Surface-to-Volume Regularized Forests for Imbalanced Data

1 Introduction

"Learning from imbalanced data" is the specialized area of machine learning which is open for research exploration in resolving imbalanced data-set problems and fascinates a lot of attention in real-life necessity. A data set is considered "imbalanced" if one class (the majority class) vastly outnumbers the other (minority class) [14] in the training data. Modeling an imbalanced dataset problem is one of the major challenges in the classification process due to prejudice towards the majority class. Most of the datasets are treated to have balanced class distribution as most algorithms fail to recognize the distributive characteristics of the data[13].

Motivation:

Various solutions proposed ranged from a new sampling approach to a new learning algorithm. The sampling approach consists of two main categories which are oversampling and under-sampling. Oversampling means higher weight or cost for minority instances and increased computational cost while under-sampling reduces the cost of learning rate by providing compact balanced training sets[10]. Useful data are lost due to under-sampling, when the ratio of imbalance is high and more data is removed then it causes a lack of data[8]. Besides, SMOTE follows a different technique that creates synthetic samples for each minority class.

To tackle the curse of data imbalance there are several methods such as "Data Based Methods"- manipulate the input data to balance the skewness of different class distributions using various sampling techniques e.g.- SMOTE and "Algorithm Based Methods"- modify the training procedure to get better result for the minority class using cost-sensitive and ensemble techniques e.g.- surface to volume regularization. Data-level approaches towards addressing class imbalance follow the notion of firstly balancing the imbalanced data using techniques such as oversampling or undersampling and thereafter using classical algorithms for predictive analysis. Algorithm-level approaches on the other hand seek to develop algorithms that are able to withstand the data imbalance while not relying upon any alteration to the original class distribution of the data. Algorithm-level approaches can be further categorized into two groups: cost-sensitive algorithms, which uses different weights for each class i.e. different amount of misclassification penalty for each class, and the ensemble method algorithms, which implements learning from more than one classifier at once and average the results in the end to produce more robust predictions.

Classification trees (Breiman et al. 1984) with their extension, random forests (Breiman 2001a), and boosting have become very popular and effective methods for flexibly estimating relations in settings where out-of-sample predictive power is important. They are considered to have great out-of-the-box performance without

requiring subtle tuning. Classification trees are favored for their combined flexibility and interpretability. Again, we build up a tree by splitting a feature of the sample into two leaves, based on information gain related to a single impurity function not exceeding a threshold value. We optimize the split over the choice of feature space and splitting points. If all units in that leaf have the same label, the leaf is then pure having maximum information gain. The regularization typically works through a penalty term on the number of leaves in the tree.

Random Forest, one of the most widely used tree ensemble techniques, uses a set of decision trees to partition the data into random subspaces and learn decision mapping for them to produce a prediction averaged over the individual predictions of the trees in the ensemble. A key issue that random forests address is that the estimated regression function given a tree is discontinuous with substantial jumps. Random forests induce smoothness by averaging over a large number of trees that differ from each other for the bootstrapped sample. In boosting considering the base learners as trees, the combination of the trees does not grow independently, it is updated by the prediction error of its previous tree. Boosting faces the problem of over-fitting where as a random forest with numerous no of trees can not be over-fitted.

Contribution:

Imbalanced data-oriented algorithms are used for imbalanced class problems that deal with class imbalance without modifying the class distributions. SVR Tree comes under this algorithmic approach. The SVR Tree algorithm does not oversample or undersample the input dataset, instead, it penalizes the Surface-to-Volume Ratio of the decision set to transform the irregularly shaped boundaries to regularly shaped boundaries to defend from over-fitting. Hence, this can also be called as Surface-to-Volume Regularization technique. A tree is grown by sequentially splitting its internal nodes using the impurity and then pruning the grown tree back to avoid overfitting. Instead of a single SVR Tree, an ensemble of various SVR Trees is used wherein each tree has been built using bootstrap samples from a dataset. These mechanisms are versatile enough to deal with supervised classification problems. While individual trees can be restricted in their expressiveness due to using only axis-parallel splits, this shortcoming can be diminished by using an ensemble of decision trees as they have demonstrated statistically significant improvements over a single decision tree classifier. In this paper, SVR Forests will be introduced and these algorithms will also be compared with other states of art techniques. The proficiency of this technique will be measured by competing it with the existing algorithms.

The remaining article is organized as follows. In Section 2 we review related works on class imbalance learning and Surface-TO-Volume Ratio. Then, we describe the formulation of the proposed ensemble models for handling class imbalance problems and isometrics for different splitting criteria in Section 3. Section 4 discusses baseline models, datasets, and metrics to compare our proposed models. Section 5 reports the results of the experiments on datasets. Finally, we conclude this article with a discussion of its future research scope in Section 6.

2 Related Studies

2.1 Ensemble Methods:

An **ensemble method** is required to obtain a single very powerful and potential model combining many simple "building blocks" or "weak learners". **Bagging [2],**

[3] and Boosting[11] are very popular ensemble techniques for which the simple building block is a regression or a classification tree [4] to improve their mediocre predictions.

The bagging algorithm is a combination of many decision trees based on aggregation. In this procedure, the underlying predictors are built from bootstrapped samples or "bags" of the original data set randomly taking the training data with replacement(WR) and making use of majority vote to predict. Whereas boosting is a committee of decision trees evolving over time and the members cast a weighted vote. Boosting appears to dominate bagging on most problems due to the learning strategy from the previously grown tree. Traditional ensemble methods are constructed from a bunch of CART splitting the tree at a node, from features that have the highest information gain based on impurity function (Gini impurity or entropy). The results of decision trees fitted on the two halves of the training set are quite different. Due to the suffering from high variance, ensemble methods play a great role in reducing the variance of a statistical learning method by some particular type of averaging the decision trees. Bagging stabilizes at about 200 trees, while at 1000 trees boosting may continue to improve and slow down by the shrinkage. But it can also happen that boosting of depth 4 may have a smaller error than the stronger bagging of depth 10. The statistical view of boosting holds that it is a stage-wise optimization of an exponential loss function. One of the prime elements for ensuring the randomness of the split for the individual decision trees in the ensemble is the splitting criterion of the random forest. The original Gini splitting criterion has been found to deliver robust predictive performance for classification as well as regression in the case of balanced datasets, but for imbalanced datasets, a number of experiments have concluded that the Gini splitting criterion tends to perform poorly for them on account of its skew-sensitivity. This realization is especially fruitful leading to performing probability estimation and regression as well as adapting to different loss functions.

2.2 Surface to Volume Ratio(SVR) Trees:

Sometimes data are limited for one or more of the classes, and the estimated decision boundaries are often irregularly shaped due to the limited sample size, leading to poor generalization error. SVR Tree algorithm[21] does not over-sample or under-sample the input data set. In that case, penalizing the Surface-to-Volume Ratio of the decision set for the construction of tree classifiers is directly applied. By penalizing the SVR of the decision set, the regularly shaped decision sets much less subject to such over-fitting are favored. SVR regularization can be efficiently implemented for tree classifiers by proposing an algorithm with similar computational complexity to standard tree algorithms such as CART.

2.2.1 Notation:

The basic setting of the data is a set of training data $\mathcal{P}_n = \{(X_i, Y_i)\}_{i=1}^n$ with $X_i \in \Omega \subset \mathbb{R}^p$ a vector of features and $Y_i = m \in \{0, \ldots, M-1\}$ a class label which is output by estimating a classifier $f : \Omega \to \{0, \ldots, M-1\}$. Without loss of generality, suppose there are only two classes i.e. $M = 2$ and let us assume $m = 0$ is the majority class, $m = 1$ is the minority class, set $\{x \in \Omega : f(x) = 1\}$ is the decision set (of the minority class) and $n_1 = \sum_{i=1}^n \mathbf{1}_{\{Y_i=1\}}$ is relatively small compared to n_0 where $n_0 = \sum_{i=1}^n \mathbf{1}_{\{Y_i=0\}}$.

The jth feature of X is denoted by as $X[j]$. Let $\mathbb{P}$ denote the true distribution

of i.i.d. samples (X_i, Y_i) and let $\mathbb{P}^*$ denote the weighted distribution that up weights the minority class samples by a constant $\lambda \geq 1$ which is a standard technique to deal with the imbalance. For any measurable subset $C \subset \Omega$,

$$\mathbb{P}^*(C \times \{1\}) = \frac{\lambda \mathbb{P}(C \times \{1\})}{\mathbb{P}(\Omega \times \{0\}) + \lambda \mathbb{P}(\Omega \times \{1\})},$$

$$\mathbb{P}^*(C \times \{0\}) = \frac{\mathbb{P}(C \times \{0\})}{\mathbb{P}(\Omega \times \{0\}) + \lambda \mathbb{P}(\Omega \times \{1\})}.$$

This assigns mass $w_0\lambda/n$ to the minority class and w_0/n to the majority class to ensure the measures add up to 1.

2.2.2 Surface-to-Volume Ratio:

First, we will introduce the definition of surface-to-volume ratio (SVR) and tree impurity, and then define SVR-Tree as the minimizer of a weighted average of tree impurity and SVR adjusting with a penalty term.

Definition of Volume: For any p-dimensional Lebesgue measurable closed set (which is a subset of the input feature space), its volume is defined as the Lebesgue measure of set C: $\mathbb{V}(C) = \gamma(\partial C)$.

Definition of Surface: Surface of C is defined as $p-1$ dimensional Lebesgue measure of the boundary of C after representing C as a finite union of hyper rectangles: $\mathbb{S}(C) = \gamma_{n-1}(C)$.

If there exists $C_i, i \geq 1$ multiple Lebesgue measurable subsets within the feature space such that each subset is a finite union of hyper rectangles, all those subsets converge to C in Hausdorff distance. Now the surface of C can be expressed as: $\mathbb{S}(C) = \lim_{i \to \infty} \mathbb{S}(C_i)$, provided the limit exists. Moreover, it always exists for regular shaped subsets.

Definition Surface-to-Volume Ratio: For any set C with $0 < \gamma(c) < \infty$, the surface-to-volume ratio(SVR) can be defined as $\mathbb{R}(C) = \frac{\mathbb{S}(C)}{\mathbb{V}(C)}$. If the decision sets with same volume having multiple disconnected subsets and/or irregular boundaries have relatively high SVR whereas decision sets with regular shaped borders possess low SVR.

2.2.3 Tree Impurity Function:

A classification tree partitions the sample space into multiple leaf nodes, assigning one class label to each leaf node and the grown is in such a way to maximize homogeneity of the training sample class labels within nodes. Here, Gini Impurity $\psi(p_0, p_1) = 1 - p_0^2 - p_1^2$ is used as 'impurity function'. Let $C_1, C_2, \ldots, C_l$ be the leaf nodes of $\mathbb{T}$ and $y_j \in \{0, 1\}$ be the predictive class label for node C_j, $\forall j$. The impurity of leaf node C_j is:

$$Imp(C_j, \mathbb{P}^*) = \psi\left(\mathbb{P}^*(Y = 0 \mid X \in C_j), \mathbb{P}^*(Y = 1 \mid X \in C_j)\right)$$

under the weighted probability measure. As the **impurity of leaf node** C_j does not depend on the predictive class label y_j, a new function $\tilde{y}_j = \mathtt{I}_{\{\mathbb{P}^*(Y=1|X \in C_j) \geq 1/2\}}$ is introduced. The **signed impurity** takes into account y_j as:

$$\tilde{Imp}(Cj, \mathbb{P}^*) = \mathtt{I}_{\{y_j = \tilde{y}_j\}} Imp(C_j, \mathbb{P}^*) + \mathtt{I}_{\{y_j \neq \tilde{y}_j\}} (1 - Imp(C_j, \mathbb{P}^*))$$

In other words, the signed impurity uses quadratic loss instead of the absolute value loss function of classification accuracy.

2.2.4 SVR Tree Classifiers:

A tree classifier is formed by finite splitting criteria of features and its decision sets are a finite union of hyper-rectangles. A classification tree $\mathbb{T}$ divides the closed bounded sample space(Ω) into two disjoint subsets, one for minority class(Ω_1) and the other one for majority class(Ω_0). The surface-to-volume ratio of a classification tree is defined as the SVR of the set $\Omega_1 : \mathbb{R}(\mathbb{T}) = \mathbb{R}(\Omega_1)$.

Tree Impurity: The tree impurity is defined as:

$$Imp(\mathbb{T}, \mathbb{P}^*) = \sum_{j=1}^{l} \mathbb{P}(X \in C_j)\, Imp(C_j, \mathbb{P}^*)$$

.

Signed Tree Impurity: The signed tree impurity is:

$$\tilde{Imp}(\mathbb{T}, \mathbb{P}^*) = \sum_{j=1}^{l} \mathbb{P}^*(X \in C_j)\, \tilde{Imp}(C_j, \mathbb{P}^*)$$

.

The objective of an SVR Tree is to minimize the signed tree impurity and surface-to-volume ratio simultaneously. Therefore, an SVR Tree classifier acts as a minimizer of the weighted average and is given by:

$$\hat{\mathbb{T}} = \underset{\mathbb{T}\in\mathcal{T}}{\operatorname{argmin}} \left[\tilde{Imp}(\mathbb{T}, \mathbb{P}_n^*) + \eta_n \mathbb{R}(\mathbb{T})\right], \tag{1}$$

where $\mathcal{T}$ be the collection of possible trees for the training data and η_n is a specified penalty term.

2.2.5 Algorithm:

1. Splitting the current leaf node: For each feature, sort all samples and allow splits to occur at median of each sorting feature. After each such split we keep all other leaf nodes unchanged. The current set of trees to choose from in optimizing $\hat{\mathbb{T}}$ includes the initial $\mathbb{T}$ and all the split trees described above.

2. Feature Selection: Let, the current tree is T. We are splitting node C into two new leaf nodes C_1, C_2. The tree impurity decreases after this split is defined as:

$$\Delta Imp(\mathbb{T}, \mathbb{P}_n^*) = \mathbb{P}_n^*(C)\left[Imp(C, \mathbb{P}_n^*) - Imp(C_1, \mathbb{P}^*)\, \mathbb{P}_n^*(C_1 \mid C) - Imp(C_2, \mathbb{P}^*)\, \mathbb{P}_n^*(C_2 \mid C)\right]$$

Choose the feature which has the maximal tree impurity decrease overall splits in feature.

3. Use "Breadth-First" searching order to find which node to split in each step. After splitting, the tree is updated. The process continues up to when the further splitting of leaf nodes either does not improve the impurity or a prespecified maximum number of leaf nodes is achieved.

4. To predict the test data we start from the root node and follow the sequential branch decompositions. At the leaves, we calculate the probability of class labels and

assign the class label which has the highest probability. The probabilities of each class label at leaves are determined by sum of the indicator function by the number of points in the cell.

The SVR Tree grows with the best possible penalty term that is suitable for the current training input sample.

3 Proposed Ensemble Models:

In this paper, we have introduced two ensemble methods that use the Bagging and Boosting techniques for which the weak learners are SVR Tree. In this context we are concerned about only the classification of imbalanced data that is the number of minority samples is less than 30-20%. CART is biased towards the majority class for imbalanced data due to the skew-sensitiveness of the impurity function. To address biases and high variance of the SVR tree, several new approaches are implemented including Bagging and Boosting.

3.1 Proposed Bagged Regularized Decision Trees (BaRDT):

Bagging involves creating multiple copies of the original training data set applying the bootstrap, fitting a separate SVR tree to each copy, and then combining all of the trees in order to create a single predictive forest as a model.

3.1.1 Methodology:

In our bootstrap aggregation or bagging version, we have constructed a classifier from the given training data(the input). For any given training set $\mathcal{P}$, the proposed model forms $\mathcal{B}_h, h = 1, \ldots, H$ different bootstrap training samples by taking repeated samples from the (single) training data set where H is a parameter to our model that defines the number of trees in a forest. We then train SVR Tree Classifiers on each and every h bootstrapped training set in order to fetch the votes to form a bagged predictor. Let using each $\mathcal{B}_h$ we obtain $\hat{f}_p^1, \hat{f}_p^2, \ldots, \hat{f}_p^H$, SVR de-correlating trees. Finally, for a given test observation(x_{test}), this outputs the class predicted by each of the $\mathcal{B}_h$ trees and takes a majority vote: The overall prediction is the most commonly occurring class among the B predictions.

$$\hat{f}_{\max}(x_{test}) = \underset{h \subset H}{\operatorname{argmax}} \hat{f}_p^h(x_{test}) \tag{2}$$

Algorithm 1 shows a schematic of the Bagged Regularized Decision Trees (BaRDT) procedure. The first three lines after the for loop form the bootstrapped data with a replacement of the same size as the input training data. Then, SVR trees are fitted for "no of trees" times. For the predicting purpose test data points are predicted through all of these fitted trees resulting in each data point having "no of trees" predicted class label. Finally, taking the majority vote the predicted class label is decided.

3.2 Proposed Boosted Regularized Decision Trees (BoRDT):

Even though the learning process is highly randomized, the bootstrapped sub-samples imitate the distribution of the original data-set, if the original data set is imbalanced. Moreover, some important observations may not appear in the sample selection.

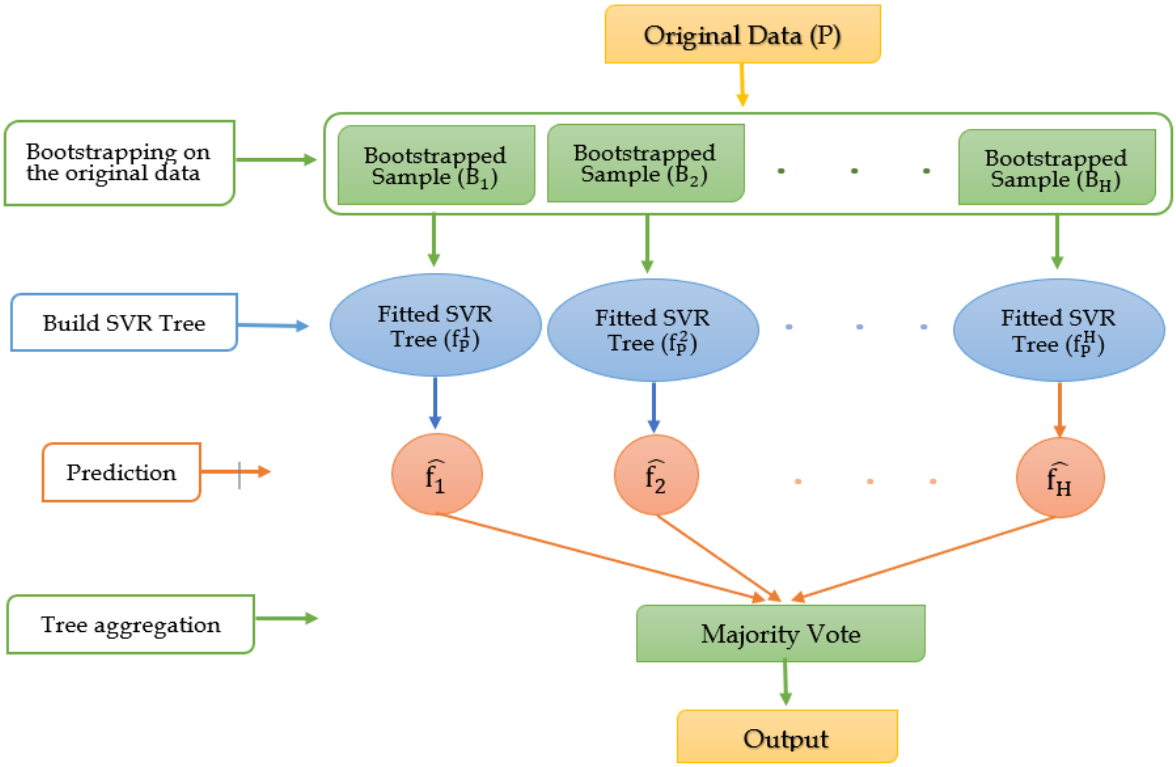


Figure 1: Flowchart of Proposed Bagging (BaRDT) Algorithm

Algorithm 1: Outline Steps of Bagged Regularized Decision Trees (BaRDT)

Data: Training Data, Testing data, no of trees(H), weight for minority class, maximal no of leaf nodes, penalty term.

Result: Output the predicted class labels for testing data.

Train Procedure:

for $1 \leq i \leq$ *no of trees* **do**

$n \leftarrow$ no of samples in the training set;

Sample $\leftarrow$ Choose random number from 1 to n of size n with replacement;

Bootstrapped samples $\leftarrow$ Select those samples from original data which have the same index no as "Sample" ;

Fit the SVR tree with these "Bootstrapped samples", penalty term and weight;

end

Test Procedure:

Predict the testing data from each of the trees ;

Fix the final class label which has the highest frequency.

Boosting is our second ensemble approach for improving the predictions resulting from an SVR tree classifier. In Boosting the trees are grown sequentially: each tree is grown taking the information from previously grown trees. Instead of involving bootstrap training samples trees are fit on a modified version of the original data set so that every observation is utilized and there is no question about losing the information.

3.2.1 Methodology:

Here we proposed Boosted Regularized Decision Trees (BoRDT) based on the methodology of **Adaboost** [18] with the SVR tree. Fit the SVR tree sequentially to the modified version of the data thereby producing a M number of trees $\mathbb{T}_m(x), m = 1, 2, \ldots, M$. The data modifications at each boosting step consist of applying weights $w_1, w_2, \ldots, w_n$ to each of the training observations $(x_i, y_i), i = 1, 2, \ldots, n$. Initially, the first tree is fitted simply as all of the weights are set to $w_i = 1/n$. For each successive iteration $m = 2, 3, ..., n$ the classification algorithm is reapplied to the modified weighted observations. At step m, those observations which have been misclassified by the fitted SVR tree $\mathbb{T}_{m-1}(x)$ induced at the previous step have their weights increased whereas the weights are decreased for those that have been classified correctly. Thus as iterations proceed, observations that are difficult to classify correctly receive ever-increasing influence. Each successive tree is thereby forced to concentrate on those training observations that are missed by previous ones in the sequence.

The predictions from all of the trees are then combined through a weighted majority vote to produce the final prediction:

$$\mathbb{T}(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m \mathbb{T}_m(x)\right) \tag{3}$$

Here, $\alpha_1, \alpha_2, \ldots, \alpha_M$ are computed by the boosting algorithm and weight the contribution of each respective $\mathbb{T}_m(x)$. The effect of α_m is to give a higher influence to the more accurate classifiers in the predicting class sequence.

Algorithm 2 shows the details of the Boosted Regularized Decision Trees (BoRDT) algorithm. Don't get confused between the two types of weights. One is w_i which implies the weights of the whole minority class to fit the SVR tree and another one is ω_i that updates depending on the wrongly classified samples in boosting. Generally, Adaboost is a collection of stumps that has only two leaves but in this paper, we have applied a full-grown SVR tree for the proper use of the SVR method. If the total no of errors is 0 but the pre-determined no of trees is not grown yet, then we assume the total no of errors is 1. The α_m is the amount of say in the algorithm, it refers to the influence assigned to each tree in the ensemble and it is determined as the logarithm of the misclassification rate. Weights are scaled by an exponential term to increase the relative influence for the next classifier in the sequence. Updated weights are higher for misclassified data points than correctly classified data points. So, running the second for loop we get a new training data in which misclassified data points occur most.

3.3 Computational Complexity:

Let the number of training samples be n, the number of features is d, the depth of the estimated tree is h and the maximal number of leaf nodes be $\bar{a}_n = O(\sqrt{n})$. The storage complexity is as same as decision trees, i.e., $O(dn)$.

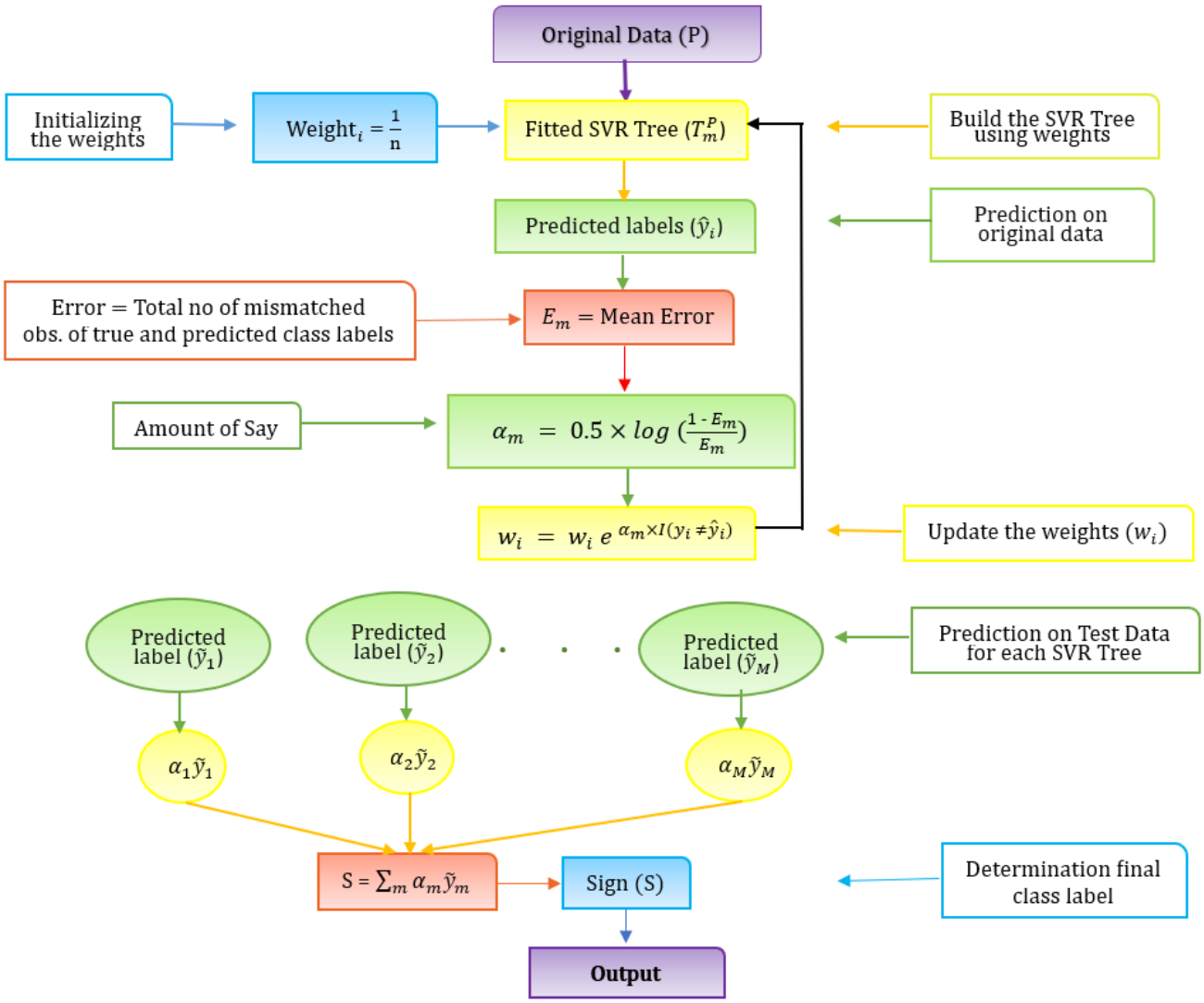


Figure 2: Flowchart of Proposed Boosting (BoRDT) Algorithm

Algorithm 2: Outline Steps of Boosted Regularized Decision Trees (BoRDT)

Data: Training Data, Testing data, no of trees, weight for minority class, maximal no of leaf nodes, penalty term

Result: Output the prediction of class labels.

Train Procedure:

$n, d \leftarrow$ dimension of input data ;

Initialize the weight: $\omega_i \leftarrow$ a vector of $1/n$ of size n;

for $1 \leq i \leq$ *no of samples* **do**

  Fit a SVR classifier $\mathbb{T}_i(x)$ to the training data ;

  Predict this training data using $\mathbb{T}_i(x)$ and record this predicting class as y_{pred} ;

  Calculate the total number of error that is $\sum_{m=1}^{n} \mathbf{1}(y_m \neq y_{pred})$;

  error $\leftarrow$ total number of error $\times$ $1/n$;

  amount of say $\leftarrow$ log((1-error)/error) $\times$ 0.5 ;

  Update the weights: $\omega_i \leftarrow \omega_i \cdot \exp\left[\alpha_m \cdot \mathbf{1}\left(y_i \neq y_{pred}\right)\right], i = 1, 2, \ldots, n$;

  Normalize updated the weight and calculate cumulative(increasing) weights ;

  for $1 \leq j \leq n$ **do**

    Select a random number between 0 to 1 ;

    Select the index no of training data for which cumulative weight $\geq$ random number is minimum.

  end

  Training data $\leftarrow$ select those samples from training data that have the same index no.

end

Test Procedure:

Calculate the predicted class label for each tree ;

Multiply these with per the amount of say of each tree ;

$S =$ Sum over all of these terms ;

Consider the final class label of the sign of S.

The volume computation takes $O(d)$ time because of involving the multiplication of d side lengths. For the computation of the surface, first compute $d-1$ dimensional volumes for all d unique faces, each can be done by dividing the volume by a side length and Computing $O(d)$ as the total time is taken. Next, adding up d-1 -dimensional volumes of all d unique faces and multiplying by 2, which takes $O(d)$ time. Thus the computation of the surface also takes $O(d)$ time in total.

Suppose, the current SVR tree has m leaf nodes $R_1, R_2, \ldots R_m$. If we split at node R_1, which has $\hat{n}$ samples to obtain two child nodes. If both child nodes are in the majority class or the minority class, the surface either does not change or changes by the overlapping surface between R_1 and some R_j s. It takes $O(d)$ time to compute the surface of R_1, and $O(md)$ time to compute all the overlapping surfaces between R_1 and R_k, $2 \leq k \leq m$. Therefore, the time complexity to compute the surface of the tree after splitting is $O(md)$. If one child node belongs to the minority class and the other belongs to the majority class, the surface of the split tree is a piece-wise linear function whose change points can only exist at the borders of the leaf nodes. So, we need to find all the borders to compute all the overlapping surfaces. Thus the process of computing the analytical forms of the surface takes $O(md)$ time. Therefore, it takes $O(md+\hat{n}d)$ time to compute the surface area for all the possible split locations and class label assignments. Thus for all the possible split locations and class label assignments at R_1, the total cost of computing SVR is $O(md+\hat{n}d)$.

Our proposed random forest is an ensemble of decision trees choosing a predictor subset let s. The time complexity of constructing a random forest is proportional to the number of trees in the forest multiplied by the time complexity of constructing a single SVR tree. So, if our SVR tree construction total cost is $O(ms+\hat{n}s)$ taking consider the selected features s, constructing a random forest with T trees would take $O(T(ms+\hat{n}s))$.

The computational complexity of constructing a boosting algorithm depends on the number of boosting iterations and the base learner's complexity. In that sense, for R boosted iteration the time taken by our boosting algorithm is $O(R(md+\hat{n}d))$.

3.4 Comparison of Isometrics:

Isometric plots show contour lines in 2D ROC space for a given metric with a skew ratio as a parameter over the range of possible values. A **ROC Space** is defined by FPR and TPR as x and y axes respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). The skew ratio indicates the relative importance of negatives over positives and is denoted by without costs: **c = Neg/Pos**. Strongly skew-insensitive metric is independent of skew ratio and isometric surfaces in 2D ROC space are vertical that can be obtained for any metric by fixing c. Weakly skew-insensitive metric has the same isometric landscape for different values of c and any collection of ROC points is ranked the same way, regardless of c. Isometric plots visualize the behavior of machine learning metrics - equivalences, skew-sensitivity, and skew-insensitive versions. There are different types of isometrics-

1. **Parallel Linear Isometrics:** Accuracy, Weighted Relative Accuracy(WRAcc).
2. **Rotating Linear Isometrics:** Precision, Lift, F-measure.
3. **Non-linear Isometrics:** Decision tree splitting criteria.

Here we discuss Non-linear Isometrics for different distance measurements applied to decision tree splitting criteria that are considered in this paper, which can then be

modeled through TPR and FPR. Thus in the isometric plots, each contour represents the combinations of TP and FP values that generate a particular value for a given decision tree splitting criterion.

3.4.1 CART Splitting Criteria:

Splitting criteria are invariant under swapping columns. They compare the impurity of the parent with the weighted average impurity of the children:

$$\text{Information Gain} = \text{Imp}(\frac{\text{Pos}}{\text{N}}, \frac{\text{Neg}}{\text{N}}) - \frac{\text{Left}}{\text{N}} \text{Imp}(\frac{\text{TP}}{\text{Left}}, \frac{\text{FP}}{\text{Left}}) - \frac{\text{Right}}{\text{N}} \text{Imp}(\frac{\text{FN}}{\text{Right}}, \frac{\text{TN}}{\text{Right}}) \tag{4}$$

where Imp is Gini Impurity function denoted as:

$$\text{Gini Impurity}(\text{p}_0, \text{p}_1) = 1 - \text{p}_0^2 - \text{p}_1^2 = 1 - \text{p}_0^2 - (1 - \text{p}_0)^2 = 2\text{p}_0\text{p}_1$$

To plot the isometrics in the ROC Space we need to find out the relationship between the information gain with TPR and FPR. Here it is,

$$\begin{aligned}
&2\frac{\text{Pos}}{\text{N}}\frac{\text{Neg}}{\text{N}} - \frac{\text{Left}}{\text{N}}2\frac{\text{TP}}{\text{Left}}\frac{\text{FP}}{\text{left}} - \frac{\text{Right}}{\text{N}}2\frac{\text{FN}}{\text{Right}}\frac{\text{TN}}{\text{Right}} \\
&= 2\left[\frac{\text{Pos Neg}}{\text{N}^2} - \frac{\text{TP FP}}{\text{N left}} - \frac{\text{FN TN}}{\text{N Right}}\right] \\
&= 2\left[\frac{1}{\text{c}+1}\frac{\text{c}}{\text{c}+1} - \frac{\text{TP}}{\text{TP}+\text{FP}}\frac{\text{FP}}{\text{N}} - \frac{\text{FN}}{\text{FN}+\text{TN}}\frac{\text{TN}}{\text{N}}\right] \\
&= 2\left[\frac{\text{c}}{(\text{c}+1)^2} - \frac{1}{1+\frac{\text{FP}}{\text{TP}}}\frac{\text{Neg FPR}}{\text{N}} - \frac{1}{1+\frac{\text{TN}}{\text{FN}}}\frac{\text{Neg TNR}}{\text{N}}\right] \\
&= 2\left[\frac{\text{c}}{(\text{c}+1)^2} - \frac{1}{1+\left(\frac{\text{FPR}}{\text{TPR}}\right)\text{c}}\frac{\text{c}}{\text{c}+1}\text{FPR} - \frac{1}{1+\left(\frac{1-\text{FPR}}{1-\text{TPR}}\right)\text{c}}\frac{\text{c}}{\text{c}+1}(1-\text{FPR})\right] \\
&= \frac{2\text{c}}{(\text{c}+1)^2}\left[\frac{1}{\text{c}+1} - \frac{\text{FPR}}{1+\left(\frac{\text{FPR}}{\text{TPR}}\right)\text{c}} - \frac{(1-\text{FPR})}{1+\left(\frac{1-\text{FPR}}{1-\text{TPR}}\right)\text{c}}\right].
\end{aligned}$$

3.4.2 Hellinger Distance:

We have discussed Hellinger Distance in sec 4.1.4. [7]. Now we may convert the integral of Eq(7) to a summation of all values and re-express our distributions within the context of the conditional probability as:

$$d_D\left(P\left(y_+\right), P\left(y_-\right)\right) = \sqrt{\sum_{i \in V}\left(\sqrt{P\left(y_+ \mid x_i\right)} - \sqrt{P\left(y_- \mid x_i\right)}\right)^2} \tag{5}$$

where V is denoted by the class labels. This presents a distance that quantifies the separability of two classes of data conditioned over the full set of feature values. The effect of class skew on the shape of these isometrics (Flach 2003) can be extended to Hellinger distance as follows:

$$d_D(\text{TPR}, \text{FPR}) = \sqrt{(\sqrt{\text{TPR}} - \sqrt{\text{FPR}})^2 + (\sqrt{1-\text{TPR}} - \sqrt{1-\text{FPR}})^2} \tag{6}$$

This is simply calculated by putting V=2 in Eq(6) considering the proper understanding of the confusion matrix.

3.4.3 Surface to Volume Ratio method:

$$
\begin{aligned}
&\tilde{\text{Imp}}(\mathbb{T}, \mathbb{P}^*_n) \\
&= \mathbb{P}^*(X \in C_0)\tilde{\text{Imp}}(C_0, \mathbb{P}^*) + \mathbb{P}^*(X \in C_1)\tilde{\text{Imp}}(C_1, \mathbb{P}^*) \\
&= \frac{\text{Right}}{\text{Neg+Pos}}[\mathtt{I}_{(y_0=\tilde{y}_0)}\,\text{Imp}(C_0, \mathbb{P}^*) + \mathtt{I}_{(y_0\neq\tilde{y}_0)}(1 - \text{Imp}(C_0, \mathbb{P}^*))] \\
&\quad + \frac{\text{Left}}{\text{Neg+Pos}}[\mathtt{I}_{(y_1=\tilde{y}_1)}\,\text{Imp}(C_1, \mathbb{P}^*) + \mathtt{I}_{(y_1\neq\tilde{y}_1)}(1 - \text{Imp}(C_1, \mathbb{P}^*))] \\
&= \frac{\text{FN+TN}}{\text{N}}[\mathtt{I}_{(0=\tilde{y}_0)}\,\text{Imp}(C_0, \mathbb{P}^*) + \mathtt{I}_{(0\neq\tilde{y}_0)}(1 - \text{Imp}(C_0, \mathbb{P}^*))] \\
&\quad + \frac{\text{TP+FP}}{\text{N}}[\mathtt{I}_{(1=\tilde{y}_1)}\,\text{Imp}(C_1, \mathbb{P}^*) + \mathtt{I}_{(1\neq\tilde{y}_1)}(1 - \text{Imp}(C_1, \mathbb{P}^*))]
\end{aligned}
$$

If $\dfrac{\text{FN}}{\text{Right}}(\text{say}, \mathtt{l}) = \dfrac{1}{1 + (\frac{1-\text{FPR}}{1-\text{TPR}})} \geq 0.5$, left term of above Eq becomes

$$\frac{\text{c}(1 - \text{FPR}) + (1 - \text{TPR})}{(\text{c} + 1)}[1 - 2(1 - \mathtt{l})\mathtt{l}]$$

otherwise, it will be $\dfrac{\text{c}(1 - \text{FPR}) + (1 - \text{TPR})}{(\text{c} + 1)}2(1 - \mathtt{l})\mathtt{l}$.

Similarly, if $\dfrac{\text{TP}}{\text{Left}}(\text{say}, \mathtt{r}) = \dfrac{1}{1 + (\frac{\text{FPR}}{\text{TPR}})} \geq 0.5$, right term of above Eq becomes

$$\frac{\text{c} \times \text{FPR} + \text{TPR}}{(\text{c} + 1)}2(1 - \mathtt{r})\mathtt{r}$$

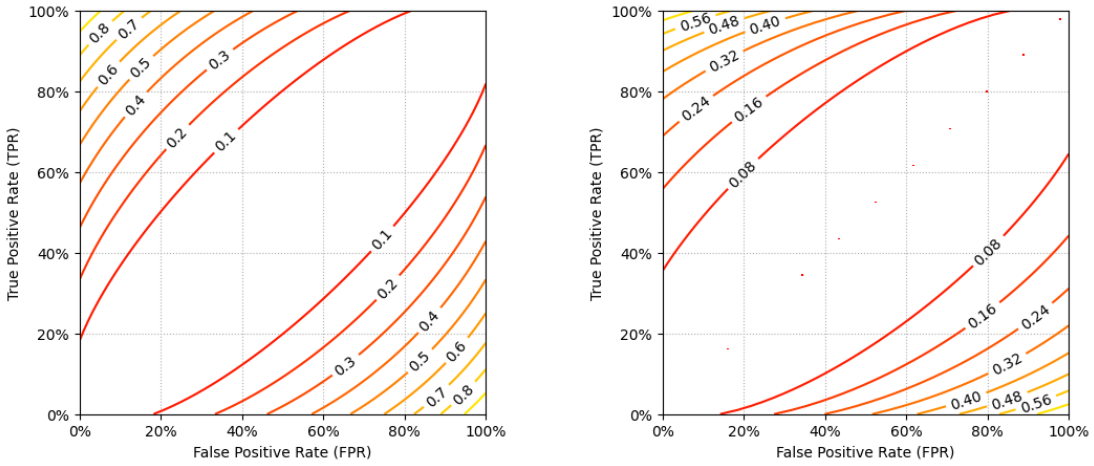
otherwise, it will be $\dfrac{\text{cFPR} + \text{TPR}}{(\text{c} + 1)}[1 - 2(1 - \mathtt{r})\mathtt{r}]$.

The 0.1 contour for Fig: 1(a) implies that for the sets of (0, 20%), (20, 60%), (80, 100%), (20, 0%), (60, 20%), (100, 80%) (tpr,fpr) values information gain is 0.1. The first plot indicates the information gain as contours under ROC space of skew (1:1) respectively. The more skew increases, the isometrics become flatter and information gain will operate more poorly as a splitting criterion that can be observed in Fig:1(b). In Fig:1(c) we observe isometric for Hellinger distance which is unaffected for varying class skew. This is because the equation is independent of class skew resulting more robust Hellinger Distance. Interestingly, Fig: 1(d) implies for (0,0%), (20,20%), (40,40%), and so on isometrics value lies between 0 to 1. Also, for the sets of (0, 20%), (20, 60%), (80, 100%) (tpr,fpr) values information gain is high (0.9) but approximately for the other reverse cases (20, 0%), (60, 20%), (100, 80%) (tpr,fpr) values information gain is low 0.1.

4 Experimental Analysis

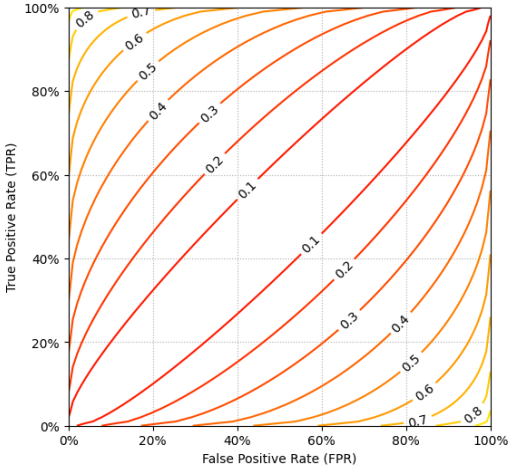
4.1 Baseline Models:

The proposed SVR ensemble methods have been compared with the other 9 popular classification algorithms including both traditional and most recent algorithms which are also efficient in dealing with both balanced and imbalanced data. Most of the algorithms assume balanced priori probabilities for every class and the model

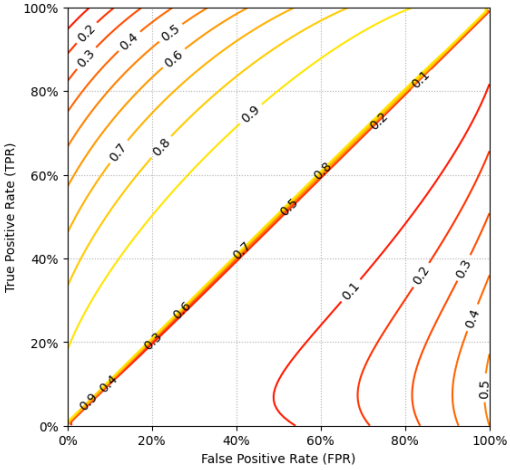


(a)

(b)



(c)



(d)

Figure 3: Isometrics for Information gain, Hellinger Distance and Surface-to-Volume Ratio over a variety of class skews. (a) Information gain isometrics for an imbalance ratio of (1:1). (b) Information gain isometrics for an imbalance ratio of (3:10). (c) Hellinger distance isometric for any imbalance ratio. (d) SVR isometrics for an imbalance ratio of (1:1).

becomes biased towards the majority class. Oversampling of minority class matches the majority class size and vice versa in order to achieve balanced distribution. In random oversampling, the minority class instances are copied and repeated in the data sets but in the case of random undersampling, the majority class instance is discarded at random. Combating this absence of minority class problems either over or under-sampling is utilized. Below we provide a brief description of the baseline models included in the experimental analysis.

Classification and Regression Tree with gini impurity as splitting criteria (CART), Random Forest with CART as the base classifier (RF), Adaboost(ADB) combination of stumps with gini impurity is applied, and the basic ideas of these methods are already discussed in the section.

4.1.1 Neural Network:

A Neural Network(NN)[17] is a two-stage regression or classification model, typically represented by a network. For classification, there are k units at the top, with the kth unit modeling the probability of class k. There are k target measurements Y_k, $k = 1, \ldots, K$ each being coded as a 0-1 variable for the kth class. Activation functions like sigmoid or Gaussian Radial Basis Function create derived features from a linear combination of inputs and then the target is modeled as a function of linear combination of derived features. The output function allows a final transformation of the output.

4.1.2 A hybrid model of CART applied with SMOTE:

SMOTE (Synthetic Minority Oversampling Technique)[5] is based on the distance of each data (usually using Euclidean Distance) and the minority class's nearest neighbors, so the generated examples are different from the original minority class and are applied to each subset of an imbalanced binary class in order to get balanced data. Now, build a usual tree with this new balanced data and this is how CART based on SMOTE is working (SMORT)[10].

The synthetic examples thus generated cause the classifier to create larger and less specific decision regions rather than smaller and more specific regions. In practice, the advantages of SMOTE under/over-sampling goes down as the number of features of the input data increases, because for each minority sample, we require at least d synthetic samples to fully describe its neighborhood so, it is necessary to fix the number of synthetic samples.

4.1.3 Extreme Gradient Boosting(XGBoost):

XGBoost(XGB)[6][20], another type of implementation of Gradient Boosting places a strong emphasis on capturing the residuals of the previous trees and focuses on the gradient descent algorithm to minimize the overall prediction error.

In gradient boosting, probabilities of the exponential function of log-odds are found out and the fitted tree also takes records of pseudo-residuals. In the chronological trees update the pseudo-residuals. It incorporates a regularization technique to prevent over-fitting and provides insights into the importance of each feature in the model. In Imbalanced XGBoost(IM-XGB), different weights to different classes are assigned to give more importance to the minority class. XGBoost produces probability scores for each class. By adjusting the threshold for classification, one can control

the trade-off between precision and recall. Setting a lower threshold can increase the sensitivity to the minority class but may also increase false positives.

4.1.4 Classification with Hellinger Distance(HD):

To address this shortcoming of the Gini index in handling class imbalance, the Hellinger Distance splitting metric is incorporated in its place. It was first implicated to be an extremely efficient splitting criterion over several other alternatives such as entropy for tree-based classifiers in the presence of class imbalance through the Hellinger Distance Decision Tree (HDDT) algorithm, which established HDDT as a robust classifier that is applicable for a variety of data and being largely skew-insensitive for the purpose of imbalanced classification. However, the HDDT algorithm tends to perform poorly when the data is more balanced than imbalanced, which posed a certain limitation to the generalizability of the Hellinger Distance splitting criterion. In order to ensure fair comparisons, uncollapsed, unpruned HDDTs with Laplace smoothing are built. The Hellinger distance is derived using the Bhattacharyya coefficient:

$$h_D\left(D_1, D_2\right) = 2\left[1 - \int_\Omega \sqrt{\frac{dD_1}{dv} \cdot \frac{dD_2}{dv}} dv\right] = \sqrt{\int_\Omega \left(\sqrt{\frac{dD_1}{dv}} - \sqrt{\frac{dD_2}{dv}}\right)^2 dv} \qquad (7)$$

where the Bhattacharyya coefficient between D_1, $D_2 \subset D$ (D is the set of all probability measures on B that are absolutely continuous with respect to v defined on the measure space (Ω, B, v)) is the integral over Ω.

The analysis of Hellinger distance as a decision tree splitting criterion(HDDT) establishes the robustness and skew insensitivity. The same idea of CART is applied to make a hierarchical axes-parallel split of the feature spaces in which each tree node corresponds to one of the segmentation subsets in the feature space. Here, we consider only a binary tree (stump) where a node has exactly two child nodes or zero child nodes. Bagged HDDTs (HDRF), that is built from a collection of trees, are recommended for dealing with imbalanced data sets without using decision trees.

4.2 Experimental Setup:

Features are linearly transformed so that samples lie in $[0, 1]^p$(available in the code) for all Surface-to-Volume methods. During an evaluation of each method, eighty percent of the data has been selected for the training phase and twenty percent of the data for the testing phase. For data sets with three or more classes, classes are combined to form binary class data sets. Preprocessing of these data sets is available in the code of the supplementary material.

We use 10×10 nested stratified cross-validation to evaluate the out-of-sample performance of all methods. It consists of two layers: the inner layer is used to choose tuning parameters and the outer layer is used to evaluate the out-of-sample performance and run this procedure 10 times. In each run, the whole data set is randomly partitioned into 10 stratified folds in which the proportions of samples of each class are roughly the same in all folds and run 10 times. Each time one of the ten folds is selected as the testing set and the other nine folds are training sets. The training sets, we further divide the training set into 10 folds and run 10 times. Each time we use nine folds as (inner) training sets and the other fold for validation.

The largest integer of the ratio of no of majority samples and no of minority samples is set to be the weight w_i to the minority class. The maximal number of leaf nodes is $2\sqrt{\text{n}}$. The penalty parameter η_n for SVR is chosen from a geometric sequence in $[2^0, 2^{10}] \times 10^{-3} \times n^{-1/3}$ and the value with the highest cross-validation F-measure across 10 runs is selected. We then train the model with the selected tuning parameter on the whole training set and evaluate its performance on the test set. The cross-validation AUC-ROC and F-measure are recorded. The mean and standard error of these statistics from 10 nested cross-validation runs are reported.

4.3 Description of Datasets:

We test our method on 27 balanced (imbalance ratio 50 % - 20 %) and 23 imbalanced data sets (imbalance ratio $\geq$ 20 %) from the UCI Machine Learning Repository (Dua and Graff, 2017) and KEEL-data-set repository (Alcal´a-Fdez et al., 2011), varying in size, number of features and level of imbalance. These data-sets are coming from different fields of real life like disease, effects of various bacteria, nature, cars, drinks, votes, and so on. Table 1 and Table 2 provide the necessary descriptions (Total Number of Sample points, Number of Minority Samples, Proportion of Minority Samples, Number of Features available, and coefficient of variation(CV)) of the balanced and imbalanced data-sets respectively arranged in a decreasing order of proportion of minority samples. Clearly, low proportion of minority samples indicates a high imbalance.

4.4 Performance Measure:

Imbalance in the data sets may vary but severe imbalance requires improvement to deal with the inconsistency. Performance metrics should be chosen taking into consideration that imbalanced data are highly skewed. For two-class classification **Confusion Matrix** is often used to assess classification performance displaying the counts of:

	Children		
Parent	TP	FN	Pos
	FP	TN	Neg
	Left	Right	N

Here, we show a Confusion Matrix.

- True Positive (TP): Correctly predicted the positive class.
- True Negative (TN): Correctly predicted the negative class.
- False Positive (FP): Incorrectly predicted the positive class when the true class is negative (Type I error).
- False Negative (FN): Incorrectly predicted the negative class when the true class is positive (Type II error).

A common criteria is **accuracy** $\dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$ but it is not suitable metric for imbalance data. In this situation, it is often important to give more importance to true positives(TP), which is accomplished using the **True Positive Rate(TPR)**

Data Set Name	No of Total Samples	No of Minority Samples	Proportions of Minority Samples	No of Features	CV
SPECTF	80	40	50%	44	0.9937
Hill valley	606	299	49.33%	100	0.986
Heart	1025	499	48.68%	14	0.9735
Monk-2	432	204	47.22%	7	0.9448
Movement libras	360	168	46.66%	90	0.9341
Sonar	208	97	46.63%	61	0.9325
Heart-c	303	139	45.87%	14	0.9173
Ringnorm	300	137	45.67%	21	0.9133
Australian	690	307	44.49%	15	0.8946
Ac inflame	120	50	41.67%	7	0.8946
Data User Modeling	258	107	41.47%	6	0.8507
Housevotes	435	168	38.62%	17	0.7923
Breast-Wisconsin	569	212	37.25%	32	0.7699
Ionosphere	351	126	35.89%	34	0.7472
Wisconsin	638	239	34.99%	10	0.7331
Pima	768	268	34.90%	8	0.7331
Tic-tac-toe	958	332	34.65%	10	0.7278
Titanic	2201	711	32.30%	3	0.7049
Car	1728	518	29.97%	7	0.6541
Breast-y	286	85	29.72%	10	0.6539
Phoneme	5404	1586	29.30%	5	0.6476
ILPD	583	167	28.64%	11	0.633
Haberman	306	81	26.47%	4	0.5991
Vehicle3	846	212	25.05%	19	0.5779
Parkinsons	195	48	24.61%	24	0.5699
Transfusion	748	178	23.79%	5	0.5584
Hayes-Roth	132	30	22.72%	6	0.5402

Table 1: Overview of Balanced Data-sets

or Recall or **Sensitivity** $\frac{\text{TP}}{\text{TP+FN}}$ and **Precision** $\frac{\text{TP}}{\text{TP+FP}}$ [1]. Combining these, the **F-measure** is often used: harmonic mean of TPR and Precision:

$$\textbf{F-measure} = \frac{2.\ \text{TPR}\ .\ \text{Precision}}{\text{TPR+Precision}}$$

It is also used to evaluate the overall performance on both classes. Another suitable measure is **AUC-ROC** (Area Under the Curve-Receiver Operating Characteristics) which is unaffected by skew and measures how well positive cases are ordered before negative cases: arithmetic mean of Sensitivity and Specificity:

$$\textbf{AUC-ROC} = \frac{(\ \text{Sensitivity}\ +\ \text{Specificity}\ )}{2}$$

where **Specificity** is **True Negative Rate(TNR)** $= \frac{\text{TN}}{\text{FP+TN}}$. Another important measures required for the next section are TNR and FNR.

$$\textbf{FPR(False Positive Rate)} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - \text{TNR},$$

$$\textbf{FNR(False Negative Rate)} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{TPR}.$$

Data Set Name	No of Total Samples	No of Minority Samples	Proportions of Minority Samples	No of Features	CV
Appendicitis	106	21	19.81%	8	0.4947
Ecoli	336	52	15.47%	6	0.4272
Segment0	2308	329	14.25%	20	0.4076
Winequality-red	1599	217	13.57%	12	0.3961
Dermatology	358	44	12.29%	35	0.3738
Fertility Diagnosis	100	12	12%	10	0.3674
Estate	5322	636	11.95%	13	0.3658
Page-blocks0	5472	559	10.20%	11	0.3372
Yeast-2vs4	514	51	9.92%	9	0.3315
Ecoli-0-2-3-4vs5	202	20	9.90%	8	0.3306
Yeast-0-3-5-9vs7-8	456	50	9.88%	9	0.3308
Ecoli-0-6-7vs5	220	20	9.09%	7	0.3155
Satimage	6435	626	8.90%	36	0.3089
Glass-0-1-4-6vs2	205	17	8.29%	10	0.2999
Glass2	214	17	8.00%	9	0.2931
Imbalance-scale	625	49	7.84%	5	0.2914
Abalone	731	42	5.40%	7	0.2711
Oil	937	41	4.37%	50	0.2193
Yeast-2vs8	482	20	4.14%	9	0.2078
car-vgood	1728	65	3.76%	7	0.1976
Yeast4	1484	51	3.46%	9	0.1885
Abalone-21vs8	581	14	2.40%	9	0.157
Abalone-20vs8-9-10	1916	26	1.35%	9	0.1172

Table 2: Overview of Imbalanced Data-sets

5 Result and Discussion

5.1 Analysis of Results:

We compute the mean and standard deviation(only for SVR methods we have shown in the first bracket) of SVR methods over 10 nested cross-validation. For each data set and evaluation measure, the method with the highest mean value ranks is highlighted in bold in Tables 3, 4, 5, and 6.

Table 3 & 4 represents AUC-ROC scores and F scores for balanced data sets. From the table, we can observe the proposed BoRDT outperforms all the other standard algorithms on 16 data sets in the case of AUC-ROC scores followed by BaRDT which shows the highest metrics value in 9 data sets and the highest F scores in 13 data sets followed by BaRDT which shows highest metrics value in 11 data-sets among 27 data-sets. In both the performance metrics BoRDT is the clear winner showing its efficiency in case of handling both balanced and imbalanced data sets and BaRDT comes after it.

From Tables 5 & 6 the proposed methods perform very well among 23 imbalanced data sets. BaRDT showed the highest AUC-ROC value in 12 data sets followed by BaRDT which shows the highest metrics value in 5 data sets and for 6 data sets SVRTree performs better than other comparative methods. BoRDT also shows the highest F scores in 10 data sets (imbalanced) followed by BaRDT which shows the

Data	CART	NN	RF	ADB	SMORT	XGB	IXGB	HDDT	HDRF	BoRDT	BaRDT	SVRTree
SPECTF	0.642	0.512	0.643	0.69	0.543	0.629	0.627	0.647	0.661	**0.797** (0.092)	0.744 (0.089)	0.688 (0.018)
Hill valley	0.546	0.5	0.581	0.481	0.536	**0.601**	0.535	0.508	0.58	0.562 (0.032)	0.557 (0.003)	0.551 (0.059)
Heart	0.725	0.827	0.795	0.791	0.498	0.814	0.795	0.733	0.81	0.907 (0.031)	**0.983** (0.012)	0.736 (0.048)
Monk-2	0.764	0.517	0.631	0.488	0.489	0.663	0.666	0.754	0.618	**1** (0)	**1** (0)	**1** (0)
Move-libra	0.85	0.857	0.867	0.846	0.76	0.789	0.745	0.791	0.795	**0.898** (0.032)	0.897 (0.052)	0.848 (0.051)
Sonar	0.698	0.564	0.698	0.692	0.598	0.694	0.693	0.782	0.784	**0.789** (0.057)	0.751 (0.089)	0.677 (0.089)
Heart-c	0.725	0.727	0.745	0.731	0.498	0.714	0.735	0.733	0.81	**0.741** (0.064)	0.678 (0.098)	0.736 (0.075)
Ringnorm	0.707	0.803	0.84	0.83	0.75	0.79	0.758	0.783	0.785	**0.862** (0.035)	0.831 (0.049)	0.769 (0.037)
Australian	0.797	0.862	0.858	0.857	0.85	0.836	0.831	0.798	0.86	0.752 (0.012)	**0.861** (0.032)	0.851 (0.022)
Ac inflame	1	0.773	1	1	0.783	1	1	1	1	**1** (0)	**1**(0)	**1** (0)
Data-Model	0.808	0.667	0.863	0.682	0.499	0.92	0.873	0.84	0.854	**0.96** (0.034)	0.959 (0.034)	0.941 (0.055)
Housevotes	0.945	0.893	0.967	0.962	0.956	0.961	0.961	0.957	0.963	**0.962** (0.012)	0.943 (0.031)	0.95 (0.014)
Breast-w	0.944	0.956	0.95	0.951	0.924	0.948	0.951	0.95	0. 956	**0.954** (0.015)	0.935 (0.022)	0.925 (0.029)
Ionosphere	0.843	0.913	0.914	0.9	0.877	0.91	0.891	0.846	0.914	**0.915** (0.024)	0.887 (0.03)	0.891 (0.026)
Wisconsin	0.947	**0.965**	0.959	0.953	0.944	0.956	0.965	0.944	0.964	0.959 (0.02)	0.958 (0.005)	0.691 (0.01)
Pima	0.69	0.728	0.704	0.735	0.699	0.714	0.718	0.679	0.712	0.723 (0.62)	**0.754** (0.026)	0.718 (0.037)
Tic-tac-toe	0.86	0.84	0.923	0.66	0.85	**0.998**	0.918	0.896	0.925	0.725 (0.11)	0.861 (0.45)	0.745 (0.034)
Titanic	0.715	0.405	0.719	0.713	0.682	0.714	0.704	0.713	0.706	**0.719** (0.23)	0.712 (0.009)	0.699 (0.27)
Car	0.834	0.838	0.846	0.629	0.821	0.819	0.851	0.841	0.85	0.79 (0.003)	**0.855** (0.005)	0.76 (0.005)
Breast-y	0.537	0.601	0.606	0.603	0.594	0.543	0.565	0.584	0.648	**0.605** (0.023)	0.592 (0.056)	0.555 (0.098)
Phoneme	0.642	0.675	0.64	0.672	0.514	0.701	0.681	0.643	0.663	0.81 (0.017)	**0.869** (0.016)	0.837 (0.009)
ILPD	0.573	0.5	0.563	0.573	0.503	0.59	0.567	0.598	0.598	**0.672** (0.32)	0.67 (0.042)	0.636 (0.052)
Haberman	0.54	0.579	0.555	0.584	0.582	0.566	0.568	0.552	0.546	**0.588** (0.4)	0.521 (0.36)	0.506 (0.016)
Vehicle3	0.667	0.598	0.637	0.637	0.629	0.667	0.677	0.685	0.666	0.7 (0.004)	0.732 (0.56)	**0.741** (0.047)
Parkinsons	0.764	0.5	0.856	0.832	0.766	0.878	0.847	0.795	0.846	**1** (0)	**1** (0)	**1** (0)
Transfusion	0.638	0.573	0.599	0.629	0.556	0.605	0.61	0.619	0.605	0.608 (0.32)	**0.655** (0.52)	0.654 (0.031)
Hayes-roth	0.89	0.848	**0.892**	0.721	0.88	0.869	0.88	0.656	0.673	0.5 (0)	0.5(0)	0.5 (0)

Table 3: Performance of AUC-ROC measure of different competitive models over Balanced Data-sets.

highest metrics value in 6 data sets.

These proposed ensemble methods have shown promising results as compared to other models and therefore adaptation of these algorithms widely is recommended as this can be experimented with a variety of data sets for the establishment of its efficacy.

We have generated 20000 synthetic Gaussian data with two class labels with a mean difference of 2.5 and the same standard deviation. Features and Frequency are represented as the x and y axes respectively in the Fig:2. Class 1 denotes the minority class and Class 0 denotes the majority class same as the rest of the paper. We can clearly understand from the plots the pink line classifies the minority class more properly than other methods. Especially for the information gain and Hellinger distance split the feature splits occur at the left side of 0 resulting in a poor classification of the minority class but the pink lines representing the SVR criteria almost perfectly classify the minority class by splitting the feature space on the right side of 0. It may not classify the majority class so accurately but for the imbalanced data we are not concerned about that and that's why we use AUC-ROC and F-measure as metrics.

Data	CART	NN	RF	ADB	SMORT	XGB	IXGB	HDDT	HDRF	BoRDT	BaRDT	SVRTree
SPECTF	0.855	0.045	0.882	0.878	0.148	0.874	0.863	0.849	**0.888**	0.813 (0.096)	0.736 (0.083)	0.62 (0.082)
Hill valley	0.532	0	0.536	0.539	0.697	0.602	0.531	0.499	0.539	**0.54** (0.007)	0.537 (0.012)	0.535 (0.22)
Heart	0.705	0.811	0.77	0.766	0.044	0.798	0.775	0.709	0.788	0.908 (0.012)	**0.984** (0.013)	0.747 (0.039)
Monk2	0.704	0.136	0.512	0.245	0.346	0.58	0.563	0.688	0.483	**1** (0)	**1** (0)	**1** (0)
Move-libras	**1**	0.62	**1**	0.98	0.756	0.812	0.835	0.825	0.892	0.831 (0.033)	0.887 (0.059)	0.829 (0.057)
Sonar	0.657	0.573	0.674	0.624	0.579	0.685	0.696	0.762	0.769	**0.771** (0.066)	0.715 (0.139)	0.643 (0.079)
Heart-c	0.705	0.811	0.77	0.766	0.044	0.798	0.775	0.709	0.788	**1** (0)	**1** (0)	**1** (0)
Ringnorm	0.69	0.8	0.832	0.826	0.752	0.788	0.76	0.764	0.779	**0.839** (0.041)	0.812 (0.4)	0.746 (0.054)
Australian	0.776	0.854	0.832	0.833	0.834	0.839	0.835	0.775	0.833	0.726 (0.034)	**0.839** (0.048)	0.831 (0.016)
Ac inflame	1	0.738	1	1	0.748	1	1	1	1	**1** (0)	**1** (0)	**1** (0)
Data-Model	0.806	0.671	0.86	0.64	0.498	0.917	0.857	0.844	0.846	**0.948** (0.034)	0.947 (0.043)	0.938 (0.055)
Housevotes	0.938	0.882	0.964	0.958	0.942	0.959	0.96	0.953	0.959	**0.962** (0.012)	0.931 (0.035)	0.933 (0.013)
Breast-w	0.929	0.936	0.94	0.941	0.905	0.942	0.941	0.937	0.937	**0.941** (0.021)	0.918 (0.035)	0.903 (0.024)
Ionosphere	0.895	0.951	0.95	0.935	0.843	0.953	0.935	0.895	**0.955**	0.891 (0.041)	0.863 (0.036)	0.854 (0.033)
Wisconsin	0.932	0.952	0.947	0.941	0.926	0.943	0.952	0.927	0.952	0.961 (0.022)	**0.975** (0.004)	0.854 (0.015)
Pima	0.594	0.639	0.605	0.65	0.6	0.627	0.627	0.584	0.617	0.657 (0.079)	**0.688** (0.044)	0.63 (0.048)
Tic-tac-toe	0.924	0.905	0.959	0.803	0.8	**0.999**	0.955	0.929	0.96	0.63 (0.016)	0.831 (0.065)	0.665 (0.051)
Titanic	0.813	0.578	0.813	0.808	0.539	0.752	0.764	**0.843**	0.82	0.615 (0.04)	0.602 (0.023)	0.589 (0.039)
Car	0.908	0.76	0.817	0.341	0.747	0.776	0.825	0.821	0.818	0.722 (0.004)	**0.827** (0.005)	0.627 (0.006)
Breast-y	0.349	0.441	0.444	0.447	0.373	0.434	0.442	0.404	0.484	**0.455** (0.027)	0.385 (0.068)	0.328 (0.132)
Phoneme	0.497	0.54	0.464	0.533	0.051	0.576	0.547	0.498	0.508	0.738 (0.022)	**0.794** (0.021)	0.754 (0.015)
ILPD	0.377	0	0.33	0.355	**0.831**	0.394	0.386	0.813	0.429	0.508 (0.054)	0.538 (0.045)	0.496 (0.069)
Haberman	0.312	0.304	0.3	0.353	0.31	0.299	0.326	0.332	0.275	0.387 (0.141)	**0.457** (0.031)	0.449 (0.017)
Vehicle3	0.499	0.344	0.433	0.438	0.396	0.494	0.513	0.528	0.49	0.459 (0.015)	0.55 (0.069)	**0.559** (0.078)
Parkinsons	0.897	0	0.499	0.927	0.897	0.955	0.929	0.91	0.934	**1** (0)	**1** (0)	**1** (0)
Transfusion	0.448	0.267	0.378	0.413	0.175	0.38	0.348	0.419	0.386	**0.464** (0.142)	0.463 (0.073)	0.458 (0.081)
Hayes-roth	0.865	0.814	0.866	0.556	0.84	0.844	0.841	0.456	0.48	**0.897** (0.02)	0 (0)	0 (0)

Table 4: Performance of F-measure of different competitive models over Balanced Data-sets.

5.2 Multi-category Classification Data Analysis:

For an interpretation of our methodology, we analyze a subset of the longitudinal cohort data on Alzheimer's disease (AD) from the Alzheimer's Disease Research Center (ADRC) at the University of Washington. The dataset is available in the R package DiagTest3Grp. In this data set, measurements of 14 neuro-psychological markers were collected from 118 independent individuals of age 75 and above among which 44 individuals were labeled as non-demented, 43 were mildly demented, and 21 individuals were labeled as demented, that is, AD. It is now commonly accepted that treatment for AD is a rather complicated process and a more clinically useful strategy is to apply appropriate interventions for the earlier-stage patients with relatively mild conditions [9]. Therefore, it is meaningful to differentiate three or even more categories of patients with ascending disease severity and subsequently offer category-specific treatments. Due to some missing observations, we delete 10 individuals from the data set for our analysis. Note that the values of these 14 biomarkers can be negative.

Data	CART	NN	RF	ADB	SMORT	XGB	IXGB	HDDT	HDRF	BoRDT	BaRDT	SVRTree
Appendicitis	0.698	0.764	0.703	0.692	0.721	0.691	0.71	0.659	0.747	**0.853** (0.089)	0.787 (0.064)	0.712 (0.095)
Ecoli	0.819	0.859	0.82	0.818	0.765	0.894	0.778	0.787	0.803	**0.898** (0.038)	0.895 (0.053)	0.862 (0.066)
Segment0	0.987	0.987	0.983	0.983	0.982	0.987	0.982	0.982	0.922	0.979 (0.008)	**0.988** (0.007)	0.986 (0.008)
Dermatology	0.463	0.5	0.494	0.492	0.563	0.492	0.469	0.579	0.569	**0.666** (0.071)	0.631 (0.099)	0.63(0.099)
FertilityD	0.519	0.5	0.607	0.586	0.519	0.592	0.608	0.546	0.564	**0.703** (0.088)	0.644 (0.102)	0.532 (0.092)
Estate	0.541	0.512	0.513	0.512	0.528	0.519	0.528	0.547	0.518	0.509 (0.025)	0.542 (0.019)	**0.55** (0.012)
WinequalityR	0.533	0.5	0.5	0.514	0.698	0.566	0.566	0.539	0.5	0.71 (0.036)	**0.798** (0.036)	0.794 (0.035)
Page-blocks0	0.904	0.923	0.919	0.879	0.914	0.931	0.917	0.912	0.925	0.898 (0.03)	**0.947** (0.033)	0.924 (0.016)
Yeast2vs4	0.795	0.821	0.83	0.841	0.801	0.834	0.851	0.819	0.832	**0.887** (0.066)	0.843 (0.044)	0.834 (0.102)
Ecoli0234vs5	0.836	0.89	0.845	0.869	0.825	0.84	0.859	0.845	0.845	**0.891** (0.091)	0.864 (0.215)	0.718 (0.125)
Yeast0359vs78	0.599	0.598	0.539	0.659	0.549	0.635	0.648	0.652	0.588	**0.687** (0.086)	0.592 (0.026)	0.5 (0)
Ecoli067vs5	0.819	0.842	0.822	0.842	0.865	0.859	0.862	0.865	0.822	**0.969** (0.037)	0.933 (0.091)	0.937 (0.92)
Satimage	0.589	0.631	0.601	0.603	0.599	0.63	0.632	0.613	0.616	0.613 (0.01)	0.811 (0.017)	**0.843** (0.014)
Glass0146vs2	0.554	0.5	0.547	0.667	0.536	0.604	0.422	0.514	0.512	**0.669** (0.147)	0.561 (0.076)	0.587 (0.139)
Glass2	0.56	0.3	0.622	0.62	0.575	0.565	0.568	0.678	0.618	0.674 (0.151)	0.636 (0.153)	**0.676** (0.157)
ImbalanceS	0.5	0.48	0.5	0.49	0.413	0.45	0.467	0.498	0.499	0.542 (0.057)	0.488 (0.047)	**0.56** (0.062)
Abalone	0.605	0.592	0.534	0.6	0.597	0.592	0.616	0.615	0.557	0.633 (0.039)	**0.643** (0.072)	0.608 (0.066)
Oil	0.683	0.591	0.572	0.641	0.64	0.651	0.691	0.697	0.57	**0.759** (0.099)	0.738 (0.072)	0.726 (0.099)
Yeast2vs8	0.739	0.749	0.649	0.699	0.583	0.697	0.749	0.671	0.674	**0.913** (0.071)	0.771 (0.079)	0.768 (0.091)
car-vgood	0.981	0.843	0.968	0.951	0.97	0.969	0.975	0.983	0.984	**0.997** (0.004)	0.947 (0.004)	0.97 (0.012)
Yeast4	0.67	0.538	0.557	0.595	0.589	0.634	0.654	0.641	0.574	0.644 (0.025)	0.613 (0.086)	**0.673** (0.074)
Abalone21vs8	0.718	0.65	0.672	0.696	0.622	0.772	0.721	0.694	0.647	0.848 (0.106)	**0.87** (0.172)	0.665 (0.083)
Abalone20vs8910	0.638	0.566	0.542	0.615	0.587	0.591	0.557	0.597	0.525	0.645 (0.16)	0.513 (0.037)	**0.656** (0.128)

Table 5: Performance of AUC-ROC measure of different competitive models over Imbalanced Data-sets.

5.3 Decision Boundary:

This section provides a comparison of our proposed ensemble methods with the base learners tree on a synthetic data set created by the Imbalanced-Learn library in Python. It aims to provide various imbalanced data sets created by various methods(Lemaitre 2017). The reason behind this topic is to illustrate the nature of decision boundaries of three various methods and to understand the improvement of the performance of our proposed techniques.

Four toy data-set(binary) are generated with proportion of minority samples 30%, 20%, 10%, 5% with weights $[0.3, 0.7], [0.2, 0.8], [0.1, 0.9], [0.05, 0.95]$ respectively. In the library sci-kit-learn the "linearly-separable" function generates 200 samples with a Gaussian noise standard deviation of 0.5. Now this test problem can learn imbalanced algorithms in complex nonlinear manifolds. In all experiments 80% data is considered for the training set and the rest of 20% data is considered as the test set. In order to assess our proposed methods we perform experiments on these data sets by employing all the algorithms and generate a graph representing each data set. The choice of tuning parameters are- for bagging we set the number of trees is to built as 40 and for boosting it is 10. The other parameters are chosen similarly as described in subsection 5.1.

From Fig:3 we can observe decision boundaries are more accurate for BoRDT (The fourth row) and wrongly classified minority samples are very less. Along with decision boundaries for BaRDT (the third row) also provides better classification than SVRTree (the first row). For decision boundaries of SVRTree not only the minority samples but also the majority samples are misclassified. BaRDT plays a middle role

Data	CART	NN	RF	ADB	SMORT	XGB	IXGB	HDDT	HDRF	BoRDT	BaRDT	SVRTree
Appendicitis	0.507	0.6	0.49	0.492	0.546	0.467	0.467	0.398	0.597	**0.634** (0.253)	0.564 (0.097)	0.562 (0.194)
Ecoli	0.622	0.756	0.726	0.716	0.612	0.823	0.652	0.601	0.7	**0.819** (0.053)	0.816 (0.074)	0.737 (0.087)
Segment0	0.983	**0.995**	0.989	0.989	0.971	0.989	0.975	0.988	0.989	0.97 (0.016)	0.982 (0.015)	0.977 (0.013)
Dermatology	**1**	0.5	**1**	0.99	0.926	0.99	0.987	0.988	0.976	0.354(0.117)	0.345 (0.167)	0.345(0.167)
FertilityD	0.15	0	0.546	0.217	0.079	0.233	0.217	0.15	0.133	**0.444** (0.068)	0.38(0.216)	0.2 (0.2)
Estate	0.202	0.055	0.084	0.055	0.169	0.096	0.169	0.215	0.102	0.163 (0.053)	0.199(0.33)	**0.219** (0.016)
WinequalityR	0.076	0	0	0.051	0.48	0.299	0.321	0.124	0	0.397 (0.064)	**0.615** (0.058)	0.547 (0.061)
Page-blocks0	0.833	0.853	0.87	0.801	0.846	0.878	0.864	0.842	0.88	0.855 (0.026)	**0.881** (0.036)	0.841 (0.022)
Yeast2vs4	0.625	0.741	0.749	0.729	0.642	0.734	0.73	0.672	0.75	**0.787** (0.095)	0.687 (0.072)	0.676 (0.164)
Ecoli0234vs5	0.683	0.813	0.73	0.767	0.677	0.65	0.737	0.73	0.73	**0.976** (0.019)	0.975 (0.017)	0.967 (0.016)
Yeast0359vs78	0.286	0.279	0.129	0.401	0.153	0.374	0.401	0.369	0.256	0.911 (0.012)	0.938 (0.013)	**0.946** (0.009)
Ecoli067vs5	0.773	0.68	0.717	0.72	0.77	0.767	0.783	0.723	0.717	0.806 (0.173)	**0.92** (0.109)	0.743 (0.149)
Satimage	0.789	0.832	0.802	0.804	0.798	0.839	0.836	0.834	0.893	0.341 (0.014)	**0.635** (0.028)	0.592 (0.026)
Glass0146vs2	0.167	0	0.133	0.317	0.111	0.34	0.068	0.5	0.067	**0.39** (0.338)	0.222 (0.192)	0.226 (0.077)
Glass2	0.263	0	0.267	0.217	0.198	0.5	0.5	0.313	0.2	0.3 (0.207)	**0.346** (0.315)	0.296 (0.204)
ImbalanceS	0.677	0.577	**0.677**	0.5	0.5	0.357	0.5	0.42	0.419	0.96 (0.092)	0.057 (0.083)	0.177 (0.051)
Abalone	0.264	0.284	0.113	0.42	0.242	0.357	0.319	0.256	0.187	0.342 (0.094)	**0.404** (0.164)	0.243 (0.123)
Oil	0.385	0.169	0.22	0.357	0.452	0.311	0.475	0.452	0.197	**0.604** (0.165)	0.584 (0.129)	0.477 (0.182)
Yeast-2vs8	0.51	0.583	0.383	0.517	0.18	0.43	0.55	0.45	0.45	**0.832** (0.073)	0.692 (0.142)	0.552 (0.149)
car-vgood	0.973	0.769	0.949	0.885	0.947	0.965	0.951	0.966	0.976	**0.98** (0.003)	0.944 (0.003)	0.969 (0.023)
Yeast4	0.352	0.12	0.179	0.234	0.189	0.342	0.405	0.289	0.202	0.341 (0.051)	0.222 (0.085)	0.327 (0.106)
Abalone-21vs8	0.407	0.333	0.4	0.4	0.261	0.573	0.453	0.363	0.317	**0.75** (0.166)	0.677 (0.222)	0.125 (0.084)
Abalone20vs8910	0.283	0.18	0.117	0.283	0.201	0.24	0.15	0.19	0.067	0.123 (0.161)	0.056 (0.125)	0.284 (0.176)

Table 6: Performance of F-measure of different competitive models over Imbalanced Data-sets

	CART	RF	ADB	SMORT	XGB	BoRDT	BaRDT	SVRTree
AUC	0.6814(0.0393)	0.7563(0.0434)	0.8072(0.0591)	0.763(0.0426)	0.7923(0.0538)	**0.8702**(0.0443)	0.8086(0.1002)	0.6327(0.0874)
F	0.5884(0.0383)	0.6657(0.0447)	0.6949(0.0842)	0.7039(0.0579)	0.7058(0.0687)	**0.7877**(0.05004)	0.7077(0.1301)	0.6327(0.0874)

Table 7: Performance of AUC-ROC and F-measure of different competitive models over Imbalanced Data-sets

to plot decision boundaries.

	BoRDT		BaRDT		SVRTree	
Data	AUC-ROC	F-measure	AUC-ROC	F-measure	AUC-ROC	F-measure
Proportion 30%	**0.835** (0.042)	**0.725** (0.071)	0.773 (0.126)	0.673 (0.193)	0.734 (0.071)	0.595 (0.123)
Proportion 20%	**0.824** (0.133)	**0.646** (0.147)	0.724 (0.128)	0.513 (0.212)	0.686 (0.148)	0.475 (0.311)
Proportion 10%	**0.811** (0.128)	**0.667** (0.167)	0.719 (0.2)	0.438 (0.361)	0.683 (0.198)	0.409 (0.323)
Proportion 5%	**0.877** (0.138)	**0.666** (0.218)	0.851 (0.194)	0.643 (0.339)	0.835 (0.238)	0.542 (0.423)

Table 8: Performance of measures of different competitive models over Synthetic Datasets of different minority sample proportions

The performance metrics are reported in Table: 8. The bold values are the highest points for each method. The above example represents data with Gaussian noise for different minority sample ratios. All these toy data sets represent binary imbalance classification problems with 2d. As shown in the plots of Table 8, our approach is able to correctly classify minority samples and achieves the highest ROC-AUC values in comparison with the other imbalanced classifiers for two data sets. This result

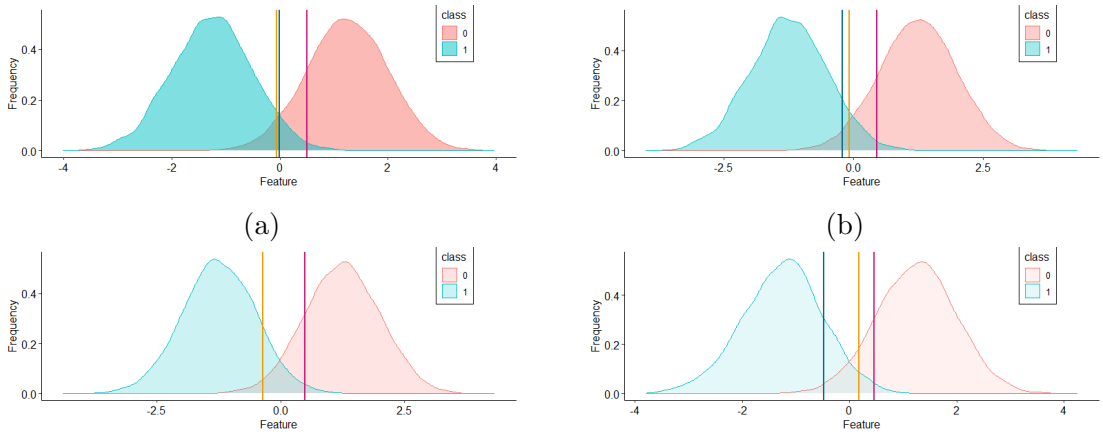


Figure 4: Comparison of the effects of various class distributions on the ability of different methods. The blue, orange, and pink vertical line denotes where the splitting occurs in the feature space to check the ability of information gain, Hellinger distance, and surface-to-volume ratio respectively. (a) Effects of an imbalance ratio of (1:1). (b) Effects of an imbalance ratio of (3:10). (c) Effects of an imbalance ratio of (2:10). (d) Effects of an imbalance ratio of (1:10).

seems to confirm that the proposed approach can deal with a highly imbalanced data structure.

5.4 Feature Selection:

Feature selection is one of the most important parts of pre-processing the data. The features which are not important to predict the class label can be ignored. Here we have shown two plots one for balanced and another for imbalanced real-life data. The variable importance is based on the impurity of a node that results from splits over that variable. The feature is more important and has higher impurity. Then we have shown the testing set's AUC-ROC measure removing each feature. Here, the higher the measure is lower the importance.

Fig:4(a) ILDP balanced data and Fig:4(b) Winequality-red imbalanced data are represented. The impurity of a feature is measured by considering the highest impurity and splitting the feature for each unique value. The impurities represented in blue bars are arranged in increasing order. Feature 4 and 3 have higher impurities so these are important features and trees are grown mostly splitting these features. For each red bar, we have removed that certain feature from the feature space, and AUC-ROC is recorded. From Fig:4(a) we can imply that if we remove feature 10 the test has a higher AUC.

The values are calculated by performing the test 10 times and taking the averages. The error bars represent the variability of two types of measures.

5.5 Conformal Prediction for classification with MAPIE:

Conformal Prediction is a set of algorithms calibrating machine learning models and assessing their uncertainty of predictions. It performs better than other calibaration methods and does not depend on the distribution of the data. Conformal Prediction usually works in the following path: A trained ML model is to be set and create a

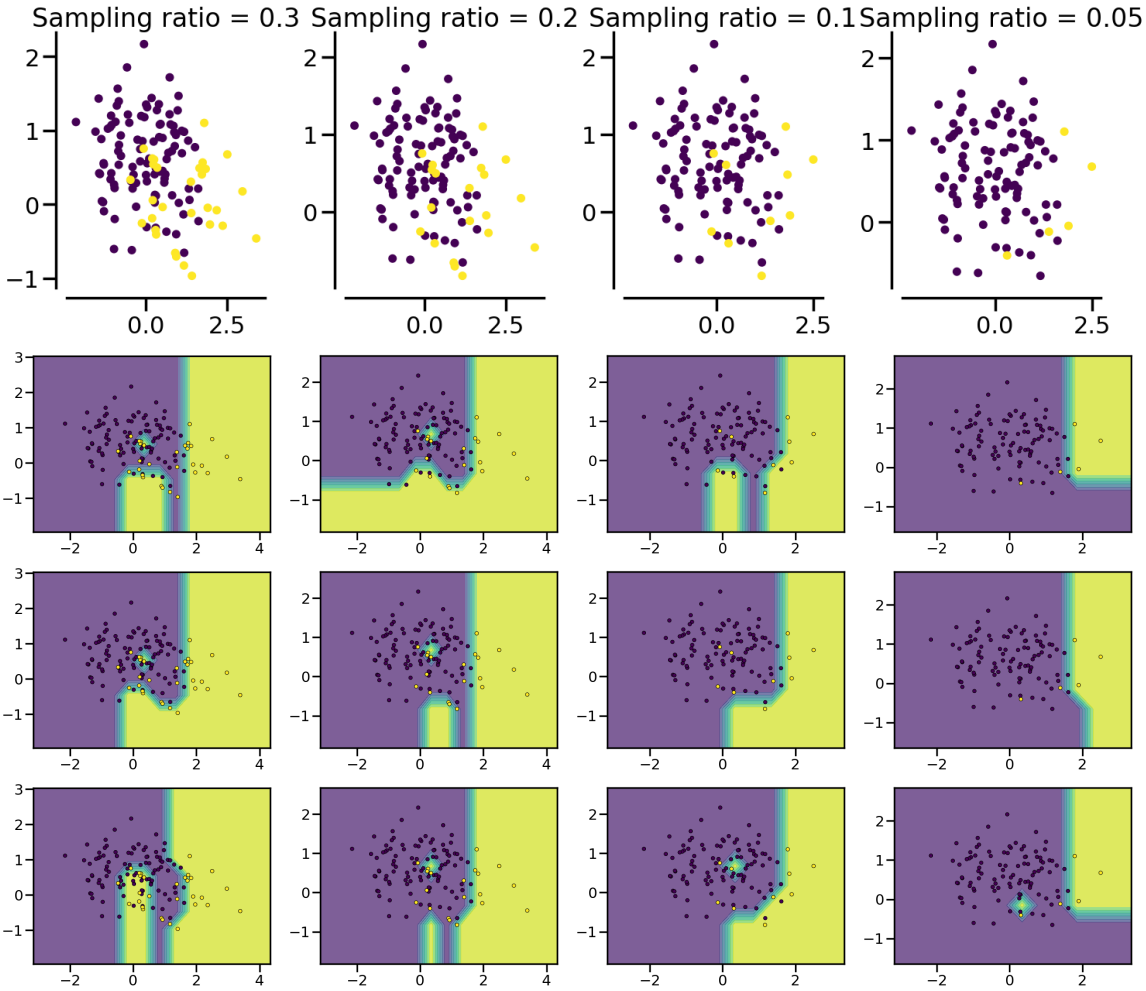
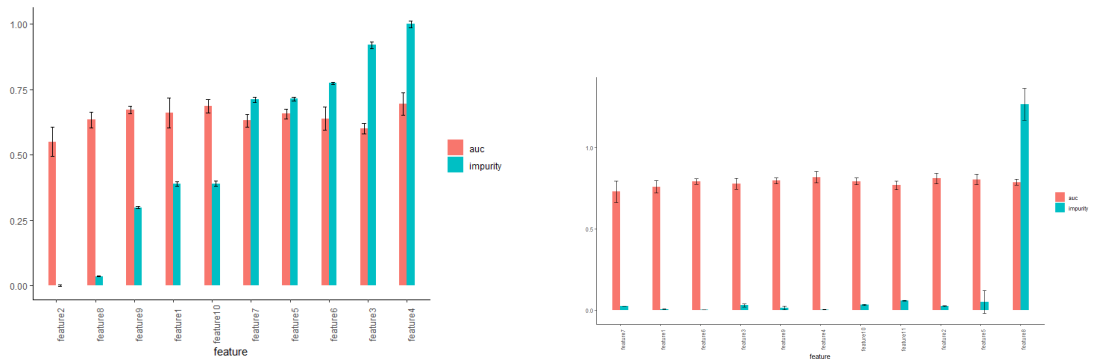


Figure 5: Decision Boundaries for SVRTree, BaRDT, and BoRDT. For all 16 plots, X and Y axes denote the first and second features of a binary toy data set with different sampling ratios to create imbalanced versions of synthetic data. The color division is based on the two-class label. The first row is the scatter plots of the particular training data. The second, third, and fourth rows are the created decision boundaries of SVRTree, BaRDT, and BoRDT respectively. The yellow color represents the minority samples and the purple is for the majority samples.



(a) (b)

Figure 6: In Fig: 4(a) and 4(b) we have illustrated ILDP and winequality-red datasets respectively. The blue bars indicate the feature's impurity measure and the orange bars specify the AUC-ROC measure dropping the certain feature. The error bars are showing how much these measures are deviating.

calibaration set unseen by the model. Apply a cutoff point on an error metric known as calibaration score. This cutoff point provides the width of the prediction interval and helps to form prediction sets for new data points. Finally, check whether most of the predictions should fall inside the prediction interval.

Model Agnostic Prediction Interval Estimator (MAPIE) is a Python library that allows you to estimate prediction intervals using any sci-kit-learn-compatible model based on conformal prediction. MAPIE works on two types of techniques:

1. **Cross Conformal Predictions**
2. **Split Conformal Predictions**

Here specifically we use Split Conformal Predictions that uses the calibrated conformity scores to estimate sets of labels associated with the desired coverage on test data. To apply Split Conformal Prediction for classification with MAPIE, we will estimate a prediction set of classes such that the probability of a true label of a new test point is always higher than the target confidence level. The softmax score of the SVR classifier is then applied as the conformity score.

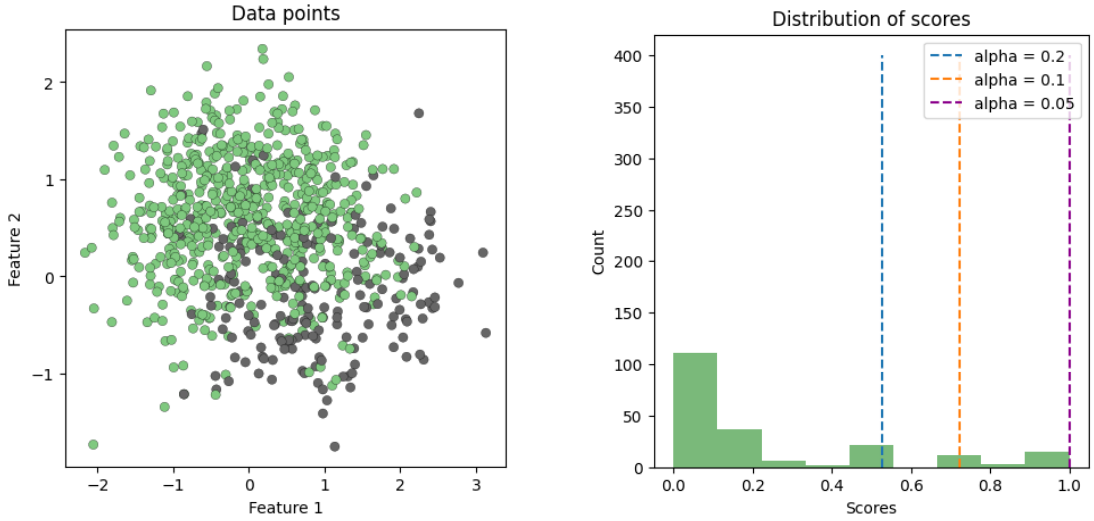
Algorithm 3: Outline steps of Conformal Prediction using MAPIE

Data: Train set, Calibaration set, Test Set, α
Result: Provide TRUE and FALSE of each class label of the given test data
Craft an imbalanced toy dataset and split it into train, calibaration and test sets;
Fit the model on the training set;
n = the number of data points in the calibaration set;
for $1 \leq i \leq n$ **do**
    Predict the class probabilities for the calibaration set;
    Score = Select that probability of the true class label for the calibaration set;
    Softmax score = 1 - Score;
end
for $1 \leq i \leq$ *no of testing data points* **do**
    Q = The α_{th} quantile of Softmax scores that is $\frac{(n+1)\times(\alpha)}{n}$;
    Create a prediction set of the testing data so that it includes all the labels with a sufficiently high Softmax output i.e. Int(test) = class label: the probability of class label > Q;
end

Here, we have created 1300 data points with the proportion of minority sample is 30%. From this synthetic data, we divide it into 80% training data and 20% testing data. Again from this training data, we have split it into 80% updated training data and 20% calibarated data.

In Fig: 5(a) we have plots 832 training data points and in Fig: 5(b) we can depict how different alpha values affect the quantile and the overall width of our conformal prediction sets. We can also observe that the spread looks good and a high value of alpha can potentially lead to a high quantile which would not necessarily be reached by any class in uncertain areas, resulting in null regions. For example, this would be the region where there is an overlap between the classes in the middle of the chart.

Fig: 6 shows the differences for different values of alpha for individual points. The top left plot presents the prediction made by the base estimator. The other



(a) (b)

Figure 7: Fig: 5(a) denotes the training data points and Fig: 5(b) denotes distribution of scores for different α.

3 subplots will feature each alpha score. We don't have any red points with high uncertainty while the highly satiated points in green shared multiple possible classes in the prediction set (higher coverage). As the alpha decreases the interval gets wider creating a higher chance to fall within the interval.

5.6 Statistical significance of the results:

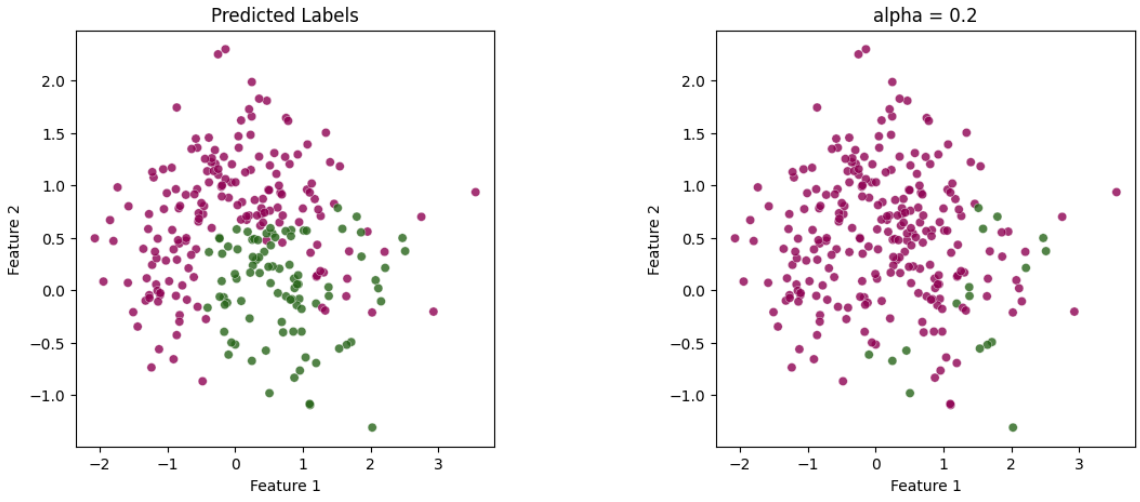
Consistent good Performance of algorithms in a maximum number of data sets is used as an evaluation criterion for determining the best out of all.

5.6.1 Multiple Comparison with the Best:

The method of multiple comparisons with the best or champion(MCB) proposed by [16] is used in situations in which the best group (we will assume the best is the largest, but it could just as well be the smallest) is desired. We concentrate on determining the statistical significance of the prediction of testing data obtained from our proposed models compared to various baseline algorithms. We compute the model's average ranks and corresponding critical distances based on the AUC-ROC and F1 measure for provided balanced and imbalanced data sets to determine the relative performance of all methods discussed in this paper.

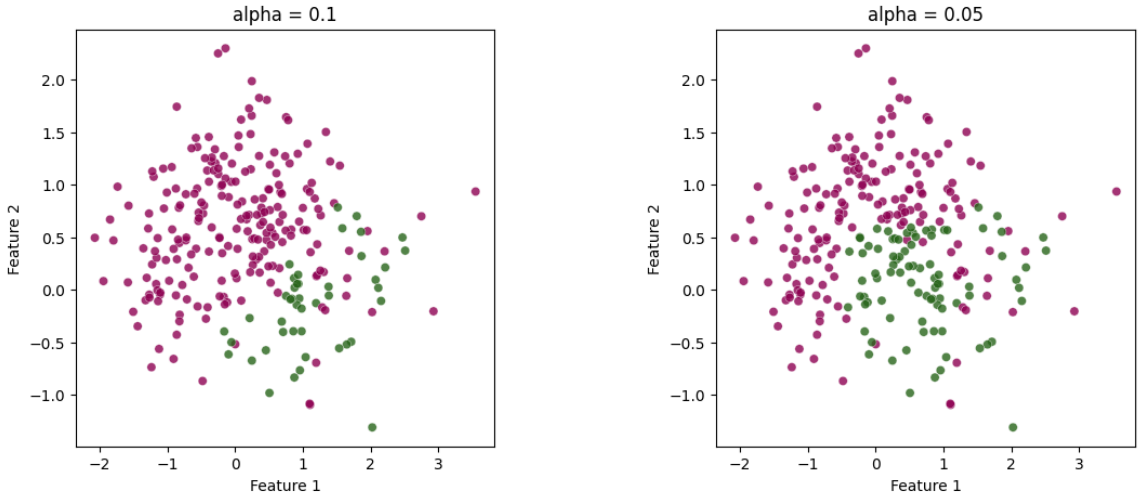
		CART	NN	RF	ADB	SMORT	XGB	IXGB	HDDT	HDRF	SVRTree	BaRDT	BoRDT
Balanced	AUC	8.04	7.63	5.15	6.85	9.74	6.26	6.44	7.39	5.17	6.80	4.72	**3.81**
	F1	7.50	8.41	5.98	6.70	9.61	5.30	5.69	7.09	5.43	7.30	4.69	**4.31**
Imbalance	AUC	7.30	8.04	8.37	7.09	8.65	6.48	6.24	6.39	8.11	4.43	3.78	**3.11**
	F1	6.43	7.85	7.41	6.46	8.57	5.17	5.65	7.33	7.61	6.07	5.07	**4.39**

Table 9: Allotted Ranks for all 12 methods for separately balanced and imbalanced data sets.



(a)

(b)



(c)

(d)

Figure 8: The top left plot shows true prediction labels of testing data and the other three plots show the prediction interval for different alphas.

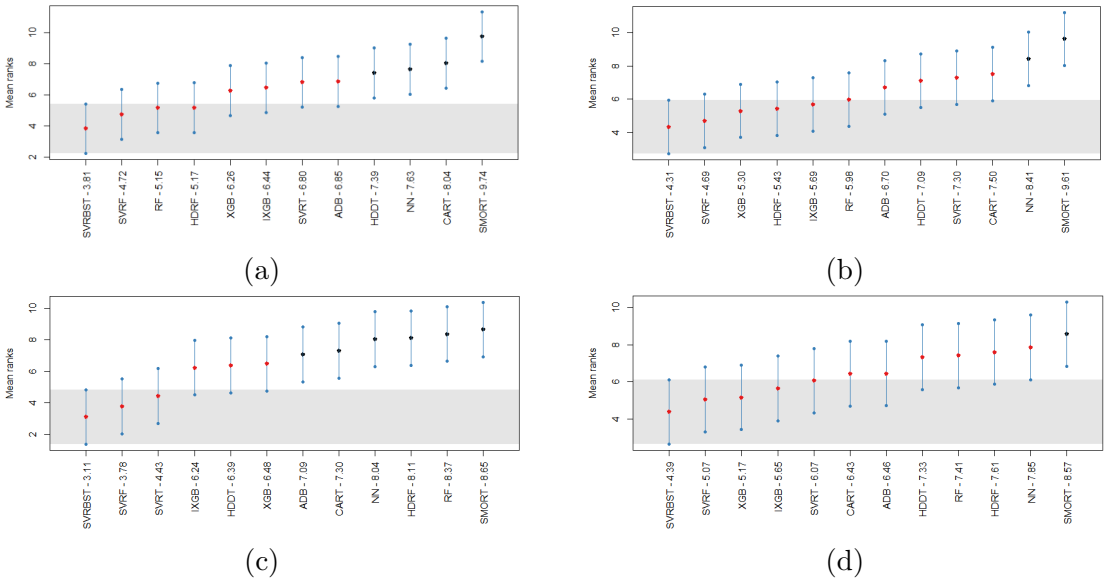


Figure 9: Visualization of MCB analysis. The figures demonstrate w.r.t. (a)AUC-ROC of balanced data, (b)F-measure of balanced data, (c)AUC-ROC of imbalanced data, and (d)F-measure of imbalanced data.

The results of the MCB test presented in Table 9 can be interpreted as follows: The proposed BoRDT model has the least rank (3.81), (4.31), (3.11), and (4.39); in terms of AUC-ROC and F1 measure for balanced and imbalanced data-sets respectively followed by the another proposed bagging technique. It has got ranks 4.72, 4.69, 3.78, and 5.07 in the above-defined order. Among the rest of the methods XGB and Imbalanced-XGB have lower ranks. HDRF also lies along with this. Moreover, from Fig: 7 the upper boundary of the critical distance for the BoRDT model (marked by the shaded region) is the reference value for the test. The black dots in the middle of the critical distance implies there is an overlap between the least rank model with others and this indicates the performance has significantly worse than the proposed methods. Though BoRDT is best, BaRDT has no significant difference from BoRDT. But SMORT, CART, and Neural Networks have non-overlapping critical intervals with the best ones almost in the plots.

5.6.2 Friedman Test:

A non-parametric Friedman test[12] can determine the robustness of our experimental evaluation by comparing the average rank. Friedman tests the null hypothesis that all models are equivalent based on their rankings across various accuracy measures for different data sets. The ranking mechanism assigns rank in increasing order from best to worst. The average of the ranks across all the data sets is then computed for different models. This distribution-free test rejects the null hypothesis of model equivalence if the value of the test statistic is greater than the critical value[15]. Let $r_{m,d}$ denote the assigned rank to mth model out of M models for the dth data-set out of D data-sets. Let R_m is the average rank $\forall m$, the Friedman statistic defined as:

$$\chi_F^2 = \frac{12\mathrm{D}}{\mathrm{M}(\mathrm{M}+1)} \left[\sum_m R_m^2 - \frac{\mathrm{M}(\mathrm{M}+1)^2}{4} \right] \tag{8}$$

follows χ^2 distribution with $(\mathrm{M}-1)$ degrees of freedom, when M and D are large.

A modification of Friedman test statistic was proposed in Iman and Davenport, 1980 as:

$$F_F = \frac{(\mathrm{D}-1)\chi_F^2}{\mathrm{D}(\mathrm{M}-1)-\chi_F^2} \tag{9}$$

which is distributed as F-distribution with $(\mathrm{M}-1)$ and $(\mathrm{M}-1)(\mathrm{D}-1)$ degrees of freedom.

		AUC-ROC	F1
Balanced	χ_F^2	60.6623	6.6736
	F_F	56.2055	6.0688
Imbalanced	χ_F^2	65.8095	7.7344
	F_F	32.1404	3.2015

Table 10: Values for Friedman test over the previous rankings.

We summarize the value of the Friedman test statistics χ_F^2 and F_F obtained for the 12 models across 27 and 23 datasets in Table: 10. The R_m in the Eq (8) is the allotted ranks in Fig: 5. Since for the balanced case, the observed value of the statistic F_F is greater than the critical value $F_{11,268} = 1.8244$, so we reject the null hypothesis at 5% level of significance and conclude that the performance of the algorithms considered in our study is significantly different across all the performance measures. For the imbalanced data set, the observed value of the statistic F_F is greater than the critical value $F_{11,242} = 1.8283$ so, we can conclude the same.

5.6.3 Wilcoxon Signed-Rank Test:

Lastly, we conduct a post-hoc non-parametric Wilcoxon Signed-Rank Test[19] to check the null hypothesis that no significant difference exists between the proposed ensemble boosting model and remaining approaches at a 95% significance level. If the calculated p-value for the test is below 0.05 and concludes that there is a significant difference. Thus from the above performed statistical tests, we can infer at a 5% significance level that our proposed framework's average improvement is robust and statistically significant.

From Table: 11 overall CART, Neural Network, and Adaboost have less than 0.05 p-values so, we can definitely replace these methods with our new approaches. In most scenarios, SVR Forest has a high p-value, implying the closest method to BoRDT. The other methods play a moderate role in all aspects.

6 Conclusion:

Class imbalance learning is a daunting challenge towards the predictive analysis of data that many of the real-world classification datasets pose, where the class distribution of the data is not unproportionate. Class imbalance learning is a daunting challenge towards the predictive analysis of data that many of the real-world classification datasets pose, where the class distribution of the data is not unproportionate. As a direct result of this, most of the classical algorithms for classification fail to model the imbalanced data properly as they are supposed to be used only for the balanced data, they suffer from this skew sensitivity where the minority class is apparently

	Balanced		Imbalanced	
Methods	AUC-ROC	F1	AUC-ROC	F1
CART	0.008889	0.1096	6.032e-05	0.1186
NN	0.008221	0.0003807	2.623e-05	0.01631
RF	0.0946	0.2133	7.206e-05	0.06982
ADB	0.004341	0.04086	5.573e-05	0.1045
SMORT	0.00202	0.004757	0.0001053	0.05223
XGB	0.02893	0.2531	0.0002622	0.3765
IXGB	0.02152	0.1318	0.0002775	0.2416
HDDT	0.01109	0.134	0.0004013	0.07272
HDRF	0.2228	0.2133	4.53e-06	0.02345
BoRDT	0.9273	0.4201	0.04488	0.4274
SVRTree	0.02079	0.02166	0.0614	0.1245

Table 11: Statistical Significance values (p-values) for BoRDT and other models for Wilcoxon Signed-Rank Test

interpreted as an outlier or noise by the algorithm and more emphasis is given to the majority class. This results in a significant hindrance to the model performance, where often the predictive performance for the minority class has been found to be significantly poorer as compared to the majority class. Regularizing classification trees is an old idea; for example, Breiman et al. (1984) proposed penalizing the number of leaf notes in the tree. Other classification trees like C4.5 (Quinlan, 2014) and Ripper (Cohen, 1995) also prune overgrown trees. While individual trees can be limited in their expressiveness due to using only axis-parallel splits, this shortcoming can be mitigated by using an ensemble of decision trees as they have demonstrated statistically significant improvements over a single decision tree classifier (Breiman 1996, 2001; Freund and Schapire 1996; Banfield et al. 2007).

Pioneering research work is carried out and rapid progress is made to tackle imbalanced data problem but dealing with these data sets still remain a challenge. The majority classifiers do not perform well in the case of highly imbalanced data as it assumes the data sets to have balanced class distribution and equal misclassification cost. Class distribution and classification performance are directly related as a relatively balanced distribution has shown better classification results. The level of imbalance along with sample size, and separability are the factors accountable for the classification accuracy. The proposed algorithm uses a novel approach that penalizes the Surface-to Volume Ratio (SVR) of the decision set, SVR regularized trees effectively control the regularity of decision sets. Various methods are chosen from the existing literature to compare with this novel technique to prove its efficiency to work with both balanced and imbalanced data sets.

References

[1] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl*, 3(10), 2013.

[2] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[3] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[4] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. Cart. *Classification and regression trees*, 1984.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[7] David A Cieslak, T Ryan Hoens, Nitesh V Chawla, and W Philip Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24:136–158, 2012.

[8] Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi. When is undersampling effective in unbalanced classification tasks? In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15*, pages 200–215. Springer, 2015.

[9] Bruno Dubois, Howard H Feldman, Claudia Jacova, Harald Hampel, José Luis Molinuevo, Kaj Blennow, Steven T DeKosky, Serge Gauthier, Dennis Selkoe, Randall Bateman, et al. Advancing research diagnostic criteria for alzheimer's disease: the iwg-2 criteria. *The Lancet Neurology*, 13(6):614–629, 2014.

[10] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[11] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[12] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[13] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[14] T Ryan Hoens and Nitesh V Chawla. Imbalanced datasets: from sampling to classifiers. *Imbalanced learning: Foundations, algorithms, and applications*, pages 43–59, 2013.

[15] Ronald L Iman and James M Davenport. Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595, 1980.

[16] Alex J Koning, Philip Hans Franses, Michele Hibon, and Herman O Stekler. The m3 competition: Statistical tests of the results. *International journal of forecasting*, 21(3):397–409, 2005.

[17] Lili Mou, Ge Li, Lu Zhang, Tao Wang, and Zhi Jin. Convolutional neural networks over tree structures for programming language processing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[18] Liwei Wang, Masashi Sugiyama, Cheng Yang, Zhi-Hua Zhou, and Jufu Feng. On the margin explanation of boosting algorithms. In *COLT*, pages 479–490, 2008.

[19] Robert F Woolson. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3, 2007.

[20] Ping Zhang, Yiqiao Jia, and Youlin Shang. Research and application of xgboost in imbalanced data. *International Journal of Distributed Sensor Networks*, 18(6):15501329221106935, 2022.

[21] Yichen Zhu, Cheng Li, and David B Dunson. Classification trees for imbalanced data: Surface-to-volume regularization. *Journal of the American Statistical Association*, pages 1–11, 2021.