# Improvement of random undersampling to avoid excessive removal of points from a given area of the majority class*

Małgorzata Bach[0000−0002−6239−7790] and Aleksandra Werner[0000−0001−6098−0088]

Department of Applied Informatics
Faculty of Automatic Control, Electronics and Computer Science
Silesian University of Technology, Gliwice, Poland
{malgorzata.bach,aleksandra.werner}@polsl.pl

**Abstract.** In this paper we focus on class imbalance issue which often leads to sub-optimal performance of classifiers. Despite many attempts to solve this problem, there is still a need to look for better ones, which can overcome the limitations of known methods. For this reason we developed a new algorithm that in contrast to traditional random undersampling removes maximum $k$ nearest neighbors of the samples which belong to the majority class. In such a way, there has been achieved not only the effect of reduction in size of the majority set but also the excessive removal of too many points from the given area has been successfully prevented. The conducted experiments are provided for eighteen imbalanced datasets, and confirm the usefulness of the proposed method to improve the results of the classification task, as compared to other undersampling methods. Non-parametric statistical tests show that these differences are usually statistically significant.

**Keywords:** Classification · Imbalanced dataset · Sampling methods · Undersampling · K-Nearest Neighbors methods.

## 1 Introduction

Uneven class distribution can be observed in datasets concerning many areas of human life – medicine [21, 29], engineering [12, 24], banking, telecomunications [15, 35], scientific tasks such as pattern recognition [30], etc. Many other examples along with the exhaustive state-of-the-art which refers to development of research in learning of imbalanced problem is included in [16, 17]. Unfortunately, a lot of learning systems are not adapted to operate on imbalanced data, and although many techniques have already been proposed in literature it is still an unresolved issue and requires further studies. The charts presented in [16] confirm these facts and show that the number of publications on the problem of class imbalance has increased in recent years.

* This work was supported by Statutory Research funds of Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

We have dealt with this issue for several years, which resulted in the publication of the scientific papers on that subject [2, 4]. The problem has not only been analyzed in the context of specific real data using well-known balancing algorithms, but we have also tried to develop our own data sampling methods that could help reduce problems arising from the skewed data distribution. For example, our method presented in paper [3] is oriented toward finding and thinning clusters of examples from the majority class. However, while this method in many cases outperformed the other compared ones, the results were not fully satisfactory. Therefore, we decided to do further research and analyses, the effect of which is KNN_RU algorithm presented in the article. The combination of random undersampling and the idea of the nearest neighbors allows to remove maximum $k$ nearest neighbors of the samples which belong to the majority class, and thus prevent an excessive removal of too many points from the given area. To investigate the impact of such a method on the result of the binary classification task, we conducted experiments, where 6 classifiers were applied for eighteen datasets of various imbalanced ratio. We also confronted our approach with four other methods belonging to the data-level balancing category and test whether the differences between them are statistically significant. To assess classifiers' accuracy, we applied a number of metrics advisable for classification of imbalanced data. Decision of analyzing scores of multiple estimation methods instead of one was motivated by the fact that no single metric is able to comprise all the interesting aspects of the analyzed model.

The structure of this paper is as follows. Section 2 overviews the ideas which address the imbalanced data challenge. In Section 3 the proposed algorithm of undersampling is outlined. The experiment details are described in Section 4. There are also a short characteristic of analyzed data together with information about applied classifiers and performance metrics used for evaluation. The results of the performed tests and the discussion of the outcomes are given in this part of the paper too, while the conclusions are given in Section 5.

## 2 Learning from imbalanced data

Related literature, e.g. [3, 14, 22, 31], provides information on solutions that counteract the effects of data imbalance. They can be categorized into three major groups: data-level, algorithmic level and cost-sensitive methods.

Data-level approaches tackle class imbalance by adding (oversampling) or removing (undersampling) instances to achieve a more balanced class distribution. Solutions of this category can be used at a preprocessing stage before applying various learning algorithms. They are independent of the selected classifier. Compared to the methods of the other two groups, data level solutions usually require significantly less computing power. They are also characterized by simplicity and speed of operation, which is especially important in the case of large datasets. Importantly, resampling methods can also extend standard ensemble classifiers to prepare data before learning component classifiers. The results presented in

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

[14] show that such extensions applied e.g. to the bagging method significantly improve the outcomes.

The simplest method from the group of undersampling techniques is random undersampling (RU) which tries to balance class distribution by random elimination of majority class examples. However, one of the drawbacks is, it can discard data that is potentially important for learning. To overcome this limitation heuristic approaches are used to identify and remove less significant training examples. These may be borderline examples or examples which are suspicious of being noisy, and their removal can make the decision surface smoother.

One of the most commonly used classes of heuristic undersampling methods is based on *k-Nearest Neighbors* algorithm (KNN). In Wilson's *Edited Nearest Neighbor* (ENN) method undersampling of the majority class is done by removing samples whose class label differs from the class of the majority of their $k$ nearest neighbors [5, 36].

*Neighborhood Cleaning Rule* (NCL) algorithm for a two-class problem can be described as follows: for each example in the training set its three nearest neighbors are found. If tested example $x_i$ belongs to the dominant class and the classification given by its three nearest neighbors contradicts the original class of $x_i$, then $x_i$ is removed. Otherwise, if $x_i$ belongs to the minority class and its three nearest neighbors misclassify $x_i$ as a dominant, then the nearest neighbors that belong to the majority class are removed [27].

*Tomek link* (T-link) algorithm can also be used to reduce majority class [34]. Tomek link can be defined as a pair of minimally distant nearest neighbors of the opposite classes. Formally, a pair of examples $x_i$ and $x_j$ is called a Tomek link if they belong to different classes and are each other's nearest neighbors. Tomek link can be used both as a method of undersampling and data cleaning, in the first case only the majority class examples being a part of Tomek link are eliminated, while in the second case the examples of both classes are removed.

Many other informed undersampling methods can be found in the literature, but because our new solution is a kind of the hybrid of random undersampling and the k-Nearest Neighbor algorithm, we decided to present only the solutions based on mentioned concepts.

## 3 KNN_RU algorithm outline

One of the problems with random undersampling is that there is no possibility to control what objects are removed and thus there is a danger of losing valuable information about the majority class. Accordingly, the method works well only when the removal does not change the distribution of the majority class objects. In other case heuristic methods should be used, which try to reject the least-significant examples of the majority class. Unfortunately, these methods also have some drawbacks, namely they usually do not allow to influence the number of removed elements because it only comes from the nature of the dataset. Therefore, sometimes only a small number of observations meets the

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

**Algorithm.** KNN_RU method for undersampling

**function KNN_RU** ($S_{maj}$, $P$, $k$)
  $l = |S_{maj}|$; // $l$ is the number of examples from the majority class
  ToRemove ← matrix (nrow=$l$, ncol=$k$);
  **for** i = 1 **to** $l$
    Calculate the distance between $i^{th}$ element of $S_{maj}$ and other samples;
    Sort the distance and determine nearest neighbors based on
    the $k$ minimum distance;
    Save indexes of the found neighbors in the $i^{th}$ row of ToRemove matrix;
  $Z = \lfloor P * l \rfloor$; //$Z$ is the number of examples to be removed from $S_{maj}$
  **if** (length(unique(ToRemove)) >= $Z$) **then**
    $R \leftarrow$ sample(unique(ToRemove), $Z$, replace = FALSE)
  **else** $R \leftarrow$ unique(ToRemove);
**return** $S_{maj} - R$ // The subset of the majority class

criteria taken into account in the individual algorithm and is removed from the set.

In order to solve the described problems an attempt was made to create the method which would reduce undesirable effects occurring during random elimination and at the same time allow to determine the number of observations which should be removed from the majority class.

The proposed solution KNN Random Undersampling (KNN_RU) is similar to the traditional random undersampling. The difference is that removing instances is not based on the full set of majority objects, but on $k$ nearest neighbors of each of the samples belonging to the majority class. The ability to control the number of analyzed neighbors and the percentage of undersampling let you fine-tune the algorithm to find such a set of majority objects which allows to achieve the satisfactory accuracy of classification.

The following notations are established to make presentation of the algorithm more clear. $S$ is the training dataset with $m$ examples (i.e., $|S| = m$) defined as: $S = \{(x_i, y_i)\}$, $i = 1, ..., m$, where $x_i \subset X$ is an instance in the $n$-dimensional feature space, and $y_i \subset Y = \{1, ..., C\}$ is a class identity label associated with instance $x_i$[1]. $S_{min} \subset S$ and $S_{maj} \subset S$ are the set of minority and majority class examples in $S$, respectively.

The arguments of the function are: the set of elements belonging to the dominant class – $S_{maj}$, the number of nearest neighbors analyzed for each majority object – $k$, and the percentage of undersampling to carry out – $P$. The result of the function is a subset of the majority class.

Algorithm works as follows. For each element of $S_{maj}$ subset its $k$ nearest neighbors are found and their indexes are stored in the auxiliary matrix. In the next step the duplicated indexes are identified. If the total number of unique indexes is greater than $Z$, where $Z$ is the number of examples which should be removed, then $Z$ random objects from the found set are selected for discarding. However, if due to the nature of the dataset too many objects' indexes are

[1] For the two-class classification problem $C = 2$.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

repeated and consequently the number of unique indexes is less than or equal to $Z$, then all found nearest neighbors are removed. In this case the assumed percentage of undersampling – P may not be achieved. Such a situation can occur primarily when the value of parameter $Z$ is very large while parameter $k$ relatively small.

Due to the fact that in the proposed method for each sample at most $k$ of its neighbors are removed, it reduces the risk of removing too many points from a certain area in comparison with the standard random undersampling method. Consequently, it also decreases danger of losing important information.

## 4 Experiments

The KNN_RU method was compared with several generally known balancing techniques in order to verify whether the proposed algorithm can effectively solve the problem of class imbalance in practice. To make the comparisons the original random undersampling and three heuristic methods: *ENN*, *NCL*, and *Tomek links* were used. All these methods are briefly described in the previous section.

The outline of the performed experiments was as follows:

- Each analyzed dataset was undersampled with five methods. The obtained subsets were treated by six classifiers, such as Naive Bayes, Rule Induction, k-Nearest Neighbor, Random Forests, Support Vector Machines and Neural Networks, and the precision of classification was measured by 6 metrics.
- The tested undersampling methods used parameters $k$ (number of nearest neighbors) and/or $P$ (percentage of undersampling). The sampling of the datasets was performed for the odd values of $k = 1, 3, 5, 7$ and $P = 10\%$, 20%, 30% ... until full balance was achieved. It allowed to find the balancing level that gave the best precision of classification.
- To make the analyses more complete, the results were also compared with those based on the original set of data (i.e. without balancing).

**Experimental environment**

The presented research was performed using the RapidMiner ver. 9.8 and R software environments.

A lot of conventional classification algorithms are often biased towards the majority class and consequently cause higher misclassification rate for the minority examples. All objects are often assigned to the dominant, i.e. negative, class regardless of the values of the feature vector. In [33] authors included a brief introduction to some well-developed classifier learning methods and indicated the deficiency of each of them with regard to the problem of class imbalance. Unfortunately, there are no clear guidelines which classifiers should be used in relation to imbalanced data, therefore descriptions of tests carried out using various classifiers can be found in the literature. Thus, six classifiers based on different paradigms and varying in their complexity were used in presented

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

study: Naive Bayes [20], Rule Induction [25], k-Nearest Neighbor [1], Random Forests[6], Support Vector Machines [9], and Neural Networks [8].

With regard to the classifiers used, the following parameter values were set: (a) Laplace correction was used for NB classifier; (b) The entropy was taken into account as the criterion for selecting attributes and numerical splits for RI classifier; (c) One neighbor was selected for determining the output class in the case of KNN. Additionally Mixed Measures were used to enable the calculation of distances for both nominal and numerical attributes. For numerical values the Euclidean distance was calculated. For nominal values a distance of 0 was taken if both values were the same, and a distance of 1 was taken otherwise; (d) 'Number of trees' parameter which specifies the number of random trees to generate in RF was set to 100; (e) There was used a feed-forward neural network trained by a back propagation algorithm (multi-layer perceptron). 'Training cycles' parameter which specifies the number of cycles used for the neural network training was set to 500. The 'hidden layer size' parameter was set to -1[2]. The default settings were applied for the remaining parameters.

Five independent 5-fold cross-validation experiments were conducted and the final gained results were the average values of these tests[3]. It was stratified validation, which means that each fold contained roughly the same proportions of examples from each class. To optimize parameters of used undersampling algorithms double cross-validations were carried out – the inner one to guide the search for optimal parameters while an outer one to validate those parameters on an independent validation set.

To evaluate tested methods 18 datasets which considered clinical cases, the biology of the shellfish population, proteins in yeast's cell and criminological investigations, was taken from KEEL[4] and UCI[5] repositories. When a dataset was not two-class, each class was successively considered as the positive, while the remaining were merged, thus forming one negative majority class. For that reason some file names have consecutive numbers in their suffix (Table 1). The ratio between the number of negative and positive instances, IR, ranged from 1.82 to 41.4 depending on the dataset.

Regarding performance measures, we chose ones, which are recommended as the most valuable for evaluating imbalanced data classifications [23]: sensitivity, specificity, Balanced Accuracy (BAcc), Geometric Mean (GMean), F-Measure, and Cohen's Kappa statistic.

**Results and discussion**

As mentioned in the paragraph outlining the experiments, the classification tasks were performed for the original datasets as well as for the ones which were undersampled with the use of various methods.

[2] I.e. the layer size was calculated as: 1+(Number of attributes+Number of classes)/2.
[3] We used 5-fold cross-validation instead of 10-fold cross-validation because one of the tested datasets (Glass5) had fewer than 10 examples of the minority class.
[4] http://www.keel.es/datasets.php.
[5] http://archive.ics.uci.edu/ml/index.html.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

Table 1: Datasets summary descriptions

| Name | Instances | Features | IR | Name | Instances | Features | IR |
|---|---|---|---|---|---|---|---|
| Abalone | 731 | 8 | 16.4 | Glass4 | 214 | 9 | 15.46 |
| Breast | 483 | 9 | 18.32 | Glass5 | 214 | 9 | 22.78 |
| Ecoli1 | 336 | 7 | 3.36 | Glass6 | 214 | 9 | 6.38 |
| Ecoli2 | 336 | 7 | 5.46 | Vowel0 | 988 | 13 | 9.98 |
| Ecoli3 | 336 | 7 | 8.6 | Yeast1 | 1484 | 8 | 2.46 |
| Ecoli4 | 336 | 7 | 15.8 | Yeast3 | 1484 | 8 | 8.1 |
| Glass0 | 214 | 9 | 2.06 | Yeast4 | 1484 | 8 | 28.1 |
| Glass1 | 214 | 9 | 1.82 | Yeast5 | 1484 | 8 | 32.73 |
| Glass2 | 214 | 9 | 11.59 | Yeast6 | 1484 | 8 | 41.4 |

The results in terms of F-Measure showed that KNN_RU got the best result in 73 out of 108 tested combinations (18 datasets * 6 classifiers). In the next 5 cases KNN_RU gave the best result ex aequo with the other tested methods. Considering the Kappa metrics the proposed method outperformed the remaining ones in 70 cases and in 11 other cases more than one method achieved the same best result as KNN_RU. Table 2 summarizes the number of the best results of the balancing methods in terms of the F-Measure, Kappa, BAcc, and GMean metrics. More detailed results for the BAcc and Kappa metrics can be found on the Gitlab[6].

Table 2: Summary of the classification best results in terms of the analyzed metrics between KNN_RU and the other balancing methods

| Metrics | KNN_RU | Equal results | Other methods |
|---|---|---|---|
| F-Measure | 73 | 5 | 30 |
| Kappa | 70 | 11 | 27 |
| BAcc | 61 | 7 | 40 |
| GMean | 56 | 8 | 44 |

The proposed KNN_RU solution in most cases gives better results than the other tested methods. However, the obtained results are not always equally good. There may be several possible reasons for this state of affairs. One of them is the fact of different data characteristics. In [19, 28] it is concluded that class imbalance itself does not seem to be a big problem, but when it is associated with highly overlapped classes it can significantly reduce the number of correctly classified examples of minority class.

To analyze the problem more thoroughly the scatterplots were generated for all tested datasets. Two exemplary plots for Glass2 and Glass4 are presented in Figure 1(A)-(B)[7]. Both sets have the same collection of attributes and are

[6] https://gitlab.aei.polsl.pl/awerner/knn_ru

[7] To facilitate visualization and enable the presentation of an exemplary dataset in a two-dimensional space there was performed dimensionality reduction via principal component analysis.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15


Fig. 1: Scatterplot for Glass2 (A), Glass4 (B), Yeast1 (C) and Yeast5 (D) datasets

characterized by a similar degree of imbalancing (IR= 11.59 and IR=15.46, respectively), but they vary in level of class overlapping (higher for Glass2). The results obtained for the Glass2 dataset are much worse than those for Glass4, although the imbalance ratio for the first set is slightly lower. The maximum value of BAcc that was achieved for the tested combinations of classifiers and undersampling methods does not exceed 0.8, while for Glass4 the values are much higher (they reach even 0.9516).

Considering the Kappa metrics, for the Glass2 set KNN_RU algorithm gives the best results for 5 out of 6 tested classifiers. However, the differences are small. None of the tested undersampling methods gives satisfactory results for the Glass2 set. Particularly poor results are obtained for Naive Bayes and SVM classifiers. In these cases, Kappa does not exceed 0.1 for any of the tested undersampling methods.

Figure 1(C)-(D) presents the scatterplots for Yeast1 and Yeast5 datasets. In this case, the sets differ substantially in degree of imbalancing (IR= 2.46 and IR=32.73 respectively). Although the size of the minority class in the Yeast1 set is over 13 times larger than in the Yeast5 one, the results obtained for the first of mentioned dataset are much worse in many of the analyzed cases. This confirms that the difficulty in separating the small class from the dominant one is a very important issue and that the classification performance cannot be stated explicitly taking into consideration only degree of imbalancing, since

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

other factors such as sample size and separability are equally valid. In regard to the Yeast1 set, the maximum achieved value of Kappa is a little over 0.4 (0.416 for NN classifier) while for Yeast5 more then 0.6 (0.625 for RF classifier). It is also noteworthy that for the Yeast5 set the proposed KNN_RU algorithm gives the best results for 5 out of 6 tested classifiers. On the remaining one KNN_RU gives the best result ex aequo with RU methods.

It should be emphasized that in about 80% of the analyzed cases the results which were found to be optimal for the KNN_RU method were achieved for a lower level of undersampling in comparison with the RU method. It is important because any intrusion of the source dataset by its under- and/or over-sampling can cause undesirable data distortion. The major drawback of undersampling is that it can discard potentially useful data that could be important for the learning process. Therefore, it is significant to obtain satisfactory classification accuracy with the least possible interference in input data. The average levels of undersampling with the use of the RU and KNN_RU methods for the Glass2 and Abalone sets are presented in Table 3.

In the case of the T-link, ENN, and NCL methods the number of removed elements comes from the nature of data and the user has no possibility to influence the level of undersampling. For the analyzed datasets these methods tended to remove less number of examples than the RU and KNN_RU methods. However, it can be concluded that in most cases the number of samples removed using the analyzed heuristic methods was not optimal. It means that the tested classifiers often achieved weaker performance than in the case of undersampling using the RU and/or KNN_RU methods. Taking into consideration BAcc measure only in 6 of the analyzed cases the tested heuristic methods of undersampling gave the best results. There were: (a) the combination of the Glass2 dataset, the KNN classifier and the T-link undersampling method; (b) the Ecoli1 dataset, the RI classifier and T-link; (c) the Ecoli2 dataset with the NN classifier sampled using NCL; (d) the Yeast4, the RI classifier and ENN; (e) the Yeast5 dataset, the NB classifier and NCL ex aequo with ENN undersampling method; (f) the Abalone dataset, the KNN classifier and NCL.

It is well known that the choice of the evaluation metrics can affect the assessment of which tested methods are considered to be the best. The results

Table 3: Average level of undersampling (%)

| Classifier | Glass2 | | Abalone | |
|---|---|---|---|---|
| | RU | KNN_RU | RU | KNN_RU |
| NB | 75 | 60 | 10 | 10 |
| RI | 70 | 55 | 80 | 60 |
| KNN | 60 | 30 | 45 | 55 |
| RF | 80 | 40 | 90 | 90 |
| SVM | 90 | 40 | 80 | 80 |
| NN | 60 | 55 | 45 | 20 |

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

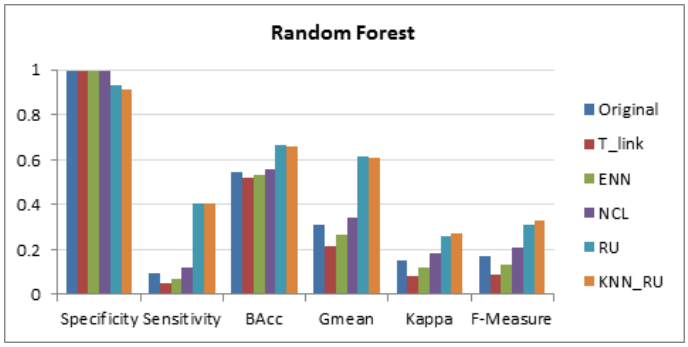


Fig. 2: Performance measures for Random Forest classifier and Abalone dataset.

presented in Table 2 confirm this fact. The various combinations of the classifiers and undersampling methods are often ranked differently by various evaluation measures. For example, according to BAcc for the Abalone dataset and the Random Forest classifier the traditional method of random undersampling (RU) proved to be the best. However, according to the Kappa metrics the proposed KNN_RU algorithm gave better results than RU. Figure 2 presents the results of all analyzed performance measures for the mentioned Random Forest classifier and the Abalone dataset.

To conduct a more complete comparison of the obtained results, the statistical tests [10] were applied. The undersampling methods for each dataset and classifier were ranked separately. The best method received rank 1, the second best – rank 2, and so on. In case of ties, average ranks were assigned. For instance, if two methods reached the same best result they both got rank 1.5. Then, average ranks across all datasets were calculated.

To compare the obtained average ranks, the omnibus Friedman test [13, 32] with Iman-Davenport extension [18] was applied. It allowed to check whether there were any significant differences, with 95% confidence level, between the tested methods. There was used the Conover-Iman post-hoc test[8] to find the particular pairwise comparisons which caused these differences.

Table 4 shows the results of the comparison of the tested undersampling methods calculated for the Kappa metrics. It presents the average ranks for each method and the Conover-Iman p-value between KNN_RU and the method from a given row. In all cases the average ranks for KNN_RU are lower than those obtained for the other methods, and that allows to treat the proposed solution as a control method. The Iman-Davenport test for each used classifier gives p-value less than 0.05. It means that it rejected the hypothesis that the compared undersampling methods were equivalent. The results in bold indicate the methods which are significantly worse than the KNN_RU one. It can be seen that the proposed undersampling method is significantly better than the heuristic ones. However, the comparison with random undersampling (RU) shows

[8] This test is considered to be more powerful than the Bonferroni-Dune one.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

Table 4: Average ranks for the Kappa metrics

| Classifier | Under-sampling method | Avarage rank | p-Conover-Iman | Classifier | Under-sampling method | Avarage rank | p-Conover-Iman |
|---|---|---|---|---|---|---|---|
| NB Iman-Davenport =32.435773; p<0.000001 | KNN_RU | 1.583 | | RF Iman-Davenport =42.389222; p<0.000001 | KNN_RU | 1.361 | |
| | Original | 3.917 | **<0.000001** | | Original | 4.833 | **<0.000001** |
| | T-link | 4.944 | **<0.000001** | | T-link | 4.722 | **<0.000001** |
| | ENN | 3.889 | **<0.000001** | | ENN | 4.389 | **<0.000001** |
| | NCL | 4.667 | **<0.000001** | | NCL | 3.694 | **<0.000001** |
| | RU | 2 | 0.23156 | | RU | 2 | **0.04982** |
| RI Iman-Davenport =9.96058; p<0.000001 | KNN_RU | 1.861 | | SVM Iman-Davenport =38.42044; p<0.000001 | KNN_RU | 1.306 | |
| | Original | 4.44 | **<0.000001** | | Original | 5 | **<0.000001** |
| | T-link | 4.194 | **0.000003** | | T-link | 4.139 | **0.000001** |
| | ENN | 3.806 | **0.000068** | | ENN | 4.472 | **<0.000001** |
| | NCL | 4.111 | **0.000006** | | NCL | 4.028 | **<0.000001** |
| | RU | 2.583 | 0.123512 | | RU | 2.056 | **0.027813** |
| KNN Iman-Davenport =8.040611; p=0.000003 | KNN_RU | 1.806 | | NN Iman-Davenport =13.31763; p<0.000001 | KNN_RU | 1.361 | |
| | Original | 4.361 | **0.000001** | | Original | 4.694 | **<0.000001** |
| | T-link | 4.167 | **0.000001** | | T-link | 4.444 | **<0.000001** |
| | ENN | 3.583 | **0.000459** | | ENN | 3.722 | **0.000003** |
| | NCL | 4.139 | **0.000007** | | NCL | 3.889 | **0.000001** |
| | RU | 2.944 | **0.021884** | | RU | 2.889 | **0.001783** |

that although KNN_RU is better, i.e. has lower average rank in 2 cases: for the NB and RI classifiers, the statistical significance of the observed differences cannot be confirmed.

Better results obtained with the KNN_RU method compared to heuristic ones can be explained by the fact that it allows to influence the number of objects that should be removed. On the other hand, compared to the classic random method of undersampling, the proposed solution reduces the risk of removing too many points from a certain area.

It should be noted that although the method used in KNN_RU to remove objects is slightly more sophisticated than in the case of RU, it does not guarantee cleaning the decision surface, reducing class overlapping or removing noisy examples. Therefore, we decided to create a hybrid solution, which would first use a method that better detects and removes borderline or noisy cases and would apply the KNN_RU solution in the next step.

At the beginning, tests were carried out using the Glass2 dataset, which appeared to be one of the most problematic. Combinations T_link + KNN_RU and NCL + KNN_RU were tested. The obtained results confirmed validity of the conception of hybridization the methods. Table 5 presents the results with the use of basic version of KNN_RU and its combinations. One can see that KNN_RU preceded by T_link or NCL improves values of the classification metrics. Better results are highlighted in bold.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

Table 5: Classification results for the Glass2 dataset

| Classifier | KNN_RU | | | | T_link + KNN_RU | | | | NCL + KNN_RU | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F-measure | BAcc | Kappa | Gmean | F-measure | BAcc | Kappa | Gmean | F-measure | BAcc | Kappa | Gmean |
| NB | 0.196 | 0.634 | 0.065 | 0.583 | **0.197** | **0.637** | **0.067** | **0.587** | **0.198** | **0.637** | **0.067** | **0.587** |
| RI | 0.263 | 0.606 | 0.114 | 0.52 | 0.219 | 0.604 | **0.136** | **0.573** | 0.252 | **0.620** | **0.162** | **0.584** |
| KNN | 0.432 | 0.705 | 0.327 | 0.665 | 0.426 | **0.799** | **0.35** | **0.798** | 0.428 | **0.746** | **0.365** | **0.729** |
| RF | 0.182 | 0.559 | 0.112 | 0.456 | **0.305** | **0.664** | **0.224** | **0.635** | **0.283** | **0.609** | **0.217** | **0.521** |
| SVM | 0.105 | 0.527 | 0.073 | 0.242 | **0.187** | **0.597** | 0.054 | **0.552** | **0.173** | **0.578** | 0.034 | **0.492** |
| NN | 0.457 | 0.765 | 0.393 | 0.756 | **0.542** | **0.827** | **0.486** | **0.824** | **0.474** | 0.712 | **0.432** | 0.670 |

The averages for each measure (average across a column) were calculated. Analyzing the values obtained in this way the improvement was observed for both tested combinations. It was approximately from 3% for NCL + KNN_RU and BAcc up to 23% for T_link + KNN_RU and GMean. The use of the hybrid version of KNN_RU in most cases improves the results also in the remaining datasets.

## 5 Conclusions

Many researchers suggest that random undersampling is one of the more effective resampling methods [11, 26]. However, this method has a drawback, namely it does not allow to control which samples from the majority class are thrown away. To reduce – at least partially – this disadvantage we propose the KNN_RU algorithm which combines random approach with k-nearest neighbors analysis.

Six metrics for classification performance evaluation were examined and as it was shown in the experimental section the choice of quality metrics had an impact on the way the various undersampling methods were ranked. Nevertheless, the outcomes of classification experiments conducted with KNN_RU on eighteen datasets in most cases outperformed the results obtained for four compared undersampling methods.

To make a comparison of the KNN_RU method more comprehensive we also contrasted it with the previously developed one [3], which was mentioned in the Introduction section. For the majority of tested data sets, the advantage of the current solution over the previous one has also been confirmed[9].

In our experiments all resampling methods were used only to perform the undersampling task, which means that chosen samples were removed exclusively from the majority class. However, some methods based on the idea of k-nearest neighbors could be used to remove examples from both classes. It means that each example misclassified by its nearest neighbors can be removed from the training set, regardless of the class it belongs to. It should insure more detailed data cleaning, and in consequence improve the accuracy of the classifications. Therefore, we plan to analyze the next combinations of the proposed method with other resampling methods which provide undersampling and have data cleaning capabilities.

[9] The complete comparisons are on Gitlab: https://gitlab.aei.polsl.pl/awerner/knn_ru

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

## References


1. Aha, D., Kibler, D. (1991): Instance-based learning algorithms. Machine Learning, Vol. 6, pp. 37-66.
2. Bach, M., Werner, A. (2018): Cost-Sensitive Feature Selection for Class Imbalance Problem, 182-194. Springer, Advances in Intelligent Systems and Computing, ISSN: 2194-5357, DOI: 978-3-319-67220-5_17.
3. Bach, M., Werner, A., Palt, M. (2019): The Proposal of Undersampling Method for Learning from Imbalanced Datasets. Procedia Computer Science, Vol. 159 (2019), pp. 125-134, DOI: https://doi.org/10.1016/j.procs.2019.09.167.
4. Bach, M., Werner, A., Żywiec, J., Pluskiewicz, W. (2016): The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis, Information Sciences (2016) Life Sci. Data Analysis, vol. 381, Elseviere, pp. 174-190, DOI: 10.1016/j.ins.2016.09.038, ISSN: 0020-0255.
5. Beckmann, M., et al. (2015): A KNN Undersampling Approach for Data Balancing. Journal of Intelligent Learning Systems and Applications, Vol. 7, 104-116 DOI: http://dx.doi.org/10.4236/jilsa.2015.74010.
6. Breiman L. (2001): Random Forest, Machine Learning, Vol. 45(1), Springer, pp. 5-32.
7. Chawla, N. (2005): Data mining for imbalanced datasets: An overview. The Data Mining and Knowledge Discovery Handbook, Springer, pp. 853-867.
8. Cheng. B, Titterington, D.M. (1994): Neural Networks: A review from a Statistical Perspective. Statistical Science 9, pp. 2-54.
9. Cortes. C, Vapnik V (1995): Support-vector network. Machine Learning 1995, Vol. 20, pp. 273-297.
10. Derrac, J., et al. (2011): A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. Swarm and Evolutionary Computation 1, pp. 3-18.
11. Dittman D., et al. (2014): Comparison of Data Sampling Approaches for Imbalanced Bioinformatics Data. Proceedings of the 27 International Florida Artificial Intelligence Research Society Conference.
12. Duan, L., et al. (2016): A new support vector data description method for machinery fault diagnosis with unbalanced datasets. Expert Systems with Applications, Vol. 64, pp. 239-246.
13. Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association, Vol. 32, No 200, pp. 675-701.
14. Galar, M., et al. (2012): A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man, Cybern., Part C: Appl. Rev., Vol. 42(4), pp. 463-484.
15. Gui Chun (2017): Analysis of imbalanced data set problem: The case of churn prediction for telecommunication, Artificial Intelligence Research 6(2):93 DOI: https://doi.org/10.5430/air.v6n2p93.
16. Haixiang, G., et al. (2017): Learning from class imbalanced data: Review of methods and applications. Expert Syst. Appl. 73, pp. 220-239, DOI: 10.1016/j.eswa.2016.12.035.
17. Harsurinder Kaur, et al. (2019): A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions, ACM Computing Surveys, DOI: https://dl.acm.org/doi/abs/10.1145/3343440.


ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15

18. Iman, R., Davenport, J. (1980): Approximations of the critical region of the fbietkan statistic. Commun. Stat.-Theory Methods, Vol. 9(6), pp. 571-595.
19. Japkowicz, N. (2003): Class Imbalances: Are we Focusing on the Right Issue? ICML-KDD'2003 Workshop: Learning from Imbalanced Data Sets.
20. John, G., Langley, P. (1995): Estimating Continuous Distributions in Bayesian Classifiers. 11th Conference on Uncertainty in Artificial Intelligence, San Mateo, pp. 338-345.
21. Krawczyk, B., et al. (2016): Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. Appl. Soft Comput. 38, pp. 714–726.
22. Lopez, V., et al. (2013): An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. 250, pp 113-141, DOI: 10.1016/j.ins.2013.07.007.
23. Luque, A., et al. (2019): The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, Vol. 91.
24. Mao, W., et al. (2017): Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine. Mechanical Systems and Signal Processing, Vol. 83, pp. 450-473.
25. Michalak, M., Sikora, M., Wróbel, Ł. (2015): Rule Quality Measures Settings in a Sequential Covering Rule Induction Algorithm - an Empirical Approach. Proceedings of the Federated Conference on Computer Science and Information Systems pp. 109-118, DOI: 10.15439/2015F388.
26. Mishra, S. (2017): Handling Imbalanced Data: SMOTE vs. Random Undersampling. IRJET, Vol. 04(08), p-ISSN: 2395 0072.
27. Prati, R.C., Batista, G.E., Monard, M.C (2009): Data mining with imbalanced class distributions: concepts and methods. 4th Indian International Conference on AI. ISBN 9780972741279.
28. Prati, R.C., et al. (2004): Class imbalance versus class overlapping: an analysis of a learning system behaviour. Mexican International Conference on Artificial Intelligence. Vol. 2972 of LNAI, Mexico City, Springer-Verlag.
29. Richardson A., Lidbury B. (2017): Enhancement of hepatitis virus immunoassay outcome predictions in imbalanced routine pathology data by data balancing and feature selection before the application of support vector machines. BMC Med. Info. Decis. Mak. 17, 1, 121.
30. Sandhan, T., Choi, JY. (2014): Handling imbalanced datasets by partially guided hybrid sampling for pattern recognition. 22nd International Conference on Pattern Recognition, pp. 1449-1453, DOI: 10.1109/ICPR.2014.258.
31. SCI2S Research Material on Classification with Imbalanced Datasets (October 2020), A University of Granada Research Group, http://sci2s.ugr.es/imbalanced.
32. SCI2S Research Material on the Use of Non-Parametric Tests for Data Mining and Computational Intelligence (October 2020), A University of Granada Research Group, http://sci2s.ugr.es/sicidm.
33. Sun, et al. (2009): Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 23, No. 4, pp. 687-719, World Scientific.
34. Tomek, I. (1976): Two modifications of CNN. IEEE Transactions on Systems Man and Communications SMC-6, pp. 769-772.
35. Wen-hui Hou, et al.(2020):A novel dynamic ensemble selection classifier for an imbalanced data set: An application for credit risk assessment Knowledge-Based Systems, DOI: https://doi.org/10.1016/j.knosys.2020.106462.
36. Wilson, D.L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Trans. Systems, Man, and Cybernetics, Vol. 2(3), pp. 408-420.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_15