

RESEARCH ARTICLE

An empirical evaluation of sampling methods for the classification of imbalanced data

Misuk Kim, Kyu-Baek Hwang *

Department of Computer Science and Engineering, Graduate School, Soongsil University, Seoul, Korea

* kbhwang@ssu.ac.kr

Abstract

In numerous classification problems, class distribution is not balanced. For example, positive examples are rare in the fields of disease diagnosis and credit card fraud detection. General machine learning methods are known to be suboptimal for such imbalanced classification. One popular solution is to balance training data by oversampling the underrepresented (or undersampling the overrepresented) classes before applying machine learning algorithms. However, despite its popularity, the effectiveness of sampling has not been rigorously and comprehensively evaluated. This study assessed combinations of seven sampling methods and eight machine learning classifiers (56 varieties in total) using 31 datasets with varying degrees of imbalance. We used the areas under the precision-recall curve (AUPRC) and receiver operating characteristics curve (AUROC) as the performance measures. The AUPRC is known to be more informative for imbalanced classification than the AUROC. We observed that sampling significantly changed the performance of the classifier (paired *t*-tests $P < 0.05$) only for few cases (12.2% in AUPRC and 10.0% in AUROC). Surprisingly, sampling was more likely to reduce rather than improve the classification performance. Moreover, the adverse effects of sampling were more pronounced in AUPRC than in AUROC. Among the sampling methods, undersampling performed worse than others. Also, sampling was more effective for improving linear classifiers. Most importantly, we did not need sampling to obtain the optimal classifier for most of the 31 datasets. In addition, we found two interesting examples in which sampling significantly reduced AUPRC while significantly improving AUROC (paired *t*-tests $P < 0.05$). In conclusion, the applicability of sampling is limited because it could be ineffective or even harmful. Furthermore, the choice of the performance measure is crucial for decision making. Our results provide valuable insights into the effect and characteristics of sampling for imbalanced classification.

OPEN ACCESS

Citation: Kim M, Hwang K-B (2022) An empirical evaluation of sampling methods for the classification of imbalanced data. PLoS ONE 17(7): e0271260. <https://doi.org/10.1371/journal.pone.0271260>

Editor: Ryan J. Urbanowicz, University of Pennsylvania, UNITED STATES

Received: March 2, 2022

Accepted: June 28, 2022

Published: July 28, 2022

Copyright: © 2022 Misuk, Kyu-Baek. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The data underlying the results presented in the study are available from the UC Irvine Machine Learning Repository (<https://archive.ics.uci.edu>) and Kaggle (<https://www.kaggle.com>). We provide the name in the source repository and URL for each of the 31 datasets in [S3 Table](#) of the manuscript.

Funding: K.-B.H. was supported by the Soongsil University Research Fund. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Introduction

In classification problems, an imbalanced dataset is a dataset in which the number of data examples of some classes is much smaller than that of other classes. Imbalanced datasets are common in many fields such as chemical and biomedical engineering, financial management, and information technology [1]. Typical imbalanced classification problems include fraud detection [2], fault diagnosis [3, 4], anomaly detection [5, 6], disease diagnosis [7], e-mail

Competing interests: The authors have declared that no competing interests exist.

foldering [8], face recognition [9], and oil spill detection [10]. Most machine learning methods assume equal misclassification costs between the majority and minority class examples. Therefore, they are based on the assumption that the class distribution is balanced. Thus, general machine learning algorithms show suboptimal performance on imbalanced datasets, resulting in a classifier that is biased toward the majority class [10–14].

There are two categories of imbalanced data: intrinsic and extrinsic [15]. Intrinsic imbalanced data are generated because of the nature of problem domains. Examples of such domains include credit card fraud detection [2] and disease diagnosis [14]. In contrast, an extrinsic imbalance is caused by other factors such as biased sampling processes. For example, even if data are obtained from a balanced continuous stream, data transfer could be interrupted for a specific period, resulting in an imbalanced dataset [15]. We focused on the classification of intrinsic imbalanced data where the true class distribution is imbalanced.

Three main approaches can be employed to tackle the imbalanced classification problem. The first is the algorithm level approach, which modifies or creates machine learning algorithms for imbalanced classification [16–22]. It requires an in-depth knowledge of both algorithms and application domains [14, 23]. Second, the data level approach balances the imbalanced class distribution by sampling before applying machine learning algorithms. Examples of data level approaches include oversampling to increase data in the minority classes and undersampling to reduce data in the majority classes [13, 24, 25]. Finally, the cost-sensitive learning-based approach addresses the imbalanced classification problem by assigning different misclassification costs to each class [26, 27]. In this approach, the most important and challenging process is defining the misclassification costs of different classes whose true values are unknown [28].

This work focused on the data level approach, which is the easiest and most popular of the three. This approach is easy to use because machine learning algorithms need not be created or modified [23]. Furthermore, it can be used with any machine learning algorithm. Because of its simplicity and applicability, data sampling has been more widely employed than the other two approaches [1, 14, 23]. Despite its popularity however, its effectiveness has not been rigorously validated. Rigorous validations are required because sampling inevitably distorts the class distribution of training data, increasing the discrepancy between the training data balanced by sampling and inherently imbalanced test data. This increased discrepancy could reduce the test performance of a learned classifier. In addition, undersampling could increase the variance of a learned model due to the reduced training data, resulting in reduced test performance. Therefore, a tradeoff exists between these adverse effects of sampling and their positive effect in reducing bias toward the majority class.

Numerous studies have suggested that data balancing by sampling helps improve the classifiers' performance [12, 24, 25, 29–38]. More than half of these studies [24, 25, 29–33] mainly used decision tree classifiers such as C4.5 and C5.0. Some studies used the naïve Bayes classifier [25] and the k nearest neighbor method [25, 33]. Other studies evaluated the sampling method using linear discriminant analysis (LDA) [35, 36], random forests (RFs) [37], and convolutional neural networks [34]. Only a few studies have compared several classifiers [12, 38]. Japkowicz and Stephen compared C5.0, multilayer perceptrons (MLPs), and support vector machines (SVMs), showing that MLPs and SVMs were less affected by sampling than C5.0 [12]. Khushi et al. compared SVMs, logistic regression, and RFs to analyze the effect of sampling for imbalanced classification [38]. They recommended oversampling combined with RFs for the prediction of lung cancer incidence.

One limitation of previous studies is that they did not comprehensively analyze the effect of sampling using a wide range of popular classifiers including regularized logistic regression and boosting. Another limitation is that most previous studies used two famous performance

measures for classification: accuracy and area under the receiver operating characteristics curve (AUROC) [1, 12, 24, 25, 29–31, 34–38]. However, several studies [39–41] suggested that the area under the precision-recall curve (AUPRC) is more informative for evaluating the results of imbalanced classification. To the best of our knowledge, no study has validated the effectiveness of sampling using AUPRC.

Unlike previous studies, our work comprehensively analyzed the effect of seven widely-used sampling methods combined with eight machine learning methods—56 combinations in total—on imbalanced classification using AUPRC and AUROC as the performance indicators. We empirically analyzed the impact of sampling using 31 real-world imbalanced datasets. In this work, we sought to answer the following questions. First, to what extent is the sampling method effective? More precisely, what is the chance of a sampling method to improve the performance of a classifier? Second, which sampling method is more effective than others? Third, which machine learning classifier is likely to be enhanced by sampling? Fourth, do we need sampling to obtain the optimal classifier for imbalanced classification? Finally, what is the effect of performance measures on evaluation of the sampling method? We present our empirical findings related to the questions above and other aspects of imbalanced classification, providing insights on using sampling to tackle the imbalanced classification problem.

Materials and methods

Imbalanced datasets

We used 31 binary classification datasets whose imbalance ratios, i.e., the majority class size divided by the minority class size, ranged from 1.14 to 577.88 [42–44] (see [S1 Table](#) for more details). Among the 31 datasets, 29 were obtained from the UC Irvine Machine Learning Repository (<https://archive.ics.uci.edu>). Among these 29 binary classification datasets, seven were originally binary and the other 22 were obtained from 14 multiclass datasets from the UC Irvine Machine Learning Repository. To develop a binary classification dataset, we grouped a set of classes of a multiclass dataset and designated them as the minority class. The remaining classes of the multiclass dataset were grouped and set as the majority class. According to the grouping procedure, more than one binary classification dataset was prepared from a multiclass dataset (see [S2 Table](#) for more details). This approach of converting multiclass datasets to imbalanced binary classification datasets has been widely employed in previous studies to increase the number of datasets used in the experiment [23–25, 32, 35]. The 30th (Creditcard in [S1 Table](#)) and 31st (Fraud_Detection in [S1 Table](#)) datasets were credit card fraud detection datasets from Kaggle (<https://www.kaggle.com>), which were originally binary. We downloaded all datasets on September 13, 2020, except for the Fraud_Detection dataset, which was downloaded on March 11, 2022. The access URLs for the 31 datasets are shown in [S3 Table](#). Among the 31 datasets, nine were originally binary, and the remaining 22 were made from multiclass datasets (see [S2 Table](#)).

The Fraud_Detection dataset had 395 features. Among them, we removed two features, for which more than 90% of the examples contained missing values. Missing values of the remaining features were imputed by the median if the feature was numerical. Missing values of a categorical feature was denoted by a new category called “missing.” Then, categorical features of the Fraud_Detection dataset were encoded by one-hot encoding if the number of categories was less than five, or by weight of evidence encoding otherwise. For one-hot encoding, we used OneHotEncoder of the scikit-learn package of Python (version 1.1.1). For weight of evidence encoding, we used WOEEncoder of the category_encoders Python package (version 0.1.4). Through the encoding process, the number of features of the Fraud_Detection dataset increased to 410 (see [S1 Table](#)). The 410 features were normalized by removing the mean and

scaling to unit variance using StandardScaler of the scikit-learn package of Python (version 1.1.1). We calculated each feature's mean and variance values using the training dataset. The detailed experimental procedure including the division of the dataset into training and test sets is described in the Performance evaluation and comparison section. All datasets, except for Fraud_Detection, were processed as follows. Categorical features were represented using dummy variables. We used RobustScaler of the scikit-learn package of Python (version 0.24.1) to normalize the features. All features, including dummy variables, were normalized by subtracting the median and dividing by the interquartile range (IQR). We calculated each feature's median and IQR values using the training dataset.

Sampling methods for imbalanced classification

We evaluated three oversampling methods: random oversampling, synthetic minority oversampling technique (SMOTE) [25], and borderline SMOTE [32]. Three undersampling methods, i.e., random undersampling, condensed nearest neighbors undersampling [45], NearMiss2 [33], were also examined. In addition, we tested SMOTETomek [24], which is a hybrid method that combines oversampling and undersampling techniques.

In random oversampling, a set of randomly selected examples of the minority class are duplicated to increase the size of the minority class. SMOTE [25] synthesizes a new instance of the minority class using a randomly selected example of the minority class and its k nearest neighbors of the same class. Borderline SMOTE [32] is a variant of SMOTE that increases the number of examples of the minority class along the borderline in the feature space to focus on the “difficult-to-classify” region. Random undersampling randomly removes examples of the majority class. Condensed nearest neighbors undersampling [45] excludes the majority class example from training if its nearest neighbor is from the same class. NearMiss2 [25, 33] selects examples of the majority class with the shortest average distance from the three farthest examples of the minority class. SMOTETomek [24] is a hybrid method in which SMOTE is applied first and then, an undersampling method (the Tomek links method) is performed. The Tomek links method removes a pair of the majority and minority class examples if they are the nearest neighbors of each other. A detailed description of the seven sampling methods is provided in [S1 File](#).

All sampling methods, except for condensed nearest neighbors undersampling, were used to balance the class distribution, making the majority to minority classes ratio one to one. The resulting class distribution by condensed nearest neighbors undersampling varies depending on the composition of a given dataset. We show the class distribution of the 31 datasets balanced by condensed nearest neighbors undersampling in [S4 Table](#).

For conciseness and clarity, we denote random oversampling by O_Random, SMOTE by O_SMOTE, borderline SMOTE by O_Border, random undersampling by U_Random, condensed nearest neighbors undersampling by U_Condensed, and NearMiss2 by U_NearMiss. We used the implementation of the seven sampling methods from version 0.7.0 of the imbalanced-learn package of Python (<https://imbalanced-learn.org>) [46].

Machine learning methods

We evaluated the seven sampling methods using eight machine learning methods, which included four non-linear methods—adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), RFs, and SVMs—and four linear methods—the LDA and three regularized logistic regression methods, i.e., ridge, lasso, and elastic net. For XGBoost, we used the XGBoost Python package (version 1.4.2) (<https://xgboost.readthedocs.io>). For the other machine learning methods, we used scikit-learn package of Python version 0.24.1 (<https://scikit-learn.org>)

[47]. We adopted the radial basis function kernel for SVMs. The hyperparameters of each machine learning method were optimized using cross-validation (CV) on the training dataset. We used the whole training dataset to optimize the hyperparameters except when optimizing the hyperparameters of SVMs for the largest dataset, i.e., Fraud_Detection (590,540 examples x 410 features) (see S1 Table). This dataset was more than seven times larger than the second largest dataset, i.e., Covtype4 (581,012 examples x 54 features). The training time for SVMs is affected more by the size of the training dataset than the other machine learning methods. Therefore, 10% of the training dataset of Fraud_Detection was randomly selected and used for hyperparameter optimization of SVMs. The hyperparameter optimization process is detailed in the next subsection.

Performance evaluation and comparison

We used both AUPRC [48] and AUROC [49] to evaluate the classification performance. These performance measures were calculated using version 0.24.1 of the scikit-learn package of Python (<https://scikit-learn.org>) [47]. To compare the performances of the two classifiers on an imbalanced classification dataset, we performed the 5x2 CV paired *t*-test, as described in [50], in which a stratified two-fold CV is repeated five times. The 5x2 CV procedure for evaluating the effectiveness of the sampling methods is as follows, and a schematic diagram is shown in Fig 1.

First, an imbalanced classification dataset was randomly divided into two folds: fold 1 and fold 2. The class ratio in each fold was similar to that of the original imbalanced dataset. Fold 1 was used as the training dataset. A sampling method was applied to fold 1, thereby balancing the class ratio. Then, a machine learning classifier was learned from the balanced training dataset, i.e., balanced fold 1. When training the classifier, we optimized its hyperparameters using 3-fold CV on balanced fold 1. A grid search over a range of hyperparameter values was performed using `sklearn.model_selection.GridSearchCV` from the scikit-learn package of Python (version 1.1.1). A list of the optimized hyperparameters of each machine learning method and their search ranges are described in S5 Table. The same performance measure was used for evaluating the classifier—AUPRC or AUROC—when optimizing the hyperparameters. The classifier learned with sampling was evaluated using fold 2 as the test dataset. We denote the test performance with sampling by $P_1^{(1) \text{ sampling}}$. For comparison, a classifier without sampling was learned using the original fold 1 as the training dataset. The hyperparameters of the classifier were optimized on fold 1. The learned classifier without sampling was evaluated using fold 2. We denote the test performance without sampling by $P_1^{(1) \text{ nosampling}}$. Then, the first performance difference value due to sampling was calculated as $P_1^{(1)} = P_1^{(1) \text{ nosampling}} - P_1^{(1) \text{ sampling}}$. The above procedure was repeated using fold 2 as the training dataset and fold 1 as the test dataset, resulting in the second performance difference value due to sampling: $P_1^{(2)} = P_1^{(2) \text{ nosampling}} - P_1^{(2) \text{ sampling}}$. The procedure for obtaining two performance difference values was iterated five times, with repeated random division of the imbalanced dataset, producing five pairs of performance difference values: $(P_1^{(1)}, P_1^{(2)})$, $(P_2^{(1)}, P_2^{(2)})$, $(P_3^{(1)}, P_3^{(2)})$, $(P_4^{(1)}, P_4^{(2)})$, and $(P_5^{(1)}, P_5^{(2)})$. Then, the *t*-statistic for the test was calculated as follows [50]:

$$t = \frac{P_1^{(1)}}{\sqrt{\frac{1}{5} \sum_{i=1}^5 s_i^2}}. \quad (1)$$

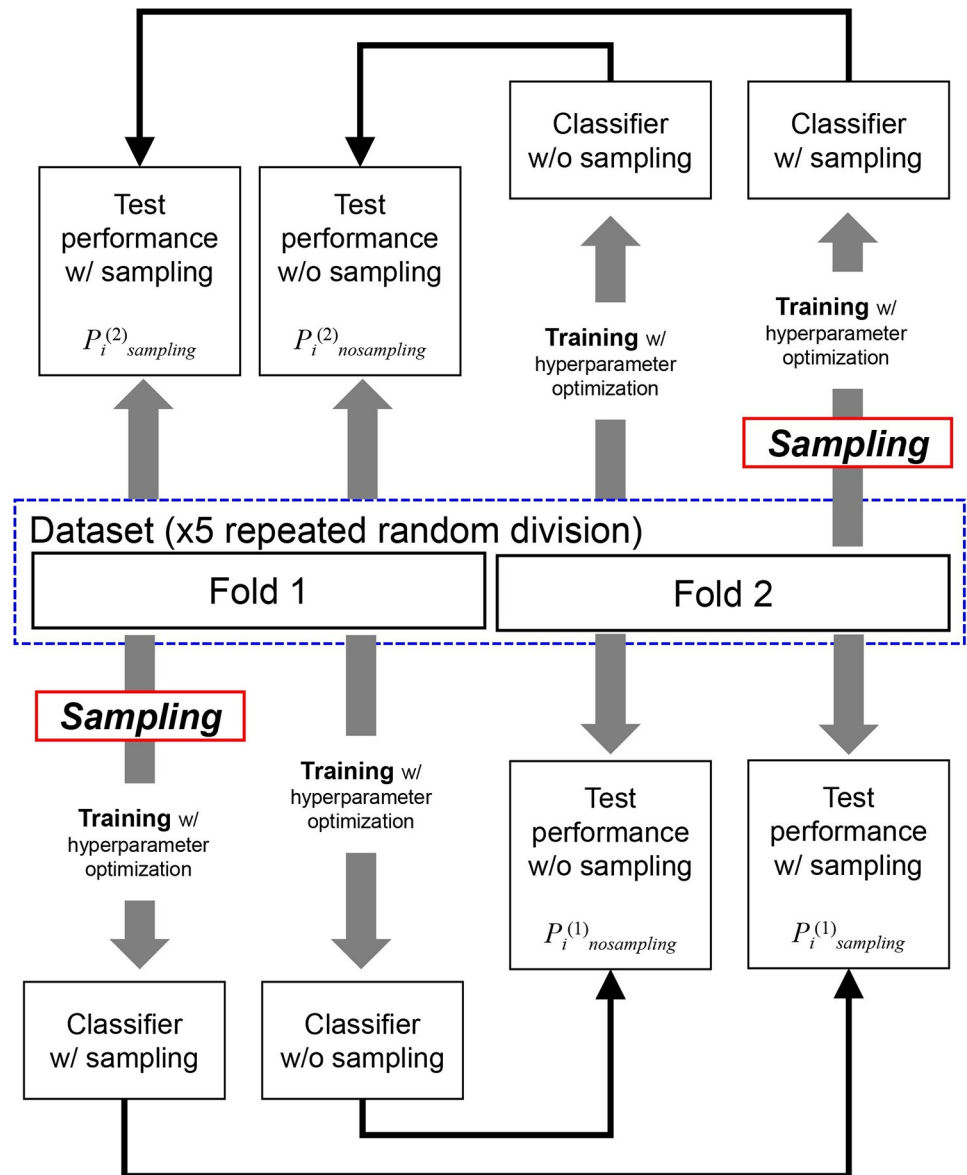


Fig 1. Schematic diagram of the workflow for evaluating the effectiveness of sampling for imbalanced classification. The process is repeated five times ($i = 1, 2, 3, 4, 5$) with repeated random division of an imbalanced classification dataset.

<https://doi.org/10.1371/journal.pone.0271260.g001>

In Eq (1), s_i^2 denotes the estimated variance of the i th iteration ($i = 1, 2, 3, 4, 5$), calculated as follows.

$$s_i^2 = (P_i^{(1)} - \bar{P}_i)^2 + (P_i^{(2)} - \bar{P}_i)^2, \tag{2}$$

where $\bar{P}_i = (P_i^{(1)} + P_i^{(2)})/2$. To evaluate the effectiveness of sampling for imbalanced classification, we performed a two-tailed test using the t -statistic.

Results

Effectiveness of sampling methods

To investigate the effectiveness of the sampling methods for imbalanced classification, we checked whether sampling changed the classification performance of a machine learning algorithm. The hyperparameters of each machine learning algorithm (see [S5 Table](#)) were optimized as described in the Materials and Methods section, and the optimized hyperparameter values are provided as [S1](#) and [S2](#) Datasets. The comparison results for each combination of machine learning and sampling methods on the 31 datasets are shown in [Figs 2](#) (for AUPRC) and [3](#) (for AUROC).

[Figs 2](#) and [3](#) show that there are few cases of statistically significant performance changes due to sampling. Among the 1736 (combinations of seven sampling and eight machine learning methods, applied to 31 datasets) cases, only 211 (12.2%) and 173 (10.0%) showed statistically significant differences (paired t -tests $P < 0.05$) in the AUPRC and AUROC, respectively. Surprisingly, we observed more cases of performance degradation than improvement, indicating that sampling could be more harmful than beneficial for imbalanced classification. Among the cases of performance changes, the proportion of cases with decreased AUPRC and AUROC was 78.7% (166 of 211) and 61.3% (106 of 173), respectively. These observations also suggest that sampling is more effective when measuring the performance using AUROC than when using AUPRC. While more cases with decreased AUPRC (166 cases) were observed than those with decreased AUROC (106 cases), the number of cases with increased AUROC (67 cases) was greater than that with increased AUPRC (45 cases).

We divided the 31 imbalanced datasets used in the experiment into two categories: nine originally binary datasets and 22 binary datasets made from multiclass datasets by merging two or more classes (see [Materials and Methods](#) and [S2 Table](#)). The numbers of cases in the nine originally binary datasets and the other 22 datasets were 504 (combinations of seven sampling and eight machine learning methods, applied to nine datasets) and 1232 (combinations of seven sampling and eight machine learning methods, applied to 22 datasets), respectively. Interestingly, sampling effectiveness differed by the category of datasets. The proportion of cases showing statistically significant performance changes due to sampling was much higher for the originally binary datasets [17.3% (87 of 504) in AUPRC and 15.3% (77 of 504) in AUROC] than for the other datasets [10.1% (124 of 1232) in AUPRC and 7.8% (96 of 1232) in AUROC]. However, both dataset categories showed a similar pattern, i.e., sampling was more likely to reduce than improve the classification performance. In AUPRC, 73.6% (64 of 87) of the cases for the originally binary datasets and 82.3% (102 of 124) of the cases for the other datasets exhibited performance degradation. The AUROC was reduced in 54.5% (42 of 77) of the cases for the originally binary datasets and 66.7% (64 of 96) of the cases for the other datasets. Furthermore, the effect of performance measures was the same for both categories of datasets. The number of cases with increased AUROC (35 cases for the originally binary datasets and 32 cases for the other datasets) was greater than that with increased AUPRC (23 cases for the originally binary datasets and 22 cases for the other datasets), whereas more cases with decreased AUPRC (64 cases for the originally binary datasets and 102 cases for the other datasets) were observed than those with decreased AUROC (42 cases for the originally binary datasets and 64 cases for the other datasets).

We examined the relationship between the effectiveness of the sampling methods and the degree of imbalance of the data. [S1 Fig](#) shows the scatterplots of the number of cases of performance reduction or improvement against the ratio (in a logarithmic scale) of the majority to minority classes for the 31 datasets. To each dataset, 56 combinations of sampling and machine learning methods were applied (see [Materials and Methods](#)). We observed small but

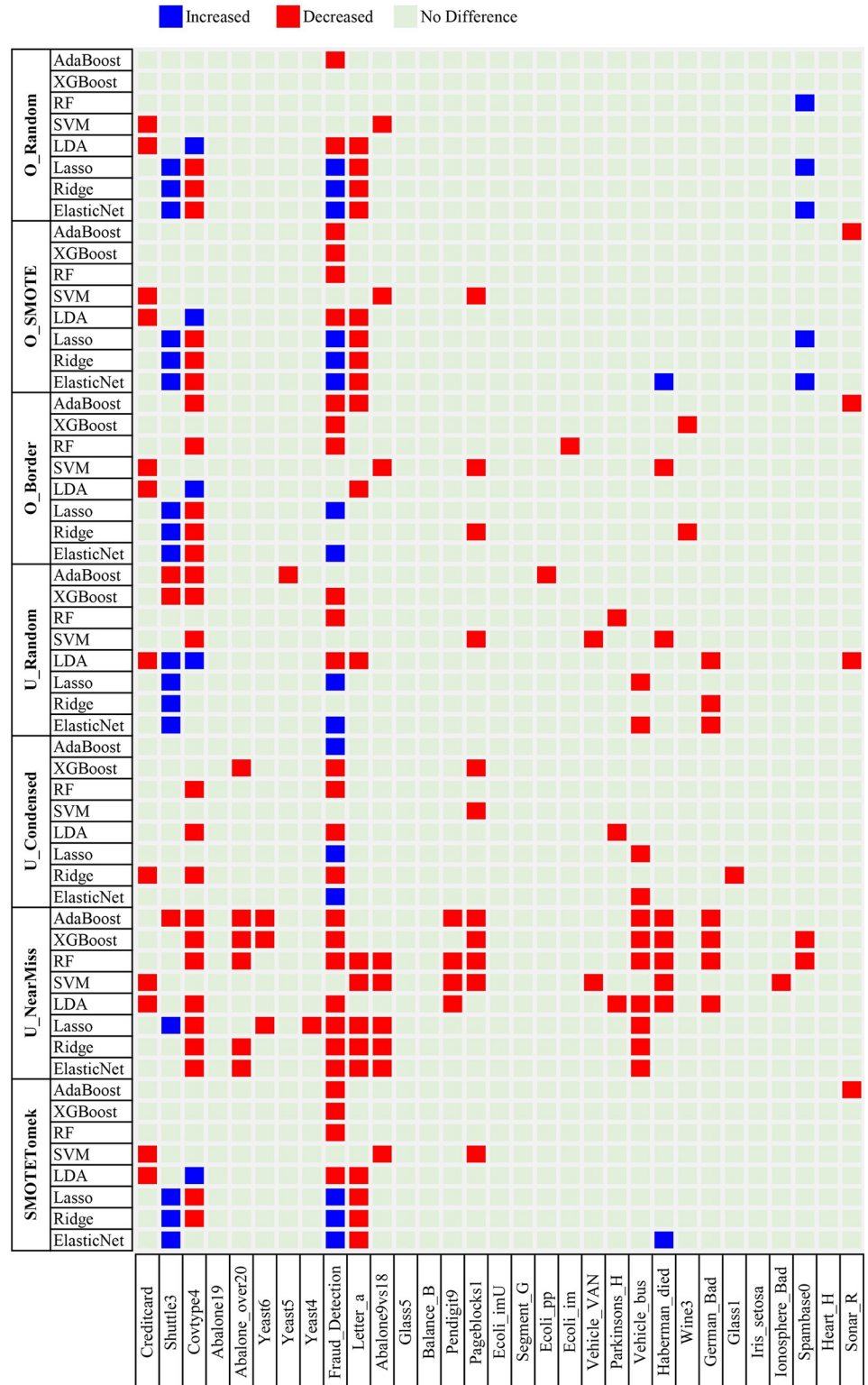


Fig 2. Heatmap of the difference in the area under the precision-recall curve between classification with and without sampling on the 31 imbalanced datasets. Combinations of the seven sampling methods [i.e., random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), borderline synthetic minority oversampling technique (O_Border), random undersampling (U_Random), condensed nearest neighbors undersampling (U_Condensed), NearMiss2 (U_NearMiss), and SMOTETomek] and eight machine learning methods

[i.e., adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), random forests (RFs), support vector machines (SVMs), the linear discriminant analysis (LDA), lasso, ridge, and elastic net] were compared using the 31 datasets.

<https://doi.org/10.1371/journal.pone.0271260.g002>

statistically significant positive correlations between the logarithmic imbalance ratio and the number of cases of performance improvement or degradation due to sampling (the Pearson correlation coefficient tests $P < 0.05$). The R^2 value was 0.13 for an increase in AUPRC, 0.14 for a decrease in AUPRC, 0.11 for an increase in AUROC, and 0.11 for a decrease in AUROC. The positive correlation is attributed to the fact that the amount of training data modification caused by sampling is proportional to the imbalance ratio.

Effectiveness comparison of sampling methods

We evaluated and compared the effectiveness of seven sampling methods using the number of cases in which each sampling method enhanced or reduced the classification performance. The total number of cases for each sampling method was 248 (eight machine learning methods and 31 datasets). The results of the comparison are shown in Fig 4.

The best method to improve the AUPRC and AUROC was O_Random. O_SMOTE was also the best for increasing the AUPRC. U_NearMiss and U_Condensed were the least effective in improving the classification performance in terms of AUPRC and AUROC, respectively. Notably, U_NearMiss decreased both AUPRC and AUROC in much larger numbers of cases compared with the others. U_NearMiss was also observed to perform poorly in a previous study [38]. On average, undersampling reduced the classification performance in more cases than oversampling and hybrid methods. One explanation for the relatively stronger negative effect of undersampling is that some characteristics of the data that are helpful for discriminating the majority class are removed during the process of eliminating examples of the majority class [16]. Moreover, the application of undersampling could increase the model variance by reducing the training dataset size. Furthermore, we found that AUPRC was more likely to be reduced by sampling than AUROC regardless of the sampling methods applied.

Comparison of machine learning methods by the effectiveness of sampling

Next, we investigated whether machine learning model selection made a difference in the effectiveness of sampling. We compared the number of cases in which sampling improved or reduced the performance of each machine learning classifier. A total of 217 cases were studied for each classifier (seven sampling methods and 31 datasets). Fig 5 shows the comparison results.

We found that sampling was much more effective in improving the linear machine learning methods (i.e., the LDA, lasso, ridge, and elastic net) than the non-linear methods (i.e., AdaBoost, XGBoost, RFs, and SVMs). The elastic net and LDA classifiers showed the largest number of AUPRC and AUROC enhancement cases, respectively. In contrast, sampling did not enhance the AUPRC (AUROC) of XGBoost and SVMs (AdaBoost). The effectiveness of sampling in improving the AUROC of the LDA agreed with the findings of a previous study [36].

Regarding the performance reduction by sampling, regularized logistic regression was less affected than the others. In terms of AUPRC, we found that lasso, ridge, and elastic net were more resistant to performance degradation due to sampling than the others, on average. The reduction in AUROC was less pronounced for the linear classifiers, i.e., the LDA and regularized logistic regression methods, compared with the non-linear ones. SVMs were by far the most negatively affected by sampling in AUROC. Indeed, SVMs have been reported to underperform when coupled with undersampling [12].

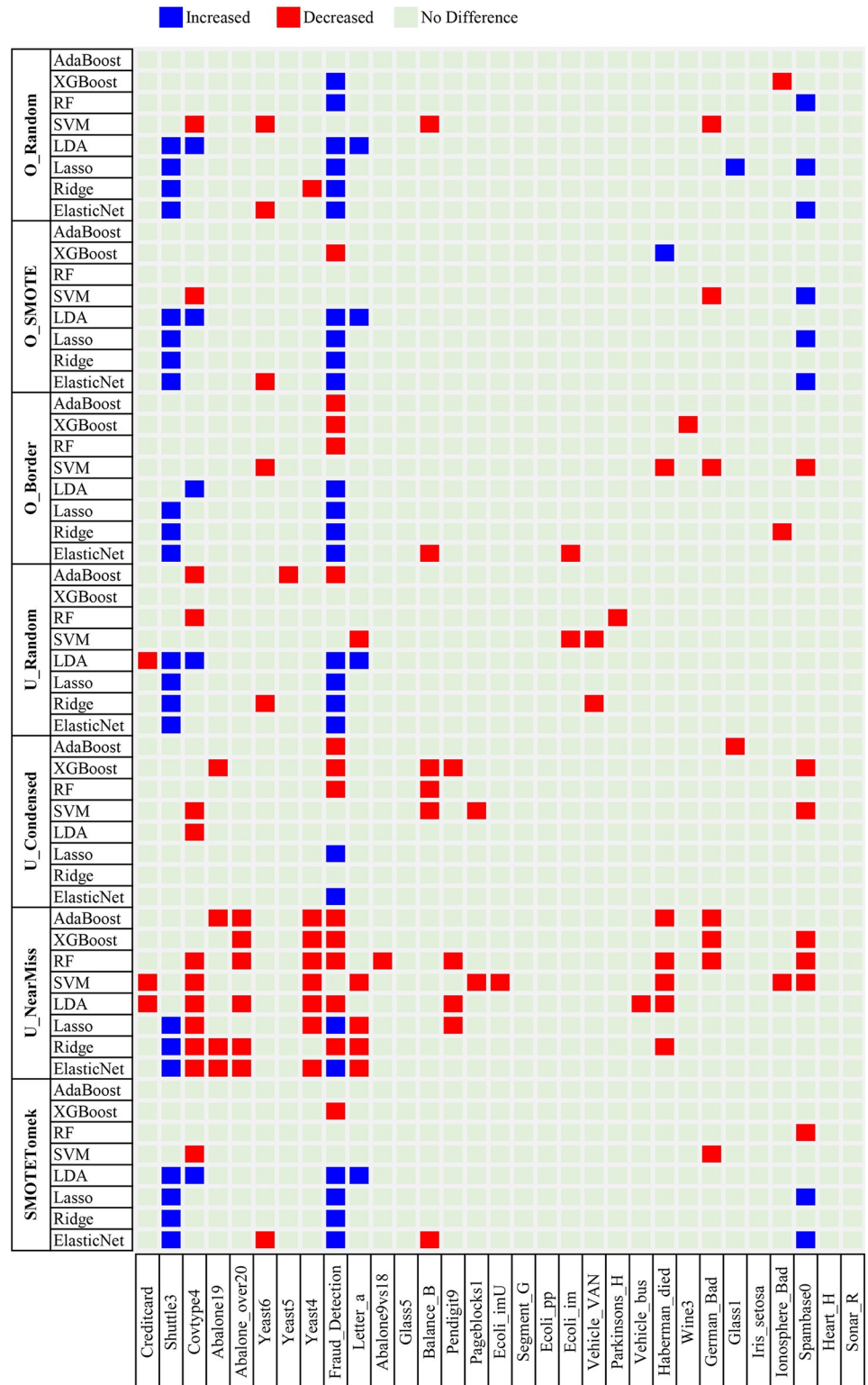


Fig 3. Heatmap of the difference in the area under the receiver operating characteristics curve between classification with and without sampling on the 31 imbalanced datasets. Combinations of the seven sampling methods [i.e., random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), borderline synthetic minority oversampling technique (O_Border), random undersampling (U_Random), condensed nearest neighbors undersampling (U_Condensed), NearMiss2 (U_NearMiss), and SMOTETomek] and eight machine

learning methods [i.e., adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), random forests (RFs), support vector machines (SVMs), the linear discriminant analysis (LDA), lasso, ridge, and elastic net] were compared using the 31 datasets.

<https://doi.org/10.1371/journal.pone.0271260.g003>

Performance comparison of combinations of machine learning and sampling methods

In the previous subsections, we focused on whether sampling improved or reduced the classification performance of a particular machine learning algorithm. Although sampling enhances

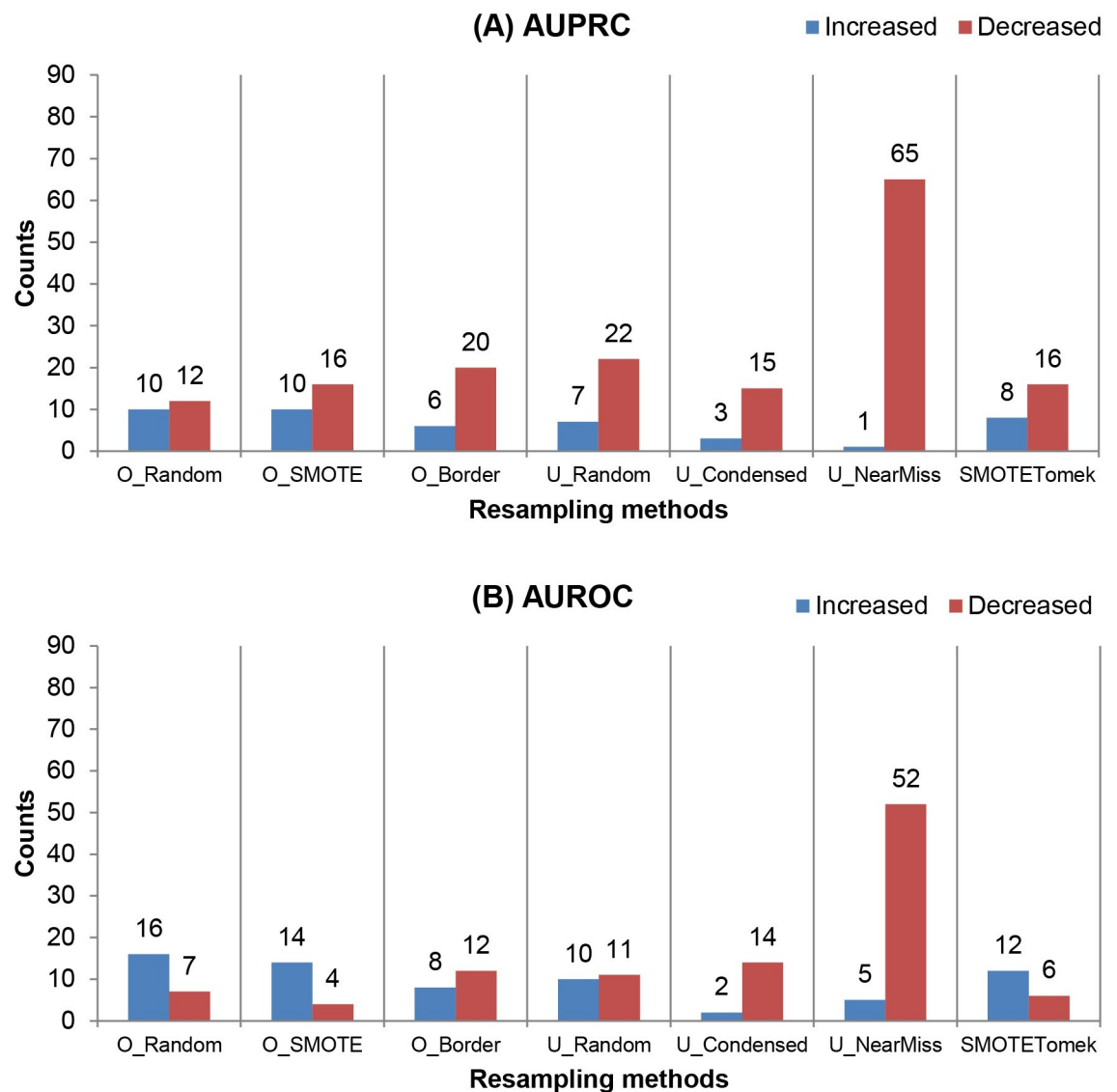


Fig 4. Comparison of the effectiveness of the seven sampling methods. The number of cases in which a sampling method enhanced (blue) or reduced (red) the performance in (A) the area under the precision-recall curve (AUPRC) and (B) the area under the receiver operating characteristics curve (AUROC) is shown. Seven sampling methods—random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), borderline synthetic minority oversampling technique (O_Border), random undersampling (U_Random), condensed nearest neighbors undersampling (U_Condensed), NearMiss2 (U_NearMiss), and SMOTETomek—were compared.

<https://doi.org/10.1371/journal.pone.0271260.g004>

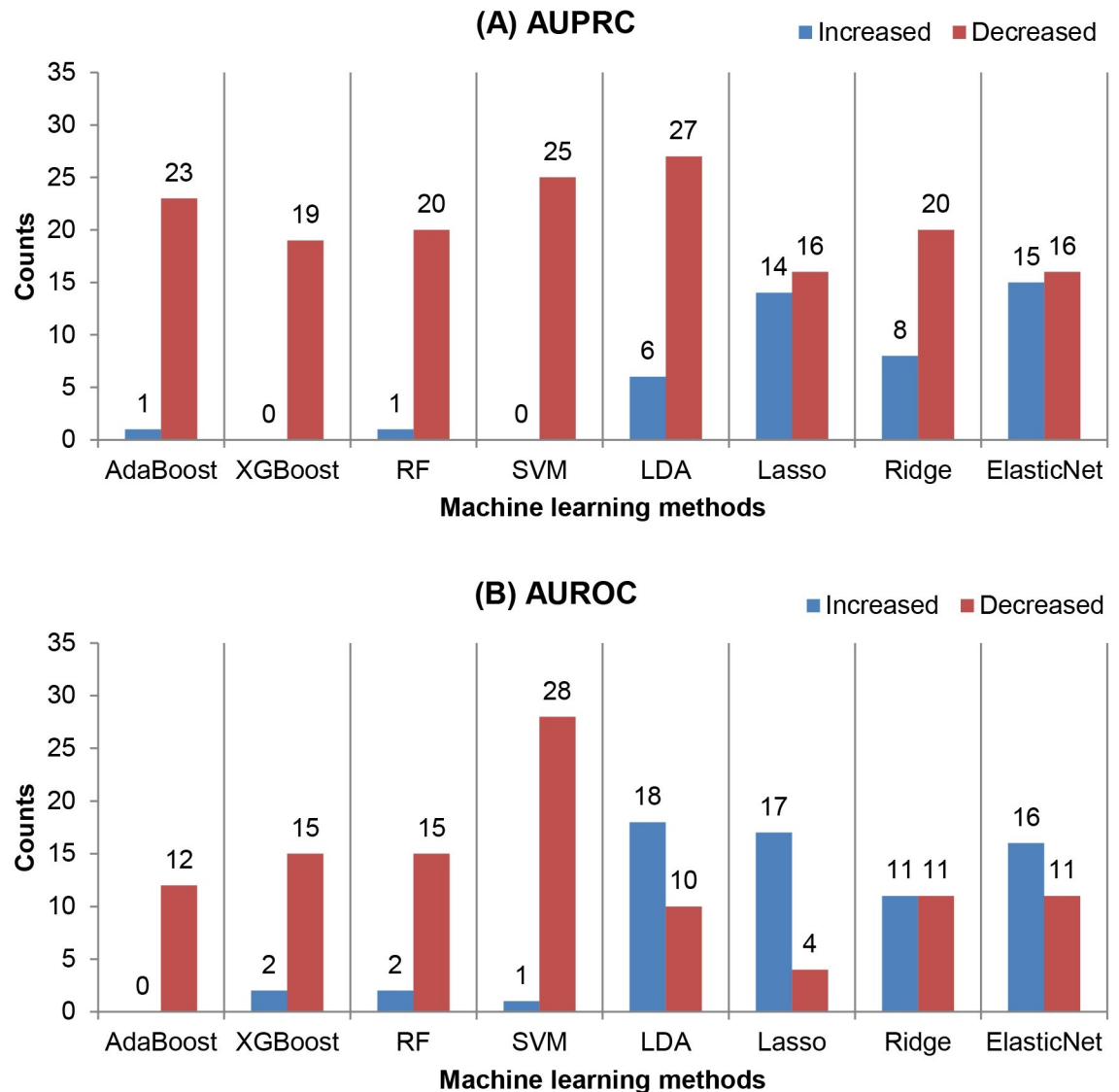


Fig 5. Comparison of machine learning methods by the effectiveness of sampling. The number of cases in which (A) the area under the precision-recall curve (AUPRC) and (B) the area under the receiver operating characteristics curve (AUROC) of a machine learning method were improved (blue) or reduced (red) by sampling. Eight machine learning methods—adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), random forests (RFs), support vector machines (SVMs), the linear discriminant analysis (LDA), lasso, ridge, and elastic net—were compared.

<https://doi.org/10.1371/journal.pone.0271260.g005>

the performance of machine learning methods, the method could perform worse than other machine learning methods without sampling. To investigate whether sampling was helpful for achieving the best performance for imbalanced classification, we compared the classification performance of each combination of machine learning and sampling methods, including machine learning without sampling. Tables 1 and 2 summarize the comparison results by showing the number of datasets on which a combination achieved the best classification performance in terms of the AUPRC and AUROC, respectively. Note that more than one combination could perform best on a dataset. Furthermore, a combination was regarded to be the best if its performance was not significantly different from that of the best one by the two-tailed 5x2 CV paired *t*-test ($P \geq 0.05$).

Table 1. Number of datasets on which a combination of machine learning and sampling methods performed the best in terms of the area under the precision-recall curve.

	Ada Boost	XG Boost	RF	SVM	LDA	Lasso	Ridge	Elastic net	All
Without sampling	14	21	23	17	13	10	10	11	29
O_Random	11	22	23	9	13	11	11	11	29
O_SMOTE	13	20	23	13	15	11	13	11	29
O_Border	10	16	22	11	12	9	11	9	28
U_Random	9	11	16	10	11	11	10	10	23
U_Condensed	9	14	19	12	10	8	9	9	22
U_NearMiss	5	11	9	8	5	6	6	6	13
SMOTETomek	14	20	22	13	15	11	13	11	28
All	18	25	26	19	16	13	15	14	

Seven sampling methods were compared: random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), borderline synthetic minority oversampling technique (O_Border), random undersampling (U_Random), condensed nearest neighbors undersampling (U_Condensed), NearMiss2 (U_NearMiss), and SMOTETomek. Eight machine learning methods were compared: adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), random forests (RFs), support vector machines (SVMs), linear discriminant analysis (LDA), lasso, ridge, and elastic net. 'All' means the number considering all sampling methods (including without sampling) or machine learning methods.

<https://doi.org/10.1371/journal.pone.0271260.t001>

For most datasets, sampling was not essential for attaining the best classification performance in terms of the AUPRC (29 of 31 datasets) and AUROC (30 of 31 datasets). The appropriate choice of machine learning algorithms without sampling was enough to obtain the optimal result for those datasets. Moreover, no sampling method showed the best performance on more datasets than “without sampling.” O_Random and O_SMOTE achieved the highest AUPRC values on the same number (29) of datasets as “without sampling.” In terms of AUROC, only O_SMOTE performed best on the same number (30) of datasets as “without sampling.” Other sampling methods achieved the best performance on fewer datasets, suggesting that many sampling methods could be more harmful than beneficial for imbalanced classification. For building the optimal classifier, undersampling was worse than oversampling and hybrid methods, as shown in Tables 1 and 2. Regarding these results, we did not observe a

Table 2. Number of datasets on which a combination of machine learning and sampling methods performed the best in terms of the area under the receiver operating characteristics curve.

	Ada Boost	XG Boost	RF	SVM	LDA	Lasso	Ridge	Elastic net	All
Without sampling	14	19	24	17	13	14	16	14	30
O_Random	13	19	21	8	13	17	18	17	29
O_SMOTE	15	18	25	11	13	17	18	17	30
O_Border	11	15	21	9	13	13	15	14	28
U_Random	10	16	20	11	14	16	15	16	27
U_Condensed	7	13	17	12	10	13	13	12	23
U_NearMiss	7	9	8	7	7	7	7	7	14
SMOTETomek	16	18	25	12	13	17	18	17	29
All	19	26	28	23	18	17	18	17	

Seven sampling methods were compared: random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), borderline synthetic minority oversampling technique (O_Border), random undersampling (U_Random), condensed nearest neighbors undersampling (U_Condensed), NearMiss2 (U_NearMiss), and SMOTETomek. Eight machine learning methods were compared: adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), random forests (RFs), support vector machines (SVMs), linear discriminant analysis (LDA), lasso, ridge, and elastic net. 'All' means the number considering all sampling methods (including without sampling) or machine learning methods.

<https://doi.org/10.1371/journal.pone.0271260.t002>

considerable difference between the categories of datasets (see [Materials and Methods](#)). Sampling was not required for achieving the best performance on most originally binary and multiclass datasets (see [S6–S9 Tables](#)).

Two examples showing contradictory evaluation results depending on the performance measure

In the previous subsections, we found that the evaluation results of the effectiveness of sampling differed according to the performance measures. More precisely, the positive (negative) effect of sampling for imbalanced classification was more (less) pronounced when we measured the performance in terms of AUROC than in terms of AUPRC (see [Figs 2 and 3](#)). Because a classifier with the optimal AUROC is not guaranteed to attain the optimal AUPRC [48], using the inappropriate performance measure could mislead the decision-making process.

In this regard, we show two remarkable examples in which the direction of performance changed, i.e., improvement or reduction was reversed depending on the performance measure. The AUROC of the LDA on the Letter_a and the Fraud_Detection datasets (see [S1 Table](#)) was significantly improved (paired t -tests $P < 0.05$) due to four sampling methods, i.e., O_Random, O_SMOTE, U_Random, and SMOTETomek. In comparison, the AUPRC of the same classification method on the same dataset was significantly reduced (paired t -tests $P < 0.05$) by the same four sampling methods. [Tables 3 and 4](#) respectively show the performance of the

Table 3. Performance of linear discriminant analysis (LDA) on the Letter_a dataset with and without the four sampling methods.

Sampling methods	AUPRC		AUROC	
	Mean±standard deviation	P values	Mean±standard deviation	P values
Without sampling	0.8917±0.0097	N/A	0.9765±0.0041	N/A
O_Random	0.8457±0.0217	(D) 0.0169	0.9851±0.0038	(I) 0.0017
O_SMOTE	0.8412±0.0209	(D) 0.0123	0.9848±0.0038	(I) 0.0016
U_Random	0.8506±0.0217	(D) 0.0050	0.9851±0.0037	(I) 0.0012
SMOTETomek	0.8412±0.0209	(D) 0.0123	0.9848±0.0038	(I) 0.0016

Four sampling methods were compared: random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), random undersampling (U_Random), and SMOTETomek. Performances were evaluated by the areas under the precision-recall curve (AUPRC) and the receiver operating characteristics curve (AUROC). The means and standard deviations of 5x2 cross-validation (CV) are shown. P values were calculated by the one-tailed 5x2 CV paired t -test. (D) and (I) in front of the P values represent “decreased” and “increased” compared to the performance of the LDA without sampling, respectively.

<https://doi.org/10.1371/journal.pone.0271260.t003>

Table 4. Performance of linear discriminant analysis (LDA) on the Fraud_Detection dataset with and without the four sampling methods.

Sampling methods	AUPRC		AUROC	
	Mean±standard deviation	P values	Mean±standard deviation	P values
Without sampling	0.4474±0.0038	N/A	0.8697±0.0014	N/A
O_Random	0.3978±0.0056	(D) 0.0029	0.8813±0.0010	(I) 0.0015
O_SMOTE	0.4005±0.0043	(D) 0.0056	0.8802±0.0013	(I) 0.0031
U_Random	0.3547±0.0086	(D) 0.0003	0.8796±0.0010	(I) 0.0031
SMOTETomek	0.4005±0.0043	(D) 0.0056	0.8802±0.0013	(I) 0.0031

Four sampling methods are compared: random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), random undersampling (U_Random), and SMOTETomek. Performances were evaluated by the areas under the precision-recall curve (AUPRC) and the receiver operating characteristics curve (AUROC). The means and standard deviations of 5x2 cross-validation (CV) are shown. P values were calculated by the one-tailed 5x2 CV paired t -test. (D) and (I) in front of the P values represent “decreased” and “increased” compared to the performance of the LDA without sampling, respectively.

<https://doi.org/10.1371/journal.pone.0271260.t004>

LDA with and without the four sampling methods on the Letter_a and the Fraud_Detection datasets.

The LDA without sampling achieved a higher AUPRC value (0.8917 ± 0.0097) than the best sampling method, i.e., U_Random (0.8506 ± 0.0217), on the Letter_a dataset, which is a binary dataset obtained from a multiclass dataset (see [Materials and Methods](#) and [S2 Table](#)). In contrast, the AUROC value without sampling (0.9765 ± 0.0041) was lower than those obtained using O_SMOTE and SMOTETomek (0.9848 ± 0.0038), which was the lowest of the four sampling methods on the same dataset. We observed the same results on the Fraud_Detection dataset, which was originally binary (see [S2 Table](#)). The LDA without sampling achieved a higher AUPRC value (0.4474 ± 0.0038) than O_SMOTE and SMOTETomek, which performed the best on the Fraud_Detection dataset in terms of AUPRC (0.4005 ± 0.043). On the same dataset, the AUROC value without sampling (0.8697 ± 0.0014) was lower than U_Random (0.8796 ± 0.0010), which was the lowest among those for the four sampling methods. To understand this difference intuitively, we compared the precision-recall (PR) and receiver operating characteristics (ROC) curves on the test dataset of the first fold of the first iteration of the 5x2 CV run (see [Materials and Methods](#) and [Fig 1](#)) on the two datasets. We show the two curves respectively in [Figs 6](#) and [7](#).

[Figs 6](#) and [7](#) show that the sampling dramatically decreases the precision, especially when the recall is close to zero, leading to decreased AUPRC values. In comparison, the AUROC is improved by sampling because the false positive rate decreases more slowly by sampling as the true positive rate (i.e., recall) approaches one. We observed similar trends from the other folds and iterations of the 5x2 CV run (see [S2](#) and [S3 Figs](#)).

Conclusions and discussion

Imbalanced classification is a critical issue in machine learning that is observed in numerous application areas. Sampling methods are among the most popular approaches to solve imbalanced classification problems owing to their ease of use and broad applicability. Although data balancing by sampling could help to build a classifier that is not biased toward the majority class, it distorts the distribution of training data, potentially reducing the test performance. However, their effectiveness has not been comprehensively tested using a wide array of machine learning algorithms.

In this study, we evaluated seven sampling methods for imbalanced classification combined with eight machine learning methods on 31 real-world imbalanced datasets. We observed that sampling affected only a small number of cases. Notably, sampling was more likely to deteriorate than improve the performance of a classifier. This result seems to disagree with those of previous studies [[12](#), [24](#), [25](#), [29–38](#)] which suggested that sampling is a remedy for the imbalanced classification problem. Our results were largely in accordance with the results of these studies. Many of the previous studies [[24](#), [25](#), [29–33](#)] used decision trees which were not covered in our study. Thus, sampling could be a good option when using decision trees for imbalanced classification. Some previous studies used the machine learning methods covered in our study such as LDA [[35](#), [36](#)], RFs [[37](#), [38](#)], and SVMs with the radial basis function kernel [[12](#)]. The previous results on LDA [[14](#), [35](#), [36](#)] partly agree with ours as we also observed the relatively positive effect of sampling on LDA, especially when the classification performance was measured by AUROC (see [Fig 5B](#)). The results of the other previous studies were not consistent with ours possibly due to the small number of datasets used in the experiments [[12](#), [38](#)] or some uncharacterized differences in the experimental setting [[37](#)].

Among the sampling methods, O_Random and O_SMOTE performed better than the others to improve classification performance. Undersampling reduced the performance more

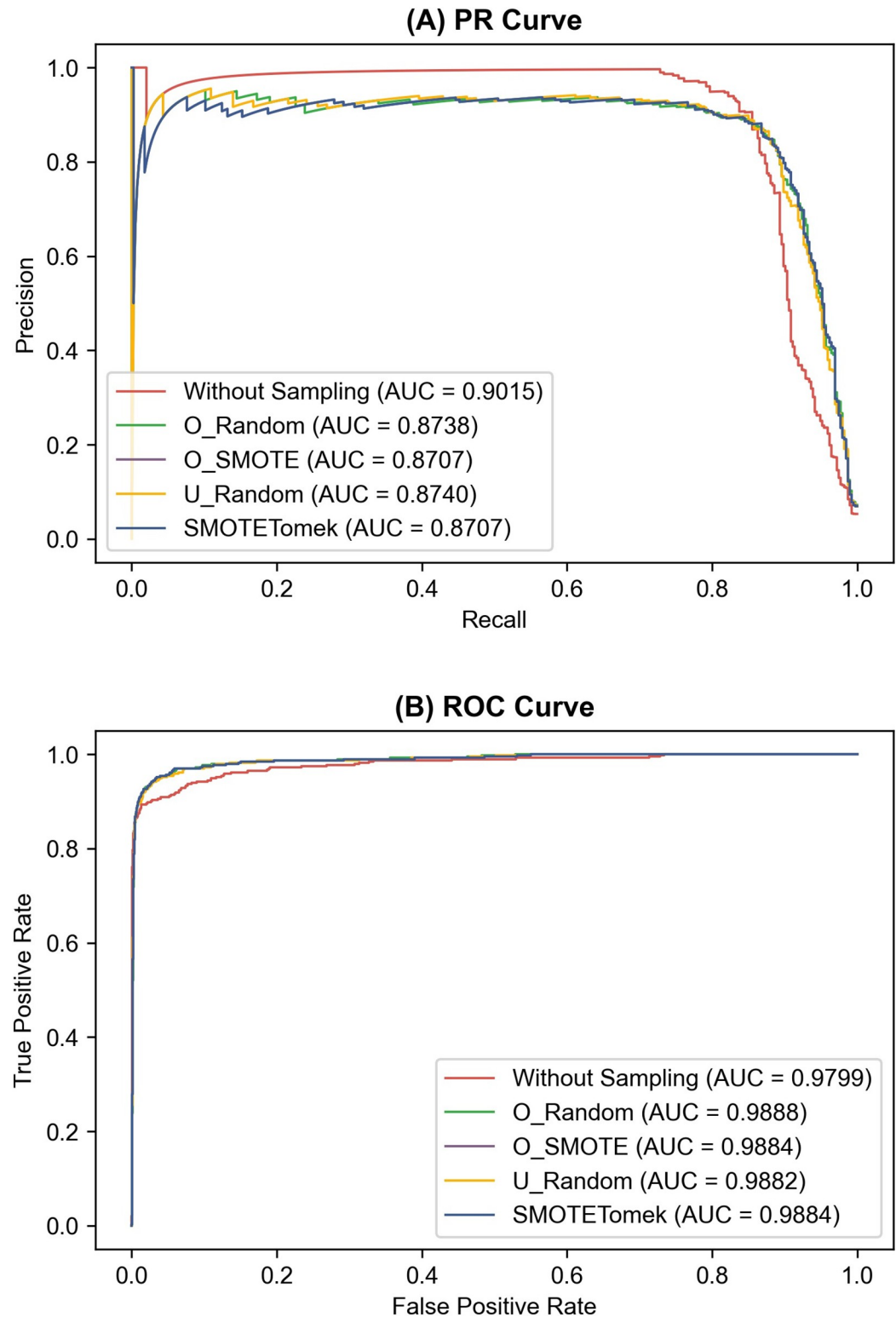


Fig 6. (A) Precision-recall (PR) and (B) receiver operating characteristics (ROC) curves of linear discriminant analysis with and without the four sampling methods on the Letter_a dataset. The PR and ROC curves on the test dataset of the first fold of the first iteration of the 5x2 cross-validation run are shown. Four sampling methods were compared: random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), random undersampling (U_Random), and SMOTETomek. AUC indicates the area under the PR or ROC curve.

<https://doi.org/10.1371/journal.pone.0271260.g006>

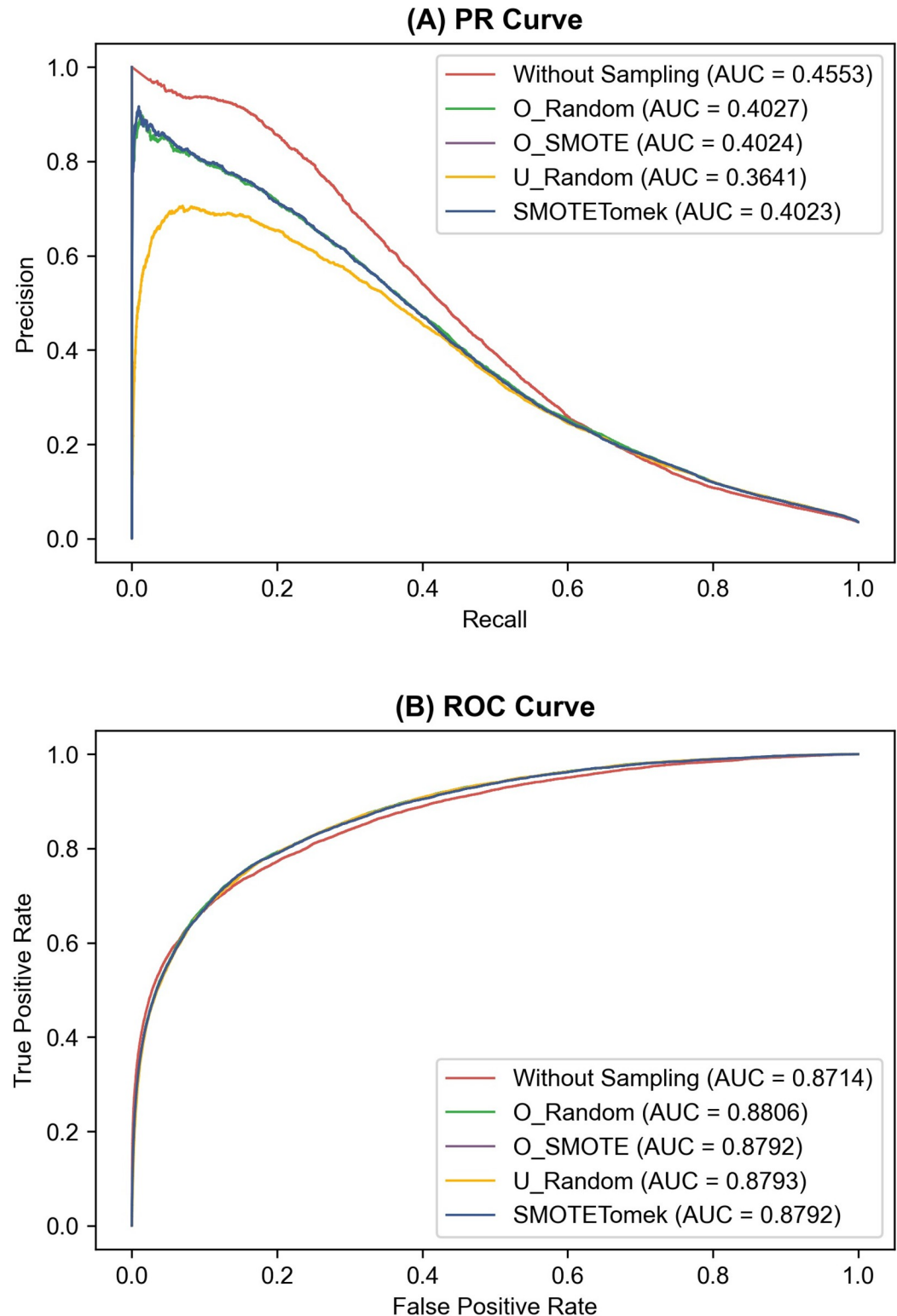


Fig 7. (A) Precision-recall (PR) and (B) receiver operating characteristics (ROC) curves of linear discriminant analysis with and without the four sampling methods on the Fraud_Detection dataset. The PR and ROC curves on the test dataset of the first fold of the first iteration of the 5x2 cross-validation run are shown. Four sampling methods were compared: random oversampling (O_Random), synthetic minority oversampling technique (O_SMOTE), random undersampling (U_Random), and SMOTETomek. AUC means the area under the PR or ROC curve.

<https://doi.org/10.1371/journal.pone.0271260.g007>

often than others. Thus, we conclude that oversampling is generally better than undersampling in terms of the classification performance. Another important aspect to consider when adopting a sampling method is its efficiency. The analysis of time complexity of each of the seven sampling methods is detailed in [S1 File](#). Among the sampling methods, the most efficient one was U_Random, the every-case time complexity of which is $T(n_{minor})$, where n_{minor} denotes the number of minority class examples, which is usually much smaller than the training dataset size. U_Condensed and SMOTETomek are the least efficient. The every-case time complexity of U_Condensed is $T((n_{major}-1)((n-1)d+(n-1)))$, where n_{major} denotes the number of majority class examples, $n = n_{major} + n_{minor}$ and d is the number of features. U_Condensed would be especially inefficient when applied to a severely imbalanced dataset, where n_{major} is similar to n . The every-case time complexity of SMOTETomek is $T((n_{major}-n_{minor})((n_{minor}-1)d+k(n_{minor}-1))d+2n_{major}((2n_{major}-1)d+(2n_{major}-1)))$, where k denotes the number of nearest neighbors used for synthesizing a new example (see [S1 File](#)). Among the two best performing sampling methods, i.e., O_Random and O_SMOTE, O_Random performed better in terms of time complexity: $T((n_{major}-n_{minor})d)$ vs $T((n_{major}-n_{minor})((n_{minor}-1)d+k(n_{minor}-1))d)$. Thus, we propose using O_Random if a sampling method is required.

Compared with non-linear classifiers, the linear ones were more likely to be enhanced by sampling. The choice of performance measure had a crucial impact on the evaluation of the sampling methods. The adverse effect of sampling was more pronounced when the performance was measured using the AUPRC than AUROC. In this regard, we found two interesting examples in which the validation results were reversed depending on the performance measure used. This finding is important because AUPRC is known to be preferable to AUROC when measuring the performance of classifiers on imbalanced datasets [39–41]. It has not been observed from the previous studies. To the best of our knowledge, we are the first to comprehensively evaluate sampling methods using the AUPRC.

Our study provides useful insights on the effectiveness of data balancing by sampling for imbalanced classification. We found that sampling could be ineffective or harmful and is not essential to achieve the optimal classifier from an experiment on a large number of imbalanced classification datasets. These findings have not been identified from the previous studies because of differences in the machine learning method evaluated and the limited number of datasets. Based on our findings, we propose to validate the effectiveness of the sampling methods using multiple machine learning algorithms and an appropriate performance measure before using it. Instead of sampling, one could also use the algorithm level and cost-sensitive learning-based approaches to alleviate the problems caused by imbalanced class ratio, although developing these techniques is challenging. Several such approaches [19–22] have been proposed that are applicable to various imbalanced classification problems.

Directions for future work include evaluating the effectiveness of sampling on multiclass imbalanced datasets and comparing the sampling methods with other approaches such as the algorithm level and the cost-sensitive learning-based approaches. Multiclass imbalanced problems are known to be more difficult than their binary counterparts [51] because more factors, e.g., the configurations with classes of intermediate sizes, have a considerable impact on the classification result. A set of sampling approaches to multiclass imbalanced classification have been proposed [52, 53]. It would be helpful to comprehensively evaluate these approaches using a large number of multiclass imbalanced datasets. Recently, a set of algorithm level approaches, e.g., density-weighted support vector machines [19], the intuitionistic fuzzy kernel ridge regression classifier [20], and kernel-target alignment based fuzzy Lagrangian twin bounded support vector machines [22], and a cost-sensitive learning-based approach, i.e., the robust twin bounded support vector machine [21], were proposed for imbalanced

classification problems. Because these methods were shown to perform well on many imbalanced datasets, it would be interesting to compare them with sampling approaches.

Supporting information

S1 Dataset. Hyperparameter values of the machine learning methods optimized with respect to the area under the precision-recall curve.

(XLSX)

S2 Dataset. Hyperparameter values of the machine learning methods optimized with respect to the area under the receiver operating characteristics curve.

(XLSX)

S1 Table. Descriptions of the 31 datasets used in the experiments.

(DOCX)

S2 Table. Descriptions of the minority class of the 31 datasets.

(DOCX)

S3 Table. URL for the 31 datasets.

(DOCX)

S4 Table. Ratios of the majority to minority classes (in fraction form) of the training dataset modified using the condensed nearest neighbors undersampling method.

(DOCX)

S5 Table. Optimized hyperparameters of the eight machine learning methods.

(DOCX)

S6 Table. Number of originally binary datasets on which a combination of machine learning and sampling methods performed the best in terms of the area under the precision-recall curve.

(DOCX)

S7 Table. Number of originally multiclass datasets on which a combination of machine learning and sampling methods performed the best in terms of the area under the precision-recall curve.

(DOCX)

S8 Table. Number of originally binary datasets on which a combination of machine learning and sampling methods performed the best in terms of the area under the receiver operating characteristics curve.

(DOCX)

S9 Table. Number of originally multiclass datasets on which a combination of machine learning and sampling methods performed the best in terms of the area under the receiver operating characteristics curve.

(DOCX)

S1 Fig. Relationship between the imbalance ratio and the number of cases of performance changes for the 31 datasets.

(DOCX)

S2 Fig. (A), (C), (E), (G), (I), (K), (M), (O), and (Q) Precision-recall (PR) and (B), (D), (F), (H), (J), (L), (N), (P), and (R) receiver operating characteristics (ROC) curves of linear

discriminant analysis with and without four sampling methods on the Letter_a dataset. (DOCX)

S3 Fig. (A), (C), (E), (G), (I), (K), (M), (O), and (Q) Precision-recall (PR) and (B), (D), (F), (H), (J), (L), (N), (P), and (R) receiver operating characteristics (ROC) curves of linear discriminant analysis with and without four sampling methods on the Fraud_Detection dataset. (DOCX)

S1 File.
(DOCX)

Author Contributions

Conceptualization: Kyu-Baek Hwang.

Data curation: Misuk Kim.

Formal analysis: Misuk Kim.

Funding acquisition: Kyu-Baek Hwang.

Investigation: Misuk Kim, Kyu-Baek Hwang.

Methodology: Misuk Kim, Kyu-Baek Hwang.

Project administration: Kyu-Baek Hwang.

Resources: Kyu-Baek Hwang.

Supervision: Kyu-Baek Hwang.

Validation: Misuk Kim, Kyu-Baek Hwang.

Visualization: Misuk Kim.

Writing – original draft: Misuk Kim, Kyu-Baek Hwang.

Writing – review & editing: Misuk Kim, Kyu-Baek Hwang.

References

1. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: Review of methods and applications. *Expert Syst Appl.* 2017; 73:220–39.
2. Bolton RJ, Hand DJ. Statistical fraud detection: A review. *Stat Sci.* 2002; 17(3):235–55.
3. Yang Z, Tang W, Shintemirov A, Wu Q. Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers. *IEEE Trans Syst Man Cybern C Appl Rev.* 2009; 39(6):597–610.
4. Zhu Z-B, Song Z-H. Fault diagnosis based on imbalance modified kernel Fisher discriminant analysis. *Chem Eng Res Des.* 2010; 88(8):936–51.
5. Khreich W, Granger E, Miri A, Sabourin R. Iterative Boolean combination of classifiers in the ROC space: An application to anomaly detection with HMMs. *Pattern Recognit.* 2010; 43(8):2732–52.
6. Tavallaee M, Stakhanova N, Ghorbani AA. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Trans Syst Man Cybern C Appl Rev.* 2010; 40(5):516–24.
7. Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Netw.* 2008; 21(2–3):427–36. <https://doi.org/10.1016/j.neunet.2007.12.031> PMID: 18272329
8. Bermejo P, Gámez JA, Puerta JM. Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. *Expert Syst Appl.* 2011; 38(3):2072–80.
9. Liu Y-H, Chen Y-T. Total margin based adaptive fuzzy support vector machines for multiview face recognition. 2005 IEEE International Conference on Systems, Man and Cybernetics: IEEE; 2005. p. 1704–11.

10. Kubat M, Holte RC, Matwin S. Machine learning for the detection of oil spills in satellite radar images. *Mach Learn.* 1998; 30(2):195–215.
11. López V, Fernández A, García S, Palade V, Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf Sci (N Y).* 2013; 250:113–41.
12. Japkowicz N, Stephen S. The class imbalance problem: A systematic study. *Intelligent Data Analysis.* 2002; 6(5):429–49.
13. Chawla NV, Japkowicz N, Kotcz A. Special issue on learning from imbalanced data sets. *SIGKDD Explor.* 2004; 6(1):1–6.
14. Sun Y, Wong AK, Kamel MS. Classification of imbalanced data: A review. *Intern J Pattern Recognit Artif Intell.* 2009; 23(4):687–719.
15. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng.* 2009; 21(9):1263–84.
16. Quinlan JR. Improved estimates for the accuracy of small disjuncts. *Mach Learn.* 1991; 6(1):93–8.
17. Zadrozny B, Elkan C. Learning and making decisions when costs and probabilities are both unknown. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining;* 2001. p. 204–13.
18. Wu G, Chang EY. KBA: Kernel boundary alignment considering imbalanced data distribution. *IEEE Trans Knowl Data Eng.* 2005; 17(6):786–95.
19. Hazarika BB, Gupta D. Density-weighted support vector machines for binary class imbalance learning. *Neural Comput Appl.* 2021; 33(9):4243–61.
20. Hazarika BB, Gupta D, Borah P. An intuitionistic fuzzy kernel ridge regression classifier for binary classification. *Appl Soft Comput.* 2021; 112:107816.
21. Borah P, Gupta D. Robust twin bounded support vector machines for outliers and imbalanced data. *Appl Intell (Dordr).* 2021; 51(8):5314–43.
22. Gupta U, Gupta D. Kernel-Target Alignment Based Fuzzy Lagrangian Twin Bounded Support Vector Machine. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.* 2021; 29(05):677–707.
23. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern C Appl Rev.* 2011; 42(4):463–84.
24. Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor.* 2004; 6(1):20–9.
25. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res.* 2002; 16:321–57.
26. Chawla NV, Cieslak DA, Hall LO, Joshi A. Automatically countering imbalance and its empirical relationship to cost. *Data Min Knowl Discov.* 2008; 17(2):225–52.
27. Freitas A, Costa-Pereira A, Brazdil P. Cost-sensitive decision trees applied to medical data. *International Conference on Data Warehousing and Knowledge Discovery:* Springer; 2007. p. 303–12.
28. Krawczyk B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence.* 2016; 5(4):221–32.
29. Weiss GM, Provost F. The effect of class distribution on classifier learning: an empirical study. Rutgers University, 2001.
30. Laurikkala J. Improving identification of difficult small classes by balancing class distribution. *Conference on Artificial Intelligence in Medicine in Europe:* Springer; 2001. p. 63–6.
31. Estabrooks A, Jo T, Japkowicz N. A multiple resampling method for learning from imbalanced data sets. *Comput Intell.* 2004; 20(1):18–36.
32. Han H, Wang W-Y, Mao B-H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. *International Conference on Intelligent Computing:* Springer; 2005. p. 878–87.
33. Mani I, Zhang I. kNN approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of Workshop on Learning from Imbalanced Datasets: ICML;* 2003. p. 1–7.
34. Buda M, Maki A, Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* 2018; 106:249–59. <https://doi.org/10.1016/j.neunet.2018.07.011> PMID: 30092410
35. Xie J, Qiu Z. The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern Recognit.* 2007; 40(2):557–62.
36. Xue J-H, Hall P. Why does rebalancing class-unbalanced data improve AUC for linear discriminant analysis? *IEEE Trans Pattern Anal Mach Intell.* 2014; 37(5):1109–12.

37. Paing MP, Pintavirooj C, Tungjitkusolmun S, Choomchuay S, HAMAMOTO K. Comparison of sampling methods for imbalanced data classification in random forest. 2018 11th Biomedical Engineering International Conference (BMEICON): IEEE; 2018. p. 1–5.
38. Khushi M, Shaukat K, Alam TM, Hameed IA, Uddin S, Luo S, et al. A comparative performance analysis of data resampling methods on imbalance medical data. *IEEE Access*. 2021; 9:109960–75.
39. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*. 2015; 10(3):e0118432. <https://doi.org/10.1371/journal.pone.0118432> PMID: 25738806
40. Liu Z, Bondell HD. Binormal precision-recall curves for optimal classification of imbalanced data. *Stat Biosci*. 2019; 11:141–61.
41. Movahedi F, Antaki JF. Limitation of ROC in evaluation of classifiers for imbalanced data. *J Heart Lung Transplant*. 2021; 40(4 Suppl):S413.
42. Dal Pozzolo A, Caelen O, Johnson RA, Bontempi G. Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on Computational Intelligence: IEEE; 2015. p. 159–66.
43. Dua D, Graff C. UCI Machine Learning Repository: University of California, Irvine, School of Information and Computer Sciences; 2017. Available from: <http://archive.ics.uci.edu/ml>.
44. Little M, Mcsharry P, Roberts S, Costello D, Moroz I. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Nature Precedings*. 2007:1–. <https://doi.org/10.1186/1475-925X-6-23> PMID: 17594480
45. Hart P. The condensed nearest neighbor rule (corresp.). *IEEE Trans Inf Theory*. 1968; 14(3):515–6.
46. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res*. 2017; 18(1):559–63.
47. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. *J Mach Learn Res*. 2011; 12:2825–30.
48. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*; 2006. p. 233–40.
49. Fawcett T. An introduction to ROC analysis. *Pattern Recognit Lett*. 2006; 27(8):861–74.
50. Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput*. 1998; 10(7):1895–923. <https://doi.org/10.1162/089976698300017197> PMID: 9744903
51. Lango M, Stefanowski J. What makes multi-class imbalanced problems difficult? An experimental study. *Expert Syst Appl*. 2022; 199:116962.
52. Koziarski M, Woźniak M, Krawczyk B. Combined cleaning and resampling algorithm for multi-class imbalanced data with label noise. *Knowl Based Syst*. 2020; 204:106223.
53. Zhu T, Lin Y, Liu Y. Synthetic minority oversampling technique for multiclass imbalance problems. *Pattern Recognit*. 2017; 72:327–40.