Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 125–134

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

The Proposal of Undersampling Method for Learning from Imbalanced Datasets

Małgorzata Bach[a,*], Aleksandra Werner[a,*], Mateusz Palt

[a]*Silesian University of Technology, Gliwice, Poland*

Abstract

Highly imbalanced data, which occurs in many real-world applications, often makes machine-based processing difficult or even impossible. The over- and under-sampling methods help to tackle this issue, however they often have serious shortcomings. In this paper different methods of class balancing, especially those obtained by undersampling, are analyzed. Besides, a new solution is presented. The method is oriented toward finding and thinning clusters of majority class examples. Removing observations from high-density areas can lead to a less loss of information than in the case of removing individual examples or these from less-density areas. Such approach makes the distribution of examples more even. The effectiveness of the method is demonstrated through extensive comparisons to other undersampling methods with the use of eighteen public datasets. The results of experiments show that in many cases the proposed solution allows to achieve better performance than other tested techniques.

Keywords: classification, imbalanced dataset, sampling methods;

1. Introduction

Class imbalance situations are pervasive in many fields and applications. Typical examples can be the diagnosis of rare diseases where the number of patients suffering from such diseases is very low in the population [1, 3], the detection of fraud in card transactions where the number of legitimate transactions is much higher than the number of fraudulent ones [4, 7] and many other areas of human life [5, 6].

Data mining from skewed datasets can result in models that are strongly predictive for the majority class and poorly predictive for the minority one. This is due to the fact that many classifiers attempt to return the most correct predictions based on the entire dataset, which results in categorizing all data as belonging

* Małgorzata Bach; Aleksandra Werner. Tel.: +48-032-237-1339.
E-mail address: malgorzata.bach@polsl.pl; aleksandra.werner@polsl.pl

1877-0509
10.1016/j.procs.2019.09.167

to the majority class. It is worth pointing out that this class is usually less interesting for the data mining problem than the minority one. In other words the cost of misclassification of minority examples is typically much higher than the cost of misclassification of the majority ones.

Unfortunately, a lot of learning systems are not prepared to cope with imbalanced data and though many techniques have already been proposed in literature the problem is still relevant and needs further studies. This is confirmed by the chart presented in [8] which shows a sharp rise in the number of publications concerning the class imbalance problem in 2013-2016. The trend implies that learning from imbalanced data remains a valuable research topic.

Therefore, in this paper we present the new algorithm (KNN_Order) which is based on the k-Nearest Neighbor idea. The gathered results of the tests show that in many cases the proposed solution allows to obtain better performance in comparison with other methods. Some representative undersampling algorithms were chosen, namely: Edited Nearest Neighbor (ENN) [18], Neighborhood Cleaning Rule (NCL) [11], Tomek link (T-link) [14] and Random Undersampling (RU) [27]. We also selected six learning algorithms with different prediction and classification methods: Naive Bayes (NB) [24], Rule Induction (RI) [19], k-Nearest Neighbor (KNN) [25], Random Forests (RF) [23], Support Vector Machines (SVM) [21] and Neural Networks (NN) [22] to evaluate the obtained results. Classification accuracy was evaluated using various metrics such as: Sensitivity, Specificity, balanced accuracy (BAcc), geometric mean (G-Mean) and Cohen's Kappa statistic (Kappa).

It is obvious that any intrusion in the source dataset by its under- and/or over-sampling can cause the distortion of data. In the case of undersampling methods there is the risk of removing important concepts related to the majority class. On the other hand, when adding minority instances there is the risk of overfitting, i.e. a classifier can construct rules that seem to be precise but in fact only cover the replicated examples. Taking into account all these issues, an attempt was made to create a method that would allow to determine the number of examples which should be removed from the majority class as well as the number of nearest neighbors that should be examined. Thanks to this, we could test various combinations of values of these parameters and chose such those that provided satisfactory classification accuracy.

The rest of the paper is organized as follows. Section 2 presents the matters concerning the class imbalance problem and gives an overview of the related work. In Section 3 the proposed algorithm of undersampling is outlined. The experiment details are described in Section 4. There is also a short description of analyzed data as well as information about applied evaluation metrics. The results of the performed tests and the discussion of the outcomes are given in this part of the paper, too. Finally, Section 5 presents the conclusions of the conducted research and the plans for future work.

2. The class imbalance problem

The problem of imbalanced datasets occurs when each class does not constitute an equal part of dataset, but they vary significantly in the number of samples belonging to them. In this situation the predictive model developed using conventional machine learning algorithms could be biased and inaccurate. This happens because machine learning algorithms are usually designed to improve accuracy by reducing the error. Thus, they do not take into account the class distribution or balance of classes. All examples are often assigned to the dominant, e.g. negative, class regardless of the values of the feature vector. So, it is necessary to remove skewed class distribution.

A lot of various methods for handling the class imbalance problem have been reported in the literature. They can be categorized into three major groups:

- Data-level: In this approach the training instances are modified by adding or removing instances to achieve a more balanced class distribution. There are three approaches for resampling: (a) undersampling the majority class – i.e. creating a subset of the original data by removing chosen samples from the class, (b) oversampling minority class – i.e. creating a superset of the original dataset by generating new samples from existing ones or by replicating the existing ones and (c) a hybrid approach which combines both (a) and (b) [8, 9].

- Algorithmic-level: This approach consists of creating new algorithms or modifying existing ones to be more attuned to class imbalance issues [10].
- Cost-sensitive: These methods consolidate approaches of data- and algorithmic- levels considering higher costs for the misclassification of samples from the positive class with respect to the negative ones [2, 9].

Due to the fact that our new solution is based on the k-Nearest Neighbor algorithm the further part of this section is mainly focused on this class of methods.

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. In other words, an example from the majority class is removed if the number of neighbors from the minority class is predominant [13, 18].

Neighborhood Cleaning Rule (NCL) modifies the ENN in order to improve data cleaning. For a two-class problem the algorithm 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 [11].

To reduce majority class *Tomek link* (T-link) algorithm can be also used [14]. 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. In other words there is no example x_l, such that $\mathrm{d}(x_i; x_l) < \mathrm{d}(x_i; x_j)$ or $\mathrm{d}(x_j; x_l) < \mathrm{d}(x_i; x_j)$, where $\mathrm{d}(x_i; x_l)$ is the distance between x_i and x_l. If two examples form a Tomek link, then either one of them is a noise or both are in the borderline. T-Link algorithm can be used as an undersampling method or as a data cleaning one. As an undersampling only the majority class examples being a part of Tomek link are eliminated while as a data cleaning examples of both classes are removed.

Obviously, there are many other methods based on the KNN algorithm e.g. KNN-Und (KNN Undersampling), CNN (Condensed Nearest Neighbor), OSS (One-sided selection) and so on [13, 31]. Generally, heuristic methods of undersampling, also called focused or informed ones, unlike random undersampling try to reject the least significant examples of the majority class and thus minimize risk of losing important information. Unfortunately, these methods have also some drawbacks, namely they usually do not allow to influence the number of removed elements which depends only on the nature of the dataset. Therefore, sometimes only a small number of observations meets the criteria taken into account in the individual algorithm and is removed from the set.

In order to solve the described problem an attempt was made to create the method that would allow for parametric determination of the number of observations which should be removed from the majority class. The algorithm we propose is described in detail in the next section.

3. Proposed algorithm

The informed resampling methods modify the class distribution taking into account local characteristics of examples. The proposed solution is based on the removal of the nearest observations from the pool of examples selected as k nearest neighbors of each example belonging to the majority class. The idea is to find and thin clusters of examples from majority class. Removing observations from high-density areas can lead to a less loss of information than in the case of removing individual examples or these from less-density areas. Such approach makes the distribution of examples more even.

To make presentation of the algorithm more clear, the following notations are established. The training dataset S with m examples (i.e. $|S| = m$) is defined as: $S = \{(x_i, y_i)\}$, $i = 1, ..., m$, where $x_i \in X$ is an instance in the n-dimensional feature space and $y_i \in Y = \{1, ..., C\}$ is a class identity label associated with instance x_i. For the two-class classification problem $C = 2$. Subsets $S_{min} \subset S$ and $S_{maj} \subset S$ are the sets of minority and majority class examples in S, respectively. Additionally, R is the subset of examples to remove.

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 example – k and the percentage of undersampling to carry out – P. The result of the function is the subset of samples belonging to the majority class.

The outline of the proposed algorithm is presented below.

```
Algorithm KNN_Order:
  # Input
     S_maj, P, k
  # Output
     S_maj−R  // The subset of the majority class

  BEGIN
    l = |S_maj| // l is the number of examples from the majority class
    exToRemove ← matrix (nrow=l * k, ncol=2)

    FOR i = 1 to l
      Find k nearest neighbors for i^th element of S_maj. Save indexes
      of the neighbors and distances between them and the analyzed
      example in the subsequent rows of exToRemove matrix;
    exToRemove ← exToRemove[order(exToRemove[,2]),] // Sorting in terms of ascending distance
    exToRemove ← exToRemove[!duplicated(exToRemove[,1]),] // Removal of repetitive indexes
    Z = ⌊P*l⌋; // The number of examples to be removed from S_maj
    IF (nrow(exToRemove) >= Z)
    THEN R ← exToRemove[1:Z, 1]
    ELSE R ← exToRemove[, 1]
   RETURN S_maj−R
   END
```

For each majority example its k nearest neighbors are found and information about them are stored in the *exToRemove* auxiliary matrix. The matrix contains index of the searched neighbors as well as their distance from the analyzed object. Each row corresponds to one neighbor. In the next step the matrix is sorted according to the increasing value of the distance, and the repetitive indexes are removed. Only the indexes associated with the shortest distance remain in the auxiliary matrix. If the total number of unique indexes stored in the matrix is greater than the number of examples which should be removed, then first Z examples are selected for discarding. However, if due to the nature of the dataset too many examples' indexes occur repeatedly and consequently the number of unique indexes is less than or equal to Z, then all found nearest neighbors are removed.

An appropriate choice of parameter values is important for the proper operation of this method. The problem of index repetition may occur if too few neighbors are declared and consequently samples lying far away from one another can also be removed. Nevertheless, even in such cases the proposed algorithm can give the desired effect of thinning examples' clusters.

4. Experiments

The main purpose of the research was to compare the KNN_Order method with the several balancing techniques published in the literature in order to verify whether the proposed method can in practice effectively cope with the problem of class imbalance. For comparison, three heuristic methods based on KNN algorithm were used: ENN, NCL and Tomek links.

The experiments were carried out according to the following scenario:

- For each analyzed dataset four methods of undersampling were applied and the obtained subsets were submitted to six classifiers[1], namely Naive Bayes (NB) [24], Rule Induction (RI) [19], k-Nearest Neighbor (KNN) [25], Random Forests (RF)[23] Support Vector Machines (SVM) [21] and Neural Networks (NN)[2] [22].
- For the undersampling algorithms in which parameters: k (the number of nearest neighbors) and/or P (percent of undersampling) were used, the tests were repeated in order to find the balancing level that allowed to reach the best precision of classification. The tests were performed for the odd values of $k = 1, 3, 5, 7$ and $P = 10\%, 20\%, 30\%$ ... until full balance was achieved.
- To get the analyzis more complete, the results were also compared with those based on the original set of data (i.e. without balancing).

The presented research was performed using the RapidMiner and R software environment [15, 16]. 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 double cross-validations were carried out – an inner one to guide the search for optimal parameters and an outer one to validate those parameters on an independent validation set.

Input data characteristics

To evaluate tested methods 18 datasets were used. Their characteristics are summarized in Table 1, namely the number of examples *Ex.*, number of attributes *Atts.* and imbalance ratio *IR* which is the ratio between the number of negative and positive instances. All used sets of data were downloaded from UCI machine learning repository [12] and from the KEEL repository [17]. For datasets with more than two classes (e.g. Ecoli, Glass, Yeast), one class was chosen as the positive class and the rest as the negative one. In addition, in the case of Breast set data was slightly modified to simulate a higher degree of class imbalance.

Table 1. Datasets summary descriptions

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

Evaluation metrics for data classification

A lot of evaluation metrics which allow to assess the performance of a classification can be found in the subject literature. Due to the fact that the analyzed datasets can differ in many aspects, including the costs

[1] In [32] it was shown that the default values of classifiers' parameters are not chosen accidentally and attempts to optimize them, not always improve the quality of classification. Therefore for majority of the used classifiers default values of parameters were applied unless otherwise specified.

[2] In the presented experiments 1 neighbor was taken into account.

[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.

of misclassification for individual classes or the degree of asymmetry in the class distribution, no single metric is able to close all the interesting aspects. Various metrics give different, valuable details. Therefore, it is suggested to analyze scores of multiple estimation methods [10].

In presented study the following metrics adapted into imbalanced data problems were analyzed: Sensitivity, Specificity, Balanced Accuracy (BAcc), Geometric Mean (G-Mean also called G-measure) and Cohen's Kappa statistic. Sensitivity can be regarded as the percentage of positive instances correctly classified as belonging to the positive class $\frac{TP}{(TP+FN)}$, while Specificity as the percentage of negative instances correctly classified as belonging to the negative class $\frac{TN}{(TN+FP)}$[4]. G-Mean for binary classification is the squared root of the product of the sensitivity and specificity. BAcc is the arithmetic mean of the sensitivity and the specificity. Kappa informs how much better the performance of the tested classifier is than the performance of the classifier which simply guesses randomly, according to the frequency of each class.

Results and discussion

The classification tasks were performed for the original datasets as well as for the ones, which were undersampled with the use of various methods: ENN, NCL, T-link and KNN_Order. We assessed the values of the individual measures: Sensitivity, Specificity, BAcc, G-Mean, Kappa for each dataset and for each tested classifier.

Summarizing, in this phase 6 classifiers and 4 balancing methods plus version without balancing were tested for 18 datasets. This gave a total of 6*(4+1)*18 = 540 runs of the classification task. Each time we evaluated 5 measures. In fact, there were more runs because for the undersampling algorithms using k and/or P parameters to find the balancing level which gave the best precision of classification, the tests were repeated for various values of these parameters. Information about the results of experiments for Yeast4 dataset and four selected classifiers is presented in Table 2. The juxtaposition does not include all data sets and classifiers due to the volume limitation.

The large number of results obtained in described stage of the experiment made it difficult to present and analyze them, therefore the ranking of the balancing methods was created. The average values over all tested datasets were calculated for each performance metrics and for each balancing method. The values were the basis for determining the ranking for each classifier. The number one in ranking was assigned to the balancing method, which achieved the greatest value for the analyzed measure. The further positions were occupied by the methods with descending values of the measure. In case of ties, average ranks were assigned. For instance, if two methods reached the same best result they both got rank 1.5. The set of ranking positions was the basis for the calculation of average ranking values for each undersampling method (Table 3).

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. This fact was also pointed out by Raeder et al. in [20] and Diez-Pastor et al. in [26]. The various combinations of the classifiers and undersampling methods are often ranked differently by various evaluation measures. It can be seen in relation to the raw results as well as the averaged values used to create the above described rankings. Analyzing the values in Table 2, one can notice, for example, that the RI classifier according to the Kappa measure achieved the best results when using ENN undersampling method, while according to the G-Mean measure the best result was reached by the KNN_Order method.

Looking at Table 3 it can be observed that T-link takes the last place among the tested undersampling methods with regard to the Kappa measure, while in the case of BAcc and G-Mean metrics, ENN takes this position. In our tests the KNN_Order method reaches the highest average ranking position according to BAcc, G-Mean, Kappa and Sensitivity measures. Analyzing the percentage of negative examples correctly classified as belonging to the negative class (Specificity) the best results is achieved for raw data. The subsequent positions are taken by ENN, T-link and ex aequo KNN_Order and NCL, respectively.

[4] TN (True Negatives) is the number of negatives correctly classified, FP (False Positives) is the number of negative examples incorrectly classified as positives, while TP (True Positives) and FN (False Negatives) are respectively the numbers of positive examples correctly/incorrectly classified.

Table 2. The values of the performance measures for Yeast4 dataset

		Specificity	Sensitivity	BAcc	G-Mean	Kappa
RF	Original	1	0.016	0.508	0.128	0.026
	T-link	0.994	0.135	0.565	0.367	0.187
	ENN	0.998	0.098	0.548	0.313	0.145
	NCL	0.996	0.131	0.564	0.361	0.186
	KNN_Order	0.954	0.455	0.704	0.659	0.461
RI	Original	0.961	0.418	0.69	0.634	0.43
	T-link	0.947	0.471	0.709	0.668	0.461
	ENN	0.946	0.529	0.737	0.707	0.517
	NCL	0.95	0.475	0.713	0.672	0.469
	KNN_Order	0.874	0.586	0.73	0.716	0.434
SVM	Original	0.978	0.316	0.647	0.556	0.382
	T-link	0.966	0.102	0.534	0.313	0.44
	ENN	0.973	0.361	0.667	0.592	0.416
	NCL	0.961	0.434	0.697	0.646	0.459
	KNN_Order	0.933	0.553	0.743	0.719	0.505
NN	Original	0.956	0.533	0.744	0.714	0.537
	T-link	0.948	0.557	0.753	0.727	0.544
	ENN	0.956	0.537	0.746	0.716	0.54
	NCL	0.932	0.57	0.751	0.729	0.516
	KNN_Order	0.953	0.562	0.757	0.732	0.559

In the first stage of our tests we compared the KNN_Order undersampling method to other heuristic approaches based on KNN idea. However, in order to extend the evaluation of the proposed solution, we also decided to include random undersampling (RU) in the study. RU is the simplest method of undersampling which tries to balance class distribution by random elimination of majority class examples. Despite its simplicity, it has been empirically demonstrated by many researchers (among others in [28, 27]) that this is one of the more effective resampling methods and it is often very difficult to outperform RU results even using more sophisticated undersampling techniques.

The results of the experiments, augmented by RU, in terms of the Kappa metric[5], are summarized in Figure 1. One can state that proposed KNN_Order method obtained the best result in 50 out of 108 tested combinations (18 datasets * 6 classifiers). In the next 17 cases KNN_Order gave the best result ex aequo with the other tested methods. Only in 10 of the analyzed cases the other tested heuristic methods of undersampling gave the best results. There were: (a) the combination of the Glass0 dataset, the KNN classifier and the ENN undersampling method; (b) the combination of the Glass0 dataset, the SVM classifier and the NCL; (c) the Glass2 dataset, the KNN classifier and the T-link undersampling method; (d) the Glass2 dataset, the NN classifier and NCL; (e) the Ecoli1 dataset with the KNN classifier sampled using NCL; (f) the Ecoli2 dataset with the KNN classifier and T-link; (g) the Ecoli3 dataset with the KNN classifier sampled using NCL; (h) the Yeast1 dataset, the KNN classifier and T-link; (i) the Yeast4 dataset, the RI classifier sampled using ENN; (j) the Abalone dataset, the KNN classifier and the NCL undersampling method.

[5] The similar comparisons were prepared for all of the analyzed metrics, but unfortunately due to volume limitations it is presented only for Kappa.

The RU method outperformed the remaining ones in 31 of cases, but it should be emphasized that in 24 (77.4%) of the analyzed cases the results which were found to be optimal for the RU method were achieved for higher level of undersampling in comparison with the KNN_Order method. It is important because any intrusion in the source dataset by its under- and/or over-sampling can cause undesirable data distortion. Therefore, it is significant to obtain satisfactory classification accuracy with the least possible interference in input data.

Table 3. Average ranks for the analyzed undersampling methods

	Specificity	**Sensitivity**	**BAcc**	**G-Mean**	**Kappa**
Original	1.5	5	4.67	4.33	4.67
T-link	3.67	2.83	3.33	2.67	3.5
ENN	1.83	4	3.67	4.17	3
NCL	4	2.17	2.33	2.83	2.83
KNN_Order	4	1	1	1	1

It can be seen that the proposed KNN_Order solution in many 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. The majority class sometimes consists of many smaller groups, called sub-concepts. These subgroups may be skewed, in themselves, which is described as 'within class imbalance'. Such situation further complicates the data mining task. In addition, datasets may contain noisy examples and/or class overlapping regions. All this make various undersampling approaches effective for various datasets and it is difficult, or even impossible, to identify one which fits them all.

5. Conclusions

The class imbalance is a problem in a number of different areas such as quality assurance, medical diagnosis, credit scoring, fraud detection, and so on. One of the methods to improve this situation is data sampling. There are many different data sampling algorithms, each with their own strengths and weaknesses, which makes choosing the best one difficult.

In the presented research a new undersampling algorithm is proposed. The method used in this solution is based on the removal of the nearest observations from the pool of examples selected as k nearest neighbors of the examples from majority class.

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

Many authors, e.g. [29, 30] show that a combination of oversampling the minority class and undersampling the majority one can provide better classifier performance than only undersampling the majority class. Our previous research confirmed that good results can be achieved thanks to the SMOTE technique, therefore it is planned to combine this oversampling method with the proposed KNN_Order.

It should also be noted that the proposed method used in KNN_Order to remove examples is more sophisticated than e.g. in the case of RU, however, it does not guarantee cleaning the decision surface. Therefore, it would be interesting to create a hybrid solution, which will first use a method that better detects and removes borderline or noisy examples and then will apply the KNN_Order solution. In this regard, we plan to analyze the combinations of the proposed method with other ones which provide data cleaning capabilities.

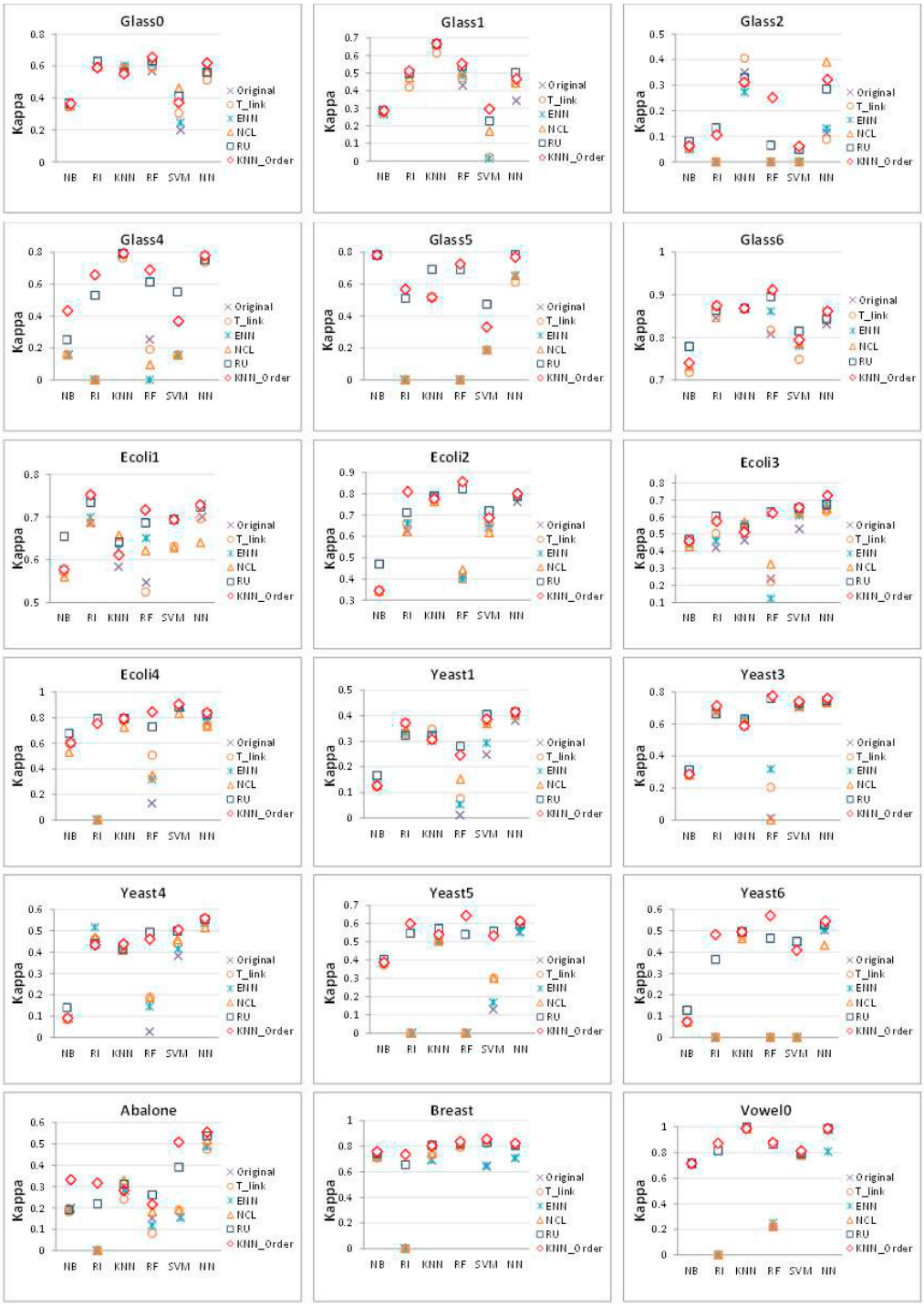

Fig. 1. Kappa results for different datasets

Acknowledgements

This work was supported by Statutory Research funds of Institute of Informatics, Silesian University of Technology, Gliwice, Poland (BK/204/RAU2/2019).

References

[1] Fotouhi, S., Asadi, S., and Kattan MW. (2019) "A comprehensive data level analysis for cancer diagnosis on imbalanced data." *Journal of Biomedical Informatics* 90 DOI:10.1016/j.jbi.2018.12.003

[2] Bach, M., and Werner, A. (2018) "Cost-Sensitive Feature Selection for Class Imbalance Problem" Advances in Intelligent Systems and Computing, Proceedings of 38th ISAT Conference: 182-194.

[3] Adamczyk, P., Werner, A., Bach, M., et al. (2017) "Risk Factors for Fractures Identified in the Algorithm Developed in 5-Year Follow-Up of Postmenopausal Women from RAC-OST-POL Study"*Journal of Clinical Densitometry* (21): 213-219.

[4] Panigrahi, S., Kundu, A., Sural, S., and Majumdar, A.K. (2009) "Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning"*Information Fusion* (10): 354-363.

[5] Duan, L., Xie, M., Bai, T., and Wang, J. (2016) "A new support vector data description method for machinery fault diagnosis with unbalanced datasets" *Expert Systems with Applications* (64): 239-246.

[6] Mao, W., He, L., Yan, Y., and Wang, J. (2017) "Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine" *Mechanical Systems and Signal Processing* (83): 450-473.

[7] Olszewski, D. (2012) "A probabilistic approach to fraud detection in telecommunications" *Knowledge-Based Systems* (26): 246-258.

[8] Haixiang, G., Yijing, L., Shang, J., et al. (2016) "Learning from class imbalanced data: Review of methods and applications" *Expert Systems with Applications* (73): 220-239.

[9] Galar, M., Fernandez, A., Barrenechea, E., et al. (2012) "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches" IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) (42): 463-484.

[10] Lopez, V., Fernandez, A., Garcia, S., et al. (2013) "An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics" *Information Sciences* (25): 113-141.

[11] Prati, R.C., Batista, G.E., and Monard, M.C. (2009) "Data mining with imbalanced class distributions: concepts and methods", Proceedings 4th Indian International Conference on Artificial Intelligence.: 359-376.

[12] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/index.html

[13] Beckmann, M., Ebecken, N.F., and Pires de Lima B.S.L. (2015) "A KNN Undersampling Approach for Data Balancing" *Journal of Intelligent Learning Systems and Applications* (7): 104-116.

[14] Tomek, I. (1976) "Two Modifications of CNN" IEEE Transactions on Systems, Man, and Cybernetics (SMC-6): 769-772.

[15] RapidMiner https://rapidminer.com

[16] The R Project for Statistical Computing https://www.r-project.org

[17] Knowledge Extraction based on Evolutionary Learning http://www.keel.es/datasets.php

[18] Wilson, D.L. (1972) "Asymptotic properties of nearest neighbor rules using edited data" IEEE Transactions on Systems, Man, and Cybernetics (SMC-2): 408-421.

[19] Michalak, M., Sikora, M., and 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: 109-118.

[20] Raeder, T., Forman, G., and Chawla, N.V. (2012) "Learning from imbalanced data: evaluation matters" Data Mining: Foundations and Intelligent Paradigms Springer-Verlag.

[21] Cortes, C., and Vapnik, V. (1995) "Support-vector network" *Machine Learning* (20): 273-297.

[22] Cheng, B., and Titterington, D.M. (1994) "Neural Networks: A review from a Statistical Perspective" *Statistical Science* (9): 2-54.

[23] Breiman, L. (2001) "Random Forest" *Machine Learning* (45): 5-32.

[24] John, G., and Langley, P. (1995) "Estimating Continuous Distributions in Bayesian Classifiers" 11 Conference on Uncertainty in Artificial Intelligence: 338-345.

[25] Aha, D., and Kibler, D. (1991) "Instance-based learning algorithms" *Machine Learning*: 37-66.

[26] Diez-Pastor, J.F., Rodriguez, J.J., Garcia-Osorio, C.I., and Kuncheva, L.I. (2015) "Diversity techniques improve the performance of the best imbalance learning ensembles" *Information Sciences* (325): 98-117.

[27] Mishra, S. (2017) "Handling Imbalanced Data: SMOTE vs. Random Undersampling" *International Research Journal of Engineering and Technology* (04): 317-320.

[28] Dittman, D., Khoshgoftaar, T., Wald, R., and Napolitano, A. (2014) "Comparison of Data Sampling Approaches for Imbalanced Bioinformatics Data" Proceedings of the 27 International Florida Artificial Intelligence Research Society Conference: 268-271.

[29] Estabrooks, A., Jo, T., and Japkowicz, N. (2004) "A Multiple Resampling Method For Learning From Imbalanced Data Sets" *Computational Intelligence* (20): 18-36.

[30] Dubey, R., Zhou, J., Wang, Y., et al.,num (2014) "Analysis of Sampling Techniques for Imbalanced Data: An n=648 ADNI Study" *Neuroimage* (87): 220-241.

[31] Batista, G., Prati, R., and Monard, M. (2004) "A Study of the Behavior of Several Methods for Balancing machine Learning Training Data" *ACM SIGKDD Explorations Newsletter* 6(1): 20-29 DOI:10.1145/1007730.1007735

[32] Kostrzewa, D., Brzeski, R. (2017) "Parametric Optimization of the Selected Classifiers in Binary Classification" Advanced Topics in Intelligent Information and Database Systems, *Studies in Computational Intelligence* 710: 59-69.