Extending Bagging for Imbalanced Data

Jerzy Błaszczyński, Jerzy Stefanowski and Łukasz Idkowiak

Abstract Various modifications of bagging for class imbalanced data are discussed. An experimental comparison of known bagging modifications shows that integrating with undersampling is more powerful than oversampling. We introduce Local-and-Over-All Balanced bagging where probability of sampling an example is tuned according to the class distribution inside its neighbourhood. Experiments indicate that this proposal is competitive to best undersampling bagging extensions.

1 Introduction

Class imbalance is one of obstacles for learning accurate classifiers. Standard learning algorithms tend to show a strong bias toward the majority classes and misclassify too many examples from the minority class. Several methods to address class imbalance have been proposed so far (see, e.g., [8] for a review). In general, they are categorized in *data level* and *algorithm level* ones. Methods within the first category try to re-balance the class distribution inside the training data by either adding examples to the minority class (*oversampling*) or removing examples from the majority class (*undersampling*). The other category covers methods modifying the learning algorithm, its classification strategy or adapting the problem to cost sensitive framework. New type of ensembles of component classifiers is also visible among these methods. They modify sampling strategies (e.g., in bagging), integrate the ensemble with specialized pre-processing method (e.g. SMOTEbagging [13]) or use different cost weights in generalizations of boosting (see, e.g., [7]).

Although these ensembles are presented as a remedy to a certain imbalanced problems, there is still a lack of a wider study of their properties.

Institute of Computing Science, Poznań University of Technology, ul. Piotrowo 2, 60–965 Poznań, Poland

Authors often compare their proposals against the basic versions of other methods. The set of considered imbalance data is usually also limited. Up to now, only two studies were carried out in different experimental frameworks [7, 11]. In [7] authors presented a wide study of 20 different ensembles (all with C4.5 tree classifiers) over 44 data sets. They considered a quite wide range of ensembles from simple modifications of bagging to complex changes of cost Adaboost or hybrid approaches. Their main conclusions said that SMOTEbagging, RUBoost and RUBagging presented the best AUC results. It was also shown that simple versions of undersampling or SMOTE re-sampling combined with bagging worked better than more complex solutions. In [11], two best boosting and bagging ensembles were compared over noisy and imbalanced data. Different amounts of class noise were randomly injected in the training data. The experimental results showed that bagging significantly outperformed boosting. The difference was more significant when data were more noisy. Another surprising conclusion said that it was better to implement sampling without replacement in bagging.

We focus our interest on bagging extensions for class imbalance - following both these related works and its potential usefulness for better handling massive data streams than more complex ensembles. We want to study behavior of bagging extensions more precisely than in [7, 11]. In particular, roughly balanced bagging [9] was missed in [7], although it is appreciated in the literature. The study presented in [11], was too much oriented to the noise level and only two versions of random undersampling in bagging were considered.

Our first objective is to study more precisely a wider set of known extensions of bagging. We will consider exactly balanced bagging and rough balanced bagging but also more variants of using oversampling in bagging – including new type of integrating SMOTE method. We want to check experimentally whether undersampling is better than oversampling in extended bagging. The other contribution is to introduce new extensions of bagging. They are based on the analysis of local neighborhood of each example, which affect the probability of its selection into bootstrap sample – which is a different perspective than the known integrations with pre-processing methods.

2 Adapting Bagging for Imbalanced Data

Bagging introduced by Breiman [4] is an ensemble of T base classifiers induced by the same learning algorithm from T bootstrap samples drawn from the original training set. The predictions of component classifiers form the final decision as the result of equal weight majority voting. The key concept is *bootstrap* aggregation, where the training set for each classifier is constructed by random uniformly sampling (with replacement) instances from the original training set (usually keeping the size of the original data).

2.1 Related Modifications of Bagging

The majority of proposals modify the bootstrap sampling by integrating it with data pre-processing. As the original data is imbalanced the bootstrap sampling will not change significantly the class distribution in the training sample. Its bias toward the majority class could be changed in different ways.

Exactly Balanced Bagging (**EBBag**) is based on a kind of undersampling, which reduces the number of the majority class examples in each bootstrap sample to the cardinality of the minority class (S_{min}) [5]. While creating each training sample, the entire minority class is copied and combined with randomly chosen subsets of the majority class.

Roughly Balanced Bagging (**RBBag**) results from a critique of EBBag [9]. Instead of fixing the sample size, it equalizes sampling probability of each class. Size of the majority class in each bootstrap sample (BS_{maj}) is determined probabilistically according to the negative binomial distribution. Then, S_{min}, and BS_{maj} examples are drawn with or without replacement from the minority class and majority class, respectively. The class distribution in the bootstrap samples maybe slightly imbalanced and varies over iterations. Authors of[9], said that RBBag is more consistent with the nature of bagging and performs better than EBBag. However, authors of [11] claim that there is no significant difference in performance between RBBag and EBBag.

Another way to overcome class imbalance in the bootstrap sampling is oversample the minority class. OverBagging (**OverBag**) is the simplest version which applies oversampling to create each training bootstrap sample. S_{maj} of minority class examples is sampled with replacement to exactly balance the cardinality of the minority and the majority class in each sample. Majority examples are sampled with replacement as in the original bagging.

Another methodology is used in SMOTEBagging (**SMOTEBag**) to increase diversity of component classifiers [13]. Firstly, SMOTE is used instead of random oversampling of the minority class. Then, the SMOTE resampling rate (α) is stepwise changed in each iteration from small to high values (e.g,. from 10% till 100%). This ratio defines the number if minority examples ($\alpha \times S_{min}$) to be additionally re-sampled in each iteration. Quite similar trick to construct bootstrap training samples is also used in the "from undersampling to oversampling" ensemble. According to [7] SMOTEBag gave slightly better results than other good random undersampling ensembles.

2.2 New Proposals

We propose to consider another approaches to bagging based on analyzing the local characteristics of examples and focusing sampling toward more consistent ones. Following [12], type of an example (with respect to its classification consistency) can be identified by analyzing its local neighborhood. It can be

modeled by class assignments of its k-nearest neighbours. For each example, the ratio n/k representing the consistency weight is defined, where n is the number of neighbor examples that have the same class label (the Heterogeneous Valued Distance Metric is used and k is relatively small k=5).

Firstly, we consider modifications of Roughly Balanced Bagging, where the probabilities of sampling the majority class examples are changed. Instead of using equal probabilities they are tuned to reflect the consistency weight of examples. We consider two variants. In **RBBagV1**, weights of the examples correctly re-classified by their neighbours are set to 1 (to focus more interest on safer regions while reducing the role of border examples). On the other hand, in **RBBagV2** the probability of drawing the example is proportional to n/k instead of equal probabilities. In both variants, weights of outlier examples (with all neighbours from the other class) are set to 0.1. The minority examples are drawn with uniform probabilities (as in RBBag).

In the other approach, we prioritize changing probability of example drawing, without modifying the size of the bootstrap samples. In this way, we drop the idea to integrate either undersampling nor oversampling and stay with the original boostrap idea. Our intuition is still to focus the sampling toward more safe, consistent examples. The approach also results in two variants. In Over-All Balanced Bagging (**O-ABBag**), we perform overall (global) balancing. For each majority example, its weight is reduced according to proportion of majority class in the original data, i.e., $\frac{S_{min}+S_{maj}}{S_{maj}}$. The weights of minority examples are increased analogously, with respect to proportion $\frac{S_{min}+S_{maj}}{S_{min}}$. Local-and-Over-All Balanced Bagging (**L-O-ABBag**), on the other hand, uses a mixture of local and global balancing. For each example, its weight is tuned, w.r.t. (with respect to) O-ABBag, in the way to reflect the *local imbalance*, i.e., imbalance in its neighborhood. The tuning takes a form of product of the local and global proportions under the assumption that the local imbalance is independent on the global imbalance. The first component of the product is global proportion, which is calculated exactly as in O-ABBag. The second component is the local proportion that has the same mathematical form as global proportion, however it is calculated taking into account only the examples from neighborhood. The neighborhood is composed of all similar examples, i.e., examples, which share the same description on nominal features and have the value in the same interval on numerical ones (i.e., local discretization) [3]. We consider neighborhoods which are constructed on random subsets of the original feature set (see [2]). More precisely, for each example, we calculate the local proportion w.r.t. random subset of features of size equal to ln of the feature set size. This idea is unlike the previously considered k-nearest neighbours. Instead of taking given k similar examples w.r.t. all features, we take all similar examples, however, w.r.t small random subset of features. In this way, the local proportion promotes examples that are distinct (having different description than any other example) or that have the common description for the class. Taking random subsets of fea-

tures is a way to consider larger neighborhoods, which may help to obtain more diversified bootstrap samples used in bagging.

3 Experiments

In the first experiments we compare literature best extensions of bagging, while in the second experiments we evaluate our new extensions proposed in the previous section. All implementations are done for WEKA framework. Component classifiers b are learned with C4.5 tree learning algorithm (J4.8), which uses standard parameters except disabling pruning.

We choose 22 real-world data sets representing different domains, sizes and imbalance ratio. Most of them come from the UCI repository and are often used in other works on class imbalance, e.g. in [1]. Other data sets come from our medical applications. For data sets with more than two decision classes, we chose the smallest class as a minority class and aggregated other classes into one majority class.

The performance of classifiers is measured using: *sensitivity* of the minority class, its *specificity* and their aggregation to the *geometric mean* (G-mean). For their definitions see, e.g. [8, 10]. They are estimated in stratified 10-fold cross-validation repeated several times to reduce variance.

3.1 Comparison of Known Bagging Extension

The following bagging variants are considered: Exactly Balanced Bagging (denoted further as EBBag), Roughly Balanced Bagging (RBBag) as the best representatives of undersampling generalization. OverBagging (OverBag) and SMOTEBagging (SMOBag) for oversampling perspectives. In case of using SMOBag, we used 5 neighbours and oversampling ration α was stepwise changed in each sample starting from 10%. Moreover, we decided to use SMOTE in yet another way. In the new ensemble, called BagSMOTE, the bootstrap samples were drawn in a standard way, and than SMOTE was applied to balance majority and minority class distribution in each sample (but with the same α, invariant between samples). For all bagging variants, we tested the following numbers T of component classifiers: 20, 50 and 100. Due to space limit, we present detailed results for $T = 50$ only. Results for other T lead to similar general conclusions. The average values of the sensitivity measure are presented in Table 1. The last row contains averaged ranks calculated as in the Friedman test [10]. The test with post-hoc analysis (the critical difference CD = 1.61) shows that EBBag and RBBag leads to significantly better sensitivity than all other bagging variants. However, the

Table 1 Sensitivity [%] for known bagging extensions

data set	Bag	EBBag	RBBag	OverBag	SMOBag	BagSMOTE
abdominal pain	72.05 (5)	81.65 (1)	80.08 (2)	74.22 (4)	71.57 (6)	76.86 (3)
acl-m	83.33 (6)	94.17 (1)	88.5 (2)	84.17 (5)	85.0 (4)	88.33 (3)
balance-scale	0 (6) (6)	49.33 (1)	44.2 (2)	8.83 (3)	1.0 (4)	0.67 (5)
breast-cancer	35.93 (5)	56.06 (2)	56.25 (1)	44.91 (3)	34.36 (4)	50.05 (6)
breast-w	94.88 (6)	96.01 (2)	96.27 (1)	95.84 (3)	95.02 (4)	95.17 (5)
bupa	60.48 (5)	66.97 (3)	68.49 (1)	63.27 (4)	57.02 (6)	67.21 (2)
car	73.97 (6)	100 (1.5)	100 (1.5)	92.62 (3)	92.54 (4)	92.13 (5)
cleveland	9.72 (6)	77.22 (1)	73.5 (2)	16.11 (5)	20.83 (4)	36.11 (3)
cmc	36.67 (6)	66.61 (1)	63.62 (2)	46.47 (4)	40.05 (5)	53.10 (3)
german credit	48.89 (5)	72.5 (2)	91.67 (1)	52.89 (4)	45.89 (6)	63.11 (3)
ecoli	56.67 (6)	78.2 (1.5)	78 (1.5)	60.85 (5)	71.67 (4)	77.11 (3)
flags	0 (6)	70 (1)	67.4 (2)	65.27 (3)	55.6 (4)	20 (5)
haberman	26.38 (6)	60.56 (2)	58.39 (3)	49.86 (4)	48.91 (5)	66.25 (1)
hepatitis	49.44 (6)	81 (2)	81.5 (1)	61.67 (3)	54.44 (5)	67.25 (4)
ionosphere	81.79 (6)	85.73 (2)	85.86 (1)	84.7 (3)	83.7 (4.5)	83.76 (4.5)
new-thyroid	87.5 (6)	95.5 (1.5)	95.5 (1.5)	93.06 (4)	92.22 (5)	93.89 (3)
pima	61.28 (6)	76.7 (1)	75.64 (2)	67.38 (3)	65.13 (4)	63.38 (5)
scrotal pain	58.11 (6)	73.78 (2)	74.6 (1)	65.89 (4)	73.8 (3)	58.56 (5)
solar-flareF	7 (6)	86 (2)	86.7 (1)	42.3 (3)	37.33 (4)	34.4 (5)
transfusion	34.62 (6)	65.45 (2)	64.36 (3)	61.88 (4)	56.54 (5)	68.66 (1)
vehicle	91.29 (6)	91.16 (2)	96.78 (1)	93.46 (4)	92.14 (5)	94.97 (3)
yeast-M2	32.22 (6)	90.22 (1)	89.8 (2)	39.9 (5)	41.18 4)	57.94 (3)
average rank	5.54	1.57	1.52	3.54	4.34	3.52

more precised Wilcoxon test (with α=0.05) shows that differences between these two classifiers are not significant.

While using SMOTE to oversample the minority class, the new integration BagSMOTE performs better than the previously known SMOTE+Bag and OverBag. We also analysed sampling with or without replacement. Conclusions are not univocal. For best undersampling variants like EBBag differences are insignificant while for oversampling standard replacement sampling works much better.

Similar analysis is performed for G-mean (we have to skip details). All extensions are significantly better than the standard version and the ranking of best performing classifiers is the same as for the sensitivity. Again, undersampling methods EBBag and RBBag are better than oversampling bagging variants. However, RBBag seems to be slightly better than EBBag and this trend is more visible for a higher number of component classifiers and using bootstrap sampling with replacement. Bag+SMOTE is also the best performing classifier among oversampling variants. For EBBag and RBBag, we calculated F-measure as yet another popular measure. In this case, RBBag with replacement is better than EBBag with in the Wilcoxon test.

For all bagging variants average values of Q statistics are also calculated to analyze the diversity of component classifiers. Generally, its values are high

positive which indicates that diversity is not high. Sampling with replacement improves the diversity. RBBag produces more diverse classifiers than EBBag.

3.2 Newly Proposed Extensions

Then, we compare newly proposed extensions of bagging for class imbalance (see Section 2.2) to Roughly Balanced Bagging, which show good properties. We will consider: two modifications of RBBag (RBBagV1 and RBBagV2), Over-All Balanced Bagging (O-ABBag), and Local-and-Over-All Balanced Bagging (L-O-ABBag). In case of L-O-ABBag, a random subset of features of size ln of the number of features in the data set is used.

Table 2 Sensitivity [%] calculated for newly proposed extensions of bagging

data set	RBBag	RRBBagV1	RRBBagV2	O-ABBag	L-O-ABBag
abdominal pain	80 (4)	82.26 (1)	82.08 (2)	75.35 (5)	80.4 (3)
acl-m	88.5 (3.5)	89.5 (2)	90 (1)	86 (5)	88.5 (3.5)
balance-scale	44.2 (2)	25.1 (4)	32 (3)	8.163 (5)	99.59 (1)
breast-cancer	56.56 (2)	56.06 (3)	54.97 (4)	51.76 (5)	63.29 (1)
breast-w	96.27 (3)	97.11 (2)	97.52 (1)	95.35 (5)	96.27 (4)
bupa	68.49 (2)	68.37 (3)	66.39 (4)	65.66 (5)	71.03 (1)
car	100 (2.5)	100 (2.5)	100 (2.5)	94.78 (5)	100 (2.5)
cleveland	73.5 (1)	66.67 (4)	67.5 (3)	32.57 (5)	70.29 (2)
cmc	63.62 (2)	61.41 (4)	61.88 (3)	48.41 (5)	68.23 (1)
german credit	89.33 (3)	91 (2)	91.67 (1)	62.6 (5)	71.8 (4)
ecoli	78 (2)	64 (4.5)	64 (4.5)	71.43 (3)	93.14 (1)
flags	66 (4)	67.4 (2)	67.27 (3)	51.76 (5)	92.94 (1)
haberman	58.39 (4)	57.58 (5)	58.58 (3)	59.75 (2)	89.14 (1)
hepatitis	81.5 (1)	76.17 (3)	78.67 (2)	64.38 (5)	71.25 (4)
ionosphere	85.86 (1.5)	85.86 (1.5)	85.4 (3)	85.4 (4)	84.76 (5)
new-thyroid	95.5 (2)	93.33 (3)	92.67 (5)	93.14 (4)	96 (1)
pima	75.64 (4)	76.7 (3)	77.38 (2)	75.3 (5)	79.18 (1)
scrotal pain	74.6 (2)	73.8 (3.5)	73.8 (3.5)	71.53 (5)	77.63 (1)
solar-flareF	86.7 (1)	86.3 (2)	85.4 (3)	41.4 (5)	83.72 (4)
transfusion	64.36 (2)	61.88 (4)	63.12 (3)	60.56 (5)	89.66 (1)
vehicle	96.78 (4)	97.38 (2)	97.29 (3)	93.97 (5)	97.99 (1)
yeast-M2	89.8 (2)	82.33 (4)	83.93 (3)	41.18 (5)	92.94 (1)
average rank	2.48	2.95	2.84	4.68	2.05

Again we show results only for $T = 50$ classifiers. The average values of sensitivity and G-mean are presented in Table 2, and Table 3, respectively (with average ranks). The results of Friedman tests (with CD = 1.3), reveals that only O-ABBg is the worst classifiers. Still, we can give some more detailed observations. According to the average ranks on sensitivity the best performing is L-O-ABBag. However, according to Wilcoxon, its difference

to RBBag is not so significant (p-value is just at 0.05). The two modified versions of RBBag perform similarly to each other.

For G-mean, RBBag is the best classifier according to average ranks. However, as Wilcoxon test indicates, its results are not significantly better than these of RBBagV1, and L-O-ABBag. The worst classifier in comparison is again O-ABBag.

Table 3 G-mean [%] calculated for newly proposed extensions of bagging

data set	RBBag	RRBBagV1	RRBBagV2	O-ABBag	L-O-ABBag
abdominal pain	81.04 (1)	80.78 (2)	80.56 (4)	79.3 (5)	80.73 (3)
acl-m	88.97 (3)	89 (2)	89.24 (1)	87.68 (5)	88.65 (4)
balance-scale	51.32 (1)	35.65 (4)	43 (2)	25.42 (5)	39.8 (3)
breast-cancer	60.28 (1)	60.14 (2)	59.68 (4)	59.53 (5)	59.7 (3)
breast-w	96.12 (4)	96.53 (2)	96.71 (1)	95.73 (5)	96.47 (3)
bupa	71.97 (2)	72.25 (1)	70.67 (3)	70.31 (4)	69.45 (5)
car	96.81 (4)	96.94 (2)	96.78 (5)	96.93 (3)	97.32 (1)
cleveland	73.33 (1)	71.23 (3)	71.14 (4)	53.56 (5)	72.06 (2)
cmc	65.29 (2)	65.25 (3)	65.68 (1)	60.12 (5)	64.18 (4)
german credit	87.07 (3)	88.61 (2)	88.71 (1)	67.67 (5)	67.94 (4)
ecoli	71.84 (3)	63.95 (4.5)	63.95 (4.5)	81.19 (2)	89.04 (1)
flags	67.23 (4)	68.81 (2)	68.6 (3)	63.8 (5)	74.04 (1)
haberman	64.17 (1)	63.04 (4)	63.29 (3)	63.53 (2)	50.74 (5)
hepatitis	80.29 (1)	77.49 (3)	78.85 (2)	74.2 (5)	75.04 (4)
ionosphere	90.75 (5)	90.87 (2)	90.84 (3)	90.79 (4)	90.91 (1)
new-thyroid	96.21 (2)	95.07 (4)	94.34 (5)	95.65 (3)	96.72 (1)
pima	74.84 (3)	75.67 (2)	75.83 (1)	74.44 (4)	73.91 (5)
scrotal pain	74.43 (1)	73.62 (4)	73.41 (5)	74.37 (2)	74.16 (3)
solar-flareF	85.03 (2)	85.05 (1)	84.43 (3)	61.33 (5)	82.12 (4)
transfusion	67.65 (2)	67.82 (1)	67.52 (3)	66.93 (4)	44.77 (5)
vehicle	95.23 (2)	94.9 (3)	94.81 (4)	94.77 (5)	96.2 (1)
yeast-M2	85.57 (1)	82.94 (4)	83.64 (3)	63.32 (5)	85.06 (2)
average rank	2.23	2.61	2.98	4.23	2.95

4 Discussion

The results of first experiments clearly show that applying simple random undersampling leads to much better classification performance than oversampling variants. Definitely, both EBBag and RBBag achieve the best results. However, the difference between them and the best oversampling bagging is much higher than shown in [7]. Moreover, according to our results, SMOTEBagging is not as accurate as it has been presented in [13]. A new oversampling bagging variant, where SMOTE is applied with the same oversampling ratio, works better than the previously promoted SMOTEBagging.

Although EBBag and RBBag performs similarly with respect to the sensitivity, RBBag seems to be slightly better than EBBag for G-mean and F-measure, in particular when sampling is done with replacement. This is a bit different observation to [11], where both classifiers worked similarly on all artificially modified noisy data. Authors of RBBag also showed its slightly better performance over EBBag [9] over 9 data sets only (4 of them was also used in our experiments). Yet another novel observation is that sampling with replacement may be profitable for RBBag unlike EBBag, where our results show no differences between sampling with or without replacement. This result is contradictory to a quite strong claim, from [11], that "bagging should be implemented without replacement".

Discussing these results, we can hypothesize that undersampling may help in random distributing majority examples among many small bootstrap samples, which may direct learning to some useful complementary classification patterns. Even though some of bootstrap samples may contain unsafe, difficult majority examples, the final voting with better components reduces their influence. We plan to carry our future experiments of studying the content and diversity of bootstraps in both EBBag and RBBag.

The main methodological contribution of our study is a new extension of bagging called Local-and-Over-All Balanced Bagging. It is based on different principles than all known bagging extensions. Firstly, instead of integrating bagging with pre-processing, we keep the standard bagging idea but we change radically probabilities of sampling examples by increasing the chance of drawing more minority examples. In this sense we somehow oversample some minority examples and go toward another distribution balance inside the bootstrap sample. Moreover, we promote sampling more safe examples with respect to analyzing class distribution in their neighborhood. The next novel contribution is modeling this nearest distribution by finding similar examples with respect to subsets of features describing them. This idea is inspired by our earlier proposal of variable consistency sampling [2] and according to our best knowledge has not been considered yet in case of imbalanced data.

The results of the second part of experiments clearly show that this novel proposal leads to competitive results to best known undersampling bagging extensions. Furthermore, using more local information about class imbalance is more powerful than using only global imbalance ratio (which was often considered in earlier works). In our future works we plan to study more precisely other ways of modeling and using local approaches to class imbalance.

We would also like to extend our comparison to other component classifiers. In final remarks we refer to problems of using these bagging variants to massive data. In our opinion, more complex solutions do not work better than these simpler bagging variants. Notice that bagging is relatively easy to implement (even in a parallel way) and to generalize, e.g., for class imbalance. Therefore, randomly balanced variants of bagging could be attractive with respect to computational costs since appropriate redistribution of the major-

ity class leads to many smaller training samples. Moreover, according to [7], all these bagging extensions lead to smaller trees than boosting variants.

References

1. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, vol. 6(1), 20–29 (2004).
2. Błaszczyński, J., Słowiński, R., Stefanowski, J.: Feature Set-based Consistency Sampling in Bagging Ensembles. Proc. From Local Patterns To Global Models (LEGO), ECML/PKDD Workshop, 19–35 (2009).
3. Błaszczyński, J., Słowiński, R., Stefanowski, J.: Variable Consistency Bagging Ensembles, Transactions on Rough Sets vol. 11, 40–52 (2010)
4. Breiman L.: Bagging predictors. Machine Learning, 24(2), 123–140 (1996).
5. Chang, E.: Statistical learning for effective visual information retrieval. Proc. of ICIP'03, 609-612 (2003).
6. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic Minority Oversampling Technique. Journal of Artifical Intelligence Research, vol. 16, 341-378 (2002).
7. Galar, M., Fernandez, A., Barrenechea, E.; Bustince, H., Herrera, F.: A Review on Ensembles for Class Imbalance Problem: Bagging, Boosting and Hybrid Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics–Part C, vol. 42 (4), 463–484 (2011).
8. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Data and Knowledge Engineering, vol. 21 (9), 1263–1284 (2009).
9. Hido S., Kashima H.: Roughly balanced bagging for imbalance data. Statistical Analysis and Data Mining vol. 2 (5-6), 412–426 (2009).
10. Japkowicz N., Shah M.: Evaluating Learning Algorithms. A Classification Perpsective. Cambridge University Press (2011).
11. Khoshgoftaar T., Van Hulse J., Napolitano A.: Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics–Part A, 41 (3), 552–568 (2011).
12. Napierała, K., Stefanowski, J.: The influence of minority class distribution on learning from imbalance data. In. Proc. 7th Int. Conference HAIS 2012, Part II, LNAI vol. 7209, Springer, 139-150 (2012).
13. Wang, S., Yao, T.: Diversity analysis on imbalanced data sets by using ensemble models. In Proc. IEEE Symp. Comput. Intell. Data Mining, 324-331 (2009).