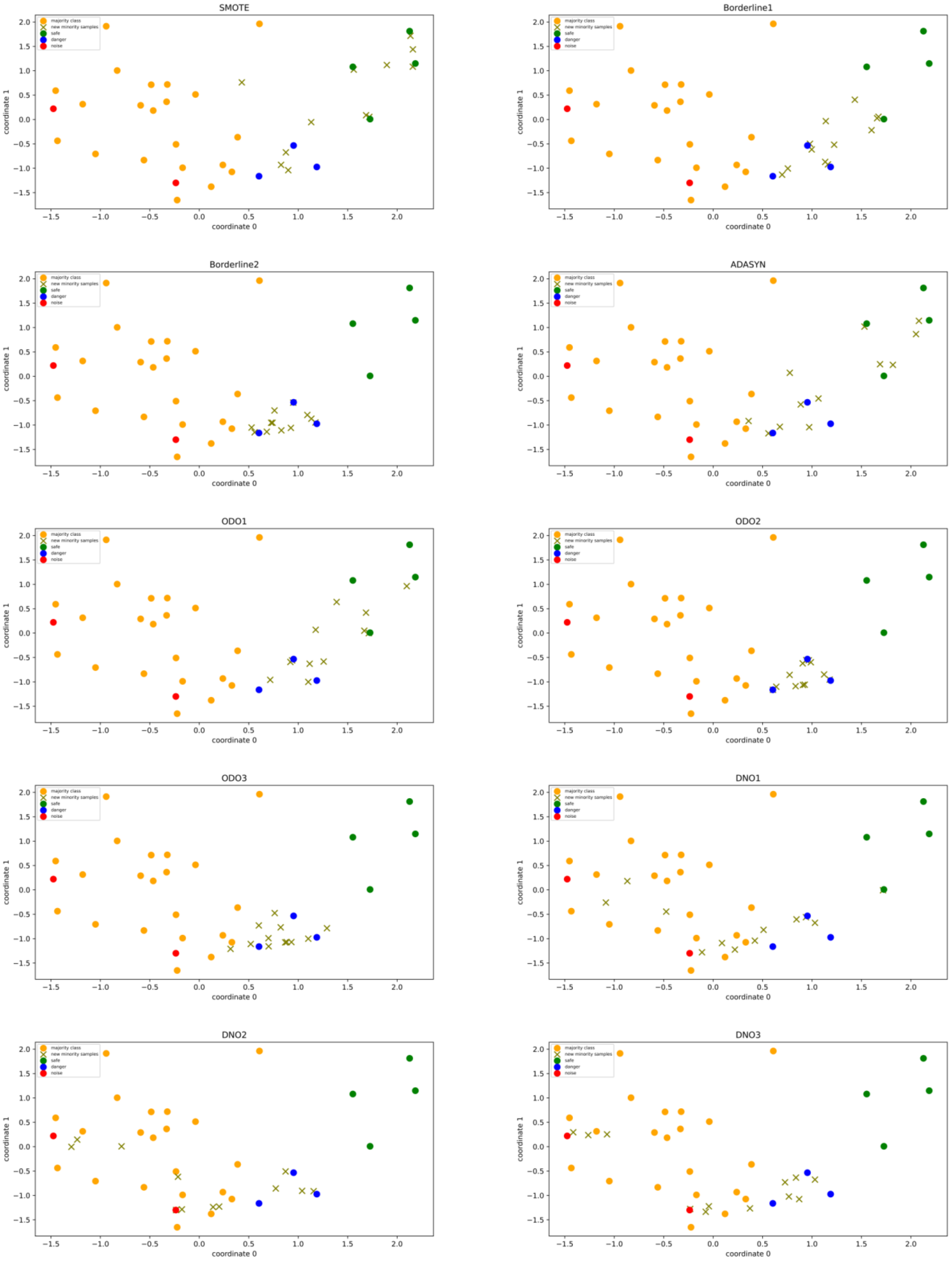

Fig. 2. The Distribution of the New Synthetic Samples using different Oversampling Methods.

TABLE II. DATASETS' DESCRIPTION

Name	**Instances**	**Attributes**	**IR**
abalone-20_vs_8-9-10	1916	8	72.69
Adult	48842	7	3.18
Covertype	38501	55	13.02
pima-indians-diabetes	768	9	1.87
glass4	214	9	15.47
Ionosphere	351	35	1.79
Mammography	11183	7	42.01
oil-spill	937	50	21.85
page-blocks0	5472	10	8.79
Phoneme	5404	6	2.41
poker-8_vs_6	1477	10	85.88
poker-8-9_vs_6	1485	10	58.4
Satimage	6435	37	9.28
Vehicle Silhouttes_0	846	19	3.25
yeast5	1484	8	32.73

TABLE III. RESULTS FOR MEAN RANKING OF THE OVERSAMPLING METHODS ACROSS THE DATASETS. THE BOLD HIGHLIGHTS THE BEST PERFORMING METHOD

Metric	**Baseline**	**SMOTE**	**BL1**	**BL2**	**ADASYN**	**ODO1**	**ODO2**	**ODO3**	**DNO1**	**DNO2**	**DNO3**
Algorithm: DT											
Recall	35.97	89.33	73.37	103.83	87.30	67.73	53.63	65.07	**121.30**	99.67	**115.80**
F1	**124.70**	101.80	96.00	63.43	74.97	**109.00**	92.10	66.03	69.10	61.80	54.07
Algorithm: LR											
Recall	9.80	80.53	80.97	**106.50**	90.40	80.00	58.53	95.83	101.50	96.50	**112.43**
F1	77.73	**114.80**	86.00	71.93	76.17	**111.93**	99.80	81.77	73.63	61.07	58.17
Algorithm: RF											
Recall	18.27	74.97	77.97	96.70	107.47	62.47	66.53	68.83	**119.07**	105.50	**115.23**
F1	105.53	101.47	**108.43**	76.43	71.20	**108.20**	94.97	55.53	75.17	70.67	45.40
Algorithm: SVM											
Recall	19.50	74.70	73.00	82.27	96.73	57.57	77.37	95.87	**106.93**	106.70	**122.37**
F1	86.50	100.67	85.43	54.43	76.93	**101.60**	**105.63**	88.77	81.87	76.47	54.70

TABLE IV. RESULTS FOR FRIEDMAN'S TEST

Metric	P value
Algorithm: DT	
Recall	0.00000
F1	0.00042
Algorithm: LR	
Recall	0.00000
F1	0.00799
Algorithm: RF	
Recall	0.00000
F1	0.00078
Algorithm: SVM	
Recall	0.00000
F1	0.04599

TABLE V. THE WINNING METHODS AMONG ALL METRICS AND CLASSIFIERS

	Recall	F1
DT	**DNO1**	None
LR	**DNO3**	SMOTE
RF	**DNO1**	BL1
SVM	**DNO3**	**ODO2**

TABLE VI. RESULTS FOR HOLMS' TEST. THE BOLD HIGHLIGHTS STATISTICAL SIGNIFICANCE (RECALL – CONTROL METHOD = DNO3)

RECALL							
DT	**adjusted p_values**	**LR**	**adjusted p_values**	**RF**	**adjusted p_values**	**SVM**	**adjusted p_values**
Baseline	**0.00005**	Baseline	**0.00000**	Baseline	**0.00000**	Baseline	**0.00000**
ODO2	**0.00329**	ODO2	**0.01803**	ODO1	**0.02240**	ODO1	**0.00183**
ODO3	**0.02908**	ODO1	0.50404	ODO2	**0.04196**	BL1	**0.03726**
ODO1	**0.04105**	SMOTE	0.50404	ODO3	0.05474	SMOTE	**0.04402**
BL1	0.09000	BL1	0.50404	SMOTE	0.12593	ODO2	0.05936
ADASYN	0.51162	ADASYN	1.00000	BL1	0.16331	BL2	0.10763
SMOTE	0.51693	ODO3	1.00000	BL2	1.00000	ODO3	0.51500
DNO2	1.00000	DNO2	1.00000	DNO2	1.00000	ADASYN	0.51500
BL2	1.00000	DNO1	1.00000	ADASYN	1.00000	DNO2	0.73831
DNO1	1.00000	BL2	1.00000	DNO1	1.00000	DNO1	0.73831

Among the common standard techniques (SMOTE, SMOTE_BORDERLINE1 (BL1), SMOTE_BORDERLINE2 (BL2), and ADASYN), the BL2 is the best in Recall results. Comparing SMOTE_BORDERLINE2's structure with DNO3 shows the importance of considering the minority noise in the base samples since SMOTE_BORDERLINE2 is not considering that, as well as the weighted distribution of the new samples used by DNO3 that creates more new samples for the most difficult samples which is not the way used in SMOTE_BORDERLINE2.

From the above analysis this study depicts that there are three factors can affect the detection of the minority class; the first is that the minority's noises and danger samples which should be considered in the initial selection / base samples, and the second factor is that the minority noises, danger, and also the majority samples should be considered in the nearest neighbors samples, and last but not least is that the distribution of the new synthetic samples should be also weighted distributed so that the more difficult samples will be given more new synthetic samples. These factors can help reducing the false negative (FN) examples and this, in turn, increases the recall.

VI. CONCLUSION

DNO'S techniques performances were the best in Recall, and specifically DNO3 that outperformed all standard techniques in recall metric. This study shows the importance of considering minority noises and danger samples whether as base samples or nearest neighbors' group. Furthermore, the majority class samples should be under concern in the nearest neighbors' group. Finally, the weighted distribution (adaptive generation) of the new samples can help to get better Recall result. Taking everything into account, next work should consider not only the minority danger and minority noise groups, but also different groups of difficult minority samples including the minority safe samples.

REFERENCES

[1] N. Chawla, N. Japkowicz, and A. Kolcz, "Workshop learning from imbalanced data sets II," in Proc. Int'l Conf. Machine Learning, 2003.

[2] 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, pp. 321-357, 2002.

[3] D. Devi and B. Purkayastha, "Redundancy-driven modified Tomek-link based undersampling: A solution to class imbalance," Pattern Recognition Letters, vol. 93, pp. 3-12, 2017.

[4] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 1999, pp. 155-164.

[5] K. M. Ting, "An instance-weighting method to induce cost-sensitive trees," IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 3, pp. 659-665, 2002.

[6] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A Review," Int. J. Advance Soft Compu. Appl, vol. 7, no. 3, 2015.

[7] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," Journal of artificial intelligence research, vol. 61, pp. 863-905, 2018.

[8] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing, 2005: Springer, pp. 878-887.

[9] 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: IEEE, pp. 1322-1328.

[10] K. Borowska and J. Stepaniuk, "Imbalanced data classification: A novel re-sampling approach combining versatile improved SMOTE and rough

sets," in IFIP International Conference on Computer Information Systems and Industrial Management, 2016: Springer, pp. 31-42.

[11] K. Borowska and M. Topczewska, "Data preprocessing in the classification of the imbalanced data," Advances in Computer Science Research, 2014.

[12] K. Borowska and M. Topczewska, "New data level approach for imbalanced data classification improvement," in Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015, 2016: Springer, pp. 283-294.

[13] 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 T. Syst. Man Cy. C, 42, 463–484," ed, 2012.

[14] N. Chawla, "Data mining for imbalanced datasets: an overview (Periodical style)," Dept. of Computer Science and Engineering, Notre Dame Univ., US, 2005.

[15] J. R. Quinlan, Programs for machine learning. 1993.

[16] P. McCullagh, "Generalized linear models," European Journal of Operational Research, vol. 16, no. 3, pp. 285-292, 1984.

[17] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[18] C.-C. Chang, "" LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology, 2: 27: 1--27: 27, 2011," http://www. csie. ntu. edu. tw/~ cjlin/libsvm, vol. 2, 2011.

[19] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, and J. Santos, "Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [research frontier]," ieee ComputatioNal iNtelligeNCe magaziNe, vol. 13, no. 4, pp. 59-76, 2018.

[20] R. Nisbet, J. Elder, and G. D. Miner, Handbook of statistical analysis and data mining applications. Academic press, 2009.

[21] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," the Journal of machine Learning research, vol. 12, pp. 2825-2830, 2011.

[22] G. Lemaıtre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," arXiv preprint arXiv:1609.06570, 2016.

[23] J. Brownlee, Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning. Machine Learning Mastery, 2020.