Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques

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

***Abstract*—Class imbalance classification has become a dominant problem in supervised learning. The bias of majority class instances dominates in quantity over minority class instances in imbalanced datasets, which produce the suboptimal classification results for classifying the minority class instances. In the last decade, several methods including sampling techniques, cost-sensitive learning, and ensemble techniques have been introduced for dealing with class imbalance classification. Among all the methods, the ensemble method performs better in compare with sampling and cost-sensitive learning. The ensemble learning uses sampling technique (either under-sampling or over-sampling) with bagging or boosting algorithms. However, which sampling techniques will work better with ensemble learning to improve class imbalance is extremely depend on problem domains. In this paper, we propose two bagging based methods: (a) ADASYNBagging, and (b) RSYNBagging for dealing with imbalanced classification. The ADASYNBagging uses ADASYN based over-sample technique with bagging algorithm. On the other hand, the RSYNBagging uses random under-sampling and ADASYN based over-sample technique with bagging algorithm. RSYNBagging utilizes both under-sampling and over-sampling in alternate iterations and thus incorporates the advantages of both techniques without introducing any extra parameter to tune or increasing time complexity. We have tested the performance of our proposed ADASYNBagging and RSYNBagging methods against existing best performing methods Underbagging, SMOTEBagging on 11 benchmark imbalanced datasets and the initial results are strongly encouraging.**

***Keywords*—Bagging; Classification; Ensemble classifier; Imbalanced datasets; Sampling**

I. Introduction

Canonical machine learning (ML) algorithms such as Support Vector Machines (SVM) [1], Artificial Neural Networks (ANN) [2] and Decision Trees (DT) [3] usually show impressive predictive accuracy in supervised classification. However, these algorithms work under the supposition that the underlying class distribution is balanced [4]. Unfortunately, this assumption does not remain valid in many real world classification domains as instances of different classes vary greatly in number, which is commonly known as the class imbalance problem [5], [6]. Some of the most significant classification tasks show this characteristic [7]. As a result, these existing ML algorithms fail to classify the minority class instances correctly and flag them incorrectly as majority class instance. There are several methods have been proposed by the computational modeling researcher to address the class imbalance problem in the last decade. These methods either altered the existing classifiers [8], [9], or pre-processed the training data before feeding it to the classifiers [10], [11] that is known as sampling techniques. The sampling techniques either decrease the number of majority class instances (under-sampling) or increase the number of minority class instances (over-sampling) to form balance distribution of the training datasets. Cost-sensitive methods have also been proposed to combine both data pre-processing and algorithmic modification while keeping in mind that the underlying class distribution is imbalanced [12], [13] by assigning higher misclassification cost to minority class instances. Recently, ensemble methods have shown promising performance in the classification of imbalanced datasets [14]. When data level re-sampling, such as under-sampling or over-sampling, is used as a pre-processing step to the base learners of ensembles, the performance seems to be most optimal [15], [16].

In this paper, we proposed two hybrid approaches namely RSYNBagging and ADASYNBagging for dealing with class imbalance problem by employing bagging [17] and sampling techniques. The RSYNBagging applies ADASYN [18] on minority class instances in bagging iteration and then random under-sampling [10] on majority class instances on the next bagging iteration. The ADASYNBagging applies ADASYN on minority class instances while keeping the majority class instances untouched using bagging. In the literature, among existing bagging and re-sampling based hybrid approaches, UnderBagging and SMOTEBagging stand out as best performers [16]. So, we have compared the performance of our proposed methods with UnderBagging and SMOTEBagging methods on 11 imbalanced benchmark datasets. Initial results suggest that our proposed methods are competent in the task of correctly classifying instances of both majority and minority classes of imbalanced datasets.

The remainder of this paper is organised as follows. Section II presents related works. Section III describes the data balancing methods, and Section IV presents the proposed methods. Section V provides the experimental results. Finally, we conclude in Section VI.

II. Related work

In recent years, the class imbalance problem has garnered huge interest in the computational intelligence research community. Various methods have been proposed which can often be used as extensions to traditional ML approaches, thus making them compatible with the imbalance classification scenario. Non-heuristic approaches random under-sampling which randomly discards majority class instances and random over-sampling which randomly duplicates minority class instances

[10] turned out to be quite effective yet simple pre-processing steps for ML algorithms.

Chawla et al. proposed an over-sampling method named, SMOTE [11], which creates new minority class instances instead of duplicating the existing ones. He et al. proposed new approach named ADASYN [18], which creates synthetic minority instances using k nearest neighbor [19]. However, ADASYN assigns weight to the minority class instances based on how many neighbors are of the majority class. This weight is used to determine how many synthetic instances to generate from which minority instance.

Breiman et al. proposed an ensemble classifier bagging [17] that combines the predictions of multiple base learners through majority vote. The bagging shows better predictive accuracy compared to the individual base classifiers in the case of balanced classification tasks. Barandela et al. proposed UnderBagging [20] that uses random under-sampling as a pre-processing step for each of the base classifiers in bagging.

Wang et al. [21] proposed three hybrid bagging based methods UnderOverBagging, OverBagging, and SMOTEBagging for dealing with class imbalance problem. OverBagging is quite similar to UnderBagging; only random over-sampling is done instead of random under-sampling in each iteration. UnderOverBagging trains the base classifiers with varying resampling rates (multiples of 10), which determine the number of instances coming from each of the classes, thus using random under-sampling and random over-sampling according to need. SMOTEBagging uses SMOTE as a pre-processing step for the base classifiers. These hybrid bagging based ensembles have shown improved predictive accuracy in imbalanced learning domains.

Lu et al. proposed HSBagging [22], which used both random under-sampling and SMOTE at every bagging iteration as a pre-processing step. HSBagging chooses an optimal sampling rate (majority : minority ratio) for SMOTE and random under-sampling based on predictive accuracy on OUT-OF-BAG instances. This method showed better performance compared to UnderBagging and SMOTEBagging. However in this method, time complexity has been increased due to sampling rate optimisation. Our proposed RSYNBagging is also combining under-sampling and over-sampling, but both are not applied in the same iteration. Hence sampling rate could be kept fixed without increasing run time cost. Additionally, since base learners trained on both under-sampled and over-sampled versions of train set has been incorporated into decision making, de-correlation of the learners has been boosted in our method.

III. Data Balancing Methods

Throughout the years, many approaches for dealing with the problem of class imbalance have been proposed by researchers. Here, we will describe some of them .

A. Sampling Techniques

1) Under-sampling: Under-sampling methods usually reduce the number of majority class instances. The most commonly used under-sampling method is random under-sampling. Random under-sampling randomly removes majority class instances until some desired ratio of majority to minority has been achieved. This random removal may alter the distribution of majority instances and thus alter their representative characteristics. If this happens, the result will be a misclassification of a huge number of majority instances. However, in spite of such drawbacks, random under-sampling usually performs well compared to other under-sampling and data cleaning approaches.

2) Oversampling: In contrast to under-sampling, oversampling works on the minority class instances and increases their quantity to reach a balanced ratio between the two. There are many approaches to achieve oversampling and the most basic of them is random oversampling where randomly selected minority class instances are duplicated until a balanced dataset is achieved. However, this leads the problem of overfitting in which the classifier becomes biased towards the duplicated data and cannot generalize the model for classifying new test instances. To overcome the limitations of random oversampling, several new oversampling techniques have been proposed by researchers. Chawla et al proposed a significant improvement to random oversampling named SMOTE [11]. This creates synthetic minority class instances by interpolation. Hence the increased diversity of the minority class instances reduces the problem of overfitting. An improvement on SMOTE named MSMOTE was proposed by Hu et al. [23]. This continues on the concept of SMOTE but additionally considers the distribution of the minority class instances before interpolating. Under-sampling method generally works better than oversampling methods as long as the imbalance ratio of the dataset is not very high [24].

B. Cost-sensitive learning

Instead of changing the distribution of class instances, cost sensitive learning attaches different misclassification costs to the minority and majority class instances and orients it towards reducing the total cost. The change in underlying cost means that it is more costly to misclassify a minority class instance than a majority class instance. Hence it overcomes the problem of traditional machine learning algorithms, which assume that the misclassification costs are equal. However, it is very challenging to create the cost matrix that defines these misclassification costs.

C. Ensemble Learning

Ensemble methods combine the decisions of multiple base classifiers with the goal of producing a more accurate prediction compared to its individual components. Accuracy and diversity of the individual base learners are two qualities that are crucial in order for the ensemble classifier to perform well. These qualities are ensured by the ensemble methods in various ways according to their working principle. Two popular ensemble methods, bagging and boosting, and their application in the domain of class imbalance will be discussed next.

1) Bagging: Bagging trains multiple base learners in parallel. The main target of bagging is to reduce variance of some high variance low bias base classifiers while retaining the low bias. Averaging the output of the base learners does this. For bagging to work, the base learners should be de-correlated as much as possible. This de-correlation is achieved

by training each base learner with a bootstrapped sub-sampled of the original training data. However, bagging itself cannot tackle imbalanced classification problems. In order to make bagging compatible in this domain, sampling methods such as under-sampling and over-sampling has been applied in each iteration before training the base learners. Some bagging based methods for dealing with the problem of class imbalance include Asymmetric Bagging [25], Roughly Balanced Bagging [26], and Bagging Ensemble Variation (BEV) [27].

2) Boosting: Boosting [28] combines multiple classifiers like bagging but trains the models in a sequential manner. Each base learner assigns weights to the instances, which is used by the upcoming base learners. Weights of the misclassified instances are increased while weights of the correctly classified instances are decreased. As a result, each base learner is trained with a different distribution of the training data that makes him or her diverse. The base learners are also assigned weights according to their performance on the training data. Boosting itself cannot deal with imbalanced datasets and so different sampling methods are applied at each iteration of boosting. This makes the training partition balanced [29], [30].

IV. Proposed Methods

In this section, we present the proposed hybrid methods: (a) ADASYNBagging in sub-section IV-A, and (b) RSYNBagging in sub-section IV-B.

A. ADASYNBagging

SMOTEBagging inspires ADASYNBagging. SMOTEBagging incorporates SMOTE over-sampling method in every bagging iteration before training the base model. For each of the minority class instances, SMOTE generate the same number of synthetic instances. On the other hand, ADASYN sorts the minority instances according to how hard they are to classify and generates comparatively more synthetic samples for harder minority ones. ADASYNBagging uses ADASYN for generating synthetic instances thus making the training data partition balanced. This balanced data partition is used for training the base learners. Since ADASYN has been shown to outperform SMOTE [18], we wanted to explore its performance when combined with bagging in place of SMOTE. Algorithm 1 presents the proposed ADASYNBagging approach.

Algorithm 1 ADASYNBagging Algorithm

Input: Training data, $D_{Imbalance}$, number of iterations, k, & C4.5 as base classifier.
Output: Ensemble model, M^*
Method:

1: divide $D_{Imbalance}$ into $D_{Majority}$ and $D_{Minority}$;
2: **for** i = 1 to k **do**
3: create bootstrap sample $D_{sampled_Majority}$ from $D_{Majority}$, while $D_{Minority}$ remains untouched;
4: create $D_{over-sampled-minority}$ from $D_{Minority}$ using ADASYN;
5: marge $D_{over-sampled-minority}$ with $D_{sampled_Majority}$ to get $D_{balanced-over-sampled}$;
6: build a classifier, M_i from $D_{balanced-over-sampled}$;
7: **end for**

To use M^* to classify a new instance, x_{New}:
Each $M_i \in M^*$ classify x_{New} and return the majority vote;

B. RSYNBagging

RSYNBagging uses random under-sampling as a pre-processing step for one-half of the base learners and ADASYN over-sampling for the other half. For example, if 200 base learners are used, 100 of them will be trained using over-sampled version of the training data partition and 100 will be trained using under-sampled version of it. Over-sampling and under-sampling both have their own advantages and disadvantages. Over-sampling may generate duplicate or almost identical instances, which will increase the probability of overfitting and under-sampling may result in huge amount of information loss. RSYNBagging reduces the individual downsides of both of these methods while benefiting from their combined strength. From Figure 1, it can be seen that oversampling is applied to minority class instances only on odd bagging iterations. As a result, the time complexity of RSYNBagging is less compared to that of OverBagging methods such as SMOTEBagging and proposed ADASYNBagging. Algorithm 2 presents the proposed RSYNBagging approach.

Algorithm 2 RSYNBagging Algorithm

Input: Training data, $D_{Imbalance}$, number of iterations, k, & C4.5 as base classifier.
Output: Ensemble model, M^*
Method:

1: divide $D_{Imbalance}$ into $D_{Majority}$ and $D_{Minority}$;
2: **for** i = 1 to k **do**
3: create bootstrap sample $D_{sampled_Majority}$ from $D_{Majority}$, while $D_{Minority}$ remains untouched;
4: **if** i is odd **then**
5: create $D_{under-sampled-majority}$ from $D_{sampled_Majority}$ using Random Under Sampling;
6: marge $D_{under-sampled-majority}$ with $D_{Minority}$ to get $D_{balanced-under-sampled}$;
7: build a classifier, M_i from $D_{balanced-under-sampled}$;
8: **else**
9: create $D_{over-sampled-minority}$ from $D_{Minority}$ using ADASYN;
10: marge $D_{over-sampled-minority}$ with $D_{Majority}$ to get $D_{balanced-over-sampled}$;
11: build a classifier, M_i from $D_{balanced-over-sampled}$;
12: **end if**
13: **end for**

To use M^* to classify a new instance, x_{New}:
Each $M_i \in M^*$ classify x_{New} and return the majority vote;

V. Experimental Results

We have used 11 benchmark imbalanced datasets from KEEL Dataset Repository [31] for comparing our proposed methods against UnderBagging and SMOTEBagging methods. The number of attributes, instances and imbalance ratio of each datasets are shown in Table I. For all the experiments, we have chosen C4.5 models and scikit learn's [32] implementation has been used.

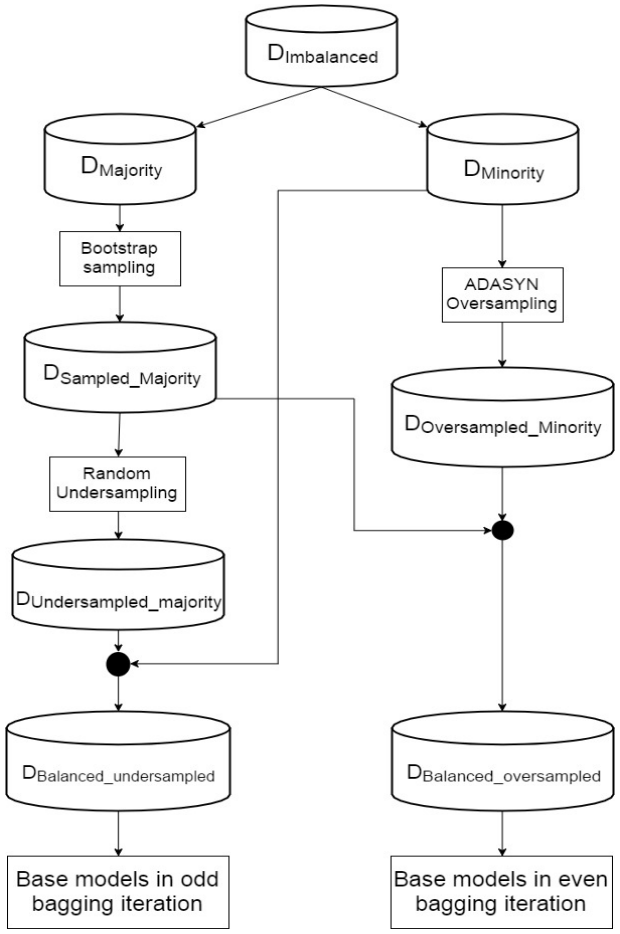

Fig. 1: Process of RSYNBagging method for classifying imbalanced data.

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 and False Positive Rate (fpr) as underlying metrics .

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

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

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

1) AUROC: ROC represents False Positive Rate (fpr) down the horizontal axis and True Positive Rate (tpr) down the vertical axis. A perfect classifier will have 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. AUC 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.

TABLE I: Datasets Description

Dataset	Attributes	Instances	Imbalance Ratio
pageblocks-13vs4	10	472	15.86
ecoli-0-3-4-7_vs_5-6	7	257	9.28
poker-9_vs-_7	10	244	29.5
yeast-0-2-5-7-9_vs_3-6-8	8	1004	9.14
ecoli-0-3-4-_vs_5	7	200	9
spambase	57	4597	1.53
new-thyroid1	5	215	5.14
glass-0-1-5-_vs_2	9	172	9.12
vehicle2	18	846	2.88
pima	8	768	1.87
abalone19	8	4174	129.44

2) AUPR: 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 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 negatives [33].

B. Results

Each algorithm has been run 6 times with 10 fold cross validation. The average auroc results are shown in TABLE II. Average AUPR results are shown in TABLE III .

With respect to AUROC, our proposed method RSYNBagging is the best performer in 7 of the datasets. On the other hand, the proposed method ADASYNBagging shows the best performance in 1 of the datasets. Similarly, when it comes to AUPR our proposed method RSYNBagging is the best performer in 5 of the datasets. In comparison, ADASYNBagging shows the best performance in 3 of the datasets. Since ADASYN itself is a modification of SMOTE, theoretically the performance of ADASYNBagging should be

TABLE II: Average AUROC Comparison

Dataset	Under Bagging	SMOTE Bagging	ADASYN Bagging	RSYN Bagging
pageblocks-13vs4	0.989	0.973	0.984	**0.994**
ecoli-0-3-4-7_vs_5-6	0.887	0.845	0.886	**0.914**
poker-9_vs-_7	**0.621**	0.547	0.617	0.597
yeast-0-2-5-7-9_vs_3-6-8	0.885	0.882	0.890	**0.895**
ecoli-0-3-4-_vs_5	0.893	0.881	0.912	**0.934**
spambase	0.915	0.911	0.912	**0.924**
new-thyroid1	0.977	**0.985**	0.983	0.982
glass-0-1-5-_vs_2	**0.691**	0.639	0.681	0.652
vehicle2	0.952	0.951	0.957	**0.962**
pima	0.659	0.661	0.656	**0.671**
abalone19	0.776	0.787	**0.798**	0.784

TABLE III: Average AUPR Comparison

Dataset	Under Bagging	SMOTE Bagging	ADASYN Bagging	RSYN Bagging
pageblocks-13vs4	0.875	0.946	0.942	**0.953**
ecoli-0-3-4-7_vs_5-6	0.703	0.815	**0.842**	0.796
poker-9_vs-_7	0.345	**0.3624**	0.339	0.262
yeast-0-2-5-7-9_vs_3-6-8	0.716	0.802	**0.806**	0.759
ecoli-0-3-4-_vs_5	0.784	0.827	0.820	**0.839**
spambase	0.912	**0.921**	0.908	**0.921**
new-thyroid1	0.921	**0.976**	0.961	0.955
glass-0-1-5-_vs_2	**0.480**	0.398	0.418	0.391
vehicle2	0.913	0.902	0.919	**0.929**
pima	0.667	0.652	0.639	**0.669**
abalone19	0.821	0.833	**0.839**	0.825

promising in cases where SMOTEBagging performs well, i.e where oversampling is the best option. This is supported by our experimental results. In some cases, ADASYNBagging outperforms SMOTEBagging which has been considered as the best performing OverBagging method to date as mentioned in the survey [16]. So, when it comes to the analysis of OverBagging techniques, ADASYNBagging should also be considered.

Between under-sampling with bagging and over-sampling with bagging based hybrid approaches, a clear winner cannot be determined. Hence a combination of under-sampling and over-sampling should be able to benefit from their alliance. Our proposed RSYNBagging does the exact same thing and outperforms both of its component sampling methods most of the times. When it comes to bagging, if the base learners are accurate yet diverse, the performance is usually better. Since half of the base learners in RSYNBagging are fed with under-sampled versions of the training set while others with synthetically over-sampled versions, different learners learn from different balanced representations which makes them more diverse compared to those in UnderBagging and OverBagging. This makes the final prediction more accurate which can be seen from both AUROC and AUPR results.

VI. Conclusion

When the dataset is imbalanced, existing classification algorithms generally focus on majority class instances while ignoring the minority class instances and thus misclassifying most of the minority class. This happens due to the way they are designed and these characteristic forces researchers to come up with some classification models that will treat majority and minority class instances equally and correctly classify both. This is because the minority class is of equal importance if not more. Throughout the last decade, several solutions have been proposed to deal with this problem of which bagging or boosting and sampling based methods have been most successful. This proves the effectiveness of the alliance between ensemble and sampling techniques. In this paper, we have come up with two novel bagging and re-sampling based approaches named ADASYNBagging and RSYNBagging in order to tackle the problem of class imbalance. We have compared our methods with two of the best performing bagging based methods: UnderBagging and SMOTEBagging. Through experiments, we have demonstrated that the performance of our proposed methods is promising when compared with that of UnderBagging and SMOTEBagging on imbalanced datasets.

UnderBagging uses only under-sampled versions of the original dataset while OverBagging only uses over-sampled versions. Between under-sampling and over-sampling, which one will be more appropriate depends on the problem. Hence none of them show stable performance throughout different datasets. However, our proposed method RSYNBagging uses both over-sampled and under-sampled versions of the dataset in separate iterations and thus is suitable for a wide variety of datasets. It's comparatively much more stable performance throughout different datasets shown in TABLE III and II proves this fact. Additionally, it's time complexity is much less compared to SMOTEBagging cause over-sampling is used in only half of the iterations thus making RSYNBagging suitable for practical purposes. Another proposed method ADASYNBagging has appeared to be a promising over-sampling based bagging method and can be a suitable alternative to SMOTEBagging. In our methods we have kept the sampling rate fixed(always majority:minority ratio 1:1) so that no extra parameter tuning is required here. In this paper, combining two different sampling approaches together in RSYNBagging and applying a popular sampling method ADASYN individually in ADASYNBagging has shown positive effect. So, in future we would like to combine other sampling approaches together with ensemble methods.

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] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, "Random balance: ensembles of variable priors classifiers for imbalanced data," *Knowledge-Based Systems*, vol. 85, pp. 96–111, 2015.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[6] N. Chawla, N. Japkowicz, and A. Kolcz, "Special issue on learning from imbalanced datasets, sigkdd explorations," in *ACM SIGKDD*, 2004.

[7] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 5, no. 04, pp. 597–604, 2006.

[8] J. R. Quinlan, "Improved estimates for the accuracy of small disjuncts," *Machine Learning*, vol. 6, no. 1, pp. 93–98, 1991.

[9] G. Wu and E. Y. Chang, "Kba: Kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 786–795, 2005.

[10] 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.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[12] 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.

[13] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63–77, 2006.

[14] B. Krawczyk and G. Schaefer, "An improved ensemble approach for imbalanced classification problems," in *Applied Computational Intelligence and Informatics (SACI), 2013 IEEE 8th International Symposium on*. IEEE, 2013, pp. 423–426.

[15] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[16] 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.

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

[18] 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.

[19] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[20] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Analysis & Applications*, vol. 6, no. 3, pp. 245–256, 2003.

[21] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*. IEEE, 2009, pp. 324–331.

[22] Y. Lu, Y.-m. Cheung, and Y. Y. Tang, "Hybrid sampling with bagging for class imbalance learning," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2016, pp. 14–26.

[23] S. Hu, Y. Liang, L. Ma, and Y. He, "Msmote: improving classification performance when training data is imbalanced," in *Computer Science and Engineering, 2009. WCSE'09. Second International Workshop on*, vol. 2. IEEE, 2009, pp. 13–17.

[24] D. M. Farid, A. Nowé, and B. Manderick, "A new data balancing method for classifying multi-class imbalanced genomic data," *25th Belgian-Dutch Conference on Machine Learning (Benelearn)*, pp. 1–2, 12-13 September 2016.

[25] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 7, pp. 1088–1099, 2006.

[26] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," *Statistical Analysis and Data Mining*, vol. 2, no. 5-6, pp. 412–426, 2009.

[27] C. Li, "Classifying imbalanced data using a bagging ensemble variation (bev)," in *Proceedings of the 45th annual southeast regional conference*. ACM, 2007, pp. 203–208.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[33] 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.