# LIUBoost : Locality Informed Underboosting for Imbalanced Data Classification

Sajid Ahmed, Farshid Rayhan, Asif Mahbub, Md. Rafsan Jani,
Swakkhar Shatabda, Dewan Md. Farid and Chowdhury Mofizur Rahman
Department of Computer Science & Engineering, United International University, Bangladesh
Email: dewanfarid@cse.uiu.ac.bd

*Abstract*—The problem of class imbalance along with class-overlapping has become a major issue in the domain of supervised learning. Most supervised learning algorithms assume equal cardinality of the classes under consideration while optimizing the cost function and this assumption does not hold true for imbalanced datasets which results in sub-optimal classification. Therefore, various approaches, such as undersampling, over-sampling, cost-sensitive learning and ensemble based methods have been proposed for dealing with imbalanced datasets. However, undersampling suffers from information loss, oversampling suffers from increased runtime and potential overfitting while cost-sensitive methods suffer due to inadequately defined cost assignment schemes. In this paper, we propose a novel boosting based method called LIUBoost. LIUBoost uses under sampling for balancing the datasets in every boosting iteration like RUS-Boost while incorporating a cost term for every instance based on their hardness into the weight update formula minimizing the information loss introduced by undersampling. LIUBoost has been extensively evaluated on 18 imbalanced datasets and the results indicate significant improvement over existing best performing method RUSBoost.

*Index Terms*—Boosting ; Class imbalance ; Undersampling ; Cost-sensitive learning ; Locality information ; RUSBoost ; SMOTEBoost

## I. Introduction

Class imbalance refers to the scenario where the number of instances from one class is significantly greater than that of another class. Traditional machine learning algorithms such as Support Vector Machines [1], Artificial Neural Networks [2], Decision Tree [3], Random Forests [4] exhibit suboptimal performance when the dataset under consideration is imbalanced. This happens due to the fact that, these classifiers work under the assumption of equal cardinality between the underlying classes. However, many of the real world problems such as anomaly detection [5], facial recognition [6] where supervised learning is used are imbalanced. This is why researchers came up with different methods that would make the existing classifiers competent in dealing with classification problems that exhibit class imbalance.

Most of these proposed methods can be categorized into sampling techniques, cost-sensitive methods and ensemble based methods. The sampling techniques either increase the number of minority class instances(oversampling) or decrease the number of majority class instances(undersampling) so that imbalance ratio decreases and the training data fed to some classifier becomes somewhat balanced [7]. The cost sensitive methods assign higher misclassification cost to the minority class instances which is further incorporated into the cost function to be minimized by the underlying classifier. The integration of these cost terms minimizes the classifiers' bias towards the majority class and puts greater emphasis on the appropriate learning of the minority concept [8]. Ensemble methods such as Bagging [9] and Boosting [10] employ multiple instances of the base classifier and combine their learning to predict the dependent variable. Sampling techniques or cost terms are incorporated into ensemble methods for dealing with the problem of class imbalance and these methods have shown tremendous success [11], [12]. As a matter of fact, these ensemble methods turned out to be the most successful ones for dealing with imbalanced datasets [13].

In order to reduce the effect of class imbalance, the aforementioned methods usually attempt to increase the identification rate for the minority class and decrease the number of false negatives. In the process of doing so, they often end up decreasing the recognition rate of the majority class which results in a large number of false positives. This can be equally undesirable in many real world problems such as fraud detection where identifying a genuine customer as fraud could result in loss of loyal clients. This increased false positive rate could be due to under-representation of the majority class(undersampling), over-emphasized representation of the minority class(oversampling) or over-optimistic cost assignment(cost-sensitive methods). The most successful ensemble based methods also suffer from such problems because they use undersampling or oversampling for the purpose of data balancing while the cost-sensitive methods suffer from over-optimistic cost assignment because the proposed assignment schemes only take into account the global between-class imbalance and do not consider the significant characteristics of the individual instances [14].

In this study, we propose a novel boosting based approach called **L**ocality **I**nformed **U**nder**boost**ing (LIUBoost) for dealing with class imbalance. The aforementioned methods have incorporated either sampling or cost-terms into boosting for mitigating the effect of class imbalance and have fallen victim to either information loss or unstable cost assignment. However, LIUBoost uses undersampling for balancing the datasets while retaining significant information about the local characteristics of each of the instances and incorporates that information into the weight update equation of AdaBoost in

the form of cost terms. These cost terms minimize the effect of information loss introduced by undersampling. We have used K-Nearest Neighbor (KNN) algorithm [15] with small $K$ value for locality analysis and weight calculation. These weights are not meant to mitigate the effect of class imbalance in any way. However, these weights are able to differentiate among safe, borderline and outlier instances of both majority and minority classes and provide the underlying base learners with a better representation of both majority and minority concepts. Additionally, LIUBoost takes into account problems such as class overlapping [16], the curse of bad minority hubs [17] that occur together with the problem of class imbalance. The aim of this study is to show the effectiveness of our proposed LIUBoost both theoretically and experimentally. To do so, we have compared the performance of LIUBoost with that of RUSBoost on 18 standard benchmark imbalanced datasets and the results shows LIUBoost significantly improves over RUSBoost.

The remainder of the paper has been arranged as follows. Section II presents related work and motivation behind our proposal, Section III presents our proposed method and Section IV provides the experimental results. Finally, we conclude in Section V.

## II. RELATED WORK

Seiffert et al. [11] proposed RUSBoost for the task of imbalanced classification. RUSBoost integrates random under-sampling at each iteration of AdaBoost [10]. In different studies, RUSBoost has stood out as one of the best performing boosting based methods alongside SMOTEBoost for imbalanced data classification [13], [18]. A major key to the success of RUSBoost is its random under-sampling technique which, in spite of being a simple non-heuristic approach, has been shown to outperform other intelligent ones [19]. Due to the use of this time-efficient yet effective sampling strategy, RUSBoost is more suitable for practical use compared to SMOTEBoost [12] and other boosting based imbalanced classification methods which employ intelligent under-sampling or over-sampling, thus making the whole classification process much more time-consuming. However, RUSBoost may fall victim to information loss when faced with highly imbalanced datasets. This happens due to its component random under-sampling [20] which discards a large number of majority class instances at each iteration, thus the majority class is often underrepresented in the modified training data fed to the base learners. Our proposed method incorporates significant information about each of the instances of the unmodified training set into the iterations of RUSBoost in the form of cost in order to mitigate the aforementioned information loss.

Fan et al. proposed AdaCost [21] which introduced misclassification costs for instances into the weight update equation of AdaBoost. They theoretically proved that introducing costs in this way does not break the conjecture of AdaBoost. However, they did not develop any generic weight assignment scheme that could be followed for different datasets. Their weight assignments were rather domain specific. Karakoulas

et al. [22] proposed a weight assignment scheme for dealing with the problem of class imbalance where false negatives were assigned higher weights compared to false positives. Sun et al. proposed three cost-sensitive boosting methods for the classification of imbalanced datasets AdaC1, AdaC2 and AdaC3 [8]. These methods assign greater misclassification cost to the instances of the minority class. If an instance of the minority class is misclassified, its weight is increased more forcefully compared to a misclassified majority class instance. Furthermore, if a minority instance is correctly classified, its weight is decreased less forcefully compared to a correctly classified majority instance. As a result, appropriate learning of the minority instances is given greater emphasis in the training process of AdaBoost in order to mitigate the effect of class imbalance. All these methods assign an equal cost to all instances of the same class considering the between-class imbalance ratio. None of them take into account local characteristics of the data points.

Most of the methods proposed for classification of imbalanced datasets only take into account the difference between number of instances from the majority and the minority class and try to mitigate the effects of this imbalance. However, this difference is only one of the several factors that make the task of classification extremely difficult. But these additional yet extremely significant factors are often overlooked while designing algorithms for imbalanced classification [23]. One of these factors is the overlapping of majority and minority classes. Prati et al. [24] studied the effect of class overlapping combined with class imbalance by varying their respective degree and deduced that overlapping is even more detrimental to the classifier performance. Garcia et al. [16] examined the performance of six classifiers on datasets where class imbalance and overlapping was high and noticed that KNN [15] with a small value of $\mathbf{K}$(local neighborhood analysis) was the best performer under such circumstances. These observations point towards the feasibility of dealing with the problem of class overlapping in imbalanced datasets through incorporating information about the local neighborhood of the instances into the training process. Another factor responsible for degrading the performance of classifiers in imbalanced datasets is the effect of bad minority hubs. These are instances of the minority class that are closely grouped together in the feature space. If such a group is close to a majority instance, that majority instance will have a high probability of being misclassified [17]. Such effects are not taken into account in the cost assignment scheme proposed by aforementioned cost-sensitive methods for imbalanced classification. However, our proposed method attempts to mitigate the effects of class-overlapping and bad minority hubs by taking into account the local neighborhood of each of the instances while assigning weights to them.

In some recent proposals, authors have incorporated locality information of the instances into their methods in different ways for dealing with imbalanced datasets. He et al. proposed ADASYN [25] over-sampling technique which takes into account number of majority class instances around the existing

minority instances and creates more synthetic samples for the ones with more majority neighbors so that the harder minority instances get more emphasis in the learning process. Blaszczynski et al. proposed Local-and-Over-All Balanced Bagging [26] which integrates locality information of the majority instances into UnderBagging. In this approach, the majority instances with less number of minority instances in their local neighborhood are more likely to be selected in the bagging iterations. Bunkhumpornpat et al. proposed Safe-Level-SMOTE [27] which only uses the safe minority instances for generating synthetic minority samples. Han et al. proposed Borderline-SMOTE [28] which only uses the borderline minority instances for synthetic minority generation. Furthermore, Napierala et al. used locality information of the minority instances to divide them into aforementioned categories such as safe,borderline,rare and outlier [23]. All these aforementioned methods suggest that locality information of minority and majority instances is significant and can be used in the learning process of classifiers designed for imbalanced classification.

### III. PROPOSED METHOD

---

**Algorithm 1** Weight_Assignment(dataset $X$,$k$)

---
1: **for** each instance $x_i \in$ in the training set, $X$ **do**
2:     find **k** nearest nearest neighbors for the instance
3:     $N_s \leftarrow$ number of neighbors with same class
4:     $N_o \leftarrow$ number of neighbors with opposite class
5:     **if** $N_s == 0$ **then**
6:         $Weight^+(i) \leftarrow \delta$
7:         $Weight^-(i) \leftarrow \frac{1}{N_o}$
8:     **else if** $N_o == 0$ **then**
9:         $Weight^+(i) \leftarrow \frac{1}{N_s}$
10:        $Weight^-(i) \leftarrow \delta$
11:     **else**
12:        $Weight^+(i) \leftarrow \frac{1}{N_s}$
13:        $Weight^-(i) \leftarrow \frac{1}{N_o}$
14:     **end if**
15: **end for**
    Return the $Weight^+$ and $Weight^-$

---

**Algorithm 2** LIUBoost($dataset = (X,Y)$)

---
1: $m$ = number of instances
2: $T$ = number of boosting iterations
3: $(Weight^+, Weight^-)$ = Weight_Assignment($dataset$,$k$)

4: **for** $i \leftarrow 1$ to $m$ **do**
5:     $D_i \leftarrow \frac{1}{m}$
6: **end for**
7: **for** $t \leftarrow 1$ to $T$ **do**
8:     $undersampled\_dataset$ = Undersampling($dataset$)
9:     $h_t \leftarrow$ Decision_Tree($undersampled\_dataset$)
10:    $mis\_sum \leftarrow \sum_{h_t(x_i) \neq y_i} D_i^t \cdot Weight^+(i)$
11:    $cor\_sum \leftarrow \sum_{h_t(x_i) = y_i} D_i^t \cdot Weight^-(i)$
12:    update parameter $\alpha_t \leftarrow \frac{1}{2}\log\frac{1+cor\_sum-mis\_sum}{1-cor\_sum+mis\_sum}$
13:    **if** $\alpha_t \leq 0$ **then**
14:       $t \leftarrow t - 1$
15:       return to statement 8
16:    **end if**
17:    **for** $i \leftarrow 1$ to $m$ **do**
18:       **if** $y_i \neq h_t(x_i)$ **then**
19:         $D_{t+1}(i) \leftarrow D_t e^{-\alpha_t \cdot y_i \cdot h_t(x_i) \cdot Weight^+(i)}$
20:       **else**
21:         $D_{t+1}(i) \leftarrow D_t e^{-\alpha_t \cdot y_i \cdot h_t(x_i) \cdot Weight^-(i)}$
22:       **end if**
23:    **end for**
24:    normalize $D$
25: **end for**
26: $g \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
27: Return $h = sign(g)$

---

The pseudo code of our proposed method LIUBoost is given in Algorithm 2. LIUBoost calls Weight_Assignment method given in Algorithm 1 before boosting iterations begin. This method returns two sets of weights $Weight^-$ and $Weight^+$ used respectively to decrease and increase the weights associated an instance. $Weight^+$ are added inside the exponent term of the weight update equation for the misclassified instances at the iteration under consideration while $Weight^-$ are added for the correctly classified instances. As a result, weight of the instances with greater $Weight^+$ grow rapidly if they are misclassified while weight of the instances with greater $Weight^-$ drop rapidly if they are correctly classified. Thus LIUBoost puts greater emphasis on learning the important concepts rapidly. Additionally, LIUBoost performs undersampling at each boosting iteration for balancing the training set.

The alpha terms determine how significant the predictions of each of the individual base learners are in the final voted classification. These terms also play an important role in the weight update formula which ultimately minimizes the combined error. Since LIUBoost has modified the original weight update equation of AdaBoost by adding cost-terms, the alpha term needs to be updated accordingly in order to preserve coherence of the learning process. The alpha term has been updated according to the recommendations from [8].

One thing to notice here is that LIUBoost combines sam-

pling method and cost-sensitive learning in a novel way. The proposed weight assignment method assigns greater $Increase\_Weight$ to borderline and rare instances while assigning less $Increase\_Weight$ to safe instances due to the way it analysis local neighborhood. Napierala et al. [23] proposed a similar method for grouping only the minority instances into four categories such as safe, borderline, rare and outlier. However, LIUBoost also distinguishes the majority instances through weight assignment. When the majority and minority classes are highly overlapped, which is often the case with highly imbalanced datasets [24], undersampling may discard a large number of borderline and rare majority instances which will increase their misclassification probability. LIUBoost overcomes this problem by keeping track of such majority instances through assigned weights and puts greater emphasis on their learning. This is its unique feature for minimizing information loss.

## IV. EXPERIMENTAL RESULTS

This section presents the details of the experimental results carried out in this paper.

### A. Evaluation Metrics

As evaluation metrics, we have used area under the Receiver Operator Curve (AUCROC) and area under the Precision Recall Curve (AUPR) . These curves use Precision, Recall (TPR) and False Positive Rate (FPR) as underlying metrics .

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

Receiver Operating Characteristic (ROC) curve represents false positive rate (fpr) along the horizontal axis and true positive rate (tpr) along the vertical axis. A perfect classifier will have Area Under ROC Curve (AUROC) of 1 which means all instances of the positive class instances have been correctly classified and none of the negative class instances have been flagged as positive. AUROC provides an ideal summary of the classifier performance. For a not so good classifier TPR and FPR increase proportionally which brings the AUROC down. A classifier which is able to correctly classify high number of both positive and negative class instances gets a high AUROC which is our goal in case of imbalanced datasets. AUPR represents tpr down the horizontal axis and precision down the vertical axis. Precision and TPR are inversely related, ie. as Precision increases, TPR falls and vice-versa. A balance between these two needs to be achieved by the classifier, and to achieve this and to compare performance, AUPR curve is used.

Both of the aforementioned evaluation metrics are held as benchmarks for the assessment of classifier performance on imbalanced datasets. However, AUPR is more informative for

TABLE II: Wilcoxon Signed Rank Test Based on Average AUROC

| RUSBoost | LIUBoost | Hypothesis (alpha=0.05) | p-value |
|---|---|---|---|
| 11.5 | 159.5 | Rejected for LIUBoost | 0.00068 |

cases of high class imbalance AUROC. This is because a large change in false positive counts can result in a small change in the FPR represented in ROC. However, the same change results in a greater change of precision since it compares the false positives to the true positives instead of the true negative instances [29].

### B. Results

We have compared the performance of our proposed method LIUBoost against that of RUSBoost over 18 imbalanced datasets with varying imbalance ratio. All these datasets are from KEEL Dataset Repository [30]. Table I contains a brief description of these datasets.

TABLE I: Dataset Description

| Datasets | Instances | Features | IR |
|---|---|---|---|
| pima | 768 | 8 | 1.87 |
| glass5 | 214 | 9 | 22.78 |
| yeast5 | 1484 | 8 | 38.73 |
| yeast6 | 1484 | 8 | 41.4 |
| ecoli-0-3-4_vs_5 | 200 | 7 | 9 |
| abalone19 | 4174 | 8 | 129.44 |
| pageblocks | 548 | 10 | 164 |
| led7digit-0-2-4-5-6-7-8-9_vs_1 | 443 | 7 | 10.97 |
| glass-0-1-4-6_vs_2 | 205 | 9 | 11.06 |
| glass2 | 214 | 9 | 11.59 |
| glass6 | 214 | 9 | 6.38 |
| yeast-1_vs_7 | 459 | 7 | 14.3 |
| poker-8-9_vs_6 | 1485 | 10 | 58.4 |
| haberman | 306 | 3 | 2.78 |
| winequality-red-8_vs_6 | 656 | 11 | 35.44 |
| glass0 | 214 | 9 | 2.06 |
| glass-0-1-5_vs_2 | 172 | 9 | 9.12 |
| yeast-0-2-5-7-9_vs_3-6-8 | 1004 | 8 | 9.14 |

The algorithms have been run 30 times using 10 fold cross validation on each dataset and the average AUROC and AUROC are presented in table III and IV respectively. Decision tree estimator C4.5 has been used as base learner. Both RUSBoost and LIUBoost have been implemented in python. All the experiments have been designed using scikit-learn [31] library.

From the results presented in Table III, we can see that with respect to AUROC, LIUBoost outperformed RUSBoost

TABLE III: Average AUROC Comparison

| Dataset | RUSBoost | Proposed Method |
|---|---|---|
| glass5 | 0.977 | **0.987** |
| yeast5 | 0.984 | **0.988** |
| yeast6 | 0.916 | **0.921** |
| ecoli-0-3-4_vs_5 | **0.987** | 0.981 |
| abalone19 | 0.784 | **0.801** |
| glass-0-1-4-6_vs_2 | 0.701 | **0.780** |
| glass2 | 0.697 | **0.794** |
| page-blocks0 | **0.988** | 0.988 |
| glass6 | 0.961 | **0.966** |
| yeast-1_vs_7 | 0.785 | **0.794** |
| poker-8-9_vs_6 | 0.791 | **0.792** |
| haberman | 0.599 | **0.647** |
| winequality-red-8_vs_6 | 0.708 | **0.727** |
| led7digit-0-2-4-5-6-7-8-9_vs_1 | 0.943 | **0.953** |
| glass0 | 0.858 | **0.869** |
| glass-0-1-5_vs_2 | 0.646 | **0.725** |
| yeast-0-2-5-7-9_vs_3-6-8 | **0.941** | 0.938 |
| pima | 0.689 | **0.704** |

TABLE IV: Average AUPR Comparison

| Dataset | RUSBoost | Proposed Method |
|---|---|---|
| glass5 | 0.766 | **0.835** |
| yeast5 | 0.690 | **0.742** |
| yeast6 | 0.457 | **0.548** |
| ecoli-0-3-4_vs_5 | **0.930** | 0.915 |
| abalone19 | **0.998** | 0.998 |
| glass-0-1-4-6_vs_2 | 0.209 | **0.258** |
| glass2 | 0.257 | **0.263** |
| page-blocks0 | 0.905 | **0.907** |
| glass6 | 0.893 | **0.923** |
| yeast-1_vs_7 | **0.403** | 0.344 |
| poker-8-9_vs_6 | 0.188 | **0.249** |
| haberman | 0.344 | **0.392** |
| winequality-red-8_vs_6 | 0.192 | **0.242** |
| led7digit-0-2-4-5-6-7-8-9_vs_1 | 0.648 | **0.759** |
| glass0 | 0.708 | **0.753** |
| glass-0-1-5_vs_2 | 0.220 | **0.263** |
| yeast-0-2-5-7-9_vs_3-6-8 | **0.835** | 0.824 |
| pima | 0.529 | **0.544** |

over 15 datasets. However, with respect to AUPR, LIUBoost outperformed RUSBoost over 14 datasets out of 15. Results can be found in Table IV.

We have performed Wilcoxon Pairwise Signed Rank Test [32] in order to ensure that the improvements achieved by LIUBoost are statistically significant. This is highly recommended for comparing the performance of two machine learning algorithms. The test results indicate that the performance improvements both with respect to aupr and auroc are significant since the null hypothesis of equal performance has been rejected at 5% level of significance in favor of LIUBoost. Wilcoxon test results can be found in Table II and Table V.

TABLE V: Wilcoxon Signed Rank Test Based on Average AUPR

| RUSBoost | LIUBoost | Hypothesis (alpha=0.05) | p-value |
|---|---|---|---|
| 23.5 | 146.5 | Rejected for LIU-Boost | 0.0037 |

## V. CONCLUSION

In this paper, we have proposed a novel boosting based algorithm for dealing with the problem of class imbalance. Our method LIUBoost is the first one to combine both sampling technique and cost-sensitive learning. Although good number of methods have been proposed for dealing with imbalanced datasets, none of them have proposed such an approach. We have tried to design an ensemble method that would be cost-efficient just like RUSBoost but would not suffer from the

resulting information loss and the results so far are satisfying. Additionally, recent research has indicated that dividing the minority class into categories is the right way to go for imbalanced datasets [23], [33]. In our opinion, both majority and minority instances should be divided into categories and the hard instances should be given special importance in imbalanced datasets. This becomes even more important when the underlying sampling technique discards some instances for data balancing.

Class imbalance is prevalent in many real world classification problems. However, the proposed methods have their own deficits. Cost-sensitive methods suffer from domain specific cost assignment schemes while oversampling based methods suffer from overfitting and increased runtime. Under such scenario, LIUBoost is cost-efficient, defines a generic cost assignment scheme, does not introduce any false structure and takes into account additional problems such as bad minority hubs and class overlapping. The results are also statistically significant. In future work, we would like to experiment with other cost assignment schemes.

## REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] J. J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, 1988.

[3] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "Iterative boolean combination of classifiers in the roc space: An application to anomaly detection with hmms," *Pattern Recognition*, vol. 43, no. 8, pp. 2732–2752, 2010.

[6] Y.-H. Liu and Y.-T. Chen, "Total margin based adaptive fuzzy support vector machines for multiview face recognition," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1704–1711.

[7] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[8] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[9] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[11] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[12] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2003, pp. 107–119.

[13] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "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)*, vol. 42, no. 4, pp. 463–484, 2012.

[14] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, "A novel ensemble method for classifying imbalanced data," *Pattern Recognition*, vol. 48, no. 5, pp. 1623–1637, 2015.

[15] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126, 2003.

[16] V. García, J. Sánchez, and R. Mollineda, "An empirical study of the behavior of classifiers on imbalanced and overlapped data sets," *Progress in Pattern Recognition, Image Analysis and Applications*, pp. 397–406, 2007.

[17] N. Tomašev and D. Mladenić, "Class imbalance and the curse of minority hubs," *Knowledge-Based Systems*, vol. 53, pp. 157–172, 2013.

[18] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Comparing boosting and bagging techniques with noisy and imbalanced data," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 3, pp. 552–568, 2011.

[19] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126, 2003.

[20] A. Liu, J. Ghosh, and C. E. Martin, "Generative oversampling for mining imbalanced datasets." in *DMIN*, 2007, pp. 66–72.

[21] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, "Adacost: misclassification cost-sensitive boosting," in *Icml*, vol. 99, 1999, pp. 97–105.

[22] G. I. Karakoulas and J. Shawe-Taylor, "Optimizing classifers for imbalanced training sets," in *Advances in neural information processing systems*, 1999, pp. 253–259.

[23] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *Journal of Intelligent Information Systems*, vol. 46, no. 3, pp. 563–597, 2016.

[24] R. C. Prati, G. Batista, M. C. Monard *et al.*, "Class imbalances versus class overlapping: an analysis of a learning system behavior," in *MICAI*, vol. 4. Springer, 2004, pp. 312–321.

[25] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 1322–1328.

[26] J. Błaszczyński, J. Stefanowski, and Ł. Idkowiak, "Extending bagging for imbalanced data," in *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*. Springer, 2013, pp. 269–278.

[27] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," *Advances in knowledge discovery and data mining*, pp. 475–482, 2009.

[28] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," *Advances in intelligent computing*, pp. 878–887, 2005.

[29] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.

[30] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[33] K. Borowska and J. Stepaniuk, "Rough sets in imbalanced data problem: Improving re–sampling process," in *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, 2017, pp. 459–469.