Pattern Recognition 102 (2020) 107262

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Radial-Based Undersampling for imbalanced data classification

Michał Koziarski

Department of Electronics, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland

ARTICLE INFO

Article history:
Received 10 June 2019
Revised 15 November 2019
Accepted 5 February 2020
Available online 5 February 2020

Keywords:
Machine learning
Classification
Imbalanced data
Undersampling
Radial basis functions

ABSTRACT

Data imbalance remains one of the most widespread problems affecting contemporary machine learning. The negative effect data imbalance can have on the traditional learning algorithms is most severe in combination with other dataset difficulty factors, such as small disjuncts, presence of outliers and insufficient number of training observations. Aforementioned difficulty factors can also limit the applicability of some of the methods of dealing with data imbalance, in particular the neighborhood-based oversampling algorithms based on SMOTE. Radial-Based Oversampling (RBO) was previously proposed to mitigate some of the limitations of the neighborhood-based methods. In this paper we examine the possibility of utilizing the concept of mutual class potential, used to guide the oversampling process in RBO, in the undersampling procedure. Conducted computational complexity analysis indicates a significantly reduced time complexity of the proposed Radial-Based Undersampling algorithm, and the results of the performed experimental study indicate its usefulness, especially on difficult datasets.

1. Introduction

The problem of data imbalance [1–3] occurs in the classification task whenever the number of observations belonging to one of the classes, the *majority class*, exceeds the number of observations belonging to one of the other classes, the *minority class*. Traditional classification algorithms are susceptible to the presence of imbalanced data, and tend to display a bias towards the majority class at the expense of the capability of minority class discrimination. This negative effect on the classification performance is further exacerbated by a presence of additional dataset difficulty factors, such as small disjuncts [4] or insufficient number of training observations [5], that can lead to model overfitting.

Most real datasets exhibit some degree of imbalance that can influence the classification process. Data imbalance heavily impacts many practical domains, such as cancer malignancy grading [6,7], industrial systems monitoring [8], fraud detection [9], behavioral analysis [10] and cheminformatics [11]. As a result, imbalanced data classification remains an active area of research [12–15]. Numerous approaches to mitigating the negative impact of data imbalance have been proposed in the literature. In particular, a family of data-level methods can be distinguished. Data-level methods manipulate with the training data to make it more suitable for classification by traditional learning algorithms, either by increasing the number of minority observations (oversampling) or reducing the number of majority observations (undersampling).

Perhaps the most widespread paradigm of imbalanced data resampling are the neighborhood-based algorithms based on Synthetic Minority Oversampling Technique (SMOTE) [16]. However, SMOTE and many of its derivatives are susceptible to the presence of difficulty factors such as small disjuncts, outliers and small number of minority observations. Recently, a novel method based on the concept of mutual class potential, Radial-Based Oversampling (RBO) [17], has been proposed with the goal of avoiding some of the pitfalls of SMOTE.

In this paper we investigate the possibility of extending the concept of mutual class potential to the undersampling procedure, with the aim of preserving some of the performance gains offered by using the potential to guide the resampling, while simultaneously reducing the computational complexity of the algorithm. The contributions of this paper can be summarized as follows.

1. Proposal of a novel Radial-Based Undersampling (RBU) algorithm based on the concept of mutual class potential.
2. Detailed analysis of the computational complexity of the proposed method, indicating a significantly reduced cost compared to the RBO algorithm.
3. Conceptual and experimental analysis of the impact of the RBUs parameters on its behavior and resulting performance.
4. Thorough experimental evaluation of the proposed method on the basis of diverse benchmark datasets and a large number of state-of-the-art, data-level reference algorithms.

E-mail address: michal.koziarski@agh.edu.pl

https://doi.org/10.1016/j.patcog.2020.107262

5. Analysis of the impact of dataset characteristics on the relative performance of the proposed method.

In other words, the paper expands on the idea of the mutual class potential and proposes a significantly faster alternative to the RBO algorithm, directly addressing one of the main drawbacks of the original oversampling approach. At the same, during the conducted experimental analysis RBU achieved performance comparable to the original RBO algorithm, with a statistically significantly better results when combined with the decision trees. Finally, presented conceptual and experimental analysis provide some insight on the strengths, weaknesses and areas of applicability of the proposed approach.

The rest of this paper is organized as follows. In Section 2 we present the related work on the neighborhood-based oversampling algorithms, guided undersampling techniques, advantages and disadvantages of the two, and the categorization of the minority object types. In Section 3 we describe the proposed method and discuss its computational complexity. In Section 4 we describe the conducted experimental study and the observed results. Finally, in Section 5 we present our conclusions.

2. Related work

In this section we discuss the research relevant to the approach proposed in this paper. We begin with a brief description of the most prevalent paradigm of imbalanced data oversampling, neighborhood-based methods, and highlight their shortcomings which originally inspired Radial-Based Oversampling. Afterwards we describe the related research on guided undersampling strategies and briefly outline the most notable algorithms. Next we discuss the advantages and disadvantages of applying either over- or undersampling, and present some of the considerations that may affect the choice of the resampling strategy. Finally, we summarize the existing research of minority object type categorization, used later in this paper to identify the areas of applicability of the proposed method.

2.1. Neighborhood-based oversampling algorithms

The most fundamental choice during the design of both oversampling and undersampling algorithms for handling data imbalance is the question of defining the regions of interest: the areas in which either the new instances are to be placed, in case of oversampling, or from which the existing instances are to be removed, in case of undersampling. Besides the random approaches, probably the most prevalent paradigm for the oversampling are the neighborhood-based methods originating from Synthetic Minority Over-sampling Technique (SMOTE) [16]. The regions of interest of SMOTE are located between any given minority observation and its closest minority neighbors: SMOTE synthesizes new instances by interpolating the observation and one of its, randomly selected, nearest neighbors.

SMOTE can be considered a cornerstone for the majority of the existing oversampling strategies [18,19]. Numerous extensions of the original algorithm were proposed, with the most notable including: Borderline-SMOTE [20], focusing on the borderline instances, placed close to the decision border; Adaptive Synthetic Sampling (ADASYN) [21], individually adjusting the oversampling ratio based on the difficulty of the given observation; and Safe-Level-SMOTE [22] and LN-SMOTE [23], limiting the risk of placing synthetic instances inside the regions belonging to the majority class. However, despite their prevalence, neighborhood-based approaches have their own shortcomings that can affect the suitability of synthesized observations for improving classification. Most importantly, in the basic variant SMOTE does not utilize the information about the distribution of the majority class objects: the regions of interest are based solely on the position of the minority observations. This can potentially lead to synthesizing minority observations overlapping the clusters of majority observations for datasets displaying factors such as a small number of minority objects, disjoint data distributions, or presence of the outliers, which was illustrated in Fig. 1. Classifiers trained on datasets resampled in that way can display, possibly unjustified, bias towards the minority class and resulting decreased performance. While some attempts at limiting the described deficiency of SMOTE have been made, such as the previously mentioned extensions, Safe-Level-SMOTE and LN-SMOTE, or combining the oversampling using SMOTE with later cleaning with either Tomek links [24] or Edited Nearest-Neighbor rule [25], a further research into the methods explicitly using the information about the distribution of both classes is required.

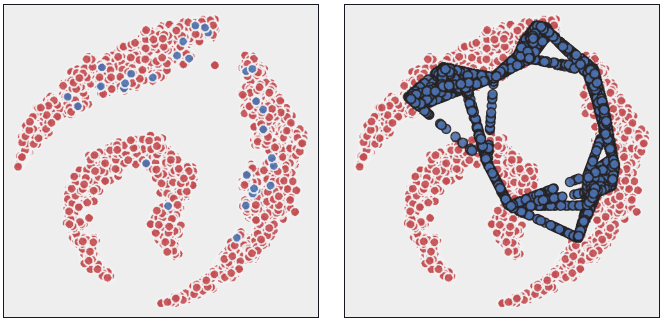

Fig. 1. An example of a difficult dataset for neighborhood-based methods, displaying factors such as a small number of minority objects, disjoint data distributions, or presence of the outliers. On the left: original data distribution. On the right: dataset after oversampling with SMOTE, with the generated observations highlighted.

2.2. Guided undersampling strategies

Similar to the case of oversampling, finding the regions of interest, in the case of undersampling indicating which observations are to be discarded, is essential choice in the algorithm design process. Besides the random methods, over the years a number of guided undersampling strategies was proposed. Many of them rely on some sort of mechanism for identifying the least informative instances, either due to a high redundancy of the given observation or a low confidence that it is not an outlier.

One of the oldest examples of the latter are the cleaning strategies, heuristics algorithms used to remove observations deemed as inconsistent with the remainder of the data: Tomek links [24], Edited Nearest-Neighbor rule [25], Condensed Nearest Neighbour editing (CNN) [26], and more recently Near Miss method (NM) [27], constitute examples of that paradigm. Notably, these methods do not allow specifying the number of observations that should be discarded: instead, they remove all the observations meeting the cleaning rule, which can leave an undesired level of data imbalance. As a result, more recent methods tend to sort the majority observations according to the chosen criterion and allow arbitrary level of balancing. For instance, Anand et al. [28] propose sorting the undersampled observations based on the weighted Euclidean distance from the positive samples. Smith et al. [29], in their study of instance level data complexity, advocate for using the instance hardness criterion, with the hardness estimated based on the certainty of the classifiers predictions.

Another family of methods that can be distinguished are the cluster-based undersampling algorithms, notably the methods proposed by Yen and Lee [30], which use clustering to select the most representative subset of data. Finally, as has been originally demonstrated by Liu et al. [31], undersampling algorithms are well-suited for forming classifier ensembles, an idea that was further extended in form of evolutionary undersampling [32] and boosting [33].

2.3. Differences between oversampling and undersampling

On the most basic level both oversampling and undersampling strategies modify the original data distribution to alleviate the problem of imbalance. However, both approaches can lead to a significantly different performance on any given dataset, and pose unique challenges during the algorithm design process. The main issue associated with undersampling is the possibility of discarding valuable information, whereas during the oversampling we are concerned with the possibility of overfitting the classifier and an increased computational cost of training for an artificially enlarged dataset [34]. The resampling process is also vastly different: while during the undersampling we are concerned only with designating a subset of majority observations that should be removed, effectively ranking a finite number of discrete objects, starting with SMOTE oversampling strategies tend to be a continuous problems, requiring finding regions in which synthetic observations should be generated.

The choice of one of these approaches is determined by a number of factors and has been, to some extent, studied in the literature. First of all, some of the classification algorithms show clear preference towards either of the resampling strategies, with a notable example of decision trees, the overfitting of which was a motivation behind the SMOTE [16]. Nevertheless, later study found the SMOTE itself to still be ineffective when combined with the C4.5 algorithm [35], for which applying undersampling led to a better performance. In another study [36] authors focused on the impact of noise, with a conclusion that especially for a high levels of noise simple random undersampling produced the best results. Finally, in another experimental study [37] authors investigated the impact of the level of imbalance on the choice of the resampling strategy. Their results indicate that oversampling tends to perform better on a severely imbalanced datasets, while for more modest levels of imbalance both over- and undersampling tend to perform similarly. In general, there is no clear consensus on which of the approaches tends to produce the better results, and while some guidelines are available, in practice an experimental evaluation of both approaches is usually required.

2.4. Categorization of minority object types

Despite the abundance of different strategies of dealing with data imbalance, it often remains unclear under what conditions a given method is expected to guarantee a satisfactory performance. Furthermore, taking into the account the no free lunch theorem [38] it is unreasonable to expect that any single method will be able to achieve a state-of-the-art performance on every provided dataset. Identifying the areas of applicability, conditions under which the method is expected to be more likely to achieve a good performance, is therefore desirable both from the point of view of a practitioner, who can use that information to narrow down the range of methods appropriate for a problem at hand, as well as a theoretician, who can use that insight in the process of developing novel methods.

In the context of the imbalanced data classification, one of the criteria that can influence the applicability of different resampling strategies are the characteristics of the minority class distribution. Napierała and Stefanowski [39] proposed a method of categorization of different types of minority objects that capture these characteristics. Their approach uses a 5-neighborhood to identify the nearest neighbors of a given object, and afterwards assigns to it a category based on the proportion of neighbors from the same class: *safe* in case of 4 or 5 neighbors from the same class, *borderline* in case of 2 to 3 neighbors, *rare* in case of 1 neighbor, and *outlier* when there are no neighbors from the same class. The percentage of the minority objects from different categories can be then used to describe the character of the entire dataset: an example of datasets with a large proportion of different minority object types was presented in Fig. 2. Note that the imbalance ratio of the dataset does not determine the type of the minority objects it consists of, which was demonstrated in the above example.

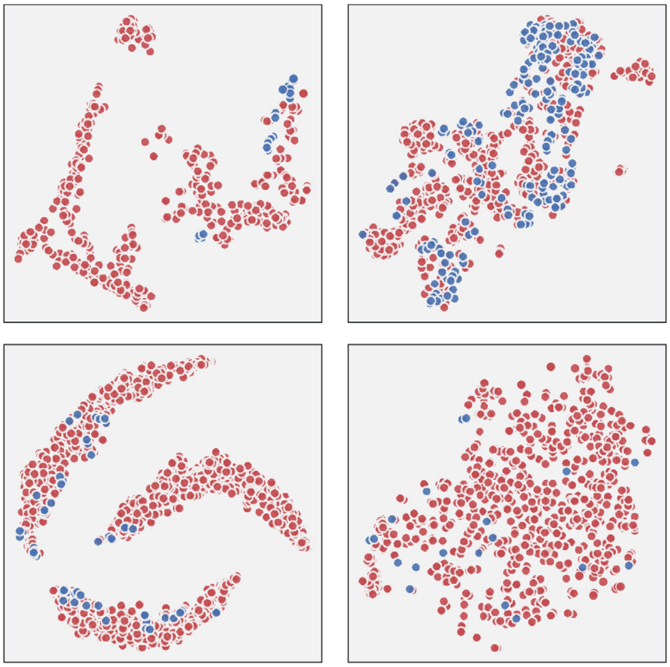

Fig. 2. An example of datasets with a large proportion of safe (top left), borderline (top right), rare (bottom left) and outlier (bottom right) minority objects.

3. Radial-Based Undersampling

In this section we describe the proposed Radial-Based Undersampling algorithm. We begin with a description of the potential estimation using radial basis functions, and the previously introduced Radial-Based Oversampling algorithm. Afterwards, we describe how the concept of mutual class potential can be applied during the undersampling of the majority objects. Finally, we discuss the computational complexity of the proposed algorithm.

3.1. Potential estimation and Radial-Based Oversampling

In the context of imbalanced data resampling, the concept of class potential was first introduced as an approach for designating the regions of interest for oversampling [17]. Specifically, it was proposed as an alternative to the regions of interest used by SMOTE and its derivatives, which did not utilize the information about the majority class distribution. Mutual class potential is a real-valued function, the value of which, in any given point in space, represents the degree of affiliation of that point to either the majority or the minority class. To calculate that potential, we assign a Gaussian radial basis function (RBF) to every observation in the considered dataset, with the polarity dependent on its class. We assume the convention of assigning a positive polarity to the majority class observations and a negative polarity to the minority class observations. More formally, given a set of majority observations K, a set of minority observations κ, and a parameter γ affecting the spread of a single RBF, we define the mutual class potential in the point x as

$$\Phi(x, K, \kappa, \gamma) = \sum_{i=1}^{|K|} e^{-\left(\frac{\|K_i - x\|_2}{\gamma}\right)^2} - \sum_{j=1}^{|\kappa|} e^{-\left(\frac{\|\kappa_j - x\|_2}{\gamma}\right)^2}, \tag{1}$$

where K_i denotes the ith object from the majority class and κ_j denotes jth object from the minority class, respectively.

Mutual class potential was used in the Radial-Based Oversampling algorithm, in which an iterative optimization was harnessed to locate the regions minimizing the absolute value of potential. Intuitively, such regions represent a low certainty towards the affiliation to either of the classes. New synthetic observations were generated in such regions with the aim of reducing the classifiers bias towards the majority class and moving the decision border in favor of the minority class. Compared with SMOTE, such approach displayed some beneficial characteristics. RBO tended to be less affected by the presence of the outliers, as well as a small number of minority objects combined with disjoint distributions. While in those cases SMOTE was likely to generate new instances overlapping the clusters of existing majority objects, using RBO resulted in a smaller, constrained regions of negative class potential in which new instances were synthesized.

The algorithm was afterwards extended to the problem of classification of noisy imbalanced data [40] and multi-class imbalanced data [41]. Furthermore, an extension omitting observations categorized as outliers was also proposed by Bobowska and Woźniak [42]. Despite leading to a favorable performance in the conducted experiments, especially in the multi-class setting, using RBO was computationally expensive due to the need of recalculating the class potential at every optimization step. Reducing the computational overhead of the algorithm was an issue identified to be essential to make the algorithm applicable to a very large datasets.

3.2. Using mutual class potential during undersampling

While originally proposed to provide the regions of interest in the process of oversampling, mutual class potential can also easily be used to guide the process of undersampling the majority class. Recall that, using the assumed convention, high value of mutual class potential in a given point in space would indicate that in its proximity there is a higher concentration of majority than minority observations. It is therefore possible to rank the existing majority observations based on their mutual class potential. We propose using such ranking mechanism to determine the order of undersampling. Specifically, we make the assumption that the majority observations with highest corresponding mutual class potential provide the least amount of information and are more redundant than the observations with lower potential. As a result, we undersample in the order of decreasing potential, updating its value for the remaining observations after each undersampled object.

We present the pseudocode of the proposed method in Algorithm 1. In addition to the collection of majority objects K and the collection of minority objects κ, algorithm has two additional parameters: spread of the individual radial basis function γ, affecting the range of impact of the associated observation on the mutual class potential, and the undersampling ratio, with radio equal to 1.0 indicating that the majority objects are undersampled up to the point of achieving balanced class distribution. Furthermore, we present a visualization of the algorithms behavior for different values of γ in Fig. 3. As can be seen, the value of γ parameter significantly impacts the shape of the resulting potential: using smaller values of γ leads to a more complex potential field, affected in a given point in space only by the observations in its close proximity, whereas using larger values of γ leads to a smooth potential. As a result, the choice of γ affects the order of undersampling. For the smaller values of γ removed observations are mostly a part of local clusters consisting of several majority and no minority observations, and these clusters are never completely removed. Furthermore, individual majority observations and majority observations located in a close proximity of minority observations tend to remain unaffected. On the other hand, for larger values of γ a single cluster with a high concentration of majority objects is identified and the undersampling is performed solely within its bounds. When combined with a significant data imbalance this can lead to a potentially undesirable behavior of a very highly centralized undersampling, which may indicate the proclivity towards using lower values of γ.

Algorithm 1 Radial-Based Undersampling.

```
Input: collections of majority objects $K$ and minority objects $\kappa$
Parameters: spread of radial basis function $\gamma$, oversampling ratio
Output: undersampled collection of majority objects $K'$

function RBU($K$, $\kappa$, $\gamma$, ratio):
$K' \leftarrow K$
for every majority object $K'_i$ in $K'$ and its associated potential $\Phi_i$ do
    $\Phi_i \leftarrow \Phi(K'_i, K, \kappa, \gamma)$
end for
while $|K| - |K'| < ratio \cdot (|K| - |\kappa|)$ do
    $x \leftarrow$ majority object $K'_i$ from $K'$ with highest potential $\Phi_i$; in case of multiple selected objects break ties arbitrarily
    discard $x$ from $K'$
    for every majority object $K'_i$ in $K'$ and its associated potential $\Phi_i$ do
        $\Phi_i \leftarrow \Phi_i - e^{-\left(\frac{\|K'_i - x\|_2}{\gamma}\right)^2}$
    end for
end while
return $K'$
```

3.3. Computational complexity analysis

Let us define the total number of observations by n, the number of majority and minority observations by n_{maj} and $n_{\min}$, respectively, and the number of features by m. Let us consider the computational complexity of RBU algorithm applied up to the point of achieving a balanced class distribution. A single calculation of the mutual class potential in any given point, as defined in Eq. (1), requires n distance calculations, each with a complexity of $\mathcal{O}(m)$, n summations and n radial basis function calculations, both with a complexity of $\mathcal{O}(1)$. Therefore, a total complexity of a single mutual class potential calculation is equal to $\mathcal{O}(mn)$. Furthermore, the procedure of removing a single observation consists of finding the observation with a highest potential in a collection of a size not exceeding n_{maj}, with a complexity equal to $\mathcal{O}(n_{maj})$, discarding it from said collection, the complexity of which we will assume to be $\mathcal{O}(n_{maj})$, and updating all of the remaining potentials, consisting of n_{maj} distance calculations, subtractions and radial basis function calculations, leading to an overall complexity of the potential update operation equal to $\mathcal{O}(mn_{maj})$. Combining the above, the complexity of the procedure of removing a single observation is also equal to $\mathcal{O}(mn_{maj})$. The complete RBU algorithm consists of the initial calculation of the potential for every majority observation, with a complexity of $\mathcal{O}(mnn_{maj})$, and a removal of $n_{maj} - n_{\min}$ observations, with a complexity of $\mathcal{O}(mn_{maj}(n_{maj} - n_{\min}))$. As a results, the total complexity of the proposed RBU algorithm can be simplified to $\mathcal{O}(mn^2)$. For comparison, the complexity of the orig-

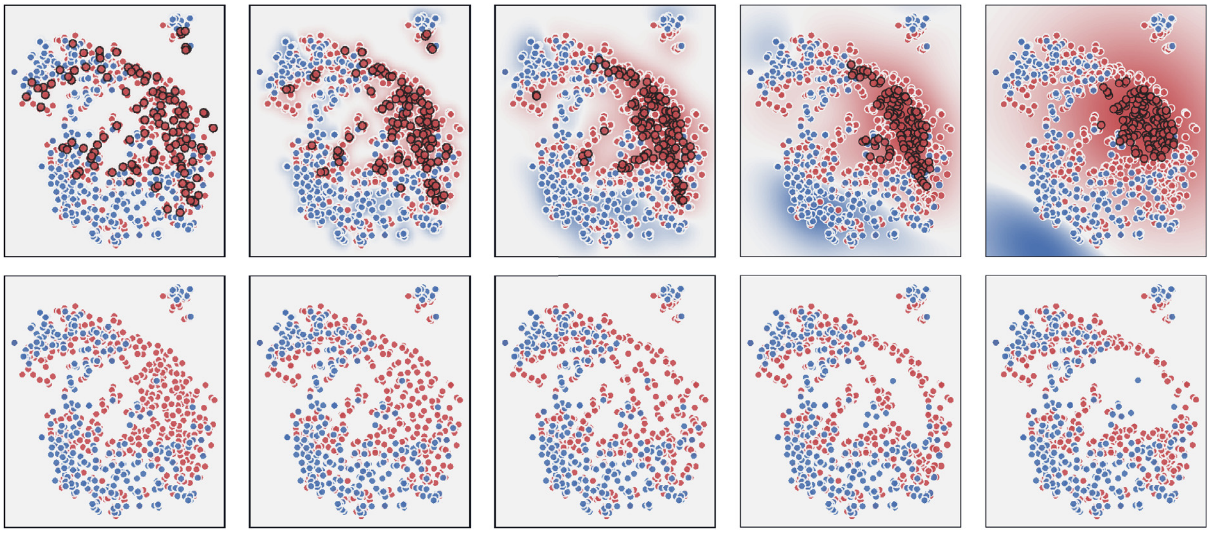

Fig. 3. Visualized impact of the γ parameter of RBU on the shape of the mutual class potential and resulting undersampling regions. Top row: original dataset with highlighted majority objects selected for undersampling. Bottom row: dataset after undersampling. Values of γ, from the left: 1.0, 2.5, 5.0, 10.0, 25.0.

inal Radial-Based Oversampling algorithm applied up to the point of achieving balanced class distributions, as discussed in [41], is equal to $\mathcal{O}(imn^2)$, with i denoting the number of algorithms iterations, the value of which was experimentally chosen to be in range from 1000 to 8000 in the conducted experiments. As a result, RBU has a significantly reduced computational overhead when compared to the original RBO algorithm.

4. Experimental study

To empirically evaluate the applicability of the proposed Radial-Based Undersampling algorithm we conducted a two-part experimental study. In its first stage we examined the impact of the algorithms parameters on its performance. In the second stage we compared the algorithm with the selected state-of-the-art resampling strategies. Finally, we analysed the parameters of the datasets on which the proposed method achieved the best results to identify possible areas of applicability. In the remainder of this section we describe our experimental set-up and present the observed results.

4.1. Set-up

Data. Conducted experimental study was based on the binary imbalanced datasets provided in the KEEL repository [43]. Specifically, from the available datasets we excluded the ones containing less than 12 minority observations to avoid issues with cross-validation, as well as the ones for which AUC greater than 0.85 was achieved with SVM without any resampling, to eliminate the datasets for which, despite data imbalance, resampling was not required. A total of 50 datasets was selected using this approach. Out of them, 20 were randomly chosen for the preliminary analysis, during which the impact of the parameters on the algorithms performance was examined. Remaining 30 datasets were used during the comparison with the reference methods.

The details of the used datasets were presented in Table 1. In addition to the imbalance ratio (IR), the number of samples and the number of features, for each dataset we computed the proportion of different types of minority class observations, proposed by Napierała and Stefanowski [39]. Specifically, the types were identified using 5-neighbourhood computed based on the Minkowski metric.

Prior to resampling and classification, categorical features were encoded as integers. Afterwards, all features were standarized by removing the mean and scaling to unit variance. No further preprocessing was applied.

Classification. Four different classification algorithms, representing different learning paradigms, were used throughout the experimental study. Specifically, we used CART decision tree, k-nearest neighbors classifier (KNN), naive Bayes classifier (NB) and support vector machine with RBF kernel (SVM). The implementations of the classification algorithms provided in the scikit-learn machine learning library [44] were used, and their default parameters remained unchanged.

Reference resampling methods. During the comparison with reference resampling algorithms we considered a total of 17 different data-level approaches. We focused on other undersampling methods, with a total of 11 algorithms of that type. Specifically, we used random undersampling (RUS), All k-Nearest Neighbors editing (AKNN) [45], Cluster Centroid undersampling (CC) [30], Condensed Nearest Neighbour editing (CNN) [26], Edited Nearest Neighbour rule (ENN) [25], Instance Hardness Threshold method (IHT) [29], Neighborhood Cleaning Rule (NCL) [46], Near Miss method (NM) [27], One-Sided Selection (OSS) [47], Repeated Edited Nearest Neighbour method (RENN) [45] and Tomek links undersampling (TL) [24]. Furthermore, we used 4 additional oversampling algorithms: random oversampling (ROS), SMOTE [16], Borderline-SMOTE (Bord) [20] and Radial-Based Oversampling (RBO) [41]. Finally, we also used two methods combining oversampling with undersampling: SMOTE combined with Tomek links (STL) [24] and Edited Nearest Neighbor rule (SENN) [25].

With the exception of RBO, the implementations of the reference methods provided in the imbalanced-learn library [48] were used.

Evaluation. For every dataset we reported the results averaged over the 5 × 2 cross-validation folds [49]. Throughout the experimental study we reported the values of precision, recall, *F*-measure, AUC and G-mean. Whenever applicable, parameter selection was conducted by further 3 × 2 cross-validation on the currently considered training data, with the optimization criterion being the average of *F*-measure, AUC and G-mean. It should be noted that in most cases observed *F*-measure was significantly lower than AUC and G-mean, leading to the choice of parameters biased towards the latter two metrics.

Table 1
Details of the datasets used during the preliminary (top) and the final (bottom) analysis.

Name	IR	Samples	Features	Safe [%]	Borderline [%]	Rare [%]	Outlier [%]
pima	1.87	768	8	30.22	44.03	16.79	8.96
glass0	2.06	214	9	54.29	38.57	1.43	5.71
vehicle3	2.99	846	18	16.51	50.94	27.36	5.19
ecoli1	3.36	336	7	53.25	31.17	9.09	6.49
yeast3	8.1	1484	8	56.44	25.15	7.36	11.04
ecoli-0-6-7_vs_3-5	9.09	222	7	40.91	31.82	9.09	18.18
yeast-0-3-5-9_vs_7-8	9.12	506	8	16.0	28.0	22.0	34.0
ecoli-0-2-6-7_vs_3-5	9.18	224	7	40.91	31.82	9.09	18.18
ecoli-0-1-4-7_vs_2-3-5-6	10.59	336	7	65.52	17.24	0.0	17.24
glass-0-1-4-6_vs_2	11.06	205	9	0.0	17.65	35.29	47.06
cleveland-0_vs_4	12.31	173	13	0.0	69.23	23.08	7.69
yeast-2_vs_8	23.1	482	8	55.0	0.0	15.0	30.0
winequality-red-4	29.17	1599	11	0.0	7.55	22.64	69.81
winequality-red-8_vs_6	35.44	656	11	0.0	0.0	50.0	50.0
kr-vs-k-zero_vs_eight	53.07	1460	6	62.96	25.93	7.41	3.7
winequality-white-3-9_vs_5	58.28	1482	11	0.0	8.0	20.0	72.0
poker-8-9_vs_6	58.4	1485	10	8.0	56.0	20.0	16.0
abalone-20_vs_8-9-10	72.69	1916	8	0.0	19.23	11.54	69.23
poker-8_vs_6	85.88	1477	10	5.88	35.29	35.29	23.53
abalone19	129.44	4174	8	0.0	0.0	12.5	87.5
glass1	1.82	214	9	44.74	32.89	14.47	7.89
yeast1	2.46	1484	8	21.91	45.69	20.75	11.66
haberman	2.78	306	3	4.94	48.15	32.1	14.81
vehicle1	2.9	846	18	23.96	55.76	16.13	4.15
ecoli3	8.6	336	7	28.57	48.57	8.57	14.29
yeast-2_vs_4	9.08	514	8	54.9	19.61	11.76	13.73
yeast-0-2-5-6_vs_3-7-8-9	9.14	1004	8	33.33	32.32	14.14	20.2
yeast-0-5-6-7-9_vs_4	9.35	528	8	7.84	43.14	15.69	33.33
ecoli-0-6-7_vs_5	10.0	220	6	45.0	35.0	0.0	20.0
glass-0-1-6_vs_2	10.29	192	9	0.0	29.41	35.29	35.29
glass2	11.59	214	9	0.0	23.53	41.18	35.29
yeast-1_vs_7	14.3	459	7	6.67	33.33	26.67	33.33
glass4	15.46	214	9	23.08	53.85	7.69	15.38
page-blocks-1-3_vs_4	15.86	472	10	78.57	17.86	3.57	0.0
abalone9-18	16.4	731	8	4.76	23.81	19.05	52.38
yeast-1-4-5-8_vs_7	22.1	693	8	0.0	6.67	33.33	60.0
flare-F	23.79	1066	11	0.0	48.84	39.53	11.63
car-good	24.04	1728	6	0.0	97.1	2.9	0.0
car-vgood	25.58	1728	6	20.0	80.0	0.0	0.0
yeast4	28.1	1484	8	7.84	35.29	17.65	39.22
yeast-1-2-8-9_vs_7	30.57	947	8	3.33	23.33	23.33	50.0
yeast5	32.73	1484	8	34.09	50.0	11.36	4.55
abalone-17_vs_7-8-9-10	39.31	2338	8	3.45	13.79	34.48	48.28
abalone-21_vs_8	40.5	581	8	14.29	35.71	21.43	28.57
yeast6	41.4	1484	8	34.29	25.71	11.43	28.57
winequality-white-3_vs_7	44.0	900	11	0.0	15.0	10.0	75.0
abalone-19_vs_10-11-12-13	49.69	1622	8	0.0	0.0	21.88	78.12
kddcup-buffer_overflow_vs_back	73.43	2233	41	73.33	13.33	6.67	6.67
poker-8-9_vs_5	82.0	2075	10	0.0	0.0	16.0	84.0
kddcup-rootkit-imap_vs_back	100.14	2225	41	54.55	27.27	9.09	9.09

Implementation and reproducibility. The experiments described in this paper were implemented in the Python programming language. Complete code, sufficient to repeat the experiments, was made publicly available at.[1] In addition to the code we also provided the cross-validation folds used during the experiments, as well as a file containing complete results, enabling any further analysis.

4.2. Analysis of the impact of parameters

In the first stage of the conducted experimental study we considered the impact of two of the proposed algorithms parameters, radial basis function spread γ and the undersampling ratio, on the algorithms performance. Specifically, we conducted two experiments: in the first one we adjusted the value of γ parameter in {0.001, 0.01, 0.1, 1.0, 10.0, 100.0} while selecting the values of undersampling ratio individually for each dataset using cross-validation, with considered values in {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}. In the second experiment we used the same parameter values, but adjusted undersampling ratio while selecting γ with cross-validation.

The results, averaged over 20 datasets, were presented in Fig. 4. As can be seen, the best performance with respect to the combined metrics (*F*-measure, AUC, G-mean) was observed for smaller values of γ, equal to 0.1 or lower. Using higher values either did not improve the average performance, or resulted in its significant decline. The exact value of γ parameter for which the best averaged performance was observed depended on the type of classifier and the chosen metric. For CART, KNN and SVM classifiers decreasing the value of γ tended to improve the recall at the cost of precision, whereas for NB the reverse was observed. It is worth noting that while the described trends were roughly monotonic for individual classifier and metric combinations, a significant peak in precision, combined with a drop in recall, was observed in the case of SVM

[1] https://github.com/michalkoziarski/RBU.

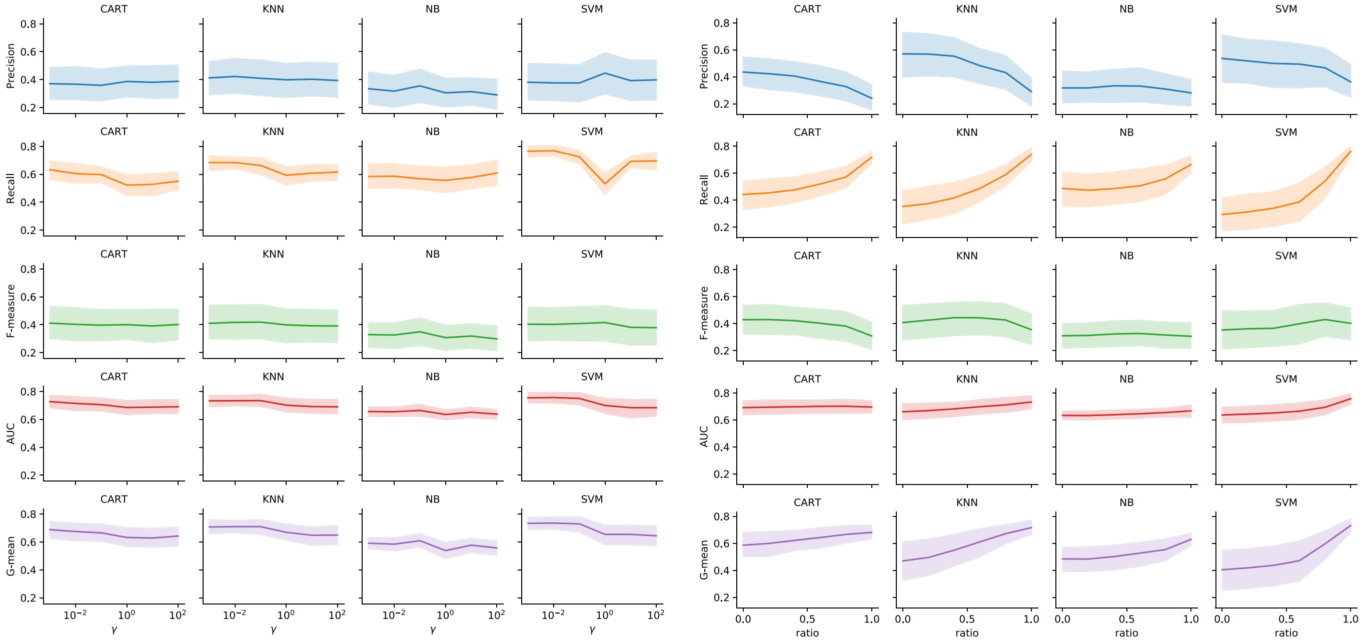

Fig. 4. The impact of γ parameter (left) and undersampling ratio (right) of Radial-Based Undersampling on various performance metrics, averaged over all datasets, with a 95% confidence intervals shown.

for $\gamma = 1.0$. During the examination of the results for individual datasets it was confirmed that this peak occurred in about half of the datasets. It is not clear what caused the peak and what is its significance.

In the case of the undersampling ratio, as can be expected, increasing the ratio led to an improvement in the classifiers recall and a corresponding drop in the precision, for all of the examined classification algorithms. When considering the combined metrics, the relation between performance of the algorithm and the undersampling ratio varied depending on the metric. In the case of the G-mean, a significantly better performance was observed for the highest value of the undersampling ratio, corresponding to undersampling up to the point of achieving balanced class distribution. A noticeable improvement in performance was also observed in the case of AUC in combination with KNN ans SVM classifiers. When combined with CART and NB classifiers, the undersampling ratio did not, on average, have a significant impact on the observed AUC. In the case of *F*-measure the peak in performance was observed for different undersampling ratios. In the case of SVM high, but not complete, undersampling led to achieving the best results, with the best performance observed for the undersampling ratio of 0.8. In the case of KNN and NB best results were achieved for medium ratios of 0.4 and 0.6, but in the case of NB the impact of the choice of the undersampling ratio was less significant. Finally, in the case of CART the best performance was observed for small or no undersampling, with ratios equal to 0.0 and 0.2. Notably, in no case was the best performance with regard to *F*-measure observed for complete undersampling, up to the point of achieving balanced class distributions.

To summarize, irregardless of the choice of the classification algorithm and the evaluation metric, the best performance was observed for smaller values of γ parameter in {0.001, 0.01, 0.1}, which could all be use as a sensible default values. The choice of the undersampling ratio, however, was dependant on the classification algorithm and evaluation metric. While using complete undersampling was a sensible default with regard to AUC and G-mean, lower undersampling ratios led to observing better performance with regard to *F*-measure, and the choice of the classifier affected the exact value of the undersampling ratio for which the best performance was observed. It is worth noting that the optimization of parameters with respect to one of the metrics could lead to suboptimal results with respect to the other. In particular, when using the scheme employed in this paper, that is choosing the parameters maximizing the average value of *F*-measure, AUC and G-mean, the parameter choice will be biased towards the AUC and G-mean: this is both because the *F*-measure tends to take lower values, and the fact that the AUC and G-mean displayed higher correlation between each other than between the *F*-measure.

4.3. Comparison with other methods

In the second stage of the conducted experimental analysis we compared the proposed Radial-Based Undersampling with a total of 17 data-level methods described in Section 4.1. For every algorithm we conducted a parameter search to adjust its hyperparameters individually for each dataset. Specifically, for all variants of SMOTE we considered the values of k neighborhood in {1, 3, 5, 7, 9}; for Bord, we additionally considered the values of m neighborhood, used to determine if a minority sample is in danger, in {5, 10, 15}; for neighborhood-based undersampling strategies, that is AKNN, CNN, ENN, NCL, NM, OSS and RENN we considered the values of respective k neighborhoods in {1, 3, 5, 7}; for RBU and RBO we considered the values of γ in {0.01, 0.1, 1.0, 10.0}. Finally, for all of the algorithms using manually specified resampling ratio, that is RUS, ROS, CC, RBO, RBU and all of the variants of SMOTE, we considered the values of that ratio in {0.5, 0.75, 1.0}. To evaluate the statistical significance of the observed results we used the Friedman test combined with the Shaffer's post-hoc. The results were reported at the significance level $\alpha = 0.10$.

We present the average rankings achieved by the respective methods for all of the performance metrics and highlight the statistically significantly different results in Table 2. Furthermore, for NB classifier we present a detailed win-tie-loss analysis, that is a visualization of the number of datasets on which RBU outperforms individual reference methods, in Fig. 5. As can be seen, in general case the usefulness of the proposed RBU algorithm, when

Table 2
Average rankings of the evaluated methods. Best performance was denoted with bold font. Methods that achieved significantly different results (according to Shaffer's post-hoc test) than RBU where denoted in subscript: with + sign for methods compared to which RBU achieved a better results, and − sign for methods compared to which RBU achieved worse results.

			Undersampling											Oversampling				Combined	
	Metric	RBU	RUS	AKNN	CC	CNN	ENN	IHT	NCL	NM	OSS	RENN	TL	ROS	SMOTE	Bord	RBO	STL	SENN
CART	Precision	11.4	15.2	8.8	15.7	14.7	8.0	14.3	8.2	17.2 $_{+}$	4.9 $_{-}$	9.8	**3.9** $_{-}$	4.4 $_{-}$	6.8 $_{-}$	6.4 $_{-}$	5.1 $_{-}$	6.8 $_{-}$	9.4
	Recall	7.9	4.6	8.0	3.8	9.2	9.0	**3.4** $_{-}$	9.3	4.7	13.4 $_{+}$	7.6	13.0 $_{+}$	15.8 $_{+}$	12.9 $_{+}$	14.1 $_{+}$	14.5 $_{+}$	12.1	7.7
	F-measure	10.1	13.5	6.3	14.2	14.5	**5.5** $_{-}$	11.8	5.9	16.7 $_{+}$	6.5	7.5	5.6	9.0	9.0	9.2	9.1	8.9	7.6
	AUC	7.6	6.7	7.1	8.7	12.7 $_{+}$	7.5	**6.0**	7.5	14.0 $_{+}$	10.8	6.3	10.2	13.6 $_{+}$	11.0	12.0	12.8 $_{+}$	10.2	6.2
	G-mean	7.2	**5.0**	7.5	7.2	10.7	8.2	5.9	8.3	11.3	12.2 $_{+}$	7.2	11.6	14.4 $_{+}$	11.3	12.5 $_{+}$	13.2 $_{+}$	10.8	6.3
KNN	Precision	11.9	14.2	7.0 $_{-}$	13.5	8.2	6.6 $_{-}$	10.8	6.6 $_{-}$	14.2	4.7 $_{-}$	8.2	**4.6** $_{-}$	8.6	9.6	9.2	8.7	10.8	13.7
	Recall	8.8	4.9	12.5	4.7	12.5	13.7 $_{+}$	7.8	12.4	8.3	17.0 $_{+}$	11.8	17.3 $_{+}$	8.9	6.1	7.6	8.6	5.1	**3.1** $_{-}$
	F-measure	11.4	11.8	9.0	11.1	9.6	9.4	9.4	6.7 $_{-}$	12.9	13.0	9.8	13.3	7.0	6.5 $_{-}$	**6.1** $_{-}$	7.0	7.3	9.7
	AUC	10.4	6.6	11.8	6.8	12.3	13.0	8.1	11.0	11.4	16.4 $_{+}$	11.2	16.4 $_{+}$	7.9	4.6 $_{-}$	6.6	7.4	**4.3** $_{-}$	4.7 $_{-}$
	G-mean	9.8	6.0	12.4	6.7	12.5	13.2	7.8	11.3	10.5	16.8 $_{+}$	11.9	17.0 $_{+}$	7.8	4.6 $_{-}$	6.7	7.4	**4.2** $_{-}$	4.5 $_{-}$
NB	Precision	**6.2**	11.0 $_{+}$	7.9	13.7 $_{+}$	8.4	8.1	10.6	10.0	15.4 $_{+}$	8.9	7.7	8.8	10.8 $_{+}$	7.6	8.0	7.9	9.7	10.4
	Recall	10.8	10.4	11.3	**4.0** $_{-}$	11.1	11.0	8.8	10.2	11.3	11.0	11.4	11.0	5.5 $_{-}$	8.2	9.8	9.6	6.9	8.7
	F-measure	**5.6**	10.8 $_{+}$	8.4	12.0 $_{+}$	8.8	8.6	10.3 $_{+}$	10.1	15.6 $_{+}$	10.3 $_{+}$	8.8	10.3 $_{+}$	10.6 $_{+}$	7.3	7.0	7.2	8.9	10.4 $_{+}$
	AUC	**6.6**	10.0	9.0	10.5	9.3	9.2	8.5	10.4	15.6 $_{+}$	11.6 $_{+}$	9.5	11.0	10.4	7.8	7.7	6.6	8.1	9.4
	G-mean	**6.0**	10.1	9.9	10.9 $_{+}$	11.0 $_{+}$	10.2	8.5	11.4 $_{+}$	12.1 $_{+}$	12.5 $_{+}$	9.8	12.1 $_{+}$	9.8	7.1	7.0	6.1	7.4	8.8
SVM	Precision	11.7	13.2	9.2	12.5	7.3	8.4	12.0	7.7	15.7	7.5	8.8	6.7 $_{-}$	7.9	6.8 $_{-}$	**6.5** $_{-}$	7.7	9.1	12.2
	Recall	7.5	**4.1**	12.0 $_{+}$	4.2	13.1 $_{+}$	13.5 $_{+}$	7.1	13.2 $_{+}$	6.0	16.6 $_{+}$	12.3 $_{+}$	17.1 $_{+}$	7.7	7.8	10.0	7.7	7.0	4.2
	F-measure	9.8	11.1	9.4	9.3	11.3	9.3	9.9	9.0	14.5 $_{+}$	15.0 $_{+}$	9.9	14.8 $_{+}$	6.4	**4.6** $_{-}$	5.5	5.7	6.8	8.9
	AUC	8.7	5.7	11.7	5.6	12.9	12.9	8.6	12.8	12.9	16.4 $_{+}$	11.6	16.3 $_{+}$	5.8	5.3	8.0	**5.2**	5.7	**5.2**
	G-mean	7.9	4.9	12.0	5.2	13.4 $_{+}$	13.2 $_{+}$	8.2	13.0 $_{+}$	11.2	16.6 $_{+}$	12.4 $_{+}$	16.7 $_{+}$	5.8	5.9	8.6	5.7	5.5	**4.7**

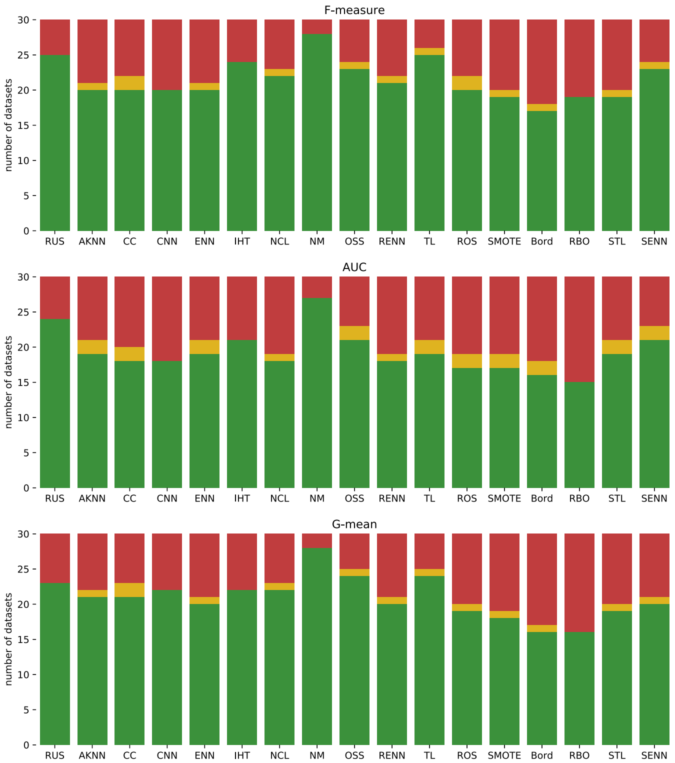

Fig. 5. Total number of datasets for which RBU achieved better (green), equal (yellow) or worse (red) performance than specific reference methods, with naive Bayes algorithm used for classification. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

compared to the reference methods, was reliant on both the choice of the classification algorithm and the metric used to evaluate the performance. RBU achieved the best results when combined with NB classifier, scoring highest average ranks with respect to all of the combined performance metrics, and statistically significantly better results with respect to at least one of them for 10 out of 17 reference methods. Furthermore, it achieved a relatively good performance when combined with CART and SVM, scoring a statistically significantly better results with respect to at least one of the combined metrics: in 6 cases for CART, and in 7 cases for SVM. At the same time, for both classifiers a statistically significantly worse results were observed in only a single case: compared to ENN for CART classifier, and compared to SMOTE for SVM, both with respect to *F*-measure. The worse performance was observed when RBU was combined with KNN classifier, in which case all of the variants of SMOTE, as well as the NCL algorithm, achieved a statistically significantly better results than RBU. In that case, the latter significantly outperformed only two of the reference methods.

When compared to the RBO, RBU achieved a statistically significantly different results only in case of CART decision tree. In that instance RBU achieved a significantly better AUC and G-mean than RBO. Out of the remaining cases the highest disproportion in average ranks was observed in combination with SVM, this time in favor of RBO, but the results were not statistically significant.

It is worth noting that for CART, KNN and SVM classifiers RBU achieved a higher rank with respect to recall than the rank achieved with respect to precision, whereas the opposite was true for NB. This indicates that undersampling with RBU affects NB classification differently than the remaining classifiers, and has less severe impact on the precision of that classifier.

To summarize, while the proposed RBU algorithm, in general case, did not achieve the best results when applied in combination with all of the considered classification algorithms, it performed best when combined with NB classifier, and to a lesser extent with CART and SVM. The areas of applicability with respect to the choice of the classification algorithm partially overlap for RBU and RBO: both algorithms displayed comparatively good performance when combined with the NB classifier, but RBU scored significantly better results when combined with CART. Finally, for CART, KNN and SVM classifiers RBU achieved comparatively better recall than precision, but the opposite was true for the NB classifier.

4.4. Analysis of the impact of dataset characteristics

In the final stage of the conducted experimental study we examined if the performance of the algorithm changes depending on the characteristics of the dataset on which it is applied. Specifically, we considered the categorization proposed by Napierała and Stefanowski [39] to evaluate the fraction of minority objects belonging to one of the categories: safe, borderline, rare and outlier, for each individual dataset. Afterwards, we examined the relation between the fraction of objects of a given type and the rank the RBU method achieved compared to the reference algorithms. In the

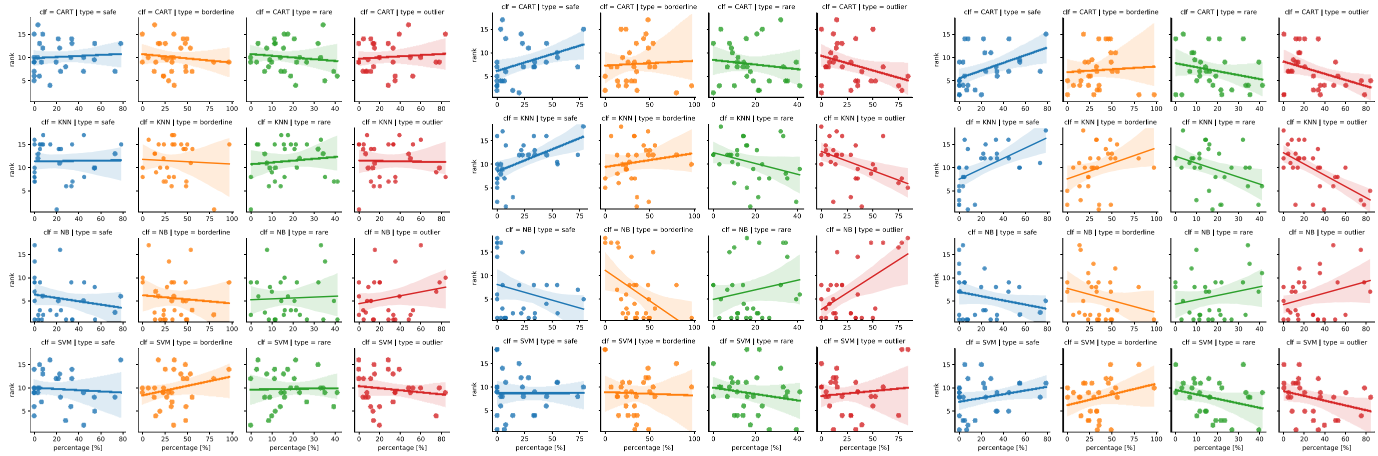

Fig. 6. Scatterplots representing relation between the percentage of the objects of a given type: safe (blue), borderline (orange), rare (green) and outlier (red), and the rank achieved by RBU on the given dataset, with regard to *F*-measure (left grid), AUC (middle grid) and G-mean (right grid). Each row contains data for a single classifier, from the top: CART, KNN, NB and SVM. 95% confidence invervals were shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Pearson correlation coefficients between the proportion of minority objects of a given type and the rank of the RBU algorithm with regard to the given metric. Statistically significant correlations denoted with bold font. Note that negative correlation indicates increasing performance.

	Metric	Safe	Borderline	Rare	Outlier
CART	Precision	**−0.3770**	−0.0434	**+0.3329**	+0.2395
	Recall	**+0.6517**	+0.1797	**−0.4036**	**−0.5895**
	F-measure	+0.0792	−0.1179	−0.1292	+0.0913
	AUC	**+0.3760**	+0.0493	−0.1294	**−0.3414**
	G-mean	**+0.4637**	+0.0632	−0.2387	**−0.3852**
KNN	Precision	−0.2934	−0.1719	**+0.4244**	+0.2299
	Recall	**+0.6305**	**+0.4247**	**−0.5032**	**−0.7397**
	F-measure	+0.0180	−0.0536	+0.1097	−0.0221
	AUC	**+0.4862**	+0.1406	−0.3007	**−0.4458**
	G-mean	**+0.5472**	+0.3050	**−0.3676**	**−0.6185**
NB	Precision	−0.1286	+0.0021	−0.0371	+0.1389
	Recall	−0.1524	−0.2858	**+0.4037**	+0.2064
	F-measure	−0.1802	−0.0823	+0.0476	+0.2227
	AUC	−0.2582	**−0.4682**	+0.1975	**+0.5690**
	G-mean	−0.2204	−0.2354	+0.2270	**+0.3114**
SVM	Precision	−0.0207	+0.0583	+0.0107	−0.0373
	Recall	**+0.4724**	**+0.4331**	**−0.4653**	**−0.6141**
	F-measure	−0.0799	+0.2325	+0.0224	−0.1414
	AUC	+0.0024	−0.0350	−0.1779	+0.1143
	G-mean	+0.2390	+0.2559	−0.2864	**−0.3189**

Table 3 we present the Pearson correlation coefficient between the percentage of the objects of a given type and the rank obtained for that dataset, with highlighted statistically significant correlations at the significance level $\alpha = 0.10$. Furthermore, in Fig. 6 we present scatterplots containing individual data points with a linear regression model fit. Note that ranking was performed in a descending order, meaning that the best performing method received the rank equal to 1. Therefore, negative correlation and regression slope indicate that the relative rank increases.

Similarly to the results observed during the comparison with reference methods, the trends observed for CART, KNN and SVM classifiers differed from the ones observed for the NB classifier. For all of the three former classifiers a statistically significantly worse recall, compared to the reference methods, was observed when the datasets contained a high proportion of safe objects. Conversely, a statistically significantly better recall was observed for the datasets containing higher proportion of rare and outlier minority objects. This, in turn, led to a significantly higher values of AUC and G-mean for the datasets containing a larger proportion of rare and outlier minority objects, and significantly lower values of these metrics for datasets containing large proportion of safe objects. In the case of NB classifier, on the other hand, a significantly worse results, compared with the reference methods, were observed with regard to AUC and G-mean for datasets with a large proportion of outliers, and significantly better results with regard to AUC for datasets with a large proportion of borderline minority objects. However, no significant relations with regard to precision or recall were observed for NB in those cases.

To summarize, the results of the analysis of the impact of dataset characteristics indicate that the proposed RBU algorithm, when used with CART, KNN or SVM classifier, is particularly well suited for resampling datasets with a high proportion of rare and outlier minority objects, but achieves a relatively worse performance for safe datasets. This is caused mainly due to the differences in the classification recall. However, these trends do not extended to the case of the NB classifier, for which the observed performance with regard to the AUC and G-mean was actually worse when datasets contained a high proportion of outliers.

5. Conclusions

Throughout this paper we proposed a novel undersampling algorithm, Radial-Based Undersampling, based on a previously introduced concept of mutual class potential. The main motivation behind the proposed algorithm was extending the notion of non-nearest neighbor based resampling, previously used in Radial-Based Oversampling, to the undersampling procedure. The proposed method offers a conceptually simple and computationally more efficient alternative to the Radial-Based Oversampling algorithm. In the conducted experimental study we empirically evaluated the usefulness of the proposed method. Through the course of the study we were able to identify the areas of applicability of the algorithm. Specifically, the observed results indicate the suitability of the algorithm to be used in combination with naive Bayes classifier and, to a lesser extent, CART decision tree and support vector machine. Compared to the Radial-Based Oversampling, RBU displayed a statistically significantly better performance when combined with CART decision tree. Furthermore, we were able to analyse the behavior of the proposed algorithm with respect to the characteristics of datasets on which it was applied. For the majority of the examined classification algorithms proposed method achieved comparatively better results when used on difficult datasets, consisting of higher proportion of rare and outlier minority instances.

Despite the relative simplicity of the proposed criterion of undersampling selection, the observed results are encouraging for further development of the algorithm. Specifically, we intend to explore the possibility of using other selection criteria based on the idea of mutual class potential, with a particular focus on more theoretically motivated choices.

Declaration of Competing Interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgments

This work was supported by the Polish National Science Center under the grant no. 2017/27/N/ST6/01705 as well as the PLGrid Infrastructure.

References

[1] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recognit. Artif. Intell. 23 (4) (2009) 687–719.
[2] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, Prog. Artif. Intell. 5 (4) (2016) 221–232.
[3] P. Branco, L. Torgo, R.P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM Comput. Surv. 49 (2) (2016) 31:1–31:50.
[4] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM Sigkdd Explorations Newsl. 6 (1) (2004) 40–49.
[5] X.-w. Chen, M. Wasikowski, Fast: a ROS-based feature selection metric for small samples and imbalanced data classification problems, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 124–132.
[6] B. Krawczyk, M. Galar, Ł. Jeleń, F. Herrera, Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy, Appl. Soft Comput. 38 (2016) 714–726.
[7] M. Koziarski, B. Kwolek, B. Cyganek, Convolutional neural network-based classification of histopathological images affected by data imbalance, in: Video Analytics. Face and Facial Expression Recognition, Springer, 2018, pp. 1–11.

[8] E. Ramentol, I. Gondres, S. Lajes, R. Bello, Y. Caballero, C. Cornelis, F. Herrera, Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: the SMOTE-FRST-2T algorithm, Eng. Appl. Artif. Intell. 48 (2016) 134–139.

[9] W. Wei, J. Li, L. Cao, Y. Ou, J. Chen, Effective detection of sophisticated online banking fraud on extremely imbalanced data, World Wide Web 16 (4) (2013) 449–475.

[10] A. Azaria, A. Richardson, S. Kraus, V. Subrahmanian, Behavioral analysis of insider threat: a survey and bootstrapped prediction in imbalanced data, IEEE Trans. Comput. Social Syst. 1 (2) (2014) 135–155.

[11] W.M. Czarnecki, K. Rataj, Compounds activity prediction in large imbalanced datasets with substructural relations fingerprint and EEM, 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 2, IEEE, 2015. 192–192

[12] A. Fernández, C.J. Carmona, M. Jose del Jesus, F. Herrera, A pareto-based ensemble with feature and instance selection for learning from multi-class imbalanced datasets, Int. J. Neural Syst. 27 (6) (2017) 1750028.

[13] M. Koziarski, M. Wożniak, CCR: a combined cleaning and resampling algorithm for imbalanced data classification, Int. J. Appl. Math. Comput. Sci. 27 (4) (2017) 727–736.

[14] M. Lango, J. Stefanowski, Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data, J. Intell. Inf. Syst. 50 (1) (2018) 97–127.

[15] P. Ksieniewicz, M. Woźniak, Imbalanced data classification based on feature selection techniques, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2018, pp. 296–303.

[16] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[17] M. Koziarski, B. Krawczyk, M. Woźniak, Radial-based approach to imbalanced data oversampling, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2017, pp. 318–327.

[18] M. Pérez-Ortiz, P.A. Gutiérrez, P. Tino, C. Hervás-Martínez, Oversampling the minority class in the feature space, IEEE Trans. Neural Netw. Learn. Syst. 27 (9) (2016) 1947–1961.

[19] C. Bellinger, C. Drummond, N. Japkowicz, Manifold-based synthetic oversampling with manifold conformance estimation, Mach. Learn. 107 (3) (2018) 605–637.

[20] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.

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

[22] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-Level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference 2009, Bangkok, Thailand, April 27–30, 2009, Proceedings, 2009, pp. 475–482.

[23] T. Maciejewski, J. Stefanowski, Local neighbourhood extension of SMOTE for mining imbalanced data, in: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining 2011, Part of the IEEE Symposium Series on Computational Intelligence 2011, April 11–15, 2011, Paris, France, 2011, pp. 104–111.

[24] I. Tomek, Two modifications of CNN, IEEE Trans. Syst. Man Cybern. 6 (1976) 769–772.

[25] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Cybern. 2 (3) (1972) 408–421.

[26] P. Hart, The condensed nearest neighbor rule, IEEE Trans. Inf. Theory 14 (3) (1968) 515–516.

[27] I. Mani, I. Zhang, kNN approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of Workshop on Learning from Imbalanced Datasets, vol. 126, 2003.

[28] A. Anand, G. Pugalenthi, G.B. Fogel, P. Suganthan, An approach for classification of highly imbalanced data using weighting and undersampling, Amino Acids 39 (5) (2010) 1385–1391.

[29] M.R. Smith, T. Martinez, C. Giraud-Carrier, An instance level analysis of data complexity, Mach. Learn. 95 (2) (2014) 225–256.

[30] S.-J. Yen, Y.-S. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Syst. Appl. 36 (3) (2009) 5718–5727.

[31] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern.Part B (Cybernetics) 39 (2) (2008) 539–550.

[32] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recognit. 46 (12) (2013) 3460–3471.

[33] W. Lu, Z. Li, J. Chu, Adaptive ensemble undersampling-boost: a novel learning framework for imbalanced data, J. Syst. Softw. 132 (2017) 272–282.

[34] R. Barandela, R.M. Valdovinos, J.S. Sánchez, F.J. Ferri, The imbalanced training sample problem: under or over sampling? in: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, 2004, pp. 806–814.

[35] C. Drummond, R.C. Holte, et al., C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling, in: Workshop on Learning from Imbalanced Datasets II, vol. 11, Citeseer, 2003, pp. 1–8.

[36] J. Van Hulse, T. Khoshgoftaar, Knowledge discovery from imbalanced and noisy data, Data Knowl. Eng. 68 (12) (2009) 1513–1542.

[37] V. García, J.S. Sánchez, R.A. Mollineda, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowl. Based Syst. 25 (1) (2012) 13–21.

[38] D.H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (7) (1996) 1341–1390.

[39] K. Napierala, J. Stefanowski, Types of minority class examples and their influence on learning classifiers from imbalanced data, J. Intell. Inf. Syst. 46 (3) (2016) 563–597.

[40] M. Koziarski, B. Krawczyk, M. Woźniak, Radial-Based Oversampling for noisy imbalanced data classification, Neurocomputing (2019).

[41] B. Krawczyk, M. Koziarski, M. Woźniak, Radial-Based Oversampling for multi-class imbalanced data classification, IEEE Trans. Neural Netw. Learn. Syst. (2019).

[42] B. Bobowska, M. Woźniak, Experimental study on modified Radial-Based Oversampling, in: The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications, Springer, 2018, pp. 110–119.

[43] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2011).

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (Oct) (2011) 2825–2830.

[45] I. Tomek, An experiment with the edited nearest-neighbor rule, IEEE Trans. Syst. Man Cybern. (6) (1976) 448–452.

[46] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, in: Conference on Artificial Intelligence in Medicine in Europe, Springer, 2001, pp. 63–66.

[47] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp. 179–186.

[48] G. Lemaitre, F. Nogueira, C.K. Aridas, Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning, J. Mach. Learn. Res. 18 (17) (2017) 1–5.

[49] E. Alpaydin, Combined 5 × 2 cv F test for comparing supervised classification learning algorithms, Neural Comput. 11 (8) (1999) 1885–1892.

Michał Koziarski received M.Sc. degree in computer science from the Wrocław University of Science and Technology, Poland, in 2016. Currently, he is a Ph.D. student at the Department of Electronics of the AGH University of Science and Technology, Poland. His research interests include computer vision, neural networks and imbalanced data classification.