Empirical Assessment of Ensemble based Approaches to Classify Imbalanced Data in Binary Classification

Prabhjot Kaur[1], Anjana Gosain[2]
Department of Information Technology, MSIT Research Scholar[1]
USICT, Guru Gobind Singh Indraprastha University, New Delhi, INDIA[1, 2]

***Abstract*—Classifying imbalanced data with traditional classifiers is a huge challenge now-a-days. Imbalance data is a situation wherein the ratio of data within classes is not same. Many real life situations deal with such problems e.g. Web spam detection, Credit card frauds, and fraudulent telephone calls. The problem exists everywhere when our objective is to identify exceptional cases. The problem is handled by researchers either by modifying the existing classifications methods or by developing new methods. This paper review ensemble based approaches (Boosting and Bagging based) designed to address imbalance in classes by focusing on binary classification. We compared 6 Boosting based, 7 Bagging based and 2 hybrid ensembles for their performance in imbalance domain. We use KEEL tool to evaluate the performance of these methods by implementing the methods on seven imbalance data having class imbalance ratio from 1.82 to as high as 129.44. Area Under the curve (AUC) parameter is recorded as the performance metric. We also statistically analyzed the methods using Friedman rank test and Wilcoxon Matched Pair signed rank test to strengthen the visual interpretations. After analysis, it is proved that RusBoost ensemble outperformed every other ensemble in the imbalanced data situations.**

Keywords—Ensemble approaches; boosting; bagging; hybrid ensembles; imbalanced data-sets; classification

I. INTRODUCTION

Classification process is very important in solving many real time problems. Various types of classifiers have been proposed in research field to solve classification problems. These classifiers only gives satisfactory results where the real time problems are represented by balanced data-set (the proportion of size of data classes is same). But sometimes, there are circumstances wherein we want to do the classification when the data-set is not balanced (proportion of size of data classes is not same) e.g. Web Spam Detection, Credit Card Frauds, Fraudulent Telephone calls etc. In such cases, if we apply the classification methods which are designed to classify balanced data sets, we will not get the accurate results. The major problem with imbalanced data set is that the data points belong to majority class (bigger class) impacts the classifier decision boundaries at the cost of minority class (smaller class) which is represented by very few points compare to majority class. This concern is known with the name as class imbalance problem in the research community. The extent of imbalance in data can be measured with class imbalance ratio (CIR). CIR is the percentage of size of majority class to the size of minority class. CIR value is indirectly related to the size of minority class. CIR with high value is considered as highly imbalanced data. Various types of solutions are developed by research community to handle this problem. Methods developed to resolve this issue can be divided into three major categories. Data level, algorithm level and the combination of data and algorithm level (hybrid) approaches. In data level approaches, data is pre-processed for balancing the dataset before classification. The biggest benefit of this category is that one can use the existing classification methods which are developed to classify balanced data-sets. Researchers have applied different logics for balancing the data. Some methods balance the data by synthetically generating the data-points within minority class either randomly copying the existing data or by applying some intelligent process to generate synthetic data [1-11]. These types of methods come under the category of oversampling methods. The limitation with random oversampling by copying the existing data may lead to overfitting. In case of the noisy data-sets, random oversampling may lead to the increase of noise within the data-set [12, 13].Some methods balance the data by removing data points from majority class either randomly or by using some intelligent concept before classification [13-19]. The biggest limitation with random undersampling is the loss of some important information. These type of methods are called undersampling methods. There is another sub-type of data-level methods wherein we combine the concept of oversampling and undersampling to balance the dataset before classification. These types of methods are known as hybrid data level methods [20, 21]. In algorithm level category, the researchers either modified the existing classification methods by working on the biasness of classifier towards the bigger class or by developing new methods to handle imbalanced data [22-38]. The third category, known as hybrid methods, combines data-level and algorithm level methods to boost the classifiers performance for imbalance data [39-53]. Many researchers combine data-level methods and algorithm level methods using ensemble concept to enhance the performance of earlier classification methods which were using only single classifier for getting results. Ensemble concept uses multiple classifiers for better predictive results compare to the methods which uses only single classifier to obtain the results. In this paper, we review ensemble based classification techniques which uses Bagging and Boosting concept to handle imbalance data-sets. We empirically assessed the methods using KEEL tool [54, 55] and statistically analyzed the results using Friedman [56] and Wilcoxon Matched pair signed rank [57] tests. Section II explains the idea of ensembles and review Boosting based and Bagging based ensembles designed to resolve class imbalance issue. It also describes the performance criteria used in this paper to assess the performance of methods. Empirical calculation of ensembles approaches and their statistical analysis is discussed in Section III followed by conclusions in Section IV.

II. Review of Ensemble Approaches

A. *Fundamentals of Ensemble Approach*

Ensemble approaches train more than one classifiers to resolve the same issue. This method is also named committee-based learning or learning through more than one classifier systems. Fig. 1 describes the model of ensemble approach. The area of ensemble approaches actually generated from three sections i.e. combining more than one classifiers, ensembles of weak classifiers and combination of experts [58]. Combining classifiers concept was mostly studied under pattern recognition area wherein the researchers works on strong classifiers and try to design powerful combining rules to get stronger combined classifiers. Ensemble of weak classifier is mostly studied by machine learning community wherein the researchers work upon weak classifiers to design powerful procedures for boosting the performance from weak to strong. This area has designed vary famous ensemble methods like AdaBoost [59] and Bagging etc. Combination of experts is studied by neural network community wherein the researchers usually consider a divide-and-conquer scheme to learn a combination of parametric prototypes jointly.

Ensemble methods are popular learning paradigm [58] since 1990’s. It is because of two main pioneering work proposed in literature. One, which has empirically proved [60], analyzed that the outcomes resulted from a set of learners are found more precise than the results given by a single finest classifier as displayed in Fig. 2. The other, theory concept proven by Schapire [61] is that the weak base learners can be enhanced to strong learners. As strong classifiers are needed to solve many real time problems which are not possible to solve using weak classifiers, this need has motivated the researchers to generate strong classifiers by using ensemble methods. Ensemble methods use multiple classification procedures to attain better predictive results. Under this approach, various classifiers are trained either parallel or sequentially to resolve the same problem. An ensemble is created using two steps, by selecting the base classifier and then joining them to make ensemble of classifiers. Performance of ensemble methods can be decided by two factors: Accuracy of the individual learner and diversity among all classifiers. Ensemble’s accuracy is directly related to the selection of base classifier. It is widely accepted [62] that improvement in the overall predictive accuracy by the ensemble can occur only if there is diversity among its components i.e. if individual classifiers don’t always agree. Diversity is the measure to which a classifier can make different decisions on a single problem. Various ways can be used to measure diversity like by manipulating training patterns (cross-validation, bagging, boosting), by manipulating input features (by considering subset of features for classifier learning) and by incorporating random noise. Major research in literature belongs to homogeneous ensembles than heterogeneous ensembles wherein we use combinations of different classifiers to produce results. But heterogeneous ensembles can produce more diversified results than homogeneous ensembles [66]. Computational complexity is very high in case of generating a single classifier than the ensemble. Because, while generating single classifier, for better performance it is essential to design various versions and tuning parameters for better model selection, whereas, the computational complexity in combining different classifiers is very less. Ensemble approaches reviewed in the paper are shown in Fig. 3.

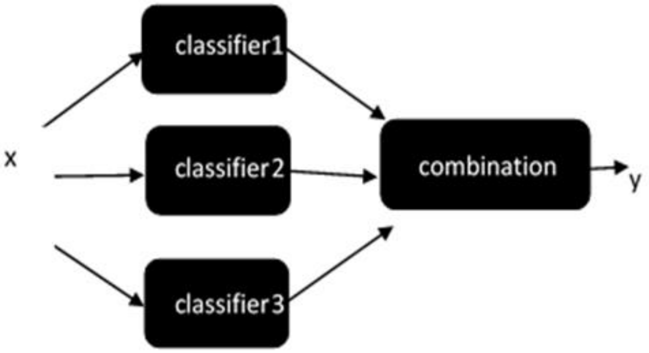


Fig. 1. A Common Ensemble Architecture [58].

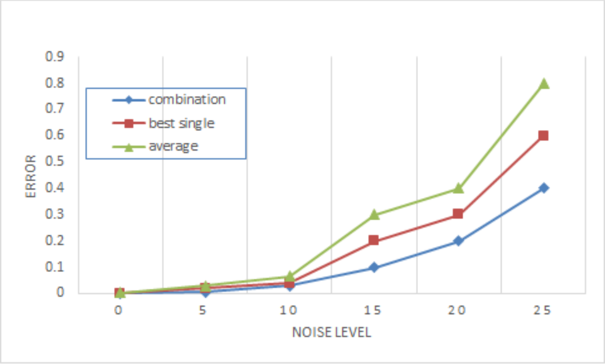


Fig. 2. Salamon and Hansen’s Observation [58].

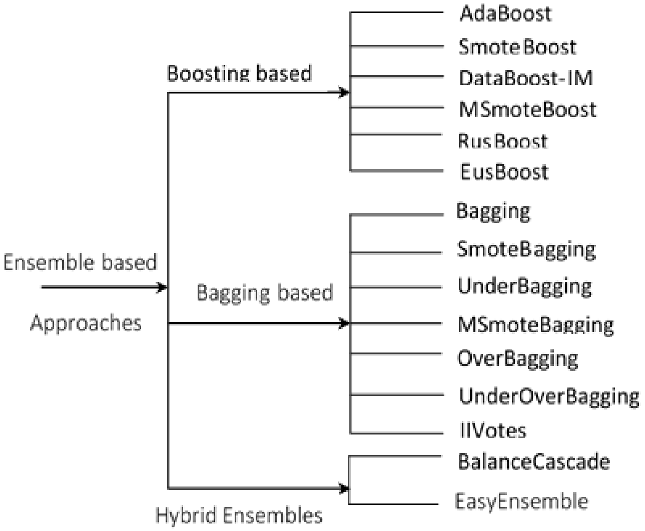


Fig. 3. Ensemble Approaches under Study.

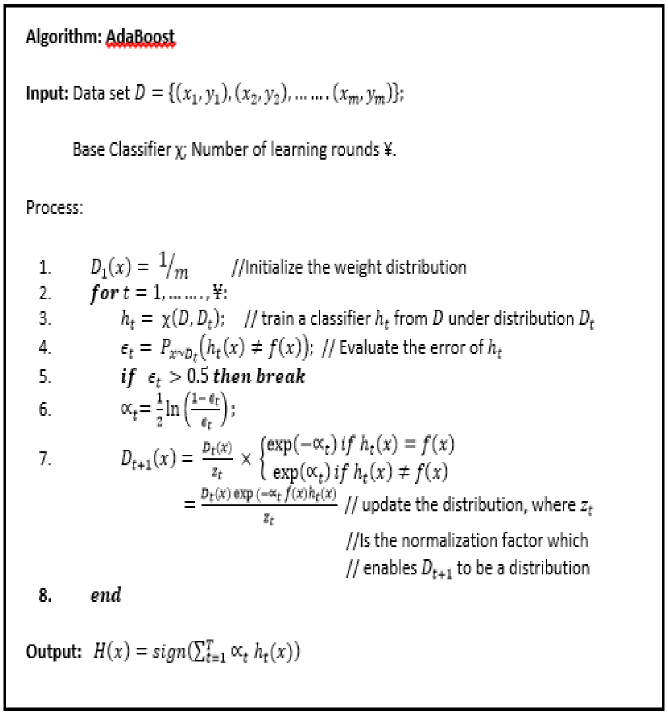
B. Ensembles based upon Boosting Concept

Ensembles are categorized into two models namely Boosting and Bagging, based on the methodology of joining base classifiers. Boosting method converts the weak classifier to strong classifier by sequentially generating the base classifier hence it goes in the category of sequential ensemble paradigm [58]. In a boosting process, initially a model is build using initial training data, then another model is created whose purpose is to correct the errors from the model generated from previous model. This process is repeated until the perfect prediction is done or a maximum number of models are generated. Various ensembles, based upon boosting concept, reviewed during current study are described as below:

1) Adaptive boosting method (AdaBoost): AdaBoost, [59] the first successful algorithm proposed by Freund and Schapire in 1996 using boosting concept for binary classification. In AdaBoost, we used complete data-set for training every classifier serially. After every iteration, the method assigns more weight to the misclassified data points, with the objective of accurately classifying the misclassified data points recognized during current iteration, in the next iteration. Hence, its main objective is to emphasize on the data points whose classification is predicted as hard. Weight allocated to the misclassified data points after every iteration is directly related to the status of misclassified data i.e. How hard it is to classify that data point. Weight is initially equally assigned to all the data. After every iteration, the weights allocated to misclassified data points are increased and allocated to correctly classified data points are decreased. Lastly, when an unknown data point is submitted, every classifier vote for it and the data point is finally allocated to the class based upon the majority votes. It is named adaptive as it is build using multiple repetitions for creating a strong classifier. The drawback of AdaBoost is that it allocates equal weights to the classes and is internally developed to detect equal size of classes (for balanced data-sets). In imbalanced scenarios, its results are always in the favor of majority class. Therefore, to handle this biasness towards majority class, many researchers updated equal weight situation of Adaboost method so as to modify the method to detect minority class accurately. Fig. 4 shows the procedure of AdaBoost.

2) Smoteboost: N. V. Chawla in 2002 proposed SmoteBoost [3] by modifying AdaBoost to address imbalance problem in classes. SmoteBoost combines an oversampling method SMOTE with standard boosting process. It generates synthetic data inside minority class using SMOTE process during every iteration of AdaBoost. The weights assigned to synthetically generated data remains constant and depend on the aggregate sum of information in the new data-set, whereas the weights assigned to the original data points are normalized so as to form a distribution with the new generated data points. When the classifier is trained, the weights assigned to the original data points are updated. Again the synthetic data is generated in another phase and weights are modified so as to match the weight distribution. This process repeats itself till we get the required predictions or extreme number of classifiers are build. Limitation with the method is that it uses oversampling method to balance the data by generating synthetic data points therefore it is more computationally expensive compare to the methods that are based on undersampling approaches. Another limitation of SmoteBoost is that in case of noisy data-sets it may end up by increasing noisy data-points by random selection of the noise points as a candidate to produce synthetic data-points [12, 13].

3) Databoost-IM: Guo and Victor in 2004 proposed another boosting based method, namely DataBoost-IM [49], by combining boosting with data-generation to improve the predictive capabilities of classifiers for binary imbalance data-sets. Its working principle is unlike SmoteBoost as it, firstly, identify and separate data points which are hard to predict, from both minority as well as majority class, to produce synthetic data-points. It also considers bias information towards hard to predict data points to produce synthetic data on which the classifier from next iteration needs to focus. In this process, the weights assigned to both the classes in the new training set are re-balanced so that boosting procedure can focus on both the classes. Hence, this method focused on refining the prediction ability of both the majority and minority class. The principle drawback [63] of this procedure is that it can’t manage very high imbalanced situations in light of the fact that it creates an extensive amount of data points which are troublesome toward oversee by the base classifier.



```
Algorithm: AdaBoost

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_m, y_m)\}$;

    Base Classifier $\chi$; Number of learning rounds ¥.

Process:

    $D_1(x) = 1/m$        //Initialize the weight distribution
    for $t = 1, \ldots\ldots$, ¥:
        $h_t = \chi(D, D_t)$;    // train a classifier $h_t$ from $D$ under distribution $D_t$
        $\epsilon_t = P_{x\sim D_t}(h_t(x) \neq f(x))$; // Evaluate the error of $h_t$
        if $\epsilon_t > 0.5$ then break
        $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
        $D_{t+1}(x) = \frac{D_t(x)}{z_t} \times \begin{cases} \exp(-\alpha_t) \text{ if } h_t(x) = f(x) \\ \exp(\alpha_t) \text{ if } h_t(x) \neq f(x) \end{cases}$
                  $= \frac{D_t(x)\exp(-\alpha_t f(x) h_t(x))}{z_t}$ // update the distribution, where $z_t$
                                              //Is the normalization factor which
                                              // enables $D_{t+1}$ to be a distribution
    end

Output: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
```

Fig. 4. AdaBoost Algorithm.

4) Modified smoteboost (MSmoteBoost): To handle the noise sensitivity of SmoteBoost in 2009, Ma and He gave an intelligent boosting approach, MSmoteBoost [41], which incorporates MSmote data level method in every iteration of AdaBoost. Unlike Smote, MSmoteBoost remove noise data-points and consider the distribution of minority class. Minority class data is divided into three groups as border, security and latent noise points. The data points are categorized based on the distance from other data points, before generating synthetic points. Security data points are those which can strengthen the performance and noise points can reduce the performance of classifier. Hard to predict data are recognized as border category. The method processes the data differently with these categories while producing synthetic data points. The weights assigned to the new data points are based on the total number of points in the new data-set. Hence, their weights always remain constant, whereas original data-set's data point's weights are normalized so that they form a distribution with the new generated data points. The assigned weights of the original data points are updated after training the classifier.

The process repeats itself till the strong classifier is build. As this classifier is also using oversampling approach, Its computational cost is also high compare to the ensembles based on undersampling.

5) Random undersampling boosting (RusBoost): In 2010, another boosting based ensemble is proposed. It is dissimilar from SmoteBoost because it incorporates undersampling data level method (Rus) in every iteration of AdaBoost with the motive of proposing a simple classifier which can work with fast speed than using any oversampling approach. RusBoost [39] removes data points randomly from majority class in every iteration of AdaBoost. RusBoost doesn't allocate new weights to the data points. It is sufficient to normalize the weights of the remaining data points in the new data-set according to the total sum of weights. After the classifier is trained, the process updates the weights of the original data-set. The process is repeated till we get the strong classifier. The inspiration of combining Random undersampling and boosting method is its simplicity, performance and speed. As the data set is balanced by removing data therefore time needed to build a model is low compare to oversampling models. Loss of required information is the major limitation because no intelligent method is used to eliminate data from the majority class. Another disadvantage is during noisy environments, it may end up removing good data from classes due to which there is more impact of noise on the classifier's performance [12, 13].

6) Evolutionary undersampling boosting (EusBoost): In 2013, an intelligent undersampling based ensemble, EusBoost, is proposed which incorporates EUS [40] preprocessing method in every iteration of AdaBoost. The basis of EusBoost [40] is RusBoost, which is simplest method compare to other oversampling approaches. EuaBoost enhances the classifier performance by the using the evolutionary undersampling approach. The key principle of EusBoost is diversity mechanism by considering different subset of data in every iteration.

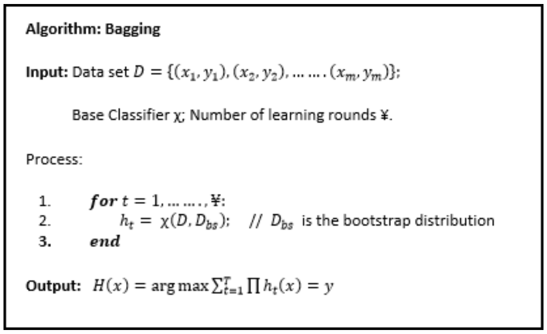
C. *Ensembles based upon Bagging (Bootstrap Aggregation) Concept*

Bagging, like Boosting, also build a strong classifier by combining multiple weak classifiers for the better performance compare to using single classifier. Bagging [64] gives the best results if the problem using single classifier is overfitting. Unlike Boosting, any data point in bagging has the same probability to appear in a new data-set. The process of bagging starts by creating sub-sets from the data-set. Then each sub-set of data-set is trained independently using classifier that results in ensemble of different models. Then average of all these different models are used to build a strong classifier. It brings diversity by using different data-sets for every classifier. Hence, bagging comes under the category of parallel ensemble methods. The inspiration behind these ensemble methods is to exploit the independence between the weak classifiers [58]. Fig. 5 shows the bagging algorithm.

1) Smotebagging: SmoteBagging [65] combines oversampling of minority class using Smote with bagging. In this method both the classes participate in creating each bag. A Smote oversamples the data with a% rate during every iteration and increased the rate with the multiple of 10 with every next iteration. This proportion characterizes the measure of positive data points which are arbitrarily resampled from the first data-set amid each iteration. The remaining positive data is generated by Smote algorithm till the data is balanced. Bootstrapping negative data points are created to make the ensemble more diverse.

2) Underbagging: This approach [65] does undersampling after creating subset of data from original data-set. Therefore, in place of removing data from the whole data-set, it does it before training each classifier. Undersampling is done by using nearest neighbor principle for balancing the data before training the classifier.

3) MSmotebagging: MSmoteBagging [65, 67] is the variation of SmoteBagging wherein minority class is oversampled using MSmote data level method. Oversample minority class data points using MSMOTE preprocessing algorithm.



```
Algorithm: Bagging

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \dots \dots .(x_m, y_m)\}$;

        Base Classifier χ; Number of learning rounds ¥.

Process:

        for $t = 1, \dots \dots, ¥$:
            $h_t = \chi(D, D_{bs})$;    // $D_{bs}$ is the bootstrap distribution
        end

Output: $H(x) = \arg\max \sum_{t=1}^{T} \prod h_t(x) = y$
```

Fig. 5. Bagging Algorithm.

4) Overbagging: In this method [65], data-set is balanced when the bags are randomly picked from the original data-set. Therefore, in place of removing the data randomly from whole data-set, the data is generated randomly within minority class of sub-set before every classifier is trained. This method includes all the majority class data points in the new bootstrap.

5) Underoverbagging: UnderBagging to OverBagging (UnderOverBagging) [65] method uses the combination of oversampling and undersampling process. It considers the resampling rate 'a%' in every iteration which is ranged from 10% to 100%. Resampling rate is the multiple of 10. Therefore, the number of data points trained by every classifier in the subsequent iterations will be different. This method introduce diversity is the process.

6) Imbalanced ivotes (IIVotes): IIVotes is the combination of SPIDER [68] and IVotes [69]. SPIDER is the preprocessing method. .IVotes is a variation of Bagging where the sampling is done according to the importance of each data point. Although SPIDER method improves the sensitivity of minority class but decrease the specificity at the same time. IIVotes modified SPIDER method by incorporating IVotes for improving the trade-off between specificity and senstivity. The main purpose of this method is to acquire a balance between the specificity and sensitivity for the minority class in contrast to a single classifier combined with SPIDER.

D. Ensembles based upon Hybrid Ensemble Concept (Bagging and Boosting)

Hybrid Ensemble based methods are the combination of bagging, boosting and pre-processing methods. Liu, Wu, and Zhou in 2009 proposed EasyEnsemble [53] and BalanceCascade [53] and named these methods as exploratory undersampling methods. These methods follow different approaches to tackle negative data points after every iteration. These methods used bagging as the key concept in building ensemble and used AdaBoost technique in place of the weak classifier. In BalanceCascade the classifiers are trained sequentially because it works in a supervised manner. During bagging iteration after the AdaBoost classifier is trained, the correctly classified majority data are removed from the data-set and is not processed in the next iterations. As EasyEnsemble approach does not execute any operation on the data from the original data-set after every AdaBoost iteration. So the classifiers are trained in parallel.

E. Performance Criteria

In case of imbalanced data-sets, the main objective is to identify the minority class so we are considering minority class as the positive class. Table I shows the confusion matrix for imbalance data-sets.

TABLE I. CONFUSION MATRIX

	Positive (Minority)	**Negative (Majority)**
True	True Positive	True Negative
False	False Positive	False Negative

We are using Area under the ROC Curve (AUC) [70, 71] as the performance metric to assess the methods. AUC is a standout amongst the most famous execution metric used to assess the execution of classifiers intended for imbalanced data sets. It is a curve in which false-positive rate and true positive rate are plotted on x-axis and y-axis respectively. AUC is the finest tool for comparing different classifiers. A classifier's performance is directly proportional to its location towards the upper left corner. AUC portrays ROC quantitatively. It is calculated as the arithmetic mean of True Positive rate and True Negative rate.

$$AUC = \frac{TP_{Rate}+TN_{Rate}}{2} \qquad (1)$$

Where TP_{Rate} is characterized as the quantity of positive data points that are accurately categorised as positive and TN_{Rate} is the total quantity of negative data points that are accurately categorised as negative. AUC reveals the global performance of every classifier for all conceivable estimation of False Positive rate.

III. EMPIRICAL ASSESSMENT OF ENSEMBLES

We have compared 15 ensemble approaches with 7 imbalanced data with the class imbalance ratio from 1.82 to 129.44. The characteristics of these data-sets are recorded in Appendix A. We used KEEL tool [54, 55] for comparing the performance of ensemble approaches by considering Decision Tree method (C4.5) as the weak classifier. C4.5 is the widely used classifier by many people to compare the algorithms in imbalance domains [72, 73]. The AUC of the methods is recorded with the following initial settings of the KEEL tool (Table II). Tables III and IV listed AUC values along with the variance. Results are visually displayed in Fig. 6, Fig. 7 and Fig. 8. Average performance of all the ensembles is shown in Fig. 9.

A. Visual Interpretations and Discussions

It is witnessed from the figures that for Boosting based approaches (Fig. 6), RusBoost stands out and outperformed every other method for extremely imbalanced data (Abalone19 having imbalance ratio of 129.44). In other cases, SmoteBoost and RusBoost almost performed equally. In case of Bagging based approaches (Fig. 7), Underbagging outperformed other methods for highly imbalanced data-set whereas the performance of SmoteBagging and UnderBagging is almost equal for other data-sets. Hybrid ensembles performed equally well for all the data-sets with minor differences for some data-sets. In case of Ecoli4, Balancecascade outperformed EasyEnsemble whereas in case of Abalone19, EasyEnsemble outperformed Balancecascade.

TABLE II. PARAMETER SETTING OF KEEL TOOL

Parameter Description	**Value**
Base Classifier	Decision Tree (C4.5)
Cross Validation	5 Fold
Data points per leaf	2
Confidence Level	0.25
Number of Classifiers	10
Pruning	True

TABLE III. AUC VALUES OF ENSEMBLE APPROACHES

Techniques	**Data-Sets (Class Imbalance ratio :: 1.82 to 129.44)**							
	Glass1(1.82)		**Vehicle3 (2.99)**		**Yeast3 (8.10)**		**Ecoli4 (15.80)**	
	AUC	***Variance***	***AUC***	***Variance***	***AUC***	***Variance***	***AUC***	***Variance***
Adaboost	**0.8093**	±0.0020	0.6812	±0.0004	0.8351	±0.0011	0.8449	±0.0115
SmoteBoost	0.7839	±0.0034	0.7442	±0.0009	0.8917	±0.0004	0.8826	±0.0057
DataBoost-IM	**Not Performing**		0.6917	±0.0020	0.8919	±0.0009	0.8489	±0.0065
MSmoteBoost	0.7625	±0.0061	0.7386	±0.0002	0.9176	±0.0010	0.8489	±0.0059
RusBoost	0.7703	±0.0041	0.7643	±0.0001	0.9198	±0.0004	**0.9146**	±0.0015
EusBoost	0.7836	±0.0036	**0.7713**	±0.0014	0.9321	±0.0004	0.8760	±0.0096
Bagging	0.7556	±0.0010	0.6602	±0.0008	0.8529	±0.0010	0.8906	±0.0069
SmoteBagging	0.7444	±0.0050	0.7488	±0.0020	0.9350	±0.0003	0.8996	±0.0030
UnderBagging	0.7547	±0.0038	0.7410	±0.0005	**0.9354**	±0.0003	0.8598	±0.0018
MSmoteBagging	0.7219	±0.0038	0.7678	±0.0003	0.9291	±0.0003	0.8632	±0.0040
OverBagging	0.7580	±0.0034	0.7207	±0.0004	0.9073	±0.0025	0.8853	±0.0069
UnderOverBagging	0.7286	±0.0024	0.7535	±0.0004	0.9293	±0.0005	0.8566	±0.0040
IIVotes	0.6745	±0.0044	0.7330	±0.0016	0.8908	±0.0004	0.8879	±0.0018
BalanceCascade	0.7491	±0.0025	0.7282	±0.0008	0.9135	±0.0008	0.9093	±0.0028
EasyEnsemble	0.7491	±0.0025	0.7282	±0.0008	0.9135	±0.0008	0.8650	±0.0022

TABLE IV. AUC VALUES OF ENSEMBLE APPROACHES

Techniques	**Data-Sets (Class Imbalance ratio :: 1.82 to 129.44)**						
	Abalone 9-18 (16.40)		**Yeast5 (32.78)**		**Abalone19 (129.44)**		**Average Performance out of 7 data-sets**
	AUC	***Variance***	***AUC***	***Variance***	***AUC***	***Variance***	***AUC***
Adaboost	0.7327	±0.0239	0.8174	±0.0013	0.5095	±0.0006	0.7471
SmoteBoost	0.7939	±0.0238	0.9554	±0.0030	0.5291	±0.0015	0.7972
DataBoost-IM	0.7226	±0.0240	0.9071	±0.0009	0.5000	±0.0000	0.7603
MSmoteBoost	0.7290	±0.0139	0.9142	±0.0004	0.4989	±0.0000	0.7728
RusBoost	0.8105	±0.0085	0.9633	±0.0005	0.6888	±0.0060	0.8330
EusBoost	0.7957	±0.0133	0.9471	±0.0005	**Not Performing**		**0.8509**
Bagging	0.6510	±0.0086	0.8744	±0.0003	0.5000	±0.0000	0.7407
SmoteBagging	0.7961	±0.0149	0.9670	±0.0007	0.5467	±0.0008	0.8054
UnderBagging	0.7733	±0.0030	0.9592	±0.0004	0.6894	±0.0048	0.8161
MSmoteBagging	0.7303	±0.0117	0.9340	±0.0011	0.4996	±0.0000	0.7780
OverBagging	0.7377	±0.0198	0.8788	±0.0033	0.5488	±0.0008	0.7767
UnderOverBagging	0.7527	±0.0210	0.9413	±0.0020	0.5264	±0.0033	0.7841
IIVotes	0.7456	±0.0228	0.8328	±0.0031	0.4990	±0.0000	0.7520
BalanceCascade	0.7456	±0.0185	0.9552	±0.0005	0.6667	±0.0069	0.8096
EasyEnsemble	0.7456	±0.0185	0.9552	±0.0005	0.6685	±0.0066	0.8036

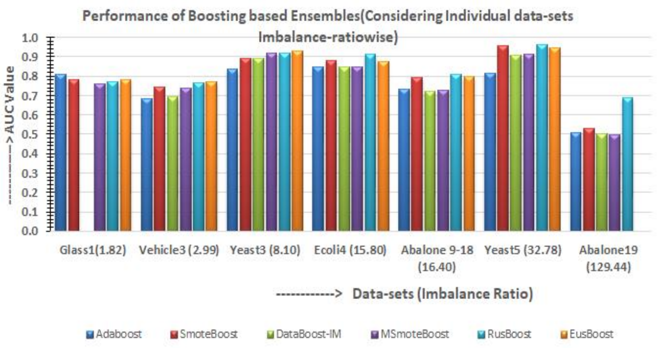


Fig. 6. AUC Results of Boosting based Ensembles.

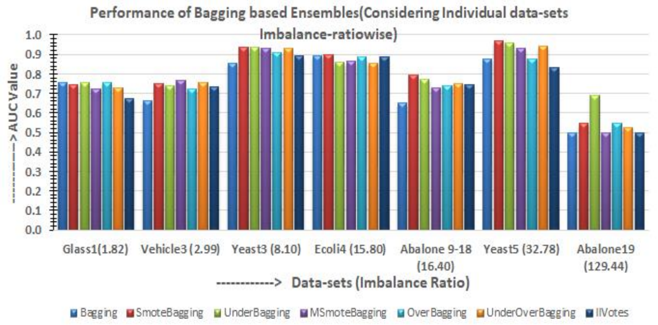


Fig. 7. AUC Results of Bagging based Ensembles.

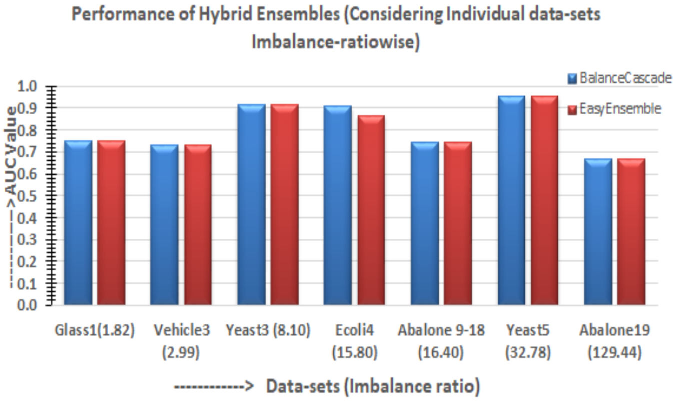


Fig. 8. AUC Results of Hybrid Ensembles.

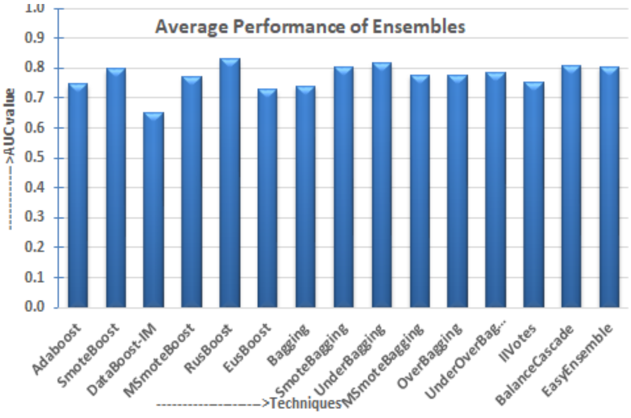


Fig. 9. Average AUC Results of all the Ensembles.

Considering the overall average performance of ensembles, it is observed that RusBoost outperformed other ensemble methods. The performance of SmoteBagging, UnderBagging, BalanceCascade and SmoteBoost performed equally well with the minor variations.

The visual interpretation about performance of these ensembles is not satisfactory and sufficient. So to prove these interpretations, we have done statistical validations.

B. *Statistical Validations*

It is very difficult to judge the performance of algorithms when their performance is tested with multiple data-sets and best performing method is not the same for every case. Statistical validation is an efficient tool when we have to compare the performance of methods with very little variation. To do better analysis we are using non-parametric tests as per the recommendation given in [72-74]. We are conducting two types of non-parametric tests. We are using Friedman rank test [75] to compare multiple methods and to know if there are any significant differences between the methods. If the 'Null hypothesis is rejected' then we are using Holm post-hoc test [75] to check if the control method (having rank 1) is significantly better than other methods (1 x N comparisons). This test computes ranks for every algorithm as per the following equation:

$$F_{AR} = \frac{(c-1)\left[\sum_{j=1}^{c}\hat{R}_j^2 - (\frac{cn^2}{4})(cn+1)^2\right]}{\{[cn(cn+1)(2cn+1)]/6\}-(1/c)\sum_{i=1}^{n}\hat{R}_i^2} \quad (2)$$

Where $'c'$ is the total number of algorithms, $\hat{R}_i$ is equal to the rank total of the ith data-set and $\hat{R}_j$ is the rank total of the jth algorithm. As per the equation the best performing algorithm will have the lowest rank. To compare two methods, we are using Wilcoxon Matched Pair signed rank test [57] to find the significant differences between two methods.

1) Statistical framework: We applied the statistical tests on the AUC performance metric as per following steps In the first step, Best performer method is selected from every group of ensembles (Boosting, Bagging and Hybrid) using Friedman test and Holm post-hoc analysis. After this step, we left with only three best methods out of all the groups. In the second step, 3 methods are assessed using Friedman test to find the final method which outperformed every other ensemble to classify imbalanced data.

2) Analysis and discussions: Firstly, we apply Friedman test on Boosting based ensembles. Fig. 10 shows the ranks assigned by Friedman test. As per the ranking, RusBoost outperformed in the family of Boosting ensembles whereas DataBoost-IM is the worst performer. Table V lists the Friedman test statistic for Boosting ensembles.

TABLE V. TEST STATISTICS USING FRIEDMAN TEST (BOOSTING ENSEMBLES)

N	07
Chi-Square (F_{AR})	16.55
Degree of Freedom (K-1)	5
p-value	0.005435

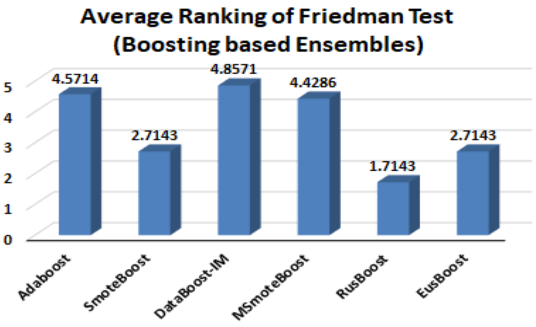


Fig. 10. Ranks Assigned by Friedman Test.

In the table, 'N' is the number of data-sets. 'k-1' is the degree of freedom (which is equal to number of algorithms minus 1). The table value of chi-square (χ2) test for '5' degree of freedom is 11.0705, which is lesser than the F_{AR} calculated value 16.55102 and the p-value is less than 0.05. Hence the null hypothesis (There is no significant difference between these groups of algorithms) is rejected. To know the difference, we did Holm post-hoc analysis by considering RusBoost as the control method (having rank 1). Holm statistics is given in Table VI. As per the statistic, the hypothesis for no significant differences is rejected for DataBoost-IM, AdaBoost, MSmoteBoost and EusBoost with the control method 'RusBoost' because the p-value is each case is less than 0.05. As the p-value of SmoteBoost is equal to 0.05, hence there are no significant differences between RusBoost and SmoteBoost. We further analyze these two algorithms using Wilcoxon Matched Pair signed rank test. The test statistics is given in Table VII. R^+ is the sum of ranks for the data-set in which the number of times first algorithm (RusBoost) outperformed other (SmoteBoost). R^- rank specify the number of times second algorithm (SmoteBoost) outperformed the other (RusBoost). It is clearly seen from the table that RusBoost performed better than SmoteBoost. So RusBoost is selected as the best performer from the Boosting based ensemble group. Friedman Test ranking for Bagging based ensembles is shown in Fig. 11 and Test statistics are shown in Table VIII. SmoteBagging outperformed other ensembles with first rank and IIVotes is the worst performer. As chi-square (χ^2) table value for 6 degree of freedom is 12.5916 which is lower than chi-square (F_{AR}) calculated value and p-value is less than 0.05, the null hypothesis is rejected. To know the difference between these ensembles, Holm post-hoc test is conducted with SmoteBagging as the control method. Table IX shows the Holm test statistics. All the methods except UnderBagging reject the null hypothesis, which means that we have to further analyze SmoteBagging and Underbagging for any significant differences. To closely analyze these two methods, we performed Wilcoxon Matched pair test. The test statistics (Table X) shows that p-value is more than 0.05 so null hypothesis for no significant differences is accepted. But the higher rank in favor of SmoteBagging proves its better performance compare to UnderBagging. Hence, SmoteBagging is selected as the best performer in the category of bagging based ensembles. As we have only two methods in hybrid ensemble category so we are performing Wilcoxon Matched pair test to analyze these methods. From the test statistics (Table XI), it is observed that the hypothesis is accepted as the p-value is more than 0.05 but the higher rank score of BalanceCascade confirms its superiority from EasyEnsemble. So, BalanceCascade is selected as the best performer from hybrid ensemble category.

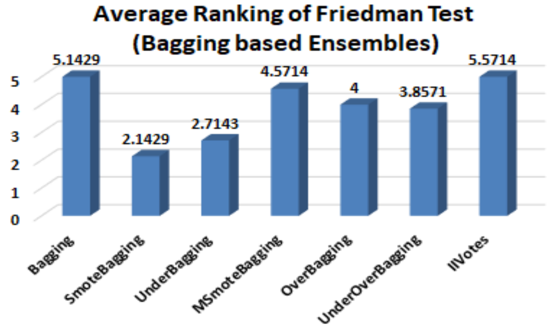


Fig. 11. Ranks Assigned by Friedman Test.

TABLE VI. STATISTICS USING HOLM TEST FOR COMPARING BOOSTING BASED ENSEMBLES

Control method: RusBoost (1.7143)				
I	**Methods**	**Z Value**	**Holm (p-value)**	**Hypothesis (α=0.05)**
5	DataBoost-IM	3.142857	0.01	**Rejected**
4	AdaBoost	2.857143	0.0125	**Rejected**
3	MSmoteBoost	2.714286	0.016667	**Rejected**
2	EusBoost	1	0.025	**Rejected**
1	SmoteBoost	1	0.05	Not Rejected

TABLE VII. STATISTICS USING WILCOXON TEST FOR COMPARING RUSBOOST AND SMOTEBOOST

Methods	**R^+**	**R^-**	**Hypothesis (α=0.05)**	**p-value**
RusBoost Vs SmoteBoost	26.0	2.0	Rejected, Significant differences between methods	0.04688

TABLE VIII. TEST STATISTICS USING FRIEDMAN TEST (BAGGING ENSEMBLES)

N	07
Chi-Square (F_{AR})	13.8367
Degree of Freedom (K-1)	6
p-value	0.031514

TABLE IX. STATISTICS USING HOLM TEST FOR COMPARING BAGGING BASED ENSEMBLES

Control method: SmoteBagging (2.1429)				
I	**Methods**	**Z Value**	**Holm (p-value)**	**Hypothesis (α=0.05)**
6	IIVotes	2.96923	0.0083	**Rejected**
5	Bagging	2.59807	0.01	**Rejected**
4	MSmoteBagging	2.10320	0.0125	**Rejected**
3	OverBagging	1.60833	0.016667	**Rejected**
2	UnderOverBagging	1.48461	0.025	**Rejected**
1	UnderBagging	0.49487	0.05	Not Rejected

TABLE X. STATISTICS USING WILCOXON TEST FOR COMPARING SMOTEBAGGING AND UNDERBAGGING

Methods	**R^+**	**R^-**	**Hypothesis (α=0.05)**	**p-value**
SmoteBagging Vs UnderBagging	16.0	12.0	Accepted, No significant differences	0.67260

TABLE XI. STATISTICS USING WILCOXON TEST FOR COMPARING BALANCECASCADE AND EASYENSEMBLE

Methods	**R^+**	**R^-**	**Hypothesis (α=0.05)**	**p-value**
BalanceCascade Vs EasyEnsemble	11.0	10.0	Accepted, No significant differences	0.83393

Next step is to analyze these three best performer methods. We again performed Friedman Test with these three methods. Ranks assigned by the test shows (Fig. 12) that RusBoost is the best performer and Balancecascade is the worst performer and Friedman Test statistic (Table XII) reveals that chi-square (χ^2) table value for 2 degree of freedom (5.9915) is less than calculated value (6.0), hence there are no significant differences between the methods. We further analyze the methods with Holm post-hoc analysis. Test statistics (Table XIII) shows that RusBoost and SmoteBagging are similar as the null hypothesis for no significant differences is accepted. As a last step to find the best performer out of all ensemble methods, we closely analyzed RusBoost and SmoteBagging with Wilcoxon matched pair test. Although, the p-value of the test statistic shown in the table (Table XIV) is more than 0.05, which means that there are no significant differences between these pair of methods but the higher rank value of RusBoost shows that its performance is better than SmoteBagging. Another advantage of RusBoost is that as it is using undersampling approach within the boosting process to classify the data-set so it is computationally less expensive compare to SmoteBagging which follows oversampling approach and bagging process for classification.

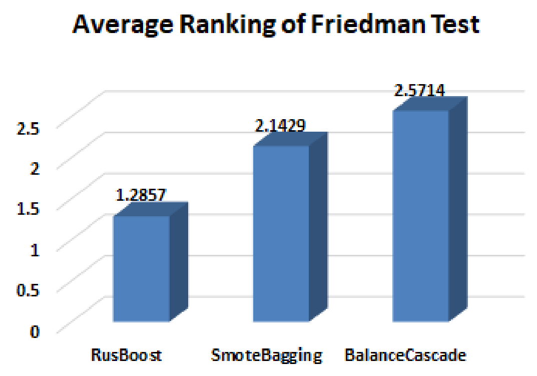


Fig. 12. Ranks Assigned by Friedman Test.

TABLE XII. TEST STATISTICS USING FRIEDMAN TEST (BEST PERFORMER ENSEMBLES)

N	07
Chi-Square (F_{AR})	6.0
Degree of Freedom (K-1)	2
p-value	0.049787

TABLE XIII. STATISTICS USING HOLM TEST FOR COMPARING THE CANDIDATE METHODS FOR BEST PERFORMER ENSEMBLES

Control method: RusBoost (1.2857)				
I	**Methods**	**Z Value**	**Holm (p-value)**	**Hypothesis (α=0.05)**
2	BalanceCascade	2.405351	0.025	**Rejected**
1	SmoteBagging	1.603567	0.05	Not Rejected

TABLE XIV. STATISTICS USING WILCOXON TEST FOR COMPARISON BETWEEN RUSBOOST AND SMOTEBAGGING

Methods	**R^+**	**R^-**	**Hypothesis (α=0.05)**	**p-value**
RusBoost Vs SmoteBagging	23.0	5.0	Accepted, No significant differences	0.108319

From the visual interpretations and the statistical analysis, we can say that RusBoost outperformed other ensemble based methods in the imbalance domains.

IV. CONCLUSION

In the current study, we review various boosting and bagging based ensemble approaches for their performance in imbalanced domains by focusing on binary classification. We empirically assessed 15 approaches using 7 imbalanced data sets (KEEL repository) with the class imbalance ratio from 1.82 to as high as 129.44. After analyzing the results through statistical analysis methods (Wilcoxon matched signed rank and Friedman test), it is reported that RusBoost has outperformed other 14 methods considering any level of imbalance ratio. In future, we are planning to propose an ensemble approach which can work efficiently in the presence of other data impurities like noise, etc. along with data-set.

APPENDIX A

TABLE AI PROPERTIES OF DATA SETS

Sr. No	**Data sets**	**Imbalance Ratio**	**Number of Dimensions**	**Minority Class %**	**Size of data-set**
1	Glass1	1.82	9	35.51	214
2	Vehicle3	2.99	18	25.06	846
3	Yeast3	8.10	8	10.98	1484
4	Ecoli4	15.80	7	5.95	336
5	Abalone9-18	16.40	8	5.75	731
6	Yeast5	32.78	8	2.96	1484
7	Abalone19	129.44	8	0.77	4174

REFERENCES

[1] A. Fernandez, V. LóPez, M. Galar, M. J. Del Jesus and F. Herrera, “Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and adhoc approaches”, Knowledge Based Systems, Vol. 42, pp 97-110, 2013.

[2] G. E. A.P.A Batista, R. C. Prati, and M. C. Monard, “A study of the behaviour of several methods for balancing machine learning training data”, SIGKDD Expl. Newl., Vol 6, No. 1, pp 20-29,2004.

[3] 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.

[4] J. A. Saez, J. Luengo, J. Stefanowski, and F. Herrera, “Managing Borderline and Noisy examples in Imbalanced Classification by combining SMOTE with Ensemble Filtering”, IDEAL2014, LNCS, Vol. 8669, pp. 61-68, 2014, Springer.

[5] R. Akbani, S. Kwek, and N. Japkowicz, “Applying Support Vector Machines to Imbalanced Datasets”, ECML 2004, LNAI 3201, pp. 39-50, 2004. Springer-Verlag Berlin Heidelberg.

[6] H. Han, W. Wang, and B. Mao, “Borderline-SMOTE: A new oversampling method in Imbalanced Data-sets Learning”, In. ICIC 2005. LNCS, Vol. 3644, pp. 878-887. Springer, Heidelberg, 2005.

[7] J. Stefanowski and S. Wilk, “Selective Preprocessing of imbalanced data for improving classification performance”, Datawarehousing and Knowledge Discovery (Lecture Notes in Computer Science Series 5182, pp 283-292), 2008.

[8] S. Hu, Y. Liang, L. Ma, and Y. He, “MSMOTE: Improving classification performance when training data is imbalanced”, In prpoc. 2nd Int. Workshop Computer Sci. Eng., Vol. 2, pp. 13-17, 2009.

[9] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, “Safe-Level-SMOTE: Safe Level- Synthetic Minority Over-Sampling Technique for handling the Class Imbalance Problem”, PADD2009, LNAI, Vol. 5476, pp. 475-482, 2009, Springer.

[10] Ying Mi, “Imbalanced classification based on Active Learning SMOTE”, Research Journal of Applied Sciences, Engg. And Tech., Vol. 5, issue 3, pp. 944-949, 2013.

[11] P. Kaur and A. Gosain, “FF-SMOTE: A Metaheuristic Approach to Combat Class Imbalance in Binary Classification”, Applied Artificial Intelligence, 2019, Tayler & Francis. DOI: 10.1080/08839514.2019.1577017.

[12] P. Kaur and A. Gosain, “Comparing the behaviour of Undersampling and Oversampling of Class Imbalance Learning by combining Class Imbalance problem with noise”, A.K. Saini et al. (eds.), ICT Based Innovations, Advances in Intelligent Systems and Computing 653, pp. 23-30, 2018. https://doi.org/10.1007/978-981-10-6602-3_3.

[13] P. Kaur and A. Gosain, “An Intelligent Undersampling technique based upon Intutionistic Fuzzy sets to alleviate Class Imbalance Problem of Classification with noisy environment”, International Journal of Intelligent Engineering Informatics, Vol. 6, No. 5, pp. 417-433, 2018.

[14] M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets:one-sided selection”, International Conference on Machine Learning,Vol. 97, pp. 179-186, July 1997.

[15] Y. M. Chyi, “Classification analysis techniques for skewed class distribution problems”, Department of Information Management, National Sun Yat-Sen University, 2003.

[16] S. Garcia and F. Herrera, “Evolutionary undersampling for classification with imbalanced data-sets: proposals and texonomy”, Evolutionary Computation, Vol. 17, pp 275-306, 2009.

[17] Show-jane Yen and Yue-Shi Lee, “Cluster-based under-sampling approaches for imbalanced data distributions”, Expert Systems with Applications, Vol. 36, pp 5718-5727,2009.

[18] M. M. Rahman and D. N. Davis, “Cluster based undersampling fAHCor unbalanced Cardiovascular Data”, WCE 2013, July 3-5, Vol III, London, UK, 2013.

[19] T. Maruthi, B. S. Raju, R. N. Hota, and P. R. Krishna, “Class imbalance and its effect on PCA pre-processing”, International Journal of Knowledge engineering and Soft Data Paradigms, Vol. 4, No. 3, pp. 272-294, 2014.

[20] D. Li, S. S. Wu, T. I. Tsai, and Y. S. Lina, “Using mega-trend diffusion and artificial samples in small data-set learning for early flexible manufacturing system scheduling”, Knowledge, Computers and Operational Research, Vol. 34, pp. 966-982, 2007.

[21] E. Ramentol, Y. Caballero, R. Bello and F. Herrera, “ SMOTE-RS B*: a hybrid pre-processing approach based on Oversampling and Undersampling for high imbalanced data-sets using SMOTE and Rough set theory”, Knowledge Information Systems, Springer,2011. DOI 10.1007/s 10115-011-0465-6.

[22] G. Wu and E. Chang, “Kba: Kernel Boundary alignment considering imbalanced dataset distribution”, IEEE Transactions on Knowledge and Data Engineering. Vol. 17, No. 6, pp. 786-795, 2005.

[23] T. Iman, K. Ting, and J. Kamruzzaman, “z-SVM: An SVM for improved classification of imbalanced data”, In proceedings of the 19th Australian joint conference on Artificial Intelligence, pp. 264-273, springer-verlag, 2006.

[24] X. Hong, S. Chen, and C. J. Harris, “A kernel based two class classifier for imbalanced data-sets”, IEEE Transactions on Neural Networks, Vol. 18, No. 1, pp. 28-41, 2007.

[25] A. Fernandez, S. García, M. J. del Jesus, and F. Herrera, “A study of the behaviour of linguistic fuzzy rule base classification systems in the framework of imbalanced data-sets”, Fuzzy Sets and Systems, Vol. 159, issue 18, pp. 2378-2398, 2008.

[26] C-Y Yang, J-S Yang, and J-J Wang, “Margin calibration in svm class imbalanced learning”, Neurocompuitng, Vol. 73, No. 1-3, pp. 397-411, 2009.

[27] A. Fernandez, M. J. del Jesus, and F. Herrera, “Hierarchical fuzzy rule base classification system with genetic rule selection for imbalanced data-sets”, International journal of Approxmate reasoning, Vol. 50, pp. 561-577, 2009.

[28] R. Batuwita and V. Palade, “FSVM-CIL: Fuzzy Support vector machine for class imbalanced learning”, IEEE Transactions on Fuzzy Systems, Vol. 18, No. 3, pp. 558-571, 2010.

[29] Z. Zhao, P. Zhong, and Y. Zhao, “Learning SVM with weighted maximun margin criterion for classification of imbalanced data”, Mathematical and Computer Modelling, Vol. 54, pp. 1093-1099, 2011.

[30] X. Gu, T. Ni, and H. Wang, “New Fuzzy Support Vector machine for the Class Imbalance Problem in Medical data-sets Classification”, The Scientific World Journal. Vol. 2014, pp. 1-12, Hindawi Publishing Corporation.

[31] P. Domingos, “Metacost: a general method for making classifiers cost sensitive”, In: Fifth International Conference on Knowledge Discovery and Data mining, Vol. 99, p. 155-164, Aug. 1999.

[32] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, “Adacost: Misclassification cost-sensitive boosting”, 6th Inter. Conf. Of machine learning”, pp 97-105, SFO, CA, 1999.

[33] 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.

[34] K. M. Ting, “A comparative study of cost-sensitive boosting algorithms”, In proc. 17th Int. Conf. On mac. Learning, Stanford CA, pp. 983-990, 2000.

[35] P. K. Chan and S. J. Stolfo, “Toward Scalable Learning with non-uniform class and Cost Distribution: A case Study in Credit Card Fraud Detection”, In. Proceedings of the fourth international Conference on Knowledge Discovery and Data Mining, pp. 164-168, 2001.

[36] M. Joshi, V. Kumar and R. Agarwal, “Evaluating Boosting algorithms to classify rare classes”, In proc. IEEE International Conference on data Mining, pp. 257-264, 2001.

[37] S-C Lin, I. C. Yuan-chin, and W. N. Yang, “Meta-learning for imbalanced data & classification ensemble in binary classification”, Neurocomputing, Vol. 73, pp. 484-494, 2009.

[38] Y. Yang and G. Ma, “Ensemble based active learning for Class imbalance problem”, J. Biomedical Science and Engineering, Vol. 3, pp.1021-1028, 2010.

[39] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “RUSBoost: A Hybrid Approach to Alleviating Class Imbalance”, IEEE Trans. On Sys. Man and Cyber.-Part A, Vol. 40, No.1, pp 185-197, 2010.

[40] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, “EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling”, Pattern Recognition, Vol. 46, pp 3460-3471, 2013.

[41] S. Hu, Y. Liang, L. Ma, and Y. He, “MSMOTE: Improving classification performance when training data is imbalanced”, In prpoc. 2nd Int. Workshop Computer Sci. Eng., Vol. 2, pp. 13-17, 2009.

[42] R. Barendela , J. S. Sa´nchez, and R. M. Valdovinos, “New applications of ensembles of classifiers”, Patterns Anal. Appli., vol 6, pp. 245-256, 2003.

[43] R. Yan, Y. Liu, R. Jin, and A. Hauptmann, “On predicting rare classes with SVM ensembles in scene classification”, IEEE International

Conference on Acoustics, Speech and Signal processing, Vol. 3, pp. 21-24, 2003.

[44] 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, 2006.

[45] S. Wang and X. Yao, “Diversity analysis on imbalanced data-sets by using ensemble models”, IEEE symp. Cpmput. Intell. Data Mining, 2009, pp. 324-331.

[46] S. Hido, H. Kashima, and Y. Takahashi, “Roughly balanced bagging for imbalanced data”, Statistical analysis of Data Mining, Vol. 2, pp. 412-426, 2009.

[47] J. Blaszczynski, M. Deckert, J. Stefanowski, and S. Wilk, “Integrating Selective pre-processing of imbalanced data with ivotes ensemble”, Rough sets and Current trends in Computing (Lecture notes in Computer Science Series 6086), Springer-Verlag, pp. 148-157, 2010.

[48] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, “SMOTBoost: Improving pridiction of the minority class in boosting”, In proc. Knowledge Discovery databases, 2003, pp. 107-119.

[49] H. Guo and H. L.Viktor, “Learning from imbalanced data-sets with boosting and data generation: The Databoost-IM approach”, SIGKDD Expl. Newsl., Vol. 6, pp. 30-39, 2004.

[50] B. X. Wang and N. Japkowicz, “Boosting Support Vector machines for imbalanced data-sets”, Knowledge Information Systems, Vol. 25, Issue 1, pp. 1-20, 2010.

[51] M. J. Kim, “Geometric mean based boosting algorithm to resolve data imbalance problem”, DBKDA2013, The fifth International conf. On Advances in databases, knowledge and data applications, pp. 15-20, 2013.

[52] K. Lokanayaki and A. Malathi, “A pridiction for classification of highly imbalanced medical dataset using Databoost.IM with SVM”, International journal of Advanced Research in Computer Science & Software Engg., Vol. 4, Issue 4, April 2014.

[53] X. Y. Liu, J. Wu, and Z. H. Zhou, “Exploratory Undersampling for class imbalance learning”, IEEE Tran. Systm. Man and Cyber. B, Appl. Rev. Vol. 39, No. 2, pp. 539-550, 2009.

[54] J. Alcalá-Fdez et al., “KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework”, Journal of Multiple- Valued Logic and Soft Computing, Vol 17, No. (2–3), pp. 255–287, 2011.

[55] J. Alcalá-Fdez et al., “KEEL: a software tool to assess evolutionary algorithms for data mining problems”, Soft Computing, Vol. 13, No. 3, pp. 307–318, 2008.

[56] J. L. Hodges and E. L. Lehmann, “Rank methods for combination of independent experiments in analysis of variance”, Annals of Mathematical Statistics, Vol. 33, pp. 482–497, 1962.

[57] F. Wilcoxon, “Individual comparisons by ranking methods”, Biometrics Bulletin, Vol. 1, No. 6, pp. 80–83, 1945.

[58] Z-H Zhau, “Ensemble Methods-Foundations and Algorithms”, CRC Press, Tayler and Francis Group, 2012.

[59] Y. Freund and R. Schapire, “Experiments with a new boosting algorithms”, In Proc.13th Int. Conf. Machine Learning, pp. 148-156, 1996.

[60] L. K. Hensen and P. Salamon, “Neural Network Ensembles”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, pp. 993-1001, 1990.

[61] R. E. Schapire, “The strength of weak learnability”, Machine Learning, Vol. 37, No. 2, pp. 197-227, 1990.

[62] R. Barendela, J. S. Sa´nchez, and R. M. Valdovinos, “New applications of ensembles of classifiers”, Patterns Anal. Appli., Vol 6, pp. 245-256, 2003.

[63] K. M. Ting, “An instance-weighting method to induce cost sensitive trees”, IEEE Trans. Of knowledge and data Engg., Vol. 14, issue 3, pp. 659-665, 2002.

[64] L. Breiman, “Bagging predictors”, Machine Learning, Vol. 24, pp. 123--140, 1996.

[65] S. Wang and X. Yao, “Diversity analysis on imbalanced data sets by using ensemble models”, in IEEE Symposium Series on Computational Intelligence and Data Mining, 2009, pp. 324-331.

[66] R. Barendela, J. S. Sa´nchez, and R. M. Valdovinos, “New applications of ensembles of classifiers”, Patterns Anal. Appli., vol 6, pp. 245-256, 2003.

[67] M. Galar, A. Ferndez, 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. doi: 10.1109/TSMCC.2011.2161285.

[68] J. Stefanowski and S. Wilk, “Selective Preprocessing of imbalanced data for improving classification performance”, Datawarehousing and Knowledge Discovery (Lecture Notes in Computer Science Series 5182, pp 283-292), 2008.

[69] L. Breiman, “Pasting small votes for classification in large databases and on-line”, Machine Learning, Vol. 36, pp. 85-103, 1999.

[70] C. E. Metz, “Basic Principals of ROC Analysis”, Seminars in Nuclear Medicine. 8(4), pp 283-298, 1978.

[71] J. A. Hanley and B. J. McNeil, “A method of comparing the areas under receiver operating characteristic curves derived from the same cases”, Radiology, 148(3), pp 839-843, 1983.

[72] J. Demšar, “Statistical comparisons of classifiers over multiple datasets”, Journal of Machine Learning Research, Vol. 7, pp. 1–30, 2006.

[73] S. García and F. Herrera, “An extension on statistical comparisons of classifiers over multiple datasets for all pairwise comparisons”, Journal of Machine Learning Research, Vol. 9, pp. 2677–2694, 2008.

[74] S. García, A.Fernández, J.Luengo, and F.Herrera, “Advanced non parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power”, Information Sciences, Vol 180, pp. 2044–2064, 2010.

[75] S. Holm, “A simple sequentially rejective multiple test procedure”, Scandinavian Journal ofStatistics, Vol. 6, pp. 65–70, 1979.