Open Access
Full Length Article

ORIGINAL ARTICLE

# SMOTEMultiBoost: Leveraging the SMOTE with MultiBoost To Confront the Class Imbalance in Supervised Learning

**Naveed Jhamat[1], Ghulam Mustafa[2], Zhendong Niu[3]**

[1]Assistant Professor, Department of Information Technology, University of the Punjab, Gujranwala Campus, Pakistan.
[2]Assistant Professor, Department of Information Technology, University of the Punjab, Gujranwala Campus, Pakistan.
[3]Professor, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China.

## ABSTRACT

Class imbalance problem is being manifold confronted by researchers due to increasing amount of complicated data. Common classification algorithms are impoverished to perform effectively on imbalanced datasets. Larger class cases typically outbalance smaller class cases in class imbalance learning. Common classification algorithms raise larger class performance owing to class imbalance in data and overall improvement in accuracy as their goal while lowering performance on smaller class. Furthermore, these algorithms deal false positive and false negative in an even way and regard equal cost of misclassifying cases. Meanwhile, different ensemble solutions have been proposed over the years for class imbalance learning but these approaches hamper the performance of larger class as emphasizing on the small class cases. The intuition of this overall degraded outcome would be the low diversity in ensemble solutions and overfitting or underfitting in data resampling techniques. To overcome these problems, we suggest a hybrid ensemble method by leveraging MultiBoost ensemble and Synthetic Minority Over-sampling TEchnique (SMOTE). Our suggested solution leverages the effectiveness of its elements. Therefore, it improves the outcome of the smaller class by reinforcing its space and limiting error in prediction. The proposed method shows improved performance as compare to numerous other algorithms and techniques in experiments.

**Keywords:** Class Imbalance Learning, Diversity, SMOTE, MultiBoost Learning

Author`s Contribution
*[1,2] Manuscript writing, Data Collection Data analysis, interpretation, Conception, synthesis, ,[3]planning of research, and discussion*

*Address of Correspondence*
*Naveed Jhamat*
*naveed.jhamat@pugc.edu.pk*

Article info.
*Received: June 11,2020*
*Accepted: November 24,2020*
*Published: December 30,2020*

***Cite this article:*** *Jhamat N, Mustafa G, Niu Z. SMOTEMultiBoost: Leveraging the SMOTE with MultiBoost to Confront the Class Imbalance in Supervised Learning. J. Inf. commun. technol. robot. appl.2020; 11(2):8-21*

***Funding Source:*** *Nil*
***Conflict of Interest:*** *Nil*

## INTRODUCTION

Widening use of e-commerce and web activities accelerates the creation of enormous quantity of raw data. Classification is the key subfield of machine learning where unlabelled data is labelled, based on learning from past labelled data. The classifiers are trained to make accurate predictions based on matching past data [1-7]. The classifiers are provided with the training data where instances or cases are already labelled with the accurate labels. This entered data is utilized to build models that can be used to

classify unlabelled data. Decision trees [8-14], naive Bayes [15], support vector machines [16], ensemble learners [17-21], etc. have been successfully used to many categorization problems.

Typically, classifiers try to achieve higher accuracy or try to reduce error rate as the evaluation benchmark [22]. Moreover, classifiers cope with type I and type II error in equal manner and consider equal cost of misclassification of cases for distinct classes. However, there is unequal misclassification costs in class imbalanced datasets where one class has lesser cases as compared to the other class. For example, it is much less likely to detect malign cell among normal cells in the body. The infrequent cases in particular class are usually more worthy and significant [5, 23-25]. The miss to predict accurate class could be an unenviable loss.

In imbalanced datasets, the class with fewer cases is referred as smaller or minority class and class with many cases is referred as larger or larger class [22]. The common classifiers with the aim of achieving maximum accuracy are generally inefficient on smaller class performance. This inadequacy in common classifiers is due to a little part from smaller class and some other errors. These classifiers achieve high prediction accuracy despite poor performance with smaller class identification [26-29]. This poor performance on smaller class is due to class imbalance; however, it has been noted that not only class imbalance but other issues also play important role in common classifiers underperformance on smaller class. The challenges include scarceness of data, class overlapping and small disjuncts, etc. [22, 30-35]. These challenges along with imbalanced data representation limit the functionality of common classifiers on imbalanced datasets. Nevertheless, the ordinary classifiers attain higher accuracy performance by correctly categorizing all larger class cases and disregarding the skewed class cases. However, this result is not adequate and recognition of smaller class cases is much desirable than larger class cases. Class imbalance problem can be experienced in fraud detection [16], risk management [36, 37], health care [38, 39], software quality assurance [40-42], sentiment classification[43] and abstract classification [1], etc.

The machine learning community has proposed various methods to address the requirements of class imbalance issue. These algorithms and techniques can be divided into two kinds of approaches. In first approach, data sampling techniques are used to equalize the class distributions. Data under sampling and data oversampling are two major kind of methods used to balance the class distributions. These methods are simple to apply but these methods are not without limitations. The oversampling methods may face over fitting and under sampling method may loss valuable information. On the other hand, various algorithms are tailored to cope with the Skewness problem in the second approach. The basic objective is to decrease the tendency towards the larger class. Additionally, cost sensitive algorithms are also used to address class imbalance problem. They choose high costs to skewed class cases with regard to the larger class cases. However, cost sensitive classifiers are domain dependent for cost estimation.

Furthermore, ensemble learning approach are also used in classification. In this approach, multiple ordinary learning algorithms are united to create a strengthened learner. For example, AdaBoost [18] and Bagging [16] are two such methods that combine the multiple weak classifiers to form a strong classifier. To manage the class Skewness problem, ensemble methods are adapted in the past. These ensemble classifiers are coalesced with external resampling techniques to address the class imbalance problem. For example: BEV [18], SMOTEBoost [44], Asymmetric Bagging [45], RareBoost [15], RUSBoost [46], EasyEnsemble [47], IIVotes [48] and others [49, 50]. The data resampling methods try to balance the class distribution and ensemble methods improve the overall performance by reducing the error. It is also reported that ensemble learners combined with data resampling techniques increases the diversity and hence improve the accomplishment of the smaller class [51]. Furthermore, it is also noted that ensemble solutions decrease the functioning of the larger class while focusing on the smaller class. Consequently, an elaborate study is required to see the effects of ensembles with regard to the class imbalance problem. This effort can provide the insight of learning prospect of the ensembles and might motivate us to use it for better performance on

both smaller and larger class. It inspires new adapted ensemble algorithms for class imbalance learning that promotes diversity and overall improved performance. It is anticipated to address class imbalance problems with significant improvement.

We are principally focused in the hybrid ensembles which integrate the data pre-processing methods with ensemble learning. The hybrid ensembles are effective in reducing bias and variance in the error and data preprocessing methods are effective to solve imbalance problem in data. In this paper, we design SMOTEMultiBoost technique for class imbalanced datasets. MultiBoost [52] ensemble is mixed with SMOTE [23] technique to enhance the results of smaller class, in addition to maintain high performance on larger class. MultiBoost is a technique that uses wagged subcommittees which consists of week boosted learners. This combination effectively reduces the error and hence enhance the outcome of the learner. While, SMOTE intelligently creates synthetic cases to extend the boundaries of scarce instance class and therefore promote the recall of smaller class cases. Our proposed method has the ability of its factors and so enhance the prediction ability of smaller class by reinforcing the smaller space and final results by limiting the error. Various benchmark datasets are utilized to appraise the functioning of the proposed class imbalanced learning algorithm. These datasets are popular binary class imbalance datasets used in class imbalance learning. These datasets are available at KEEL data repository. Evaluation on a range of benchmark datasets with different attributes provides a real scenario of the functioning of the proposed learning algorithms. The outcome of experiments show that it outperformed numerous other existing algorithms and techniques. The remainder of this paper is arranged as follows. Section 2 reports related work, and Section 3 introduces the preliminaries. Section 4 outlines the SMOTEMultiBoost algorithm. Section 5 provides the details of our experiments, and the experimental outcomes are depicted in Section 6. In section 7, we conclude our paper.

## LITERATURE REVIEW

In this part, we review commonly used class imbalanced learning algorithms and techniques. Several approaches have been presented to handle the class imbalance problem. Broadly, class imbalance learning approaches can be split into external and internal techniques. For external techniques, class distribution is adapted to support smaller class by reducing the cases of the class having more cases or adding to the cases of the class having fewer cases. For internal techniques, the class bias is altered in support of smaller class by tailoring the existing algorithms. Additionally, innovative composite techniques were also presented that merge the external and internal techniques to approach the class imbalance problem. It has been exhibited that resampling techniques are useful while treating the class imbalance problem [53].

The fundamental resampling techniques are random oversampling and random undersampling [54]. Oversampling methods add the artificial cases to smaller class to equalize the class distribution. Random oversampling is the easy method to randomly add the cases to smaller class. Moreover, different sophisticated methods have been presented to decrease the negative outcome of primitive oversampling technique [55]. Synthetic oversampling is another approach that sensibly produces the smaller class cases. For instance, Synthetic Minority Oversampling Technique (SMOTE) [23] wisely produces new smaller class cases. SMOTE technique generates new artificial cases of smaller class among the under consideration instance and its k-nearest neighbours. However, SMOTE suffers from overfitting problem due to its artificial cases creation process. Furthermore, several variants of SMOTE have been introduced that use more adaptive approaches to improve the performance of SMOTE [4, 56], etc.

Additionally, different data cleaning techniques have been proposed that are used with oversampling methods to overcome the overlapping problem. Tomek links [57] have least distanced nearest neighbours of larger and smaller class. Some techniques used data cleaning methods with resampling methods to improve the performance [54, 58]. Moreover, cluster-based sampling methods are applied to handle the class imbalance. They provided the adaptability that is not

available in other sampling methods. For example, cluster-based oversampling algorithm uses the K-mean clustering approach to effectively solve the within-class and between-class problem [32].

The second type of resampling approach is known as under sampling. Random under sampling (RUS) is a primitive data under sampling technique to adjust the class distribution. In this approach cases from the larger class are randomly vanished to reach the desired balance. The main limitation of this technique is loss of valuable information [59]. The loss of valuable cases may result for the learning algorithm to ignore substantive concepts relating to the larger class. To reduce the information loss during the under sampling process, different advanced methods were presented to make more knowledgeable under sampling [60, 61].

Additionally, cost sensitive learning is utilized to address imbalance in data. This approach assigns various costs for classifying cases. Cost sensitive learning allows more cost to smaller class cases so that cost sensitive algorithms can focus on these cases to learn a useful class boundary [62-64]. However, cost sensitive learning techniques are not without limitations. The misclassification cost is based on expert judgment and furthermore, majority of classification algorithms are not adaptable to include cost into their learning procedure [65]. Furthermore, class imbalanced learning approaches can be crosswise split up with respect to ensemble learning. These algorithms present encouraging outcome in improving the performance of weak or fragile algorithms. Ensemble approach uses ordinary learners to form a strong and robust learner. Ensemble approach is divided into cost sensitive and resampling approach. The cost sensitive approach integrates internal and external approaches. These algorithms incorporate dissimilar costs of incorrectly classified cases into their learning mechanism. Different cost sensitive ensemble has been presented to confront the imbalance difficulty. The most prominent are AdaC1, AdaC2 and AdaC3 [66]. In these techniques, cost factor is introduced into the updating portion of AdaBoost ensemble. The AdaBoost.M1 algorithm iteratively updates the distribution function. By introducing the cost factor into the updating process these cost sensitive ensembles enhance the chances of sampling costly cases iteratively. This is due to the fact that these algorithms give more chances to the costly cases for more aimed approach to induction. This enhancement in the AdaBoost increases the bias of the algorithm towards the smaller class and it ensures the inclusion of more relevant data for better classification on smaller class.

The data resampling ensembles can be further distinguished into three subclasses. The foremost subclass is boosting based. This subclass represents SMOTEBoost [44], RUSBoost [46], Asymmetric AdaBoost [45] etc. In this subclass of techniques, data resampling methods are unified with boosting based ensembles to treat the imbalance in the data. SMOTEBoost integrates SMOTE with the AdaBoost to make an ensemble technique that identifies the smaller class cases. In SMOTEBoost, SMOTE is fused with boosting procedure to handle the class imbalance by extending the smaller class. The new learner method thus enforces the border of the smaller class and raises learner's diversity. Furthermore, RUSBoost is a type of SMOTEBoost but it incorporates random under sampling (RUS) rather than intelligent oversampling technique.

Additionally, RUSBoost is easy method as compared to SMOTEBoost because random undersampling (RUS) is a computationally cheaper algorithm. Additionally, some techniques have been proposed that combine the Bagging ensemble [16] with the data resampling techniques to handle the class imbalance problem [67, 68]. Furthermore, Bagging Ensemble Variation (BEV) [18] also uses combination of undersampling and Bagging. However, in this approach the larger cases are split into separate subsets and all subsets are combined with smaller class cases individually and then, each learner is trained with one of those subsets.

In the third subclass, hybrid ensembles are combined with data resampling techniques. In hybrid ensemble approach, two ensembles are combined to take advantage of its constituent e.g. EasyEnsemble and BalanceCascade [47], etc. EasyEnsemble forms an ensemble by combining separate subsets of larger class cases with the single whole set of smaller class cases. Then, on these separate subsets, AdaBoost

ensembles are trained. EasyEnsemble combines bagging and AdaBoost as its base learner. This algorithm is designed to overcome the drawback of random undersampling where valuable information is lost. However, in this algorithm, larger class cases are used in different subsets rather than simply discarding them. The main steps of EasyEnsemble are: A subsample is drawn from the training set which contains whole smaller class cases and a subset of larger class cases in equal quantity. Then these combined subsets called as bag is used by AdaBoost ensemble. Multiple AdaBoost base learners are trained on this bag of cases. At the end, these AdaBoost ensembles are aggregated to get the final hypothesis. On the other hand, BalanceCascade works in supervised fashion as compared to EasyEnsemble. In BalanceCascade the larger class cases are lay off in organized way. The basic functionality of this algorithm act in cascade way where larger class cases are thrown out from training set on right classification. Moreover, BalanceCascade proceeds as: In the first step, bags of cases are prepared from training set by drawing instance from larger class and all instance of smaller class. Then, a AdaBoost learner is trained on this new subset. Furthermore, cases from larger class are droped which are correctly classified. At the end, the ultimate hypothesis is formed by combining all the constituent learners as in EasyEnsemble.

## METHODOLOGY

### A. Preliminaries

We use some common notations that are associated with the class imbalance problem to remove the ambiguity. We represent the base learning algorithm with L and represent a committee of learners known as the final hypothesis with H∗. We represent T subcommittees having size $\sqrt{T}$, whereas $H* = T\sqrt{T}$. Let, S is the features-class vector of size m. All pair $(x_i, y_i)$ associates features $x_i \in X$ and class $y_i \in Y$. Let a curser t is used that iterates through iterations T. Suppose, $H_t$ be the constituent learner trained and $H_t(x_i)$ be the output of learner $H_t$, for instance, $x_i$ . Suppose, $D_t(i)$ holds the value of the $i_{th}$ instance on repetition t. H∗ (x) is the hypothesis attained maximum votes from its constituents when implement to x. Ii represents the resulted index for subcommittees. We show the oversampling rate with N and nearest neighbors used in it with nn.

### B. Synthetic Minority Over-sampling TEchnique (SMOTE)

It is an intelligent method to handle the skewed data. SMOTE makes fresh artificial smaller class cases by using the existing cases. These new artificially created synthetic cases are based on k-nearest neighbors. This procedure randomly forms artificial smaller class cases which determine details from nn-nearest neighbors. It extrapolates the boundary for the smaller class and therefore reduce the over fitting problem in data.

The SMOTE works as follow: For each instance of smaller class, find and take its nn-nearest neighbors (nn is specify by user). The fresh artificial instance is placed amongst the selected smaller class instance and nearest neighbors. Firstly, find the gap between the smaller class instance and its nn-nearest neighbors. Afterwards, normalize by multiplying it with any number between 0 and 1. Furthermore, combine fresh feature vector with the base feature vector, for continues features. On the other hand, select the cases with highest votes between smaller class feature vector and its neighbors, in the event of draw take any value and assign it to freshly created instance. SMOTE behaves differently for discrete and continues features to measure functions for closest neighbours. To measure the values for discrete characteristics, the Value Distance Metric is applied and the Euclidean Distance metric is applied to the continuous.

### C. MultiBoost Algorithm

MultiBoost is a group strategy that strengthen the wagging with the AdaBoost. Bootstrapping sampling is used in bagging where diverse sets are made. However, in this bootstrapping sampling a few cases may duplicate. Continuous Poisson distribution is used in Wagging to assign weights to cases. Additionally, weak learning algorithms are gel together to constitute a robust learning algorithm. Additionally, AdaBoost combines less effective classifier to constitute a more effective classifier. At every cycle, a week algorithm is included and cases weights are recomputed in light of

their previous final hypothesis decisions. Formally, MultiBoost comprises of subcommittees and their magnitude √ T. It determines an objective subcommittee member list, Ii which permits early end of subcommittee because out of bound error. The final outcome is the aggregation of results of the subcommittees. Furthermore, the structure of MultiBoost is also suitable for distributed computing.

### D. SMOTEMultiBoost Algorithm

It is important to argue about the undermentioned essential elements to adjust MultiBoost for learning from imbalance dataset. MultiBoosting lessens bias and variance in error in order to improve performance however it is insufficient for imbalanced datasets. To overcome the imbalance problem, the goal is disinclination of larger class cases by inclining the smaller occurrences while maintaining the achievements of larger class. We conflate the SMOTE within the Adboost to establish smaller class superiority and discover more extensive zones to enhance the outcome of smaller class. SMOTE is an oversampling strategy that uses resampling instead of reweighting which is contrary to the MultiBoost which uses reweighting. Furthermore, MultiBoost employ continuous Poisson distribution to adjust weights. We allot the values to fresh engineered cases induced by the SMOTE. SMOTEBoost beats contending techniques using boosting by reweighing [69]. Along these lines, regarding the possible margin of reweighting and to blend with the MultiBoost, we put the mean values of closest neighbors values to the fresh engineered occurrences. We utilized this technique since mean valuing do superior to early weighting methods [69]. The quantity of subcommittees framed and magnitudes are decided by individual. The proposed algorithm is presented in Algorithm 1.

To conclude, new algorithm is a unification of MultiBoost with the oversampling technique, SMOTE. SMOTEMultiBoost requires a parameter R, representing subcommittees. Continuous Poisson distribution is employed by all subcommittees for adjusting example values. An AdaBoost component is required for every subcommittee having magnitude equivalent to √ R. For every cycle of AdaBoost, SMOTE method artificially produces the smaller cases, characterized by argument N. Average weights estimated from closest neighbors weights are attributed to freshly engineered cases for calibration. An imperfect hypothesis Hr is constituted and assessed. If current AdaBoost element prematurely ends due to the classification error too large or too small then the next subcommittee is framed with expanded magnitude to repay end of premature subcommittee. This procedure continues until the marked subcommittee size is attained. At last, each subcommittee is consolidated for final vote H∗.

| **Algorithm 1: SMOTEMultiBoost** | |
|---|---|
| **Input:** training dataset S, base learner B, amount of learning repetitions R, Vector Ii which define the number of repetitions for each sub-committee where z 1, SMOTE proportion N. | |
| **Output:** ensemble H* | |
| 1 | S' = S % set the weight distribution for all cases to 1. |
| 2 | Set k = 1. |
| 3 | For r = 1 to T { |
| 4 | If $I_k = r$ then |
| 5 | $continuous\ PoissonDistribution(S')$ % With continuous Poisson distribution, random weights are drawn. |
| 6 | Regularize $(S')$% sum to 1. |
| 7 | k++ |
| 8 | $S_t' = SMOTE(D_t')$ % Build a temporary data set $S_t'$ having distribution $D_t'$ by generating N synthetic cases from smaller class Cm using SMOTE |
| 9 | Regularize$(S_t')$ |
| 10 | $H_r = B(S_t')$ % train a base learner $H_r$ from dataset $S_t'$ |
| 11 | $\varepsilon_r = \frac{\sum_{x_j \in S_t' : H_r(x_j) \neq y_j} D_r'(x_j)}{m}$ % calculate the error |
| 12 | If $\varepsilon_r > 0.5$ or $\varepsilon_r = 0$ then |
| 13 | Go to step 5. |
| 14 | $\beta_t = \frac{r}{1-\varepsilon_r}$ % calculate the weight of $H_r$ |

| | |
|---|---|
| 15 | For each $x_j \in S'_t$, |
| 16 | $D_{r+1}(x_j) = \frac{D_r(x_j)}{Z_r} \times \begin{cases} \beta_r & if\ H_r(x_j) \neq y_j \\ 1 & otherwise \end{cases}$<br>% update the distribution $D_{r+1}$,where $Z_r$is a normalization constant which alter $D_{t+1}$ to be a distribution |
| 17 | } |
| **Output the final classifier:** | |

$$H^*(x) = argmax_{y \in Y} \sum_{r:H_r(x)=y} log \frac{1}{\beta_t}$$

### E. Complexity Analysis.

The computational complexity of SMOTEMultiBoost algorithm is as follows. The computational complexity of SMOTE is O(Cm.N.NNc) which is symbolized with O(Cs). Where Cm is the quantity of true class instance, N is the quantity of new engineered cases and NNc is price to discover closest neighbors. Assume that an imperfect classifier is symbolized by B. CART algorithm's complexity is O (A.nlogn), where A, n represents quantity of attributes and cases, respectively. Assume, the count and size of subcommittees is represented as √ R. Hence, the computational complexity of proposed algorithm will be O(R.B.Cs).

| Dataset | Size | IP | #attributes |
|---|---|---|---|
| Hepatitis | 155 | 3.84 | 19 |
| Glass-0-1-6-vs-2 | 192 | 10.29 | 9 |
| Glass1 | 214 | 1.82 | 9 |
| Ionosphere | 351 | 1.79 | 34 |
| Ecoli-0-1-4-7-vs-2-3-5-6 | 336 | 10.59 | 7 |
| Wisconsin | 683 | 1.86 | 9 |
| Pima | 768 | 1.87 | 8 |
| Vehicle2 | 846 | 2.88 | 18 |
| Vehicle1 | 846 | 2.88 | 18 |
| Yeast1 | 1484 | 2.46 | 8 |
| Phoneme | 5403 | 2.4 | 5 |
| Satimage | 6435 | 9.28 | 36 |
| Mammography | 11183 | 42 | 7 |

### F. Experimental Setup

Thirteen binary class imbalanced datasets are used in investigations. We implemented various contending techniques including SMOTEMultiBoost on these datasets to look at and assess the viability of our proposed technique.

### G. Datasets

We utilized binary class datasets in our trials with various proportions of the smaller and larger classes. We utilized openly accessible datasets from KEEL dataset archive [70]. Detailed insights of these datasets are depicted in Table 1. In the dataset Satimage, entire classes are collapsed into a binary imbalance dataset, except the smallest class.

### H. Evaluation Metrics

The aim of this paper was to analyze the efficiency of the proposed procedure with distinctive evaluation measures. Confusion matrix is commonly used for categorization. Accuracy is a well-known used categorization metric. Nonetheless, we cannot reliably calculate the execution by biased datasets of learning algorithms. Consequently, to start with, we utilized Geo Metric Mean (G-mean) in our analyses [71]. It is represented as:

Gmean=

$$\sqrt{True\ Positive\ Rate\ x\ True\ Negative\ Rate} \quad \text{……..(1)}$$

F-measure is also employed in tests [73]. F1-measure utilizes accuracy (p) and recall (r) to figure the score. F1-measure is computed as:

$$F_1 - \text{measure} = \frac{2\times \text{precison} \times \text{recall}}{\text{precison} + \text{recall}} \quad \text{…….. (2)}$$

Thirdly, We also used Q-statistic to measure the relationship between different class distributions and diversity [72].

$$Q = \frac{N_{11} N_{00} - N_{01} N_{10}}{N_{11} N_{00} + N_{01} N_{10}} \quad \text{…….. (3)}$$

At last, we exercise ROC curve in tests. Receiver operating characteristic (ROC) curve is depiction of classifier outcome utilizing false positive rate(fpr) and true positive rate(tpr) on x-axis and y-axis, respectively [33]. We also used AUC to compare the performance of classifiers using single scalar value which represents the area under the curve.

### I. Evaluation Algorithms

Eight separate computing algorithms have been

used. The following include: MultiBoost, BalanceCascade and SMOTE, EasyEnsemble and RUSBoost (SMB), CART, SMOTEMultiBoost (SMB) and SMOTEBoost. In addition to pruning set to false, we used CART as a benchmark. As part of our inquiries, three distinct class dispersions of 35%, 50% and 65% are included. Five closest neighbors are utilized as a part of SMOTE. For MultiBoost and SMOTEMulti- Boost (SMB), the number of subcommittees and the magnitude of those subcommittees are set at three. Subsequently, it got to be distinctly add up to nine classifiers. Also, nine classifiers are utilized for RUSBoost, SMOTEBoost, EasyEnsemble and BalanceCascade, for reasonable examination. Tests are carried out utilizing WEKA. As the evaluation component, three fold-cross validations are used and each experiment is replicated 10 times.

## RESULTS AND DISCUSSION

In specific, the class imbalance learning techniques are designed for skewed datasets. Such techniques aim to boost the outcome of the smaller class while retaining the outcome of the larger class. The objective of the issue of class imbalance is to increase the efficiency of smaller class express by true positive rate (tpr). Similarly, by true negative rate, it is beneficial to retain high performance over greater class class express (tnr). Our new approach reflects on both the smaller and larger groups defined by the G-mean metric.

In each dataset with G-mean, AUC and F1-measures respectively, we display the execution of the classifiers using 65% small class distribution in Tables 2, 3, and 4, respectively. By applying Friedman [65] on results test, we validate the statistical value of improvement. As a specific learner for experiments SMOTEMultiBoost is used. In contrast to other learners on G-mean, AUC and F1-measurement measures, SMOTEMultiBoost indicates substantial improvement (p< 0.0, 05) of the results. Performance improvements presented with asterisk sign in Tables 2, 3, 4. asterisk sign in Tables 2, 3, 4.

**Table 2**

Result analysis of various methods with suggested SMOTEMultiBoost(SMB) on different datasets utilizing G-mean. In table, content with asterisk sign indicates improvement at ($p < 0.05$) Utilizing the Friedman test for SMOTEMultiBoost(SMB) and such techniques.

**Table 2**

| Data Set | CART | MultiBoost | SMOTE | BalanceCascad | EasyEnsemble | RUSBoost | SMOTEBoost | SMB |
|---|---|---|---|---|---|---|---|---|
| Hepatitis | 0.6001(8) | 0.6671(6) | 0.8221(4) | 0.7632(5) | 0.6000(7) | 0.9134(1) | 0.8932(3) | 0.8911(2) |
| Glass-0-1-6... | 0.4006(6) | 0.3734(7) | 0.9456(2) | 0.3488(8) | 0.4702(5) | 0.8137(4) | 0.9301(3) | 0.9702(1) |
| Glass1 | 0.7311(8) | 0.7734(7) | 0.9322(2) | 0.8267(5) | 0.7897(6) | 0.9288(4) | 0.9245(3) | 0.9810(1) |
| Ionosphere | 0.8823(8) | 0.9202(6) | 0.9522(3) | 0.9324(5) | 0.9176(7) | 0.9595(2) | 0.9511(4) | 0.9823(1) |
| Ecoli-0-1-4-7... | 0.8123(7) | 0.8325(6) | 0.9510(2) | 0.4712(8) | 0.8422(5) | 0.9113(4) | 0.9422(3) | 0.9802(1) |
| Wisconsin | 0.9527(8) | 0.9732(7) | 0.9932(2) | 0.9802(5) | 0.9724(6) | 0.9902(4) | 0.9920(3) | 0.9969(1) |
| Pima | 0.6699(8) | 0.6899(7) | 0.8900(4) | 0.7312(6) | 0.7298(5) | 0.8921(3) | 0.9001(2) | 0.9745(1) |
| Vehicle2 | 0.9421(8) | 0.9732(7) | 0.9917(3) | 0.9916(4) | 0.9809(6) | 0.9812(5) | 0.9897(2) | 0.9975(1) |
| Vehicle1 | 0.6531(7) | 0.6631(6) | 0.9211(3) | 0.5565(8) | 0.7233(5) | 0.8821(4) | 0.9304(2) | 0.9841(1) |
| Yeast1 | 0.6531(8) | 0.6516(7) | 0.8923(3) | 0.5601(6) | 0.6799(5) | 0.8714(4) | 0.9003(2) | 0.9700(1) |
| Phoneme | 0.8428(8) | 0.8693(7) | 0.9396(3) | 0.9122(5) | 0.8698(6) | 0.9410(4) | 0.9496(2) | 0.9719(1) |
| Satimage | 0.7256(7) | 0.7419(6) | 0.9714(2) | 0.4096(8) | 0.8119(5) | 0.9201(4) | 0.9710(3) | 0.9895(1) |
| Mammography | 0.0301(8) | 0.5000(5) | 0.7713(1) | 0.4619(7) | 0.4804(6) | 0.6813(4) | 0.7423(3) | 0.7600(2) |
| Average Rank | 7.70 | 6.51 | 2.70 | 6.87 | 6.11 | 3.64 | 2.74 | 1.27 |
| Friedman Test | *3.2210E-3 | *3.2210E-3 | *0.0033 | *3.1232E-3 | *3.1232E-3 | *0.0033 | *3.1232E-3 | Base |

Table 2 express G-mean quantities for all algorithms. It is showed that our proposed algorithm performs superior to all other contending algorithms. EasyEnsemble and BalanceCascade accomplishments are not significant as compared to sampling methods, particularly, oversampling. The degraded performance of these methods characterizes by uncommonness of smaller class cases in the datasets. Small class has few cases and by under sampling larger class makes its

cases small too. Additionally, the performance of SMOTE and SMOTEBoost is better as compare with the under sampling methods.

Likewise, AUC and F1-measure are expressed in Table 3, 4. Tables show that our proposed algorithm perform better as compare with the other algorithms on all datasets except Hepatitis and Mammography. Additionally, SMOTE and SMOTEBoost perform equally well having no significant difference.

**Table 3**

Result analysis of various methods with suggested SMOTEMultiBoost (SMB) on different datasets utilizing AUC. In table, content with asterisk sign indicates improvement at ($p < 0.05$) Utilizing the Friedman test for SMOTEMultiBoost (SMB) and such techniques.

**Table 3**

| Data Set | CART | MultiBoost | SMOTE | BalanceCascade | EasyEnsemble | RUSBoost | SMOTEBoost | SMB |
|---|---|---|---|---|---|---|---|---|
| Hepatitis | 0.7003(8) | 0.8268(6) | 0.9072(4) | 0.8540(5) | 0.8005(7) | 0.9361(2) | 0.9249(3) | 0.9796(1) |
| Glass-0-1-6... | 0.5999(8) | 0.7675(6) | 0.9519(2) | 0.7254(7) | 0.7859(5) | 0.8442(4) | 0.9462(3) | 0.9925(1) |
| Glass1 | 0.7406(8) | 0.8574(7) | 0.9562(2) | 0.9090(5) | 0.8637(6) | 0.9556(3) | 0.9421(4) | 0.9958(1) |
| Ionosphere | 0.8850(8) | 0.9604(6) | 0.9619(5) | 0.9815(2) | 0.9628(4) | 0.9750(3) | 0.9599(7) | 0.9965(1) |
| Ecoli-0-1-4-7... | 0.8262(8) | 0.9120(7) | 0.9679(2) | 0.9275(6) | 0.9244(5) | 0.9363(4) | 0.9551(3) | 0.9946(1) |
| Wisconsin | 0.9580(8) | 0.9897(7) | 0.9955(2) | 0.9939(3) | 0.9899(6) | 0.9921(5) | 0.9922(4) | 0.9992(1) |
| Pima | 0.6950(8) | 0.7947(6) | 0.9279(3) | 0.7270(7) | 0.7967 (5) | 0.9270(4) | 0.9313(2) | 0.9879(1) |
| Vehicle2 | 0.9536(8) | 0.9952(4) | 0.9951(5) | 0.9994(2) | 0.9957(3) | 0.9863(7) | 0.9943(6) | 0.9997(1) |
| Vehicle1 | 0.7096(8) | 0.8314(7) | 0.9481(2) | 0.8683(5) | 0.8415(6) | 0.9154(4) | 0.9470(3) | 0.9953(1) |
| Yeast1 | 0.6917(8) | 0.7683(6) | 0.9347(3) | 0.7780(5) | 0.7666(7) | 0.9091(4) | 0.9348(2) | 0.9913(1) |
| Phoneme | 0.8705(8) | 0.9504(6) | 0.9672(4) | 0.9676(3) | 0.9385(7) | 0.9653(5) | 0.9689(2) | 0.9968(1) |
| Satimage | 0.8278(8) | 0.9387(5) | 0.9758(2) | 0.9666(4) | 0.9385(6) | 0.9356(7) | 0.9727(3) | 0.9994(1) |
| Mammography | 0.5367(5) | 0.5036(7) | 0.8783(2) | 0.4889(8) | 0.5096(6) | 0.7567(4) | 0.8280(3) | 0.9100(1) |
| Average Rank | 7.76 | 6.15 | 2.92 | 4.76 | 5.76 | 4.30 | 3.46 | 1 |
| Friedman Test | * 3.1149IE- | * 3.1149IE-4 | * 3.114E-3 | * 3.1149E-3 | * 3.1149E-3 | * 3.1149E-3 | * 3.1149E-3 | Base |

**Table 4**

Result analysis of various methods with suggested SMOTEMultiBoost (SMB) on different datasets utilizing F1-measure. In table, content with asterisk sign indicates improvement at ($p < 0.05$) Utilizing the Friedman test for SMOTEMultiBoost (SMB) and such techniques.

**Table 4**

| Data Set | CART | MultiBoost | SMOTE | BalanceC | EasyEnsem | RUSBoost | SMOTE | SMB |
|---|---|---|---|---|---|---|---|---|
| Hepatitis | 0.4412(8) | 0.5324(7) | 0.8413(6) | 0.8923(4) | 0.8924(2) | 0.9145(1) | 0.8829(5) | 0.8920(3) |
| Glass-0-1- | 0.24219(7) | 0.2250(8) | 0.9560(2) | 0.9512(4) | 0.9522(3) | 0.8140(6) | 0.9423(5) | 0.9712(1) |
| Glass1 | 0.6619(8) | 0.7116(7) | 0.9278(2) | 0.8939(5) | 0.8550(6) | 0.9216(4) | 0.9429(3) | 0.9910(1) |
| Ionosphere | 0.8589(8) | 0.9012(7) | 0.9621(2) | 0.9613(5) | 0.9423(6) | 0.9634(4) | 0.9601(3) | 0.9876(1) |
| Ecoli-0-1-4- | 0.6909(8) | 0.7623(7) | 0.9312(4) | 0.9756(3) | 0.9832(2) | 0.9111(6) | 0.93005) | 0.9900(1) |
| Wisconsin | 0.9345(8) | 0.9499(7) | 0.9941(3) | 0.9809(5) | 0.9841(6) | 0.9799(4) | 0.9897(2) | 0.9899(1) |
| Pima | 0.5911(8) | 0.6231(7) | 0.9234(3) | 0.7897(5) | 0.7823(6) | 0.8796(4) | 0.9234(2) | 0.9752(1) |
| Vehicle2 | 0.9214(8) | 0.9645(7) | 0.9899(4) | 0.9898(3) | 0.9845(5) | 0.9722(6) | 0.9945(2) | 0.9981(1) |
| Vehicle1 | 0.5145(8) | 0.5421(7) | 0.9515(3) | 0.8741(5) | 0.8612(6) | 0.8734(4) | 0.9513(2) | 0.9995(1) |
| Yeast1 | 0.5261(8) | 0.5425(7) | 0.9134(3) | 0.8501(5) | 0.8213(6) | 0.8634(4) | 0.9212(2) | 0.9800(1) |
| Phoneme | 0.7819(8) | 0.8312(7) | 0.9623(3) | 0.9432(4) | 0.9001(6) | 0.9546(5) | 0.9713(2) | 0.9901(1) |
| Satimage | 0.5723(8) | 0.6435(7) | 0.9814(2) | 0.9604(5) | 0.9723(4) | 0.9228(6) | 0.9702(3) | 0.9950(1) |
| Mammogra | 0.0122(7) | 0.0128(8) | 0.8723(1) | 0.4956(6) | 0.6754(4) | 0.6529(5) | 0.8270(2) | 0.7345(3) |
| Average | 8.23 | 7.70 | 3.45 | 5.01 | 5.25 | 5.08 | 3.35 | 1.80 |
| Friedman Test | * 3.2172E-3 | * 3.2172E-3 | * 0.0032 | * 3.2172E-3 | * 0.0032 | * 0.0032 | * 0.0032 | Base |

Likewise, AUC and F1-measure are delineated in Table 3, 4. It is evident from the tables that our algorithm perform better to every other method with the exception of Hepatitis and Mammography datasets. Moreover, SMOTE and SMOTEBoost are contending algorithms with equal performance on given

measurements.

Moreover, Figure 1 portrays ROC curves created by various methods including our method on thirteen datasets. CART, EasyEnsemble and BalanceCascade showed weak performance over mammography dataset. Our proposed method perform superior as compare to other algorithms on all datasets. On each dataset, SMOTE, RUSBoost and SOTEBoost outcomes are better to BalanceCascade and EasyEnsemble. SMOTE and SMOTEBoost are contending methods taken after RUSBoost. Larger class showed degraded performance on resampling methods. Yet, the results of our method in ROC space as compare to other methods demonstrates that acquired power of wagging, boosting and SMOTE decrease bias and variance.

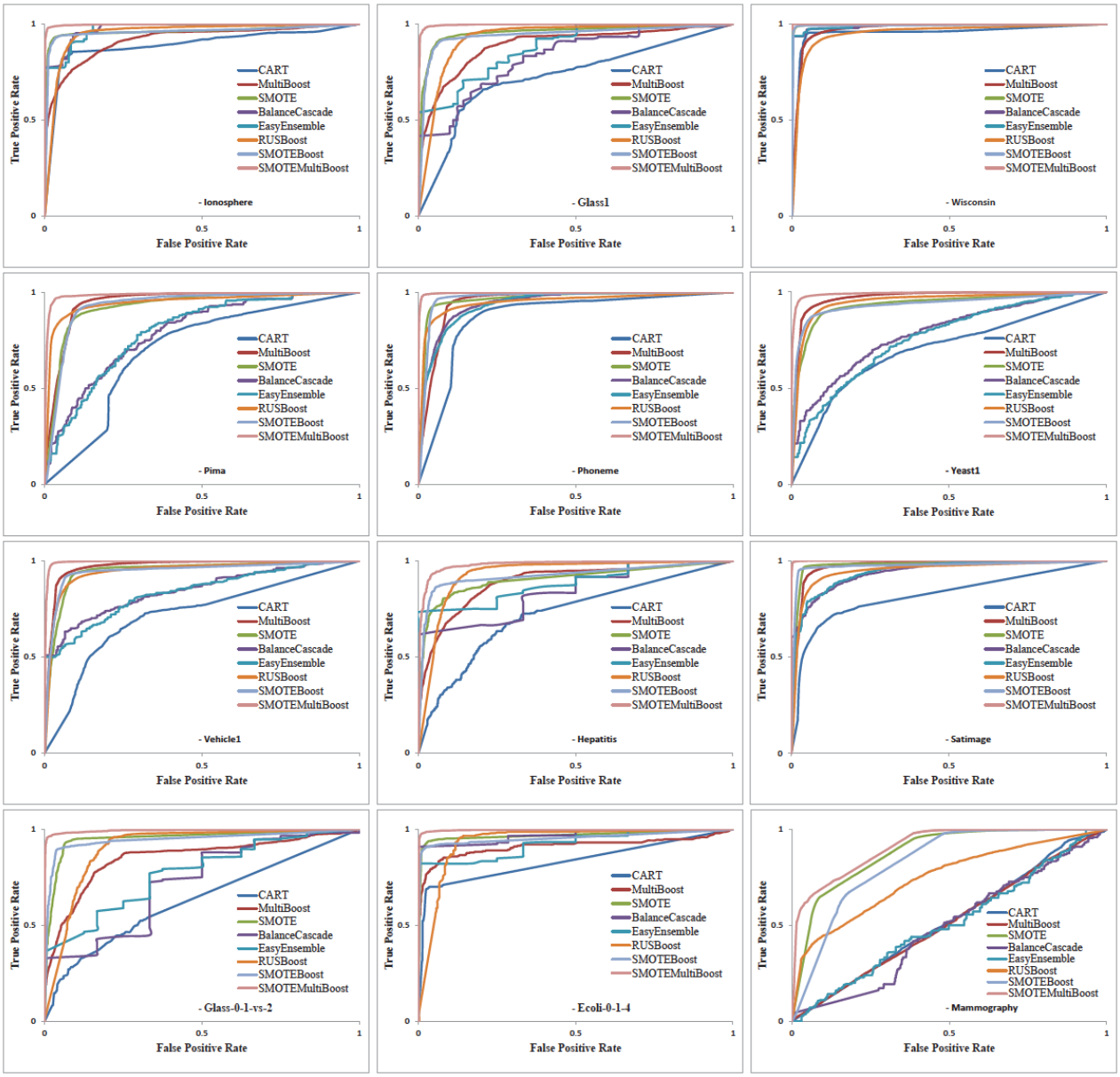


**Figure 1**. Comparison of CART, MultiBoost, SMOTE, BalanceCascade, EasyEnsemble, RUSBoost and SMOTEBoost (SMB) for all datasets. SMOTEMultiBoost (SMB) dominates in ROC space.

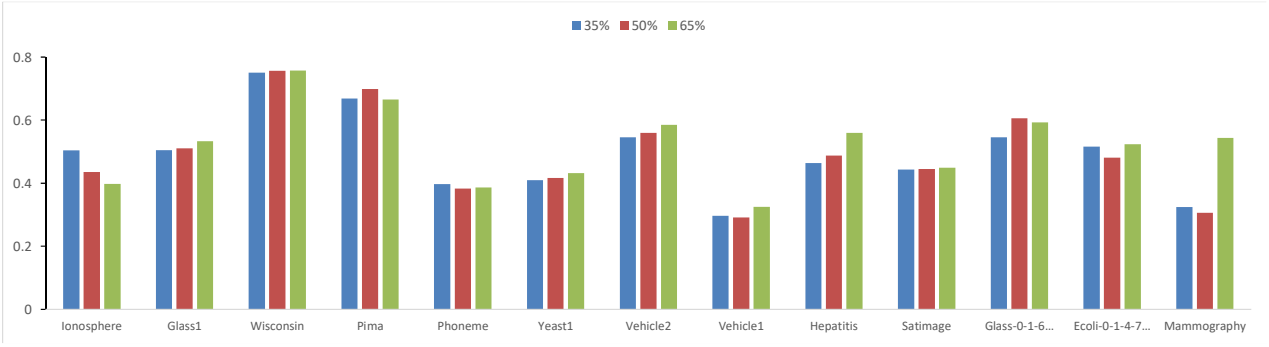


**Figure 2**. Comparison of different Q-statistic values with respect to various class distributions (35%, 50% and 65%) on all data sets.

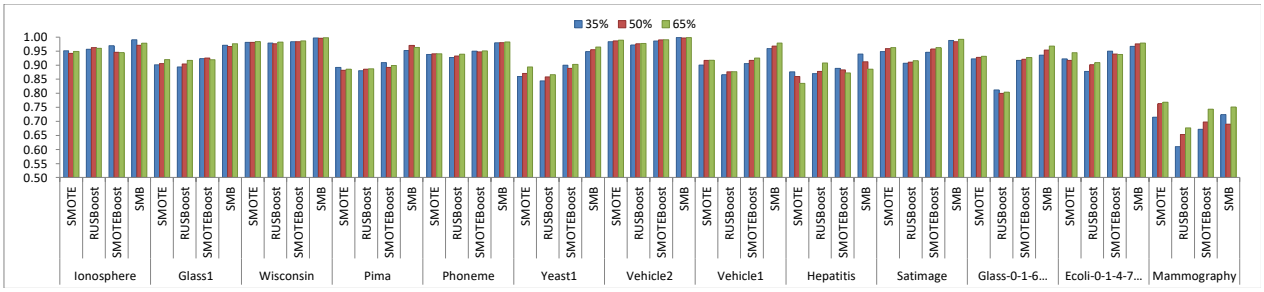


**Figure 3.** Comparison of different smaller class ratios (35%, 50% and 65%) on all data sets. The varying values of G-mean measure with respect to all methods including our SMOTEMultiBoost (SMB) are shown.

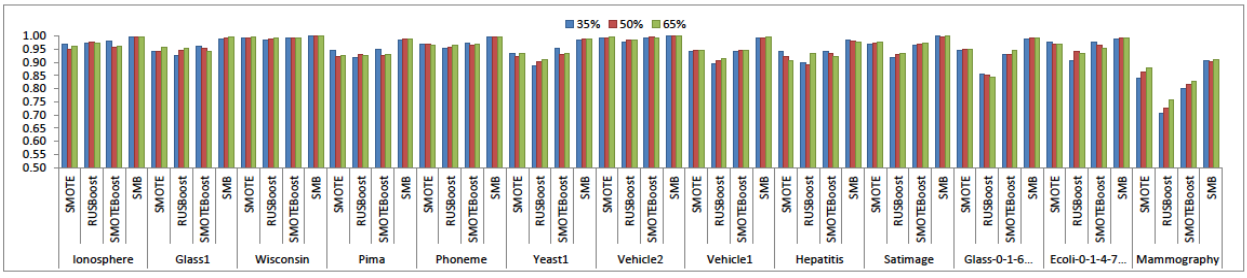


**Figure 4**. Comparison of different smaller class ratios (35%, 50% and 65%) on all data sets. The varying values of AUC measure with respect to all methods including our SMOTEMultiBoost (SMB) are shown.

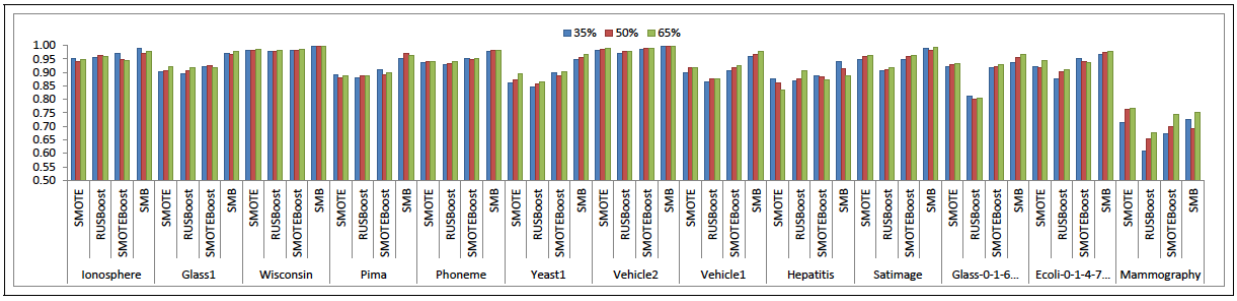


**Figure 5**. Comparison of different smaller class ratios (35%, 50% and 65%) on all data sets. The varying values of F1 measure with respect to all methods including our SMOTEMultiBoost (SMB) are shown.

Experiments are conducted using various methods including our method on thirteen datasets with three distributions (35%, 50%, 65%). The Figure 2 shows Q-statistic with respect to different class distributions for all datasets. It is apparent from the figure that there is no clear trend of Q-statistic with varying class distributions. Similarly, the Figures 3, 4, 5 show the measures used in our experiments with respect to different class distributions. All three distributions have no significant difference and they equally performed.

Nevertheless, it is obvious that appropriate distribution is the characteristic of the specific dataset. An intriguing statement to say is that the values of all metrics of the RUSBoost are superior to EasyEnsemble and BalanceCascade. The main reason of this degraded performance of EasyEnsemble and Balance-Cascade is due to loss of valuable information due to under sampling and duplication of smaller class cases create reduced diversity.

## CONCLUSION

Class imbalance problem is explored and developed effective and concrete method based on hybrid ensemble learning. Our method has quality to recognize the smaller class cases effectively while asserting the high performance for larger class cases. We presented a composite ensemble learning algorithm which adapt the MultiBoost ensemble for better overall performance and SMOTE oversampling technique for better smaller class performance. Our new hybrid algorithm not only reduce bias and variance in error by increasing the diversity but also enhance the results of smaller class, significantly. Experimental results on different commonly used datasets show that our proposed hybrid ensemble algorithm achieved better performance using G-mean, F1-measure, AUC and ROC space. We also conducted experiments with different rate of imbalance in datasets and found that all results are almost equal. Additionally, our algorithm also suitable for parallel execution. We try to implement the proposed algorithm for parallel execution for large datasets. In future work, we will additionally explore viability of our method and attempt to actualize the method for parallel execution for substantial datasets.

## REFERENCES


1. Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S.-S. Ho, "ForesTexter: an efficient random forest algorithm for imbalanced text categorization," Knowledge-Based Systems, vol. 67, pp. 105-116, 2014.
2. B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms," Information Processing & Management, vol. 56, no. 1, pp. 212-227, 2019.
3. Y. Zhong, H. Yang, Y. Zhang, and P. Li, "Online random forests regression with memories," Knowledge-Based Systems, vol. 201, p. 106058, 2020.
4. H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing, 2005: Springer, pp. 878-887.
5. J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," Journal of Big Data, vol. 6, no. 1, pp. 1-54, 2019.
6. S. Maldonado, J. López, and C. Vairetti, "An alternative SMOTE oversampling strategy for high-dimensional datasets," Applied Soft Computing, vol. 76, pp. 380-389, 2019.
7. M. Koziarski, B. Krawczyk, and M. Woźniak, "Radial-Based oversampling for noisy imbalanced data classification," Neurocomputing, vol. 343, pp. 19-33, 2019.
8. B. Schölkopf, A. J. Smola, and F. Bach, Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.
9. J. R. Quinlan, C4. 5: programs for machine learning. Elsevier, 2014.
10. A. Joshuva, R. S. Kumar, S. Sivakumar, G. Deenadayalan, and R. Vishnuvardhan, "An insight on VMD for diagnosing wind turbine blade faults using C4. 5 as feature selection and discriminating through multilayer perceptron," Alexandria Engineering Journal, vol. 59, no. 5, pp. 3863-3879, 2020.
11. A. A. Nagra et al., "Hybrid self-inertia weight adaptive particle swarm optimisation with local search using C4. 5 decision tree classifier for feature selection problems," Connection Science, vol. 32, no. 1, pp. 16-36, 2020.
12. A. Cherfi, K. Nouira, and A. Ferchichi, "MC4. 5 decision tree algorithm: an improved use of continuous attributes," International Journal of Computational Intelligence Studies, vol. 9, no. 1-2, pp. 4-17, 2020.
13. H. Zhang, L. Jiang, and L. Yu, "Attribute and instance weighted naive Bayes," Pattern Recognition, vol. 111, p. 107674, 2021.
14. L. Yu, S. Gan, Y. Chen, and M. He, "Correlation-based weight adjusted Naïve Bayes," IEEE Access, vol. 8, pp. 51377-51387, 2020.
15. P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in Aaai, 1992, vol. 90: Citeseer, pp. 223-228.

16. R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," Statistical science, vol. 17, no. 3, pp. 235-255, 2002.
17. L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123-140, 1996.
18. Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in icml, 1996, vol. 96: Citeseer, pp. 148-156.
19. M. Afifi and A. Abdelhamed, "AFIF4: deep gender classification based on adaboost-based fusion of isolated facial features and foggy faces," Journal of Visual Communication and Image Representation, vol. 62, pp. 77-86, 2019.
20. G. Biau, B. Cadre, and L. Rouvière, "Accelerated gradient boosting," Machine Learning, vol. 108, no. 6, pp. 971-992, 2019.
21. J. R. B. Junior and M. do Carmo Nicoletti, "An iterative boosting-based ensemble for streaming data classification," Information Fusion, vol. 45, pp. 66-78, 2019.
22. N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," ACM SIGKDD explorations newsletter, vol. 6, no. 1, pp. 1-6, 2004.
23. 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.
24. U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," Information Sciences, vol. 479, pp. 448-455, 2019.
25. A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," Pattern Recognition, vol. 91, pp. 216-231, 2019.
26. S. Visa and A. Ralescu, "Issues in mining imbalanced data sets-a review paper," in Proceedings of the sixteen midwest artificial intelligence and cognitive science conference, 2005, vol. 2005: sn, pp. 67-73.
27. H. Patel, D. Singh Rajput, G. Thippa Reddy, C. Iwendi, A. Kashif Bashir, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 16, no. 4, p. 1550147720916404, 2020.
28. J. Hamidzadeh, N. Kashefi, and M. Moradi, "Combined weighted multi-objective optimizer for instance reduction in two-class imbalanced data problem," Engineering Applications of Artificial Intelligence, vol. 90, p. 103500, 2020.
29. L. Pelayo and S. Dick, "Synthetic minority oversampling for function approximation problems," International Journal of Intelligent Systems, vol. 34, no. 11, pp. 2741-2768, 2019.
30. N. Japkowicz, "Class imbalances: are we focusing on the right issue," in Workshop on Learning from Imbalanced Data Sets II, 2003, vol. 1723, p. 63.
31. V. García, J. Sánchez, A. Marqués, R. Florencia, and G. Rivera, "Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data," Expert Systems with Applications, vol. 158, p. 113026, 2020.
32. R. O'Brien and H. Ishwaran, "A random forests quantile classifier for class imbalanced data," Pattern recognition, vol. 90, pp. 232-249, 2019.
33. D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," Information Sciences, vol. 505, pp. 32-64, 2019.
34. S. Liu, G. Lin, Q.-L. Han, S. Wen, J. Zhang, and Y. Xiang, "DeepBalance: Deep-learning and fuzzy oversampling for vulnerability detection," IEEE Transactions on Fuzzy Systems, vol. 28, no. 7, pp. 1329-1343, 2019.
35. R. C. Prati, G. E. Batista, and M. C. Monard, "Class imbalances versus class overlapping: an analysis of a learning system behavior," in Mexican international conference on artificial intelligence, 2004: Springer, pp. 312-321.
36. K. J. Ezawa, M. Singh, and S. W. Norton, "Learning goal oriented Bayesian networks for telecommunications risk management," in ICML, 1996, pp. 139-147.
37. Y. Wang and L. Yang, "A robust loss function for classification with imbalanced datasets," Neurocomputing, vol. 331, pp. 40-49, 2019.
38. R. M. Valdovinos and J. S. Sánchez, "Class-dependant resampling for medical applications," in Fourth International Conference on Machine Learning and Applications (ICMLA'05), 2005: IEEE, p. 6 pp.
39. M. Hosni, de Gea, J. M. C., Idri, A., El Bajta, M., Alemán, J. L. F., García-Mateos, G., & Abnane, I. , "A systematic mapping study for ensemble classification methods in cardiovascular disease.," in Artificial Intelligence Review, ed, 2020.
40. T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," IEEE transactions on software engineering, vol. 33, no. 1, pp. 2-13, 2006.
41. S. S. Rathore and S. Kumar, "A study on software fault prediction techniques," Artificial Intelligence Review, vol. 51, no. 2, pp. 255-327, 2019.
42. H. Turabieh, M. Mafarja, and X. Li, "Iterated feature selection algorithms with layered recurrent neural network for software fault prediction," Expert systems with applications, vol. 122, pp. 27-42, 2019.
43. X. Shi, Y. Li, and P. Yu, "Collective prediction with latent graphs," in Proceedings of the 20th ACM international conference on Information and knowledge management, 2011, pp. 1127-1136.
44. 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, 2003: Springer, pp. 107-119.
45. 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.
46. 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, 2009.
47. X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 2, pp. 539-550, 2008.
48. J. Błaszczyński, M. Deckert, J. Stefanowski, and S. Wilk, "Integrating selective pre-processing of imbalanced data with ivotes ensemble," in International conference on rough sets and current trends in computing, 2010: Springer, pp. 148-157.
49. G. Mustafa, Z. Niu, and J. Chen, "Adapting MultiBoost ensemble for class imbalanced learning," in 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), 2015: IEEE, pp. 12-17.
50. G. Mustafa, Z. Niu, A. Yousif, and J. Tarus, "Distribution based ensemble for class imbalance learning," in Fifth International Conference on the Innovative Computing Technology (INTECH 2015), 2015: IEEE, pp. 5-10.
51. S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in 2009 IEEE symposium on computational intelligence and data mining, 2009: IEEE, pp. 324-331.
52. G. I. Webb, "Multiboosting: A technique for combining boosting and wagging," Machine learning, vol. 40, no. 2, pp. 159-196, 2000.
53. G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," Journal of artificial intelligence research, vol. 19, pp. 315-354, 2003.
54. 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.
55. R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in European conference on machine learning, 2004: Springer, pp. 39-50.
56. S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE--majority weighted minority oversampling technique for imbalanced data set learning," IEEE Transactions on knowledge and data engineering, vol. 26, no. 2, pp. 405-425, 2012.
57. Tomek, "Two modifications of CNN," 1976.M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in Icml, 1997, vol. 97: Citeseer, pp. 179-186.
58. S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," Evolutionary computation, vol. 17, no. 3, pp. 275-306, 2009.
59. D. Mease, A. J. Wyner, and A. Buja, "Boosted classification trees and class probability/quantile estimation," Journal of Machine Learning Research, vol. 8, no. 3, 2007.
60. H. Yu, J. Ni, and J. Zhao, "ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data," Neurocomputing, vol. 101, pp. 309-318, 2013.
61. B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in Third IEEE international conference on data mining, 2003: IEEE, pp. 435-442.
62. K. M. Ting, "An instance-weighting method to induce cost-sensitive trees," IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 3, pp. 659-665, 2002.
63. P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 1999, pp. 155-164.
64. B. Das, N. C. Krishnan, and D. J. Cook, "RACOG and wRACOG: Two probabilistic oversampling techniques," IEEE transactions on knowledge and data engineering, vol. 27, no. 1, pp. 222-234, 2014.
65. 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.
66. J. Błaszczyński and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," Neurocomputing, vol. 150, pp. 529-542, 2015.
67. S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 2, no. 5-6, pp. 412-426, 2009.
68. C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Resampling or reweighting: A comparison of boosting implementations," in 2008 20th IEEE International Conference on Tools with Artificial Intelligence, 2008, vol. 1: IEEE, pp. 445-451.
69. 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 & Soft Computing, vol. 17, 2011.
70. 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.
71. C. Li, "Classifying imbalanced data using a bagging ensemble variation (BEV)," in Proceedings of the 45th annual southeast regional conference, 2007, pp. 203-208.