Improving Classification Performance on Rare Events in Data Starved Medical Applications

I. Related Work

Need to conduct a literature survey on each problem: (1) Boosting general and SMOTE boosting, (2) SMOTE using clustering, and (3) SMOTe using GAN.

Random Oversampling (ROS) [1]: In ROS, Random minority class samples are duplicated to achieve a more balanced class distribution of samples. Often upsampling can be used for avoiding overfitting issues in random oversampling.

Synthetic Minority Oversampling Technique (SMOTE) [2]: The core idea of SMOTE is that synthetic samples are generated between neighboring minority samples using k nearest neighbor samples and linear interpolation. The steps of the SMOTE method are as follows: (1) select a sample x from the minority class and find its k nearest neighbor samples within the minority class, (2) randomly select one neighboring sample y from the k nearest neighbors found in step 1, and (3) generate a new sample x_{new} by linear interpolation between x and y, x_{new} = x + $random(0, 1) * (y - x)$.

Gaussian SMOTE (G-SMOTE) [3]: G-SMOTE uses a minority sample and one of its k nearest neighbors within the minority class to generate synthetic samples. With SMOTE, it is likely that a lot of data will lie on the same line. G-SMOTE addresses this by incorporating a Gaussian distribution to generate synthetic data points that lie away from the line, creating more diverse data, but not so far from the line that the performance is degraded.

Gamma Distribution SMOTE (Gamma-SMOTE) [4]: Gamma-SMOTE utilizes the gamma distribution to create new minority class points and produces data in a non-linear fashion, thus giving rich geometric structure. Since the Gamma distribution is asymmetric, new minority points are generated close to the existing minority data sample.

Sample Density Distribution SMOTE (SDD-SMOTE) [5]: Similar to SMOTE, SDD-SMOTE generates synthetic samples but considers the total dataset distribution and local sample density to reduce fuzzy classification boundaries and control the randomness of regular SMOTE [2].

A. Generative Adversarial Network Approach

Generative Adversarial Networks (GAN) was presented in [6] as a technique to generate synthetic data that have similar properties to the existing training data. GAN is predominantly used to generate synthetic images that are similar to real images; there are only a few instances in the literature when it has been applied to solve the CIP for numerical data, such as in SMOTified GAN technique [7]. Therefore, we investigated GAN as an oversampling technique for our numerical datasets, i.e., we added the data generated by GAN to the minority class. For our experiments, we implemented a GAN method on top of the GAN implementation developed by [8], [9] and fit it to our purposes.

II. Methods

A. SMOTEBoost and Control Coefficient

Our method uses SMOTE as its core technique. SMOTEBoost breaks the data generation process of SMOTE into several iterations. Multiple subsets of minority class instances are generated in each iteration, and SMOTEBoost chooses the best one based on classification performance and adds it to the existing dataset. SMOTEBoost adds more instances of the minority class to the original dataset at each boosting iteration, thus allowing more diverse data to be generated in the next. SMOTEBoost is an oversampling method based on the SMOTE algorithm [2] and introduces a boosting technique to improve data generation. SMOTE solves the problem that ROS may cause the classifier to overfit. However, SMOTE may not be a solution to the class imbalance problem where datasets being used are very small; this is the case for our individual patient datasets.

The advantage of SMOTEBoost is that while standard SMOTE generates data based on the original dataset minority class, SMOTEBoost adds more instances of the minority class to the original dataset at each boosting iteration. Adding and using similar data from previous iterations results in generating more diverse data. Figure 1 illustrates the flow path of SMOTEBoost and the algorithm is presented in Algorithm 1.

To further improve the SMOTE algorithm, we adopt the Control Coefficient (CC) from the SDD-SMOTE algorithm to solve the limitation of the random number value used when synthesizing a new sample [5]. Thus, we apply it to our existing SMOTEBoost algorithm to get SMOTEBoostCC. We calculate CC as follows:

1) Calculate the average Euclidean distance between all minority samples as D_{pos}.
2) Calculate the average Euclidean distance between all minority samples and majority samples as Dneg.
3) During the synthesis process of a new sample, calculate the average Euclidean distance D_1 between the selected minority and its K minority class neighbors.
4) Repeat Step (3) to calculate the average Euclidean distance D_2 for K majority class neighbors.
5) Calculate the relative distance ratio μ according to D_1, D_2, D_{neg}, and D_{pos}.

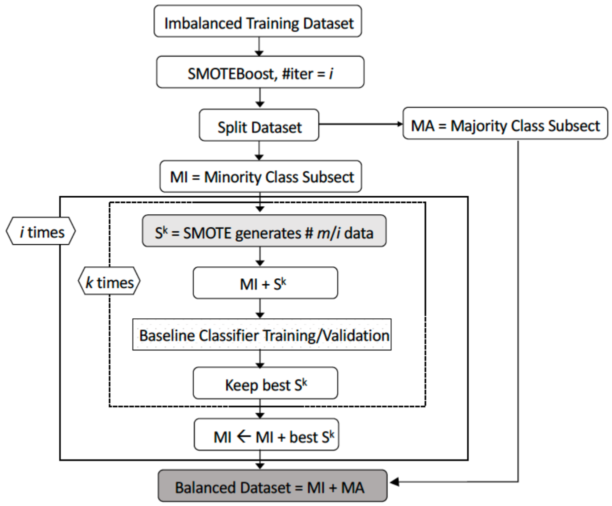


Fig. 1. Overview of boosting process

6) According to μ, calculate the value rance of the control coefficient.

B. Incremental K-means SMOTE

The basis of incremental K-Means SMOTE (iKmeans-SMOTE) isinspired by boosting techineques and K-means clustering methods.In K-means clustering, centroids are calculated which then iteratesuntil the best centroid is discovered. The letter "K" in K-meansstands for the number of clusters that the algorithm identifies fromthe data. In K-means clustering, data points are grouped into kclusters so that the total of their squared distances from the centroidis as little as it can be.

However, one disadvantage of K-means clustering is the lessunique data points within clusters result in more similar data pointswithin the same cluster. This problem would essentially affect thediversity of our balanced dataset. To solve this problem, we utilizeboosting to increase the uniqueness of data points. Similarto SMOTEBoost, iKmeans-SMOTE adds more instances of the minorityclass to the original dataset after each iteration. Addinggenerated data from previous iterations results in more diverse datapoints while keeping it within its respective area.

With the inspired methods, iKmeans-SMOTE initializes k valueand divides the minority class into k clusters. Then new points aregenerated using the average value of each cluster. Per algorithmiteration, generated data points are introduced back into the minorityclass as k-value increments by one. This process repeats untilthe minority class is approximately equal to the majority class. Details of the procedures are described in Algorithm 2.

C. Generative Adversarial Neuralnetworks (GAN)

Although GAN performed poorly in our experiments, it is a fairly recent and promising area of research, so we believe that it still has potential and have hope for future directions using

Algorithm 1 SMOTEBoost(D_{train}, k, i)

Input: D_{train} is a set of number of class-labeled training data points, k is the number of k-neighbors, i is the number of iterations
Output: A balanced training dataset augmented by synthetic data
Method:
Split D_{train} into majority class dataset (MA) and minority class dataset (MI) and record their number of instances as N_{MA} and N_{MI}, respectively.
$N_{syn} \leftarrow N_{MA} - N_{MI}$
if If $k > N_{MI}$ **then**
  $k \leftarrow N_{MI} - 1$
end if
$G \leftarrow$ Split N_{syn} into i iterations for generating synthetic data, $\lceil \frac{N_{syn}}{i} \rceil$. G is an array of storing the numbers of synthetic data that will be generated for each iteration.
repeat
  Initailization: $D' \leftarrow D_{train}$, $S_{best} \leftarrow \{\}$, $i \leftarrow 1$, $j \leftarrow 1$, $recall_{best} \leftarrow 0$
  $n \leftarrow$ the size of synthetic data for iteration i, $G[i]$
  repeat
    $s \leftarrow$ Generate n number of synthetic data points using SMOTE [2]
    $D' \leftarrow$ add s
    Conduct baseline classification training and validation using D'
    $recall \leftarrow$ Obtain recall score of the classifier
    if $recall$ ¿ $recall_{best}$ **then**
      $recall_{best} \leftarrow recall$, $S_{best} \leftarrow s$
    end if
  until j reachers k
  $D_{train} \leftarrow$ add S_{best}
  $i \leftarrow i + 1$
until All synthetic data specified in G are generated
return D_{train}, this is a balanced training data augmented by minority class synthetic data

GAN. We plan to use GAN as a pipeline for our sampling algorithms to address the small data size problem. We would first balance the data using one of our sampling algorithms. Then, we would provide the balanced data as input to GAN, and GAN would generate new data for both the minority class and the majority class, doubling the size of each class. We hope the classifiers will perform better with a larger quantity of data.

III. Experiments

Need to conduct extensive experiments on each problem: (1) Boosting general and SMOTE boosting, (2) SMOTE using clustering, and (3) SMOTe using GAN. Datasets and experimental setup can be similar. Consider including one or more small real datasets, passibly medical datasets. Performance metrics are the same as well as the presentations. I keep the

Algorithm 2 this is stakeholder and will be changed - imbalancedClassifier(D_{train}, A)

Input: D_{train} is a set of number of class-labeled training data tuples, A is a classification scheme
Output: a classifier
Method:
$D_0 \leftarrow$ tuples labled with $class_0$, $D_1 \leftarrow$ data tuples labled with $class_1$
{$class_0$: tuples with PEF value ¡= 20% quantile , $class_1$: tuples with PEF value ¿ 20% qunatile}
$n_0 \leftarrow |D_0|$, $n_1 \leftarrow |D_1|$
$D_0^{'} \leftarrow \{\}$ {$D_0^{'}$ will contain up-sampled and over-sampled tuples of $class_0$}
$D_1^{'} \leftarrow \{\}$ {$D_1^{'}$ will contain over-sampled tuples of $class_1$}
repeat
  $D_0^{'} \leftarrow$ add a data tuple t selected from D_0 using random sampling with replacement
  $m \leftarrow |D_0^{'}|$
until $m \approx n_1$ is attained
$D_0^{'} \leftarrow$ update $D_0^{'}$ with duplicates of each data tuple from D_0
$D_1^{'} \leftarrow$ duplicates of each data tuple from D_1
$D_{train}^{'} \leftarrow D_0^{'} \cup D_1^{'}$ {$D_{train}^{'}$ is an imbalanced training dataset}
classifier $C \leftarrow$ trainModel($D_{train}^{'}, A$)

experimental resuts section of our recently published paper for your reference.

A. Datasets and Experiments Setup

Our synthetic datasets vary in size (small: 200, medium: 800, large: 3,000) and the degree to which the convex hull of the points of the minority class overlaps with the convex hull of the points of the majority class. Each of the datasets have an imbalance ratio of 4 (ratio=$\frac{majority}{minority}$), and 2 features x_1 and x_2 with values between 0 and 1 from uniformly random distribution. Our real datasets include 24 nonsmoking asthma patients' physiological data and environmental exposure data collected through a case study at Soonchunhyang University Bucheon Hospital, South Korea [10]. Patient's behavioral data (house location, cooking style and income) were also obtained. Participant's exposures to environmental variables were estimated using 24-hour time window at each PEFR measurement. The experiments on the performance of classifiers that predict asthma patients' high risk zone (PEFR values in the lower 20% of a target patient's dataset) were conducted using patients' daily PEF values and their exposure variables as described in Table I. Table II outlines the characteristics of our synthetic dataset and real datasets, which are small and imbalanced.

Synthetic datasets were used to evaluate the data-level quality of ANVO, and the five SMOTE variants described in Section I. We conducted statistical analysis and data visualization on the generated data and presented the results in III-B. The effectiveness of ANVO and the SMOTE variants was tested through classification modeling where we implemented various classifiers using four conventional classification algorithms: (1) Decision Tree (DT), (2) K-Nearest Neighbors (KNN), (3) Logistic Regression (LR), and (4) Naive Bayes (NB). For each algorithm, we conducted extensive experiments for comparisons of our proposed method and the five SMOTE variants. The models were developed in Python 3.7 and Keras framework. Model hyperparameters were selected through extended training and validation processes using k-fold cross validation to avoid overfitting while to increase the performance of the models.

B. Data-level Evaluation of SMOTE Variants

Data-level evaluation criteria for evaluating SMOTE variants include: (1) maintaining the data distribution within the minority class, (2) increasing data diversity, and (3) keeping similar class boundaries. Data diversity refers to a robust dataset that is sufficiently representative of the data to prevent biasing, hence it works well with various classification algorithms. The lack of data diversity would lead to model overfitting. Factor (1) can roughly be measured by the difference between the means of the original and generated datasets, with a smaller percent difference indicating better maintenance of the data distribution. Factor (2) can roughly be measured by the difference between the standard deviations of the original and generated datasets, with a moderate increase indicating a healthy increase in diversity. Factor (3) can also roughly be measured by the difference in standard deviations with a very large increase (greater than 10%) indicating possibly excessive boundary distortions. Finally, since the goal of minority class oversampling is to improve training by diversifying the minority class, a small fraction of the generated data duplicating original data is desirable. A summary of these statistics on the synthetic datasets is presented in Table III, which shows ANVO outperforming SMOTE variants on the diversification task while remaining competitive in terms of maintaining data distribution.

Data-level evaluation of augmentation techniques can also be performed empirically by examining scatterplots and histograms of the data, as we now describe. Fig. 2 illustrates the overall pattern of class boundary in the generated data. Visually, we see that the class boundary of the data generated by each SMOTE variant follows a similar pattern to the original class boundary, while ANVO strengthens the class boundary. This stronger class boundary can be expected to help classification algorithms distinguish challenging boundary cases.

Fig. 3 shows the density-based histograms of the two variables x_1 and x_2 of the original data and the data generated by each SMOTE variant. Visually we see that each SMOTE variant achieves data diversification in different ways, increasing or decreasing data density differently with respect to neighboring densities. The differences seem large enough that each algorithm might be expected to excel for particular data distributions or with particular classification algorithms, but further study is needed to provide guidelines and best practices applicable to specific cases.

TABLE I
MAJOR VARIABLES AND MEASUREMENTS IN 24 ASTHMA PATIENTS' DATASETS

Data Category		Variables	Measurement
Physiological data		yesterday's PEFRs	twice a day (AM & PM)
Indoor	Air pollutants & Other variables	$PM_{2.5}$, CO_2 temperature, humidity	every 60 second interval via remote sensors installed at home
Outdoor	Air pollutants & Other variables	SO_2, CO, O_3 , NO_2, PM_{10} temperature, humidity, air pressure	every 30 minute interval via Korean National Data Center

* Data collected from November 1, 2017, to May 31, 2018.

TABLE II
CHARACTERISTICS OF DATASETS

Dataset	# features	# samples	# minority (MI)	# majority (MA)	Imbalance ratio
Synthetic data: small, medium, large	2	200, 800, 3,000	40, 160, 600	160, 640, 2,400	4.0
24 asthma patients' datasets	27	49 - 210	7 - 57	42 - 172	2.3 - 6.0

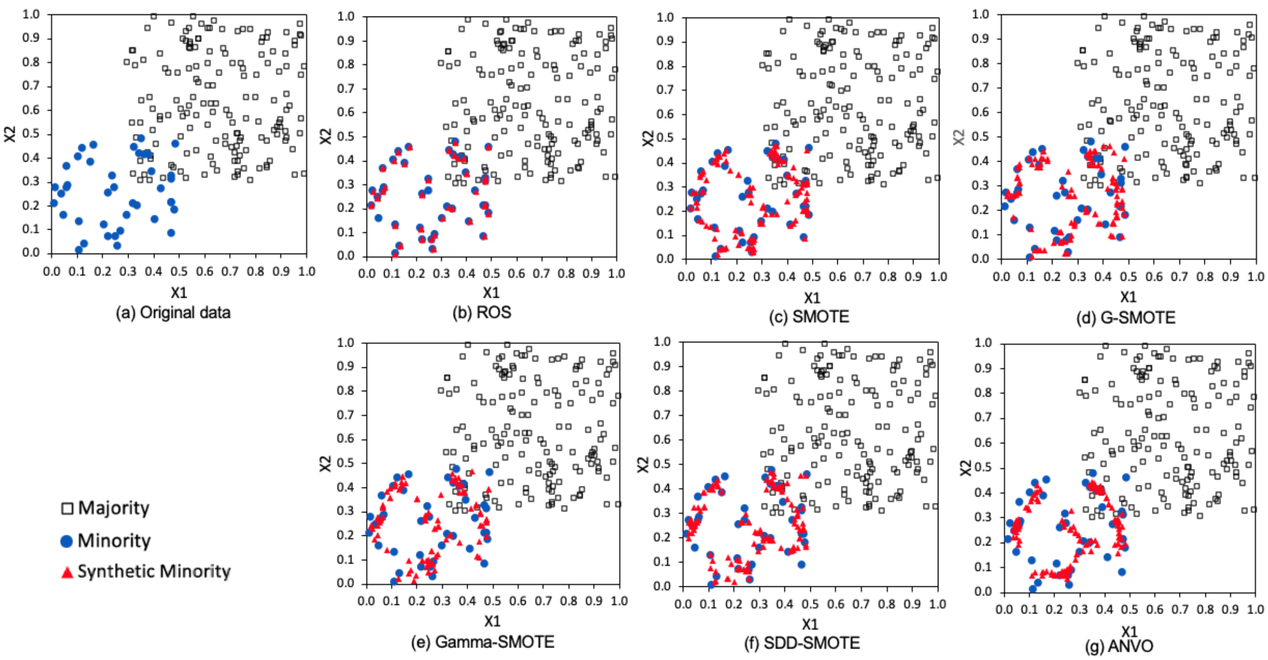
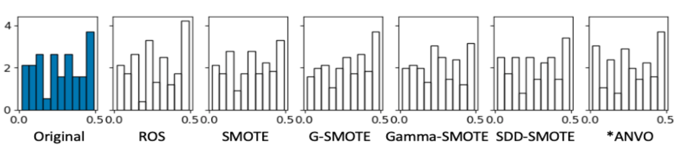
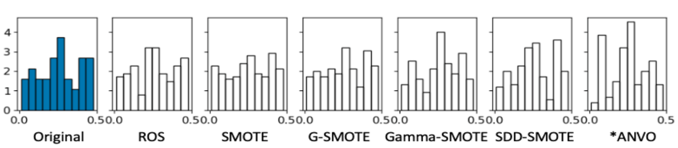


Fig. 2. Original dataset (synthetic small size data) vs. augmented datasets by ANVO and SMOTE variants



(a) Density-based histograms: x_1



(b) Density-based histograms: x_2

Fig. 3. Data distribution in minority: original data vs. SMOTE variants

TABLE III
DATA DISTRIBUTION AND DUPLICATE RATIO ON THE GENERATED DATA

Sampling method	Mean diff.		STD diff.		Duplicates
	x_1	x_2	x_1	x_2	
ROS	+1.3%	-1.5%	+0.2%	-1.2%	**75.00**%
SMOTE	+1.1%	-1.0%	-6.5%	-0.7%	0.00%
G-SMOTE	**+7.1**%	+0.3%	-7.3%	-1.8%	0.00%
Gamma-SMOTE	-1.7%	+1.4%	-6.0%	-7.2%	7.50%
SDD-SMOTE	-0.6%	+0.4%	-3.8%	-7.0%	0.00%
*ANVO	+0.9%	**-2.2**%	**-8.9**%	**-9.5**%	0.87%

* Mean diff: small is good, STD diff: large is good
Duplicates: small is good

C. Algorithm-level Evaluation of SMOTE Variants

Measuring the performance of classification algorithms in the context of imbalanced data is known to be difficult because strong performance on the majority class can easily cause algorithms that perform very poorly on the minority class to achieve very high scores. Several standard evaluation metrics attempt to evaluate performance more fairly by variously emphasizing performance on the minority class, but none perfectly measures performance in all applications. The metrics we used are: weighted accuracy (acc), sensitivity (sen), average precision (pre), and average F_1-score ($F1$).

Weighted accuracy is a better performance metric than raw accuracy for an imbalanced dataset, but it still does not fully capture minority class performance. Nonetheless relatively high weighted accuracy is still a desirable in most classification tasks. Sensitivity, on the other hand, focuses mainly on the minority class by evaluating how adept the model is in correctly identifying positive instances. Both average precision and average F_1 are well-known to be robust performance metrics on classification tasks but is prone to under-emphasize performance on the minority class in applications where misidentifying minority cases has high cost.

In order to incorporate the above considerations, we propose using a weighted evaluation function S_{rank}, defined by,

$$S_{rank} = a_1 \cdot R_{acc} + a_2 \cdot R_{sen} + a_3 \cdot R_{pre} + a_4 \cdot R_{F1},$$

where R_i is the rank of the algorithm in measure i. This metric incorporates each algorithm's rank in all metrics but weights the rank with a ranking factor a_i and $\sum a_i = 1$.

Our focus is on improving sensitivity while keeping other metrics reasonably high. With such a focus, a_2 should be taken significantly larger than other a_i values. Also, since weighted accuracy is known to be a less reliable performance metric than the others in the context of imbalanced data, a_1 should be taken significantly smaller than other values. Given these considerations, the particular a_i values for results reported here: a_1=0.1, a_2=0.5, a_3=0.2, and a_4=0.2. The results do not change significantly provided a_i values are consistent with the above discussion, meaning that,

$$a_1 \ll a_3 \approx a_4 \ll a_2.$$

For each classifier, we generated class balanced training/validation datasets by augmenting synthetic data using ANVO and the five SMOTE variants. The performance of the classifiers was evaluated by using S_{rank}. Due to space limitations, we present the results on the real datasets, but we note that the results on small synthetic dataset follow those of real datasets.

We first present the average performance analysis of ANVO, the SMOTE variants, and no oversampling with the baseline classifiers. Table IV includes S_{rank} scores and the scores of the four evaluation metrics for each oversampling technique in the four classifiers. Overall, the performance of ANVO was comparable to that of the existing SMOTE variants. While ANVO did not achieved the highest S_{rank} score in the four classifiers, its performance with all classifiers was robust by keeping S_{rank} under 3.5 out of 7. The results also show that G-SMOTE performed well with most classifiers: 2.9 with KNN, 2.1 with LR, and 3.3 with NB while the performance of other SMOTE variants varied by classifiers.

Next, we present the performance improvement by using a neural network based TL framework with ANVO and the four SMOTE variants (No oversampling and ROS were eliminated). The improvement of TL + LR over the baseline LR ranged from +23.7% to +61.6% in sensitivity. G-SMOTE and ANVO maintained high performance resulting in G-SMOTE (1.5) and ANOV (1.5), and the improvements in sensitivity were notable, G-SMOTE (+61.6%) and ANVO (+41.8%). The detail of the performance analysis is shown in Table V.

References

[1] T. R. Hoens and N. V. Chawla, "Imbalanced datasets: from sampling to classifiers," *Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley*, 2013.

[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] H. Lee, J. Kim, and S. Kim, "Gaussian-based smote algorithm for solving skewed class distributions," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 4, pp. 229–234, 2017.

[4] F. Kamalov and D. Denisov, "Gamma distribution-based sampling for imbalanced data," *Knowledge-Based Systems*, vol. 207, p. 106368, 2020.

[5] Q. Wan, X. Deng, M. Li, and H. Yang, "Sddsmote: Synthetic minority oversampling technique based on sample density distribution for enhanced classification on imbalanced microarray data," in *The 6th International Conf. on Compute and Data Analysis*, 2022, pp. 35–42.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[7] A. Sharma, P. K. Singh, and R. Chandra, "Smotified-gan for class imbalanced pattern classification problems," *Ieee Access*, vol. 10, pp. 30 655–30 665, 2022.

[8] Y. Huang, K. G. Fields, and Y. Ma, "A tutorial on generative adversarial networks with application to classification of imbalanced data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2021.

[9] ——, "Gan classification," 2021. [Online]. Available: https://github.com/yuxiaohuang/research

[10] J. Woo, G. Rudasingwa, and S. Kim, "Assessment of daily personal pm2. 5 exposure level according to four major activities among children," *Applied Sciences*, vol. 10, no. 1, p. 159, 2020.

TABLE IV

CLASSIFIER PERFORMANCE MEASURES ON ASTHMA PATIENTS' DATASETS

Classifier	Sampling method	Weighted Acc.	Sensitivity	Specificity	Precision avg.	F_1 score avg.	S_{rank}
DT	`No oversampling`	0.5801 (6)	0.2780 (7)	0.8822 (1)	0.5821 (4)	0.5663 (6)	5.6
	`ROS`	0.5779 (7)	0.3369 (6)	0.8188 (2)	0.5738 (6)	0.5655 (7)	5.8
	`SMOTE`	0.5813 (5)	0.3663 (5)	0.7963 (3)	0.5727 (7)	0.5692 (4)	4.8
	`G-SMOTE`	0.5851 (3)	0.3781 (4)	0.7880 (6)	0.5838 (3)	0.5697 (3)	3.8
	`Gamma-SMOTE`	0.5849 (4)	0.3901 (2)	0.7797 (7)	0.5798 (5)	0.5677 (5)	3.7
	`SDD-SMOTE`	0.5955 (1)	0.3980 (1)	0.7931 (5)	0.5850 (2)	0.5773 (2)	**1.8**
	`*ANVO`	0.5873 (2)	0.3803 (3)	0.7944 (4)	0.5891 (1)	0.5776 (1)	**2.3**
KNN	`No oversampling`	0.5443 (7)	0.1279 (7)	0.9607 (1)	0.5575 (7)	0.5195 (7)	6.4
	`ROS`	0.5996 (5)	0.4742 (6)	0.7250 (2)	0.5792 (3)	0.5734 (1)	4.0
	`SMOTE`	0.6188 (1)	0.5697 (1)	0.6679 (5)	0.5829 (1)	0.5673 (2)	**1.6**
	`G-SMOTE`	0.6067 (2)	0.5441 (4)	0.6692 (4)	0.5798 (2)	0.5635 (3)	**2.9**
	`Gamma-SMOTE`	0.5902 (6)	0.5262 (5)	0.6541 (7)	0.5666 (6)	0.5452 (6)	5.6
	`SDD-SMOTE`	0.6061 (3)	0.5500 (3)	0.6623 (6)	0.5783 (4)	0.5615 (4)	3.7
	`*ANVO`	0.6025 (4)	0.5601 (2)	0.6731 (3)	0.5768 (5)	0.5608 (5)	**3.3**
LR	`No oversampling`	0.5303 (7)	0.0762 (7)	0.9844 (1)	0.4656 (7)	0.4864 (7)	6.4
	`ROS`	0.6088 (6)	0.5175 (5)	0.7002 (7)	0.5716 (6)	0.5606 (6)	5.1
	`SMOTE`	0.6168 (5)	0.5028 (6)	0.7308 (4)	0.5933 (3)	0.5852 (2)	4.4
	`G-SMOTE`	0.6199 (3)	0.5260 (3)	0.7338 (2)	0.5992 (1)	0.5884 (1)	**2.1**
	`Gamma-SMOTE`	0.6215 (2)	0.5195 (4)	0.7134 (5)	0.5918 (4)	0.5812 (4)	4.1
	`SDD-SMOTE`	0.6240 (1)	0.5390 (1)	0.7089 (6)	0.5894 (5)	0.5805 (5)	**3.1**
	`*ANVO`	0.6194 (4)	0.5301 (2)	0.7315 (3)	0.5971 (2)	0.5847 (3)	**2.3**
NB	`No oversampling`	0.5383 (7)	0.0987 (7)	0.9778 (1)	0.5024 (7)	0.5019 (7)	6.4
	`ROS`	0.6072 (1)	0.5096 (1)	0.7048 (7)	0.5714 (6)	0.5608 (6)	3.6
	`SMOTE`	0.5992 (4)	0.3783 (5)	0.8201 (2)	0.5995 (2)	0.5910 (1)	**3.3**
	`G-SMOTE`	0.5935 (5)	0.3921 (3)	0.7948 (6)	0.5880 (3)	0.5823 (3)	**3.3**
	`Gamma-SMOTE`	0.5892 (6)	0.3761 (6)	0.8023 (5)	0.5838 (5)	0.5753 (5)	5.5
	`SDD-SMOTE`	0.6011 (3)	0.3880 (4)	0.8141 (3)	0.6016 (1)	0.5876 (2)	**2.9**
	`*ANVO`	0.6028 (2)	0.3952 (2)	0.8103 (4)	0.5843 (4)	0.5822 (4)	**3.0**

* indicates our new SMOTE variant. (#) indicates a rank of 1 through 7 (1 is the highest).

TABLE V

COMPARISONS OF LR VS. TL + LR CLASSIFIERS WITH 5 SMOTE VARIANTS ON ASTHMA PATIENTS' DATASETS

Classifier	Sampling method	Weighted Acc.	Sensitivity	Specificity	Precision avg.	F_1 score avg.	S_{rank}
LR	`SMOTE`	0.6168 (5)	0.5028 (5)	0.7308 (3)	0.5933 (3)	0.5852 (2)	3.8
	`G-SMOTE`	0.6199 (3)	0.5260 (3)	0.7338 (1)	0.5992 (1)	0.5884 (1)	**2.0**
	`Gamma-SMOTE`	0.6215 (2)	0.5195 (4)	0.7134 (4)	0.5918 (4)	0.5812 (4)	4.0
	`SDD-SMOTE`	0.6240 (1)	0.5390 (1)	0.7089 (5)	0.5894 (5)	0.5805 (5)	3.0
	`*ANVO`	0.6194 (4)	0.5301 (2)	0.7315 (2)	0.5971 (2)	0.5847 (3)	**2.2**
TL + LR	`SMOTE`	0.7291 (3) +18.2%	0.7090 (3) **+41.0%**	0.7492 (3) +2.5%	0.6857 (3) +15.6%	0.6824 (3) +16.6%	3.0
	`G-SMOTE`	0.8235 (1) +32.8%	0.8178 (1) **+61.6%**	0.8292 (2) +13.0%	0.7505 (2) +25.3%	0.7675 (2) +30.5%	**1.5**
	`Gamma-SMOTE`	0.7039 (4) +13.3%	0.6425 (5) **+23.7%**	0.7192 (4) +0.8%	0.6281 (5) +6.2%	0.6101 (5) +5.0%	4.9
	`SDD-SMOTE`	0.6891 (5) +9.6%	0.6891 (4) **+27.9%**	0.6791 (5) -4.2%	0.6381 (4) +8.3%	0.6148 (4) +5.9%	4.1
	`*ANVO`	0.7516 (2) +30.9%	0.7516 (2) **+41.8%**	0.8704 (1) +19.0%	0.7677 (1) +28.6%	0.7750 (1) +32.6%	**1.5**

* indicates our new SMOTE variant. (#) indicates a rank of 1 through 5 (1 is the highest rank). #% is the improvement over LR