Beinecke and Heider *BioData Mining* (2021) 14:49
https://doi.org/10.1186/s13040-021-00283-6

BioData Mining

SHORT REPORT Open Access

Gaussian noise up-sampling is better suited than SMOTE and ADASYN for clinical decision making

Jacqueline Beinecke and Dominik Heider*

* Correspondence: dominik.heider@uni-marburg.de
Department of Mathematics and Computer Science, Philipps-University of Marburg, Hans-Meerwein-Str. 6, 35043 Marburg, Germany

Abstract

Clinical data sets have very special properties and suffer from many caveats in machine learning. They typically show a high-class imbalance, have a small number of samples and a large number of parameters, and have missing values. While feature selection approaches and imputation techniques address the former problems, the class imbalance is typically addressed using augmentation techniques. However, these techniques have been developed for big data analytics, and their suitability for clinical data sets is unclear.

This study analyzed different augmentation techniques for use in clinical data sets and subsequent employment of machine learning-based classification. It turns out that Gaussian Noise Up-Sampling (GNUS) is not always but generally, is as good as SMOTE and ADASYN and even outperform those on some datasets. However, it has also been shown that augmentation does not improve classification at all in some cases.

Keywords: Machine learning, Clinical data, Data augmentation, Synthetic data

Introduction

Machine learning (ML) and artificial intelligence (AI) have entered many areas of life and will also pave the way to a new era in medicine. These models can improve medical treatment or diagnosis, identify novel subtypes, or give new insights into survival prognostics. These models consider all facets of data types, e.g., clinical health records, image, or omics data. Clinical decision-support-systems based on ML and AI have been successfully used in many different studies and medical fields, e.g., oncology [1], pathology [2–4], diabetes [5, 6], human genetics [7], and infectious diseases [8–10] as part of a growing trend toward personalized / precision medicine.

Overall, there is great potential for clinical decision-support-systems based on ML and AI techniques. However, clinical datasets have very special properties and suffer from many caveats regarding ML and AI, e.g., a high class imbalance. Moreover, clinical decision-support-systems in medicine need to be interpretable in a probabilistic manner, typically addressed by calibration methods [11]. Furthermore, small-n-large-p and missing values are addressed by feature selection (also called biomarker discovery) approaches [12] and imputation techniques [13]. The class imbalance is typically addressed by using down-sampling or data augmentation techniques [37]. ML and AI approaches perform worse when the data is imbalanced, i.e., when the proportion of positive samples (i.e., cases) and negative samples (i.e., controls) differ strongly. The resulting model will be biased towards the majority class [14]. This is frequently found in clinical datasets, e.g., for rare diseases, but also for general cohort data. While down-sampling might be a straightforward approach to balance a dataset, it is typically worse than data augmentation techniques, as important information / associations from the majority class might be lost. Moreover, down-sampling is not an option when the dataset is already relatively small, as it is often the case in clinical settings.

Methods for addressing the class imbalance by data augmentation have been developed for big data analytics. However, their suitability for clinical data sets has not been tested yet and remains unclear. In the current study, we analyzed commonly used data augmentation techniques in several imbalanced clinical datasets and different ML models.

Methods

Data

In our study, we used ten clinical datasets covering different diseases/scenarios from different medical fields, such as oncology, reproduction, psychology, or hepatology. These datasets address breast cancer, cervical cancer, fertility, drug abuse, hepatitis, cardiotocography, and fatty liver disease, to reflect different sample sizes and class imbalances. Nine of these datasets were collected from the UCI Machine Learning Repository [15]. The smallest dataset has 72 samples, and the most extensive dataset consists of 1831 samples. On average, the datasets have 540 samples, the median is 426, 1st and 3rd quartiles are 124.5 and 713, respectively. The imbalance differs between 2.23 and 37.26% concerning the cases (i.e., positive class). On average, the imbalance is 18.42% (median is 17.7%), with 1st and 3rd quartile at 9.78 and 28.49%, respectively. The number of features ranges from 3 to 32, on average 15 (median is 11), with 1st and 3rd quartile of 9 and 21, respectively.

An overview of the datasets can be found in Table 1. We removed all samples and features with missing values. Thus, the numbers may differ slightly from the original number of samples and features.

Table 1 Overview of the datasets

Name	Samples	Cases	Controls	Percentage	Features
wdbc	569	212	357	37.26%	30
wpbc	198	47	151	23.74%	32
CCRF	761	17	744	2.23%	7
fertility	100	12	88	12.00%	9
CCBR	72	21	51	29.17%	19
Haberman	306	81	225	26.47%	3
heroin	942	97	845	10.3%	11
HCV	546	20	526	3.66%	12
NAFLD	74	22	52	29.73%	9
CTG	1831	176	1655	9.61%	22

The Breast Cancer diagnostics dataset (wdbc) consists of 569 breast cancer patients [357 benign, 212 (37.26%) malignant] with 30 attributes describing histological cancer characteristics [16].

The Breast Cancer prognostics dataset (wpbc) consists of 198 breast cancer patients [151 non-recur, 47 (23.74%) recur] with 32 attributes [16].

The Haberman's survival dataset captures three numerical clinical characteristics and the survival status (binary) for 306 breast cancer patients [225 (73.5%; referred to as controls) survived 5 years or longer, 81 (26.5%; referred to as cases) died within 5 years] [17]. The models predict the survival status (case/control) as a function of three numerical features, namely the patient's age at the time of surgery, the patient's year of surgery, and the number of positive axillary nodes detected.

We used two datasets for cervical cancer, namely the Cervical cancer (Risk Factors) Data Set (CCRF) and the Cervical Cancer Behavior Risk Data Set (CCBR).

The original CCRF dataset comprises demographic information, habits, and historical medical records of 858 patients [18]. We removed all samples and features with missing values. Thus, the final CCRF dataset consists of 761 samples [744 controls and 17 cases (2.23%)] with 7 features. The CCBR dataset consists of 72 samples [51 controls and 21 cases (29.17%)] with 19 features [19].

The fertility dataset consists of 100 samples [88 controls and 12 cases (12.00%)] with 9 features, such as age, childish diseases, etc., and can be used to predict seminal quality [20].

The heroin dataset is a subset of the Drug consumption dataset [21], consisting of data from different drug users. For our analyses, we used just a subset, namely the female heroin abusers. The heroin dataset consists of 942 samples [845 non-heroin abusers and 97 heroin abusers (10.30%)] with 11 attributes, including age, different personality measurements, etc.

The HCV dataset contains laboratory values of blood donors and Hepatitis C patients as well as demographic values and consists of 546 samples [526 controls and 20 cases (3.66%)] [22].

The NAFLD dataset consists of 74 patients with fatty liver disease. Fifty two of them have a non-alcoholic fatty liver, while 22 have an alcoholic fatty liver (29.73%). The dataset has 9 features, including demographic values as well as blood parameters [23].

The Cardiotocography dataset (CTG) consists of 1831 fetal cardiotocograms, of which 1655 are normal and 176 have been classified as pathologic (9.61%). The dataset provides 22 features that have been calculated based on the cardiotocograms [24].

Augmentation techniques

An imbalance is frequently found in real-world datasets, in particular in biomedical datasets. Two main approaches can be found in the literature used to rebalance the data prior to machine learning modeling, namely under-sampling and up-sampling. In under-sampling, we downsize the actual dataset so that the ratio of the dependent categories is balanced. However, due to the nature of clinical data, which is typically relatively small, under-sampling is not an appropriate approach. Thus, our study focuses only on up-sampling techniques (also referred to as data augmentation).

We analyzed three frequently used approaches for data augmentation, namely SMOTE, ADASYN, and GNUS, and compared them to the model without any data augmentation (from now on referred to as the null model) and with each other.

SMOTE (Synthetic Minority Over-sampling Technique) is based on the k-nearest neighbor algorithm [25]. First, it finds the k-nearest neighbors in the minority class for each of the samples in the class. Then it draws a line between the neighbors and generates random points on the lines. We used SMOTE with default settings, i.e., the number of k-nearest neighbors is set to 5.

ADASYN (Adaptive Synthetic sampling approach) works similar to SMOTE and generates synthetic observations for the minority class [26]. However, it adds some noise and thus introduces some variance to the synthetic data points generated from the k-nearest neighbors. We used ADASYN with default settings, i.e., the number of k-nearest neighbors is set to 5. We used the R package *smotefamily* v.1.3.1 for SMOTE and ADASYN.

GNUS (Gaussian Noise Up-Sampling) is a very straightforward and fast technique to generate synthetic data points. It is based on up-sampling [27], i.e., it randomly selects samples from the minority class and adds them to the training data. However, in contrast to normal up-sampling, GNUS adds some noise to the synthetic data points improving variance and smoothing the class boundary to reduce overfitting. Several types of noise could be used. However, the most common one is Gaussian noise. We used GNUS with normal distribution with $\overline{x} = \overline{x}_i * 0.001$ and $sd = sd_i * 0.001$ for $i \epsilon 1, \ldots, n$, and n the number of features (i.e., columns) in the dataset.

Machine learning models

To compare the different augmentation techniques for clinical datasets on subsequent classification, we used three different statistical and machine learning algorithms that are frequently used in clinical settings, namely Logistic Regression (LR), Support Vector Machines (SVM), and Random Forests (RF). For the SVMs, we employed different kernels to capture linear and non-linear associations, namely the linear, radial basis function (rbf), and polynomial kernels. We used the R packages *randomforest* v.4.6 and *kernlab* v.0.9 [28], to train the RF and SVMs, respectively. The LR was trained with the *glm* function in R using the logit model. RFs and SVMs were trained with default parameters.

Statistical evaluation

The statistical and machine learning models were trained and evaluated based on 1000 times repeated Monte Carlo cross-validation (MCCV) [29] using the area under the receiver operating characteristics curve (AUC), area under the precision-recall curve (PR), F1, and the Matthews correlation coefficient (MCC) calculated using the R package *ROCR* v.1.0 [30]. Due to the nature of the data, we used MCCV (also referred to as repeated random sub-sampling validation) instead of the leave-one-out CV, according to Xu et al. [31]. The MCCV has a more minor variance and thus is more reliable for comparison in small datasets. However, it has a higher bias than the k-fold CV. For large sample sizes, the variance issues become less important. However, the datasets are relatively small, and bias does not play a decisive role but variance, as we do not aim at having the best model, but instead want to compare different augmentation techniques with each other. The Bias-Variance trade-off is very common in machine learning, however, in our study, a low variance is more important than a low bias.

The significance of the differences was calculated based on Student's t-tests, resulting p values were corrected for multiple testing by the method of Benjamini and Hochberg [32].

Results

The workflow of the current study is shown in Fig. 1.

GNUS shows significantly ($p < 0.001$ for all datasets) smaller runtime compared to SMOTE and ADASYN (see Fig. 2). On average, GNUS is 9.41 times ($p < 0.001$) faster than SMOTE and 8.9 times ($p = 0.009$) faster than ADASYN, while SMOTE and ADASYN show no significant differences in runtime ($p = 0.2423$). The runtime of GNUS significantly correlates with the number of features ($r = 0.9347$, $p < 0.001$), while SMOTE and ADASYN show only significant correlation with the number of cases in the datasets ($r = 0.9468$, $p < 0.001$ and $r = 0.8996$, $p < 0.001$).

The runtime was measured on a MacBook Pro with 3.5 GHz Dual-Core Intel Core i7 with 16 GB 2133 MHz LPDDR3 RAM.

Data augmentation with neither GNUS, SMOTE, or ADASYN improved subsequent classification with any of the tested statistical and machine learning models significantly for the NAFLD dataset.

For the other nine datasets, data augmentation could improve subsequent classification significantly. An overview of the nine datasets' AUC and MCC values are shown in Tables 2 and 3, as a representative selection of metrics [33].

For the CCBR dataset, SMOTE improved subsequent classification in terms of F1 for the RF from 0.88921 to 0.90024 with $p < 0.001$. ADASYN and GNUS were not able to improve predictions.

All data augmentation methods improved PR, MCC, and F1 for the RF on the fertility dataset. However, for GNUS, the differences were not significant. SMOTE increased

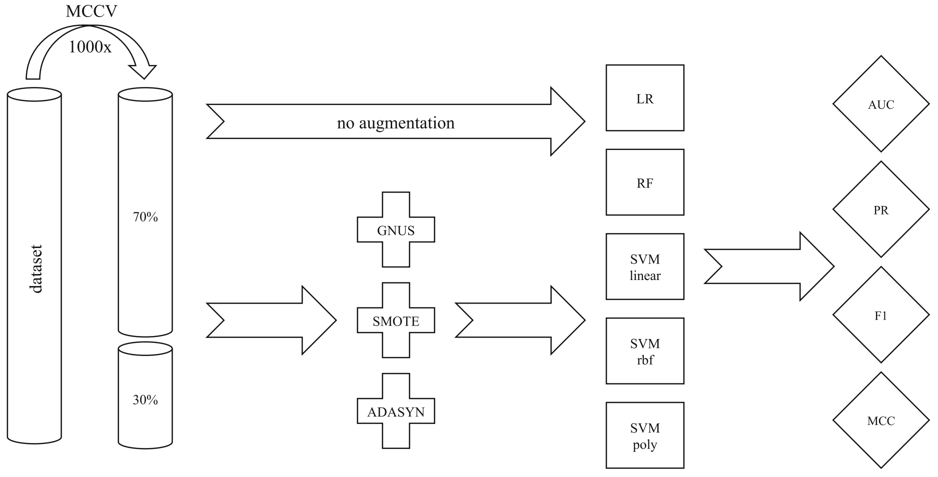

Fig. 1 The workflow of the study. Models are evaluated with 1000 times repeated MCCV. Three augmentation techniques, namely GNUS, SMOTE, and ADASYN are compared against each other and to the null model. As machine learning models we used logistic regression (LR), random forests (RF), and support-vector machines (SVM) with three different kernels (linear, rbf, and polynomial). As for performance metrics, we used AUC, PR, F1, and MCC

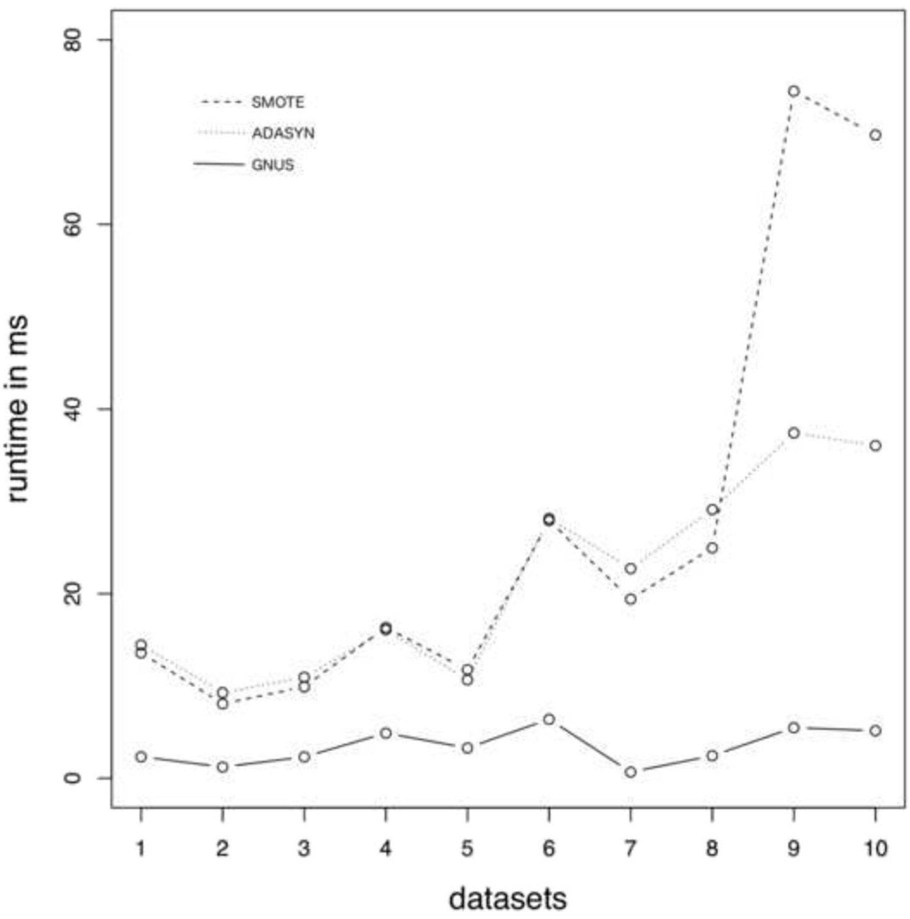

Fig. 2 The runtime of GNUS, SMOTE, and ADASYN on the datasets. GNUS is shown as a solid line, SMOTE is shown as a dashed line, and ADASYN is shown as a dotted line. Datasets are sorted according to the number of cases in the dataset. 1:fertility, 2:CCRF, 3:HCV, 4:CCBR, 5:NAFLD, 6:wpbc, 7:Haberman, 8:heroin, 9:ctg, 10:wdbc

MCC from 0.3952 to 0.43899 with $p < 0.001$ (ADASYN: $MCC = 0.43691$ with $p < 0.001$, GNUS: $MCC = 0.40475$ with $p > 0.001$). All data augmentation methods increased AUC for the RF, but no difference was significant.

For the wpbc dataset, all data augmentation methods improved subsequent classification in terms of all metrics for all SVMs and the RF. However, not all differences were significant. All data augmentation methods significantly improved ($p <$ 0.001) all metrics for the SVM with linear and polynomial kernel. For the RF, all augmentation methods significantly increased ($p < 0.001$) AUC and F1. Furthermore, GNUS and SMOTE increased the PR for the RF significantly ($p < 0.001$). GNUS significantly increased ($p < 0.001$) MCC for RF from 0.39806 to 0.42492, which was significantly larger than both SMOTE and ADASYN (SMOTE: $MCC =$ 0.40874, ADASYN: $MCC = 0.0.40389$). Moreover, GNUS performed significantly better than ADASYN for the RF in terms of PR. In addition, GNUS significantly increased ($p < 0.001$) F1 for SVM with rbf kernel from 0.51902 to 0.53424. A significant decrease for the LR in terms of PR was achieved by all data augmentation methods, AUC and F1 by ADASYN, and MCC by ADASYN and GNUS.

Table 2 Overview of the AUC values for the nine datasets. ***: $p < 0.001$, significant compared to model trained without augmentation. *P* values are adjusted using the method of Benjamini and Hochberg [32]

Data set	ML method	null model	SMOTE	ADASYN	GNUS
wdbc	LR	0.97751, CI: CI: [0.97676,0.97827]	0.9775, CI: CI: [0.97675,0.97826]	0.97743, CI: [0.97668,0.97819]	0.97746, CI: [0.97671,0.9782]
	RF	0.99012, CI: [0.98975,0.99049]	0.99015, CI: [0.98978,0.99052]	0.99096, CI: [0.9906,0.99131]	0.9907, CI: [0.99034,0.99106]
	SVM linear	0.99052, CI: [0.99015,0.99089]	0.99052, CI: [0.99015,0.99089]	0.98474, CI: [0.98421,0.98527]	0.98985, CI: [0.98946,0.99024]
	SVM rbf	0.99457, CI: [0.99435,0.99479]	0.99458, CI: [0.99436,0.9948]	0.99401, CI: [0.99377,0.99424]	0.99497, CI: [0.99474,0.99519]
	SVM polynomial	0.99052, CI: [0.99015,0.99088]	0.99051, CI: [0.99015,0.99088]	0.98474, CI: [0.98421,0.98527]	0.98985, CI: [0.98946,0.99024]
wpbc	LR	0.77109, CI: [0.76712,0.77507]	0.76292, CI: [0.75894,0.7669]	0.76018, CI: [0.75621,0.76414]	0.76225, CI: [0.75823,0.76627]
	RF	0.63627, CI: [0.63164,0.6409]	0.67232, CI: [0.66775,0.67689]***	0.67627, CI: [0.67178,0.68076]***	0.66675, CI: [0.66196,0.67155]***
	SVM linear	0.74045, CI: [0.73628,0.74463]	0.77754, CI: [0.77406,0.78102]***	0.77797, CI: [0.77446,0.78148]***	0.77747, CI: [0.77391,0.78104]***
	SVM rbf	0.70071, CI: [0.69652,0.7049]	0.70501, CI: [0.70102,0.70901]	0.70195, CI: [0.69792,0.70597]	0.70577, CI: [0.70162,0.70992]
	SVM polynomial	0.74042, CI: [0.73625,0.74459]	0.77755, CI: [0.77407,0.78102]***	0.77798, CI: [0.77447,0.78149]***	0.77749, CI: [0.77393,0.78105]***
CCRF	LR	0.59786, CI: [0.59129,0.60443]	0.5984, CI: [0.59157,0.60524]	0.59933, CI: [0.59254,0.60612]	0.58274, CI: [0.5755,0.58998]
	RF	0.61372, CI: [0.60711,0.62032]	0.65221, CI: [0.64606,0.65835]***	0.65084, CI: [0.64466,0.65701]***	0.61197, CI: [0.60524,0.6187]
	SVM linear	0.47064, CI: [0.46213,0.47914]	0.62738, CI: [0.62141,0.63335]***	0.62406, CI: [0.61804,0.63009]***	0.6026, CI: [0.59655,0.60865]***
	SVM rbf	0.44463, CI: [0.43695,0.45231]	0.55105, CI: [0.54252,0.55959]***	0.55191, CI: [0.54336,0.56047]***	0.56228, CI: [0.55475,0.56981]***
	SVM polynomial	0.46553, CI: [0.45719,0.47386]	0.62737, CI: [0.62139,0.63334]***	0.62406, CI: [0.61804,0.63009]***	0.60263, CI: [0.59658,0.60868]***
fertility	LR	0.58666, CI: [0.57774,0.59557]	0.58047, CI: [0.57155,0.58939]	0.57997, CI: [0.57107,0.58886]	0.57527, CI: [0.56631,0.58422]
	RF	0.65456, CI: [0.64764,0.66147]	0.66162, CI: [0.65432,0.66893]	0.66288, CI: [0.65564,0.67011]	0.66512, CI: [0.65833,0.67192]
	SVM linear	0.60622, CI: [0.59781,0.61464]	0.59428, CI: [0.5864,0.60216]	0.59409, CI: [0.58611,0.60208]	0.58382, CI: [0.57546,0.59218]
	SVM rbf	0.66918, CI: [0.66195,0.67641]	0.62841, CI: [0.62056,0.63625]	0.62787, CI: [0.61988,0.63586]	0.61285, CI: [0.60527,0.62044]
	SVM polynomial	0.60704, CI: [0.5986,0.61548]	0.5943, CI: [0.58642,0.60219]	0.59408, CI: [0.5861,0.60206]	0.58381, CI: [0.57545,0.59217]
CCBR	LR	0.90284, CI: [0.89757,0.90811]	0.90101, CI: [0.89559,0.90643]	0.90316, CI: [0.89781,0.90851]	0.89772, CI: [0.89223,0.90322]
	RF	0.95649, CI: [0.95419,0.95879]	0.95778, CI: [0.95545,0.96011]	0.95451, CI: [0.95202,0.957]	0.95335, CI: [0.95086,0.95585]
	SVM linear	0.87819, CI: [0.87349,0.8829]	0.86728, CI: [0.8621,0.87245]	0.86317, CI: [0.85773,0.8686]	0.86125, CI: [0.85581,0.8667]
	SVM rbf	0.9678, CI: [0.96591,0.96968]	0.96526, CI: [0.96318,0.96734]	0.96232, CI: [0.96024,0.9644]	0.96309, CI: [0.96096,0.96522]
	SVM polynomial	0.87818, CI: [0.87347,0.88288]	0.86731, CI: [0.86213,0.87249]	0.86315, CI: [0.85771,0.86859]	0.86124, CI: [0.8558,0.86669]
haberman	LR	0.67754, CI: [0.67404,0.68104]	0.6771, CI: [0.67364,0.68057]	0.67219, CI: [0.66869,0.67568]	0.67838, CI: [0.67488,0.68188]
	RF	0.68429, CI: [0.68153,0.68705]	0.6892, CI: [0.68645,0.69194]	0.68621, CI: [0.6835,0.68892]	0.69812, CI: [0.69531,0.70092]***
	SVM linear	0.67569, CI: [0.67103,0.68034]	0.69693, CI: [0.69349,0.70037]***	0.6536, CI: [0.64959,0.65762]	0.67451, CI: [0.67072,0.6783]
	SVM rbf	0.66694, CI: [0.66398,0.6699]	0.68291, CI: [0.67983,0.68598]***	0.6594, CI: [0.65633,0.66246]	0.66885, CI: [0.66577,0.67193]
	SVM polynomial	0.67535, CI: [0.67065,0.68005]	0.69705, CI: [0.6936,0.70049]***	0.6536, CI: [0.64959,0.65761]	0.67456, CI: [0.67075,0.67837]
heroin	LR	0.82829, CI: [0.82647,0.83011]	0.82977, CI: [0.82803,0.83151]	0.82959, CI: [0.82783,0.83134]	.83109, CI: [0.82937,0.83281]
	RF	0.81037, CI: [0.80864,0.81211]	0.81644, CI: [0.81477,0.81811]***	0.81448, CI: [0.81281,0.81615]	0.80724, CI: [0.80549,0.809]
	SVM linear	0.71822, CI: [0.71292,0.72353]	0.81571, CI: [0.81392,0.8175]***	0.81658, CI: [0.81478,0.81839]***	0.81647, CI: [0.8147,0.81824]***
	SVM rbf	0.7731, CI: [0.77064,0.77556]	0.81363, CI: [0.8116,0.81565]***	0.80945, CI: [0.80742,0.81147]***	0.81706, CI: [0.81509,0.81904]***
	SVM polynomial	0.71734, CI: [0.71196,0.72271]	0.81571, CI: [0.81391,0.8175]***	0.81658, CI: [0.81477,0.81838]***	0.81647, CI: [0.8147,0.81824]***
HCV	LR	0.97933, CI: [0.9773,0.98135]	0.97778, CI: [0.97559,0.97997]	0.9781, CI: [0.97593,0.98028]	0.9768, CI: [0.97452,0.97907]
	RF	0.99394, CI: [0.99355,0.99434]	0.99554, CI: [0.99515,0.99593]***	0.99501, CI: [0.99457,0.99546]	0.99609, CI: [0.99574,0.99643]***
	SVM linear	0.97107, CI: [0.96933,0.9728]	0.97108, CI: [0.96833,0.97383]	0.97018, CI: [0.96742,0.97293]	0.9702, CI: [0.96733,0.97308]
	SVM rbf	0.99169, CI: [0.99099,0.9924]	0.98137, CI: [0.98043,0.98232]	0.98112, CI: [0.98016,0.98209]	0.9841, CI: [0.98318,0.98501]
	SVM polynomial	0.97109, CI: [0.96936,0.97282]	0.97107, CI: [0.96832,0.97382]	0.97019, CI: [0.96743,0.97294]	0.9702, CI: [0.96732,0.97307]
CTG	LR	0.99981, CI: [0.99978,0.99985]	0.99998, CI: [0.99998,0.99999]***	0.99994, CI: [0.99993,0.99996]***	0.99999, CI: [0.99998,0.99999]***
	RF	1, CI: [1,1]	1, CI: [1,1]	1, CI: [1,1]	1, CI: [1,1]
	SVM linear	0.97754, CI: [0.97699,0.97809]	0.99837, CI: [0.99828,0.99846]***	0.99633, CI: [0.99603,0.99664]***	0.99858, CI: [0.99848,0.99867]***
	SVM rbf	0.99947, CI: [0.99945,0.99949]	0.99932, CI: [0.99929,0.99935]	0.99912, CI: [0.99909,0.99916]	0.99931, CI: [0.99928,0.99934]
	SVM polynomial	0.97754, CI: [0.97699,0.97809]	0.99837, CI: [0.99828,0.99847]***	0.99633, CI: [0.99603,0.99664]***	0.99858, CI: [0.99848,0.99867]***

Overview of the AUC values for the nine datasets. ***: $p < 0.001$, significant compared to model trained without augmentation. P values are adjusted using the method of Benjamini and Hochberg [31].

On the Haberman dataset, all data augmentation methods significantly increased F1 for the RF, but in this case, GNUS performed significantly better than SMOTE and ADASYN. GNUS increased F1 for the RF from 0.52521 to 0.54206 (SMOTE: F1 = 0.53165, ADASYN: F1 = 0.53117). Besides that, all data augmentation methods improved the RF in terms of AUC and MCC, but the differences were only significant for GNU, which also performed significantly better than ADASYN and SMOTE in this case. SMOTE was able to improve subsequent classification in terms of all metrics for the SVM with linear and polynomial kernel, as well as AUC for the SVM with rbf kernel. For the SVM with the linear and the polynomial kernel in terms of PR, SMOTE performed significantly better than GNUS, and for the SVM with rbf kernel in terms of AUC.

For the HCV dataset, all data augmentation methods were able to improve PR, MCC, and F1 for SVM with linear and polynomial kernel and for the RF. Furthermore, all augmentation methods increased AUC for the RF, but only GNUS and SMOTE significantly. For the RF in terms of all metrics, GNUS performed significantly better than ADASYN and for the SVM with linear and polynomial kernel SMOTE performed significantly better than GNUS in terms of PR. GNUS increased AUC for the RF from 0.99394 to 0.99609 with $p < 0.001$ (SMOTE: AUC = 0.99554 with $p < 0.001$, ADASYN: AUC = 0.99501).

ADASYN significantly increased PR, MCC, and F1 for the RF on the wdbc dataset. For MCC and F1 in terms of the RF ADASYN performed significantly better than both GNUS and SMOTE. For instance, ADASYN increased F1 from 0.95607 to 0.96095 with $p < 0.001$ (SMOTE: F1 = 0.95621, GNUS: F1 = 0.958). Furthermore, ADASYN significantly decreased all metrics for the SVM with linear and polynomial kernel, and PR for the SVM with rbf kernel. GNUS and SMOTE were able to increase all metrics for the

Table 3 Overview of the MCC values for the nine datasets. ***: $p < 0.001$, significant compared to model trained without augmentation. P values are adjusted using the method of Benjamini and Hochberg [32]

Data set	ML method	null model	SMOTE	ADASYN	GNUS
wdbc	LR	0.90294, CI: [0.90095, 0.90493]	0.90294, CI: [0.90095, 0.90493]	0.9029, CI: [0.90091, 0.9049]	0.9016, CI: [0.89962, 0.90358]
	RF	0.93111, CI: [0.92959, 0.93262]	0.93124, CI: [0.92971, 0.93277]	0.93902, CI: [0.93744, 0.94059]***	0.9342, CI: [0.93269, 0.93571]
	SVM linear	0.93125, CI: [0.9298, 0.93271]	0.93125, CI: [0.92979, 0.9327]	0.91492, CI: [0.9132, 0.91664]	0.93055, CI: [0.92907, 0.93203]
	SVM rbf	0.95128, CI: [0.95, 0.95256]	0.95133, CI: [0.95006, 0.95259]	0.94971, CI: [0.9485, 0.95092]	0.95431, CI: [0.95314, 0.95548]
	SVM polynomial	0.93117, CI: [0.92971, 0.93262]	0.9313, CI: [0.92984, 0.93276]	0.91492, CI: [0.9132, 0.91664]	0.93057, CI: [0.92909, 0.93205]
wpbc	LR	0.46946, CI: [0.46351, 0.47542]	0.45616, CI: [0.45038, 0.46195]	0.45225, CI: [0.44647, 0.45802]	0.45364, CI: [0.44785, 0.45944]
	RF	0.39806, CI: [0.3919, 0.40422]	0.40874, CI: [0.40267, 0.4148]	0.40389, CI: [0.39792, 0.40985]	0.42492, CI: [0.41859, 0.43125]***
	SVM linear	0.42625, CI: [0.4204, 0.4321]	0.47893, CI: [0.47342, 0.48443]***	0.4805, CI: [0.47483, 0.48617]***	0.47974, CI: [0.474, 0.48547]***
	SVM rbf	0.43084, CI: [0.42495, 0.43673]	0.43648, CI: [0.43044, 0.44252]	0.43651, CI: [0.43042, 0.4426]	0.44555, CI: [0.43938, 0.45172]
	SVM polynomial	0.42611, CI: [0.42025, 0.43196]	0.47892, CI: [0.47342, 0.48443]***	0.48056, CI: [0.4749, 0.48623]***	0.47962, CI: [0.47389, 0.48535]***
CCRF	LR	0.11934, CI: [0.11634, 0.12235]	0.12262, CI: [0.11945, 0.12579]	0.12308, CI: [0.11992, 0.12624]	0.11741, CI: [0.11416, 0.12065]
	RF	0.2125, CI: [0.20335, 0.22166]	0.15492, CI: [0.15074, 0.15909]	0.15241, CI: [0.14848, 0.15634]	0.17264, CI: [0.16601, 0.17926]
	SVM linear	0.08623, CI: [0.08274, 0.08972]	0.1354, CI: [0.13239, 0.13841]***	0.13483, CI: [0.13178, 0.13789]***	0.12363, CI: [0.12074, 0.12651]***
	SVM rbf	0.07546, CI: [0.0727, 0.07823]	0.15509, CI: [0.14909, 0.1611]***	0.15467, CI: [0.14862, 0.16072]***	0.16153, CI: [0.15503, 0.16803]***
	SVM polynomial	0.08467, CI: [0.08118, 0.08816]	0.13543, CI: [0.13241, 0.13844]***	0.13482, CI: [0.13176, 0.13787]***	0.12362, CI: [0.12074, 0.1265]***
fertility	LR	0.33909, CI: [0.3295, 0.34869]	0.3247, CI: [0.3154, 0.334]	0.3184, CI: [0.30939, 0.3274]	0.31553, CI: [0.30641, 0.32465]
	RF	0.3952, CI: [0.38682, 0.40359]	0.43899, CI: [0.42966, 0.44833]***	0.43691, CI: [0.42782, 0.44601]***	0.40475, CI: [0.39605, 0.41346]
	SVM linear	0.32997, CI: [0.32129, 0.33866]	0.32136, CI: [0.31302, 0.3297]	0.31935, CI: [0.31104, 0.32767]	0.31718, CI: [0.30842, 0.32595]
	SVM rbf	0.41066, CI: [0.40174, 0.41957]	0.3977, CI: [0.38848, 0.40693]	0.39862, CI: [0.38949, 0.40775]	0.39235, CI: [0.38296, 0.40173]
	SVM polynomial	0.33074, CI: [0.32186, 0.33962]	0.32143, CI: [0.3131, 0.32977]	0.31926, CI: [0.31095, 0.32758]	0.31725, CI: [0.30848, 0.32602]
CCBR	LR	0.7649, CI: [0.75588, 0.77392]	0.76381, CI: [0.75474, 0.77287]	0.76629, CI: [0.75725, 0.77533]	0.76019, CI: [0.75107, 0.76931]
	RF	0.85072, CI: [0.84487, 0.85657]	0.86343, CI: [0.8577, 0.86915]	0.86084, CI: [0.85508, 0.8666]	0.85158, CI: [0.84568, 0.85747]
	SVM linear	0.71174, CI: [0.70437, 0.7191]	0.70248, CI: [0.69411, 0.71086]	0.69732, CI: [0.68885, 0.7058]	0.69547, CI: [0.68688, 0.70406]
	SVM rbf	0.87747, CI: [0.87196, 0.88298]	0.88108, CI: [0.87566, 0.8865]	0.87338, CI: [0.86804, 0.87873]	0.87722, CI: [0.87185, 0.88259]
	SVM polynomial	0.71165, CI: [0.70429, 0.71901]	0.70253, CI: [0.69414, 0.71092]	0.69745, CI: [0.68897, 0.70593]	0.69542, CI: [0.68683, 0.70401]
haberman	LR	0.37981, CI: [0.37491, 0.38471]	0.37736, CI: [0.37244, 0.38227]	0.36763, CI: [0.36273, 0.37253]	0.37781, CI: [0.37289, 0.38273]
	RF	0.33082, CI: [0.32708, 0.33455]	0.33652, CI: [0.33284, 0.34021]	0.33521, CI: [0.33162, 0.33881]	0.35136, CI: [0.34739, 0.35532]***
	SVM linear	0.36978, CI: [0.36431, 0.37526]	0.39156, CI: [0.38682, 0.39631]***	0.33886, CI: [0.33355, 0.34417]	0.36457, CI: [0.35952, 0.36962]
	SVM rbf	0.34028, CI: [0.33622, 0.34434]	0.33367, CI: [0.32964, 0.33771]	0.30688, CI: [0.30298, 0.31078]	0.3178, CI: [0.31382, 0.32178]
	SVM polynomial	0.36856, CI: [0.36309, 0.37404]	0.39149, CI: [0.38675, 0.39624]***	0.3388, CI: [0.33348, 0.34411]	0.36468, CI: [0.35963, 0.36973]
heroin	LR	0.39173, CI: [0.3888, 0.39465]	0.39553, CI: [0.39275, 0.39831]	0.39629, CI: [0.39345, 0.39913]	0.39858, CI: [0.39581, 0.40136]
	RF	0.35277, CI: [0.35027, 0.35527]	0.37004, CI: [0.36735, 0.37273]***	0.36697, CI: [0.36428, 0.36967]***	0.35217, CI: [0.34957, 0.35477]
	SVM linear	0.28106, CI: [0.27606, 0.28606]	0.36966, CI: [0.36696, 0.37236]***	0.37128, CI: [0.3685, 0.37406]***	0.36981, CI: [0.36712, 0.3725]***
	SVM rbf	0.35601, CI: [0.35294, 0.35909]	0.3899, CI: [0.38694, 0.39286]***	0.38492, CI: [0.38196, 0.38788]***	0.39249, CI: [0.38944, 0.39553]***
	SVM polynomial	0.28045, CI: [0.27542, 0.28549]	0.36968, CI: [0.36699, 0.37238]***	0.37126, CI: [0.36848, 0.37404]***	0.36979, CI: [0.3671, 0.37248]***
HCV	LR	0.87178, CI: [0.86655, 0.87701]	0.86773, CI: [0.86222, 0.87324]	0.86907, CI: [0.86364, 0.87449]	0.86552, CI: [0.85991, 0.87113]
	RF	0.87267, CI: [0.86705, 0.87829]	0.9134, CI: [0.90819, 0.9186]***	0.91065, CI: [0.90536, 0.91593]***	0.92506, CI: [0.92032, 0.9298]***
	SVM linear	0.82229, CI: [0.81636, 0.82822]	0.86893, CI: [0.86377, 0.87409]***	0.86153, CI: [0.85589, 0.86717]***	0.85569, CI: [0.85009, 0.8613]***
	SVM rbf	0.9048, CI: [0.90016, 0.90944]	0.79017, CI: [0.78386, 0.79649]	0.78905, CI: [0.78263, 0.79547]	0.8094, CI: [0.80327, 0.81554]
	SVM polynomial	0.8224, CI: [0.81648, 0.82833]	0.86881, CI: [0.86363, 0.874]***	0.86156, CI: [0.85593, 0.86719]***	0.8558, CI: [0.85019, 0.86141]***
CTG	LR	0.99379, CI: [0.99317, 0.9944]	0.99883, CI: [0.99859, 0.99907]***	0.99716, CI: [0.99674, 0.99759]***	0.99905, CI: [0.99882, 0.99928]***
	RF	0.1, CI: [1,1]	0.1, CI: [1,1]	0.1, CI: [1,1]	0.1, CI: [1,1]
	SVM linear	0.81092, CI: [0.80884, 0.813]	0.95298, CI: [0.95153, 0.95442]***	0.94593, CI: [0.94415, 0.94771]***	0.95886, CI: [0.95737, 0.96034]***
	SVM rbf	0.96879, CI: [0.96793, 0.96965]	0.97359, CI: [0.97276, 0.97442]***	0.96917, CI: [0.96826, 0.97008]	0.97258, CI: [0.97176, 0.9734]***
	SVM polynomial	0.81093, CI: [0.80885, 0.81301]	0.95293, CI: [0.95149, 0.95437]***	0.94589, CI: [0.94411, 0.94766]***	0.95882, CI: [0.95734, 0.96031]***

Overview of the MCC values for the datasets. ***: $p < 0.001$, significant compared to model trained without augmentation. P values are adjusted using the method of Benjamini and Hochberg [31].

SVM with rbf kernel, but only GNUS significantly for PR. GNUS performed significantly better than SMOTE.

For the CCRF dataset, all data augmentation methods significantly improved subsequent classification in terms of all metrics for all SVMs. For example, SMOTE increased AUC for the SVM with linear kernel from 0.47064 to 0.62738 (ADASYN: AUC = 0.62406, GNUS: AUC = 0.6026). In terms of AUC, MCC, and F1 for SVM with linear and polynomial kernel ADASYN and SMOTE performed significantly better than GNUS. Furthermore, ADASYN and SMOTE significantly increased AUC for the RF. Lastly, SMOTE and ADASYN increased AUC, PR, MCC, and F1 for the LR and GNUS increased PR and F1 for the LR, but no significant increase. No data augmentation method was able to improve PR, MCC, or F1 for the RF.

All data augmentation methods were able to improve all metrics for the SVMs on the heroin dataset. All differences were significant except for the SVM with rbf kernel in terms of PR. Furthermore, SMOTE and ADASYN increased AUC, MCC, and F1 for the LR and the RF, with significant differences for SMOTE in terms of AUC, MCC, and F1 for the RF, and for ADASYN in terms of MCC and F1 for the RF. In addition, SMOTE and ADASYN performed significantly better than GNUS for the RF in terms F1. In contrast, GNUS performed significantly better than ADASYN for the SVM with rbf kernel in terms of AUC. Lastly, GNUS improved the LR in terms of AUC, MCC, and F1 and the RF in terms of F1, but none of the differences were significant.

For the CTG dataset, all data augmentation methods significantly improved subsequent classification in terms of all metrics for the LR and SVM with the linear and polynomial kernel. For the SVM with rbf kernel all data augmentation methods increased MCC and F1, however, only significantly for SMOTE and GNUS. Furthermore, for the SVM with linear and polynomial kernel in terms of MCC, PR, and F1 GNUS performed significantly better than ADASYN and SMOTE, and SMOTE performed significantly better than ADASYN. Additionally, for the LR in terms of all metrics SMOTE and GNUS performed significantly better than ADASYN. This also holds for the SVM with linear and polynomial kernel in terms of AUC, and the SVM with rbf kernel in terms of MCC and F1.

For the three smallest datasets, namely CCBR (72 samples), NAFLD (74 samples), and fertility (100 samples), data augmentation did not improve subsequent classification a lot. This is also true for the wdbc dataset, which is larger (569 samples) but is the least imbalanced (37.26%) out of all of the ten datasets.

The wpbc and Haberman datasets are the next larger ones with 198 samples and 306 samples. Both are similarly imbalanced (wdbc = 23.74% and Haberman = 26.47%) but the Haberman dataset only has three features while the wpbc dataset has 32. This could be why the Haberman dataset profited significantly less from the data augmentation methods than the wpbc dataset.

The HCV dataset (546 samples and 3.66%), CCRF dataset (761 and 2.23%), heroin dataset (942 samples and 10.3%), and the CTG dataset (1831 samples and 9.61%) are the largest and most imbalanced datasets, and next to the wpbc dataset they profited the most from all data augmentation methods.

Discussion

Data augmentation techniques have been developed to tackle the problem of class imbalance and subsequent bias in machine learning models. However, these methods have been developed for big data analytics, and their applicability and suitability for clinical data sets, which are relatively small, have not been evaluated widely so far.

Taneja et al. compared SMOTE and ADASYN with RFs and gradient boosting approaches on a single dataset (with 284,807 samples) with a high degree of imbalance [34]. Their results show that the metrics from all models in conjunction with SMOTE outperformed ADASYN. Another analysis on a single data set (7718 samples) with a high degree of imbalance was carried out by Barros et al. [35]. They compared SMOTE and ADASYN based on subsequent classification with decision trees and neural networks and could also show that SMOTE outperforms ADASYN on this dataset. Davagdorj et al. compared SMOTE and ADASYN based on seven different statistical and machine learning models on a single dataset (3692 samples). However, the performance of SMOTE and ADASYN varied depending on the model [36]. Amin et al. included four datasets (between 3333 and 38,162 samples) with varying degrees of imbalance and four different machine learning models [37]. However, neither SMOTE nor ADAS YN were superior to each other. To the best of our knowledge, our study marks the first comparison of SMOTE, ADASYN, and GNUS on a larger number of datasets and models and the first study that addresses clinical data and clinical decision making.

For some datasets, data augmentation did not improve overall performance in subsequent machine learning. It turned out that for small datasets (≤100 samples), data augmentation is less useful for subsequent classification than for larger datasets. For larger datasets (≥546 samples) with a high-class imbalance (≤10.3%), all three data augmentation methods significantly improved performance in subsequent classification.

In most datasets, GNUS is as good as SMOTE or ADASYN, and a significant improvement has been reached compared to the models without data augmentation. In line with some other studies, SMOTE generally performs better than ADASYN. There are, however, also some cases where GNUS is significantly better than SMOTE or ADASYN, e.g., in the wdbc dataset. Moreover, in some cases, data augmentation significantly decreased performance concerning compared to the imbalanced model. In all datasets analyzed, GNUS significantly outperforms SMOTE and ADASYN in terms of runtime.

Nevertheless, data augmentation can only help to decrease bias in imbalanced datasets for machine learning predictions. If the data has not enough variance, data augmentation will not improve subsequent predictions. In the worst case, can also decrease performance because of the additional noise introduced in the training of the models.

This study marks the first comprehensive analysis of three commonly used data augmentation techniques for use in clinical datasets with a variety of machine learning models. It turned out that simple GNUS is generally as good as SMOTE and ADASYN and on some datasets and models even outperformed them.

Acknowledgements
We are grateful to the UCI Machine Learning Repository for granting access to a great variety of datasets.

Authors' contributions
DH designed the study and the experiments, and supervised the study. DH and JB performed the analyses and jointly wrote the manuscript. The authors read and approved the final manuscript.

Funding
This work was supported by the LOEWE program of the State of Hesse (Germany) in the Diffusible Signals research cluster.

Availability of data and materials
All datasets are publicly available at the UCI Machine Learning Repository.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 17 May 2021 Accepted: 10 November 2021
Published online: 29 November 2021

References

1. Bibault J-E, Giraud P, Burgun A. Big data and machine learning in radiation oncology: state of the art and future prospects. Cancer Lett. 2016;382(1):110–7. https://doi.org/10.1016/j.canlet.2016.05.033.
2. Madabhushi A, Lee G. Image analysis and machine learning in digital pathology: challenges and opportunities. Med Image Anal. 2016;33:170–5. https://doi.org/10.1016/j.media.2016.06.037.
3. Yala A, Barzilay R, Salama L, Griffin M, Sollender G, Bardia A, et al. Using machine learning to parse breast pathology reports. Breast Cancer Research Treat. 2017;161:203–11. https://doi.org/10.1007/s10549-016-4035-1.
4. Coudray N, Ocampo PS, Sakellaropoulos T, Narula N, Snuderl M, Fenyö D, et al. Classication and mutation prediction from non-small cell lung cancer histopathology images using deep learning. Nat Med. 2018;24:1559–67. https://doi.org/10.1038/s41591-018-0177-5.
5. Chen P, Pan C. Diabetes classification model based on boosting algorithms. BMC Bioinformatics. 2018;19(1):109. https://doi.org/10.1186/s12859-018-2090-9.
6. Spänig S, Emberger-Klein A, Sowa J-P, Canbay A, Menrad K, Heider D. The virtual doctor: an interactive clinical-decision-support system based on deep learning for non -invasive prediction of diabetes. Artif Intell Med. 2019;100:101706. https://doi.org/10.1016/j.artmed.2019.101706.
7. Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. Nat Rev Genet. 2015;16(6):321–32. https://doi.org/10.1038/nrg3920.
8. Lengauer T, Sing T. Bioinformatics-assisted anti-HIV therapy. Nat Rev Microb. 2006;4(10):790–7. https://doi.org/10.1038/nrmicro1477.

9. Heider D, Dybowski JN, Wilms C, Hoffmann D. A simple structure-based model for the prediction of HIV-1 co-receptor tropism. BioData Min. 2014;7. https://doi.org/10.1186/1756-0381-7-14.
10. Spänig S, Heider D. Encodings and models for antimicrobial peptide classification for multi-resistant pathogens. BioData Min. 2019;12(1):7. https://doi.org/10.1186/s13040-019-0196-x.
11. Schwarz J, Heider D. Guess: projecting machine learning scores to well-calibrated probability estimates for clinical decision making. Bioinformatics. 2019;35(14):2458–65. https://doi.org/10.1093/bioinformatics/bty984.
12. Neumann U, Riemenschneider M, Sowa J-P, Baars T, Kälsch J, Canbay A, et al. Compensation of feature selection biases accompanied with improved predictive performance for binary classification by using a novel ensemble feature selection approach. BioData Min. 2016;9(1):36. https://doi.org/10.1186/s13040-016-0114-4.
13. Stekhoven DJ, Bühlmann P. Missforest - nonparametric missing value imputation for mixed-type data. Bioinformatics. 2012;28(1):112–8. https://doi.org/10.1093/bioinformatics/btr597.1105.0828.
14. Krawczyk B. Learning from imbalanced data: open challenges and future directions. Prog Artif Intell. 2016;5(4):221–32. https://doi.org/10.1007/s13748-016-0094-0.
15. Dua D, Graff C. UCI machine learning repository. 2017. http://archive.ics.uci.edu/ml. Accessed 1 Feb 2021.
16. Wolberg WH, Mangasarian OL. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc Natl Acad Sci U S A. 1990;87(23):9193–6. https://doi.org/10.1073/pnas.87.23.9193.
17. Haberman SJ. Generalized Residuals for Log-linear Models. In: Proceedings of the 9th International Biometrics Conference. Boston; 1976. p. 104–22.
18. Kelwin F, J.F. Jaime S. Cardoso: transfer learning with partial observability applied to cervical Cancer screening. In: Iberian Conference on Pattern Recognition and Image Analysis. Faro: Springer; 2017.
19. Sobar MR, Wijaya A. Behavior determinant based cervical cancer early detection with machine learning algorithm. Adv Sci Lett. 2016;22(10):3120–3. https://doi.org/10.1166/asl.2016.7980.
20. Gil D, Girela JL, Juan JD, Gomez-Torres MJ, Johnsson M. Predicting seminal quality with artificial intelligence methods. Expert Syst Appl. 2012;39(16):12564–73. https://doi.org/10.1016/j.eswa.2012.05.028.
21. Fehrman E, Egan V, Gorban AN, Levesley J, Mirkes EM, Muhammad AK. Personality traits and drug consumption: Springer; 2019. https://doi.org/10.1007/978-3-030-10442-9.
22. Lichtinghagen R, Pietsch D, Bantel H, Manns MP, Brand K, Bahr MJ. The enhanced liver fibrosis (elf) score: normal values, influence factors and proposed cut-off values. J Hepatol. 2013;59(2):236–42. https://doi.org/10.1016/j.jhep.2013.03.016.
23. Sowa J-P, Atmaca O, Kahraman A, Schlattjan M, Lindner M, Sydor S, et al. Non-invasive separation of alcoholic and non-alcoholic liver disease with predictive modeling. PLoS One. 2013;9(7):101444. https://doi.org/10.1371/journal.pone.0101444.
24. Ayres de Campos D, Bernardes J, Garrido A, Marques-de-sa J, Pereira-leite L. Sisporto 2.0: A program for automated analysis of cardiotocograms. J Matern Fetal Med. 2000;5:311–8. https://doi.org/10.3109/14767050009053454.
25. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002:16, 321–357. https://doi.org/10.1613/jair.953.
26. He H, Bai Y, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, Part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1–6, 2008: IEEE; 2008. p. 1322–8. https://doi.org/10.1109/IJCNN.2008.4633969.
27. Branco P, Torgo L, Ribeiro RP. A survey of predictive modeling on imbalanced domains. ACM Comput Surv. 2016;49(2):1–50. https://doi.org/10.1145/2907070.
28. Karatzoglou A, Smola A, Hornik K, Zeileis A. kernlab an S4 package for kernel methods in R. J Stat Softw. 2004;11(9):1–20.
29. Burman P. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. Biometrika. 1989;76(3):503–14. https://doi.org/10.1093/biomet/76.3.503.
30. Sing T, Sander O, Beerenwinkel N, Lengauer T. ROCR: visualizing classifier performance in R. Bioinform. 2005;21(20):3940–1. https://doi.org/10.1093/bioinformatics/bti623.
31. Qing-Song Xu YD. Yi-Zeng Liang: Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. J Chemom. 2004;18(2):112–20. https://doi.org/10.1002/cem.858.
32. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B. 1995;57(1):289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.
33. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genomics. 2020;21(1):6. https://doi.org/10.1186/s12864-019-6413-7.
34. Taneja S, Suri B, Kothari C. Application of Balancing Techniques with Ensemble Approach for Credit Card Fraud Detection. In: International Conference on Computing, Power and Communication Technologies (GUCON), New Delhi, India; 2019. p. 753–8.
35. Barros TM, Souza Neto PA, Silva I, Guedes LA. Predictive models for imbalanced data: a school dropout perspective. Educ Sci. 2019;9(4):275–92. https://doi.org/10.3390/educsci9040275.
36. Davagdorj K, Lee JS, Pham VH, Ryu KH. A comparative analysis of machine learning methods for class imbalance in a smoking cessation intervention. Appl Sci. 2020;10(9):3307–27. https://doi.org/10.3390/app10093307.
37. Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, et al. Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. IEEE Access. 2016;4:7940–57. https://doi.org/10.1109/ACCESS.2016.2619719.

Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.