# A Regularized Tree Forest for Classification in the Presence of Extreme Class Imbalance

Samir K. Safi * and Sheema Gul

Posted Date: 22 July 2024

doi: 10.20944/preprints202407.1684.v1

Keywords: machine learning; optimal tree ensemble classifier; random forest; support vector machine; artificial neural network

 doi:10.20944/preprints202407.1684.v1



*Article*

# A Regularized Tree Forest for Classification in the Presence of Extreme Class Imbalance

**Samir K. Safi * and Sheema Gul**

United Arab Emirates University; sheema@uaeu.ac.ae
* Correspondence: ssafi@uaeu.ac.ae

**Abstract:** Machine-learning methods used for classification are challenged by the class imbalance problem, where a certain class is underrepresented. Over/under-sampling the majority/minority class observations or model selection for ensemble methods alone might not be effective if the class imbalance ratio is very high. To address this concern, a novel method is presented for generating synthetic data for minority class observations in conjunction with an optimal tree ensemble classifier (OTEC). This novel synthetic method first generates minority class instances to balance the training data. Then it applies OTEC, where models are selected based on their performance using out-of-bag observations and training subsamples. A total of 20 benchmark problems on binary classification with moderate to extreme class imbalance are used to assess the efficacy of the proposed method against other well-known methods, including optimal tree ensemble, SMOTE random forest, under-sampling random forest, oversampling random forest, k-nearest-neighbor, support vector machine, tree, and artificial neural network, using classification error rate, sensitivity, specificity, precision, recall, and F1-score as performance metrics to assess the efficacy of the proposed method. The analyses presented in this study revealed that the proposed method based on data balancing and model selection yielded better results than other methods.

**Keywords:** machine learning; optimal tree ensemble classifier; random forest; support vector machine; artificial neural network

**MSC:** 68U01; 62P10; 62P20

## 1. Introduction

One of the most challenging issues when using a machine-learning method for binary classification is the uneven distribution of classes in the given training data. This problem is referred to as the class imbalance problem [1–4]. Several studies have been conducted on the issue of class imbalance [5–8]. Class imbalance problems can be addressed at four levels associated with various learning phases. To this end, ensemble learning, feature selection at the feature level, internal classifier modification at the classifier level, and resampling methods that change the class distributions can be used to tackle the issue. Among the resampling methods, random under-and oversampling is a non-heuristic method [9]. Under-and oversampling techniques are likely to improve learner generalization when datasets are significantly skewed [10–12]. Random under-sampling aims to balance the class distributions by randomly removing samples from the majority class. However, under-sampling can result in information loss and poor model performance.

By contrast, random oversampling attempts to equalize class distributions by randomly replicating minority class instances. Two problems arise from random oversampling. It first increases the likelihood of over-fitting because it perfectly replicates the minority class samples [13], and the second generation of a minority class can significantly increase the data size, resulting in a prolonged training time and high computational requirements. The synthetic minority oversampling technique (SMOTE) [14] provides more minority instances, which helps the machine-learning model learn more efficiently and assists in tackling the issue of class imbalance. However, certain disadvantages of

 doi:10.20944/preprints202407.1684.v1



SMOTE include its noise sensitivity and dependence on its nearest neighbor. An advanced sampling technique, boosting, was applied to mitigate this issue. Boosting focuses on the incorrectly classified instances. Boosting significantly affects the distribution of the training data. Other methods for finding training data close to the optimal value include heuristic, budget-sensitive, and progressive sampling methods [15–16]. OSPC, an oversampling technique based on preliminary classification, performs better in minority/majority class classification than SMOTE and under-sampling methods.

The goal of each of the aforementioned methods for changing class distributions is to address the issue of between-class imbalance. A cluster-based oversampling method was proposed [17] to simultaneously improve the accuracy of minority classes and address the issues of both within- and between-class imbalances. Both oversampling and under-sampling can be addressed using approaches, such as SMOTE, paired with the Tomek link and ENN [18] when there are only a few samples of the minority class or when the datasets are significantly skewed. Most machine-learning methods rarely yield satisfactory results in real-world applications with class imbalance problems. The key causes of the poor performance of the classifier on imbalanced datasets are accuracy, class distribution, and error costs. Numerous approaches have been proposed to solve these problems and to improve the efficiency of binary classification models [19].

Several other dedicated research endeavors have been conducted to address the class imbalance problem. The authors in [20] have provided a thorough comparison of several existing remedies for mitigating this problem. While most typical learning algorithms can achieve high global accuracy by correctly classifying majority class observations, improving the minority class accuracy remains a significant challenge [21]. To this end, the authors in [22] proposed the idea of a k-fold division of the majority class to generate a pool of classifiers that divides an imbalanced problem into numerous balanced ones, while guaranteeing that all available samples are used in the training process. In the context of ensemble learning, the model performance is significantly associated with the performance of the base classifier, weight computation method, and selection strategy when dealing with class-imbalanced datasets. This notion has led to the development of several ensemble learning methods. For example, the algorithm given in [23], which employs under-sampling of the majority class for classifying imbalanced datasets, has been proposed. Similar methods can be found in the literature that address the issue of class imbalances in binary classification problems. These techniques primarily exploit the idea of under-sampling the majority class observations or oversampling the majority class observations. Both these ideas have certain limitations, and the consequent machine-learning method may still fail to achieve the desired classification performance.

A binary classification model must perform well for both classes to achieve the desired classification performance. In certain situations, when there are insufficient instances in one or both classes, machine-learning algorithms [24] may find it challenging to identify patterns in the data and correctly classify minority class samples because of this imbalance. To address this issue, researchers developed learning algorithms to process and extract information from data with significantly skewed distributions. Specifically, instead of ignoring minority classes in the data, an imbalanced learning technique allows the prediction algorithm to predict the outcome variables with a more realistic level of accuracy. The model was trained to distinguish between instances belonging to two separate classes. Nevertheless, several factors can significantly affect the performance of a binary classification model, including class noise, class imbalance, and insufficient data.

Several tree-based methods have been modified to address class imbalance problems. However, little attention has been paid to simultaneously balancing data and selecting optimal base models for the final tree ensemble. In constructing an efficient ensemble, accuracy, and diversity are the main features that significantly regulate the overall performance of the final model. In the case of random forest, Breiman provided an upper bound for the overall prediction error as a function of the accuracy and diversity of the base tree models: i.e.

$$\boldsymbol{\varepsilon = \hat{\rho}\epsilon_t}$$

where $\boldsymbol{\varepsilon}$ is the total prediction error of the forest, $\boldsymbol{\epsilon_t}$ is the estimate of any $\boldsymbol{t}$ tree in the forest, and $\boldsymbol{\hat{\rho}}$ is the weighted correlation between the residuals from two independent classification trees expressed as the mean correlation over the entire random forest. Therefore, developing a tree ensemble [25–27]

 doi:10.20944/preprints202407.1684.v1



that ensures the accuracy and diversity of the base model is believed to yield promising results. In the presence of class-imbalanced problems, allowing the ensemble to learn from balanced data leads to further improvements. Therefore, this study proposes an ensemble method that uses the above concepts to learn effectively from datasets with class-imbalanced problems.

To build an efficient tree ensemble, this study first balanced the given training data by generating new synthetic observations for the minority class. Consequently, datasets with extreme class imbalances were considered in this study. These data sets were obtained from several publicly available sources. Once the dataset was balanced, classification trees were grown on bootstrap/subsamples from the data. The training prediction accuracy of each tree was estimated, and the top-performing trees were selected for the final ensemble. Based on the above notion, this study attempted to increase the accuracy of individual tree models in forests, in addition to randomizing their construction. The accuracy was increased by balancing the given training data with the model selection based on individual prediction performance.

The remainder of this article is organized as follows. The proposed methods are given in Section 2, and the experiments and results are described in Section 3. Section 4 presents the simulations, which include both of the simulated scenarios for the proposed method. The final section concludes by summarizing the main findings and providing suggestions for future work.

## 2. Materials and Methods

### *2.1. Balancing the Training Data*

Let $\Upsilon$= (X, Y) be the given data; X is an $\mathbf{n} \times$ p matrix, that is, $\mathbf{X} = [\mathbf{x_{i\times j}}]_{\mathbf{n\times p}}$ where $\mathbf{i} = \mathbf{1, 2, 3, \ldots, n}$, and $\mathbf{j} = \mathbf{1, 2, 3, \ldots p}$, and Y are two-class vectors, that is, $\mathbf{Y\epsilon}$ (0,1) of length n. $\boldsymbol{y_i}$ are the values of the binary response variable with a value 1for one class and a value 0 for the other class; that is,

$$\mathbf{y_i} = \begin{cases} \mathbf{1, if\ n} = \mathbf{n^1} \\ \mathbf{0, if\ n} = \mathbf{n^0}, \end{cases} \mathbf{i = 1, 2, 3, \ldots, n.}$$

Out of the total n sample points, there were $\mathbf{n^1}$majority and $\mathbf{n^0}$ minority class observations, with a severely skewed class distribution, i.e., i.e., $\mathbf{n^1} \gg \mathbf{n^0}$ . The following procedures were considered to balance the training data before building the tree ensembles. The given dataset $\Upsilon$ was balanced by selecting $\boldsymbol{\mathcal{K}_b}$=$\mathbf{n^1 - n^0}$ bootstrap samples each of size, i.e. $\mathbf{n^0}$, equal to the number of observations in the minority class, from the minority class observations. Let the sample be $\mathbf{X_v}$, where v = 1, 2, …, $\boldsymbol{\mathcal{K}_b}$ . Each bootstrap sample had $\mathbf{n^0}$ observations of p features. Each bootstrap sample was used to generate an additional row in the data by estimating the mean $\bar{\mathbf{u}}$ of each column if it was numeric and mode $\tilde{\mathbf{u}}$ if it was categorical. Let there be $\mathbf{p_1}$ continuous features and $\mathbf{p_2}$ categorical features in the bootstrap samples; calculating the means and modes of the features gives a vector of $\mathbf{p = p_1 + p_2}$ observations, that is,

$$\mathbf{X^r_{new}} = (\bar{\mathbf{u}}_{\mathbf{1}}, \bar{\mathbf{u}}_{\mathbf{2}}, \ldots, \bar{\mathbf{u}}_{\mathbf{p_1}}, \tilde{\mathbf{u}}_{\mathbf{1}}, \tilde{\mathbf{u}}_{\mathbf{2}}, \ldots, \tilde{\mathbf{u}}_{\mathbf{p_2}}),$$

where r = 1, 2, …,p, and the elements in $\mathbf{X^r_{new}}$ are arranged according to the original order of the features in the bootstrap sample. The generated rows are arranged in the matrix, and the last column comprises the class labels of the minority class, that is,

$$\acute{\mathbf{Y}} = (\hat{\mathbf{X}}, \mathbf{Y = 0})_{\mathbf{n^o\times p}},$$

where, $\hat{\mathbf{X}}$ denotes a newly generated vector. The given training data was combined with the generated data $\acute{\mathbf{Y}}$ to obtain balanced data, that is,

$$\tilde{\mathbf{Y}} = \Upsilon \cup \acute{\mathbf{Y}}.$$

For each class, there were equal numbers of data points in $\tilde{\mathbf{Y}}$. To grow optimal trees, the data $\tilde{\mathbf{Y}}$ was used instead of the original data $\boldsymbol{\Upsilon}$ . Two methods are used to grow and select optimal trees ensemble using balanced data $\tilde{\mathbf{Y}}$. The first method uses the corresponding out-of-bag (oob) observations to assess each tree individually. Trees were ranked based on how well they performed

 doi:10.20944/preprints202407.1684.v1



individually based on oob observations. The second method involved random subsets of training data for tree growth. In this study, balanced data, i.e., $\tilde{\mathbf{Y}}$ was used in conjunction with an ensemble method, i.e., optimal tree ensemble classifier using out-of-bag, i.e., OTEC (oob) and optimal tree ensemble classifier using sub-sample, i.e., OTEC (sub), on balanced data.

*2.2. Out-of-Bag based Assessment with Balanced Data*

The first proposed method, i.e., OTEC (oob) used out-of bag (oob) observations for trees grown on the balanced training data $\tilde{\mathbf{Y}}$ . Studies [28–29] have observed that samples during bootstrapping omit approximately 1/3 of the overall training data [30]. These observations play no role in model construction and can be used as independent validation data for model assessment. From the balanced data $\tilde{\mathbf{Y}}$ , we took T bootstrap samples, i.e., $\mathbf{B\tau}$, $\tilde{\mathbf{Y}}$ = 1,2,3, …,T , and let $\bar{\mathbf{B}}_{\boldsymbol{\tau}}$ be the corresponding value of oob observations from each sample. G($\mathbf{B\tau}$) represents the classification tree that was grown on $\mathbf{B\tau}$. It was further assumed that $\widetilde{\mathbf{error}}_{\boldsymbol{\tau}}$ represented the oob error, which is the error of G($\mathbf{B\tau}$) on $\bar{\mathbf{B}}_{\boldsymbol{\tau}}$, i.e.,

$$\widetilde{\boldsymbol{error}}_{\boldsymbol{\tau}} = \frac{\mathbf{1}}{|\bar{\boldsymbol{B}}_{\boldsymbol{\tau}}|}\sum_{\boldsymbol{x_i}\boldsymbol{\epsilon}\bar{\boldsymbol{B}}_{\boldsymbol{\tau}}} \boldsymbol{\phi}(\boldsymbol{y} \neq \tilde{\boldsymbol{y}}),$$

where, y is the true class label in the bootstrap sample, i.e.,$\bar{\mathbf{B}}_{\boldsymbol{\tau}}$, $\tilde{\mathbf{y}}$ is the corresponding predicted value via classification tree G($\mathbf{B\tau}$), and $|\bar{\mathbf{B}}_{\boldsymbol{\tau}}|$ is the size of the oob sample. $\boldsymbol{\phi}$ is an indicator function, expressed as:

$$\boldsymbol{\phi}(\mathbf{y} \neq \tilde{\mathbf{y}}) = \begin{cases} \mathbf{1}, & \mathbf{if\ y} \neq \tilde{\mathbf{y}} \\ \mathbf{2}, & \mathbf{Otherwise}. \end{cases}$$

G($\mathbf{B\tau}$), After growing the desired number of classification trees, they were arranged in ascending order of their error rates on the oob samples. Let the top-ranked, second-top-ranked, etc. trees be

$$\boldsymbol{G^{r1}}, \boldsymbol{G^{r2}}, \ldots, \boldsymbol{G^{rQ}}$$

A number of trees from the above-ranked trees were selected for the final ensemble. The ensemble was then used to predict the new/test data. The number of trees grown and the number of trees selected were two potential hyper-parameters for this method.

*2.3. Sub-Sample based Assessment with Balanced Data*

The second proposed method, i.e., OTEC(sub) used sub-sample based method where trees were grown on sub-samples from the balanced data ($\tilde{\mathbf{Y}}$). Unlike the oob observations, the remaining observations from each sample acted as test data for evaluating the predictive performance of each corresponding tree. Given that $\mathbf{B\tau}$, $\boldsymbol{\tau}$ = 1,2,3, …, T be the random sample of size m < n. Additionally, let $\bar{\mathbf{B}}_{\boldsymbol{\tau}}$represent the corresponding remaining subset of observations of size n - m. G($\mathbf{B\tau}$), where, $\boldsymbol{\tau}$=1,2, …,T, is the classification tree built on $\mathbf{B\tau}$ . It is also assumed that the error of G($\mathbf{B\tau}$) on $\bar{\mathbf{B}}_{\boldsymbol{\tau}}$ is represented by $\widetilde{\mathbf{error}}_{\mathbf{sub\tau}}$. On $\boldsymbol{\tau}$ , $\boldsymbol{\tau}$ = 1,2, …,T, for the T classification trees. was used to estimate $\widetilde{\mathbf{errorsub}}_{\boldsymbol{\tau}}$on each tree.

Let the top, second highest, and so on ranked trees be, that is,

$$\boldsymbol{G^{r1}}, \boldsymbol{G^{r2}}, \ldots, \boldsymbol{G^{rQ}}$$

The remaining procedure was the same as that for OTEC(sub). This method might be useful in small-sample situations in which one wants to retain large amounts of training data to build trees. This method can also be tuned by selecting the optimal values for the initial number of trees grown and the number of trees selected for the final ensemble. The pseudo-code of the proposed ensemble is provided in Algorithm 1.

 doi:10.20944/preprints202407.1684.v1



**Algorithm 1** Pseudo-code of the proposed ensembles.

1: Training data $\boldsymbol{\Upsilon}$ consisting of $\mathbf{n^1} \gg \mathbf{n^0}$ observations and p variables;
2: $\mathbf{n^1} \leftarrow$ Number of Majority class;
3: $\mathbf{n^0} \leftarrow$Number of Minority class;
4: $\tilde{\mathbf{y}} \leftarrow$Balanced data
5: $\mathbf{n^1} - \mathbf{n^0}$ 0 = $\boldsymbol{\mathcal{K}_b}$ Number of synthetic data needed for minority class observations.
6: for $\boldsymbol{\nu} \leftarrow \mathbf{1}$ $\leftarrow$ 1 : $\boldsymbol{\mathcal{K}_b}$ : do
7: Take a bootstrap sample ($\boldsymbol{\mathcal{K}_b}$) from minority class observations ($\mathbf{n^0}$) in the training data ($\boldsymbol{\Upsilon}$);
8: If a column (variable) is continuous, find its mean ($\bar{\mathbf{u}}_{\mathbf{vj}}$)
9: if categorical, find its mode ($\bar{\mathbf{u}}_{\mathbf{vj}}$)
10: Concatenate the values in Steps 8 and 9 to get a new row arranged according to the original training data.
11: Add the row to the data with class labels of minority class $\acute{\boldsymbol{\Upsilon}}$.
12: Combine the original training data ($\boldsymbol{\Upsilon}$) with generated data $\acute{\boldsymbol{\Upsilon}}$ to obtain the balanced data ($\tilde{\boldsymbol{\Upsilon}}$)
13: end for
14: for = 1 → T do
15: Take a bootstrap/sub-sample ($\bar{\mathbf{B}}_{\boldsymbol{\tau}}$) from balanced training data ($\tilde{\boldsymbol{\Upsilon}}$).
16: Store oob/out of sample observations.
17: Grow classification tree (G(B$\boldsymbol{\tau}$)) on the bootstrap/ sub-sample ($\bar{\mathbf{B}}_{\boldsymbol{\tau}}$).
18: Use oob/out of sample observations and estimate prediction error ($\widehat{\mathbf{error}}_{\boldsymbol{\tau}}$).
19: end for
20: Arrange the trees $\mathbf{G^{r1}}, \mathbf{G^{r2}}, \ldots, \mathbf{G^{rQ}}$ in ascending order with respect to oob/out-of-sample errors.
21: Select the top ranked trees ($\boldsymbol{Q}$) as the final ensemble

## 3. Experiments and Results

This study considered extremely imbalanced classification datasets to assess the performance of the proposed method for benchmark problems. The efficacy of the proposed method, i.e., OTEC(oob) and OTEC(sub) in comparison with the state-of-the-art methods, i.e., optimal tree ensemble (OTE), random forest in conjunction with using synthetic minority over-sampling technique (RF(smote)), random forest combined with over-sampling method (RF(over)), under-sampling method coupled with random forest (RF(under)), support vector machine, k-nearest neighbor (k-NN), artificial neural network (ANN), and classification tree (Tree) was assessed using 20 benchmark datasets. The evaluation metrics considered were the classification error rate, sensitivity, specificity, precision, recall, and F1 score, which is a measure that combines precision and recall into a single value, providing a balanced assessment of a model's performance. R programming software was used to perform the experiments.

Table 1 provides a concise summary of the datasets considered. The data name is displayed in the second column and the numbers of instances/observations (n) and features (p) are shown in the third and fourth columns, respectively. The fifth column provides the classwise distribution, the sixth column provides the imbalance ratio ($n^1/n^0$), and the final column contains the data source. Additional information on the remaining datasets is provided in Table A1. This section describes the experimental design of this study.

The proposed methods, OTEC(oob) and OTEC(sub), were applied to extremely imbalanced datasets, with 95% of the observations belonging to the majority class and 5% to the minority class. This indicates that the observations were made at a 19:1 ratio. This was performed by randomly discarding minority class observations in which the original data did not have an uneven ratio of 19:1. A total of 1,000 realizations were made from the resulting data, which were split into 90% training and 10% testing parts. Model fitting was performed on the training part of the data, and evaluation was performed on the testing part. To generate T = 1,000 classification trees, bootstrap samples from 90% of the training data were obtained using the OTEC(oob) and OTEC(sub) methods described in Algorithm 1. The testing part was used for external validation.

 doi:10.20944/preprints202407.1684.v1



**Table 1.** Summary of datasets with class-imbalanced problem.

| No | Dataset | Instances | Features | Class-based Distribution | Imbalance ratio $n^1/n^0$ | Source |
|---|---|---|---|---|---|---|
| 1 | Breast Cancer | 569 | 31 | 357/212 | (1.6839:1) | https://www.kaggle.com/datasets/utkarshx27/breast-cancer-wisconsin-diagnostic-dataset |
| 2 | Credit Card | 284807 | 30 | 284807/492 | (578.876:1) | https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud |
| 3 | Drug Classification | 200 | 6 | 145/54 | (2.685:1) | https://openml.org/search?type=data&status=active&id=43382 |
| 4 | Kc2 | 520 | 21 | 414/106 | (3.905:1) | https://openml.org/search?type=data&status=active&sort=runs&id=1063 |
| 5 | Eeg eye | 5856 | 14 | 5708/148 | (38.567:1) | https://openml.org/search?type=data&status=active&sort=runs&id=1471 |
| 6 | Glass Classification | 213 | 9 | 144/69 | (2.086:1) | https://openml.org/search?type=data&status=active&id=43750 |
| 7 | Pc4 | 1339 | 37 | 1279/60 | (21.316:1) | https://openml.org/search?type=data&status=active&sort=runs&id=1049 |

 doi:10.20944/preprints202407.1684.v1



The final ensemble contained all of the top $Q$ trees that were chosen. The final result was the average of all 1,000 runs. The training and testing parts were the same for all methods under consideration. The findings presented in Tables 2 and 3, along with the box plots displayed in Figures 1–4, show the results of the proposed method, that is, OTEC(oob) and OTEC(sub), in terms of the classification error rate and precision compared with the other methods. The results for the rest of the methods, that is, OTEC(oob) and OTEC(sub), in terms of sensitivity, specificity, recall, and F1 score, are given in Tables A2–A5 of Appendix A. The plots are shown in Figures A1–A14 in Appendix A. Table 2 shows the results of the proposed method's classification error rate for the datasets with those of the other state-of-the-art methods. The OTEC(oob) results are shown in bold numbers, and the OTEC(sub) results are italicized. It is evident from the tables that the proposed methods, that is, OTEC(oob) and OTEC(sub), outperform all other procedures in terms of the classification error rate, having the lowest error rate compared to the other methods. The proposed methods, that is, OTEC(oob) and OTEC(sub), showed minimal classification error rates ranging from 0 to 0.0005. OTEC(sub) did not perform well in breast cancer, drug classification, glass classification, or KDD. By contrast, RF (smote) yielded a low error rate on the glass classification dataset. The RF(under), k-NN, SVM, and ANN did not perform well on any of the datasets. In terms of the classification error rate, OTEC(oob) performed better on 19 out of 20 datasets than the other methods. For 17 of the 20 datasets, OTEC(sub) performed better in terms of classification error rate. However, the other methods failed to yield satisfactory results.

Table 3 presents a detailed comparison of the proposed method with other state-of-the-art methods on the datasets. OTEC(oob) yielded better results for the majority of datasets in terms of precision. The results given in Table 4 show that OTEC(oob), in terms of precision, yields promising results ranging from 93.36% to 100% for several datasets. This demonstrated the effectiveness of the proposed method. In contrast, OTEC(sub) provided better results in terms of precision for 14 datasets, except for the breast, liver disorder, and ionosphere. Moreover, RF (smote) and RF (over) demonstrate high precision in kc2 and glass classification. The box plots revealed the best findings of the proposed method, that is, OTEC(oob) and OTEC(sub), in terms of the classification error rate and precision with other methods given in Figures 1-4. A box plot of the remaining datasets in terms of the classification error rate and precision is shown in Figures A1–A2 of Appendix A.

**Table 2.** The proposed methods, i.e., OTEC(oob), OTEC(sub), and other state-of-the-art methods, in terms of classification error rate.

| Dataset | OTEC(*oob*) | OTEC(*sub*) | OTE | RF(*smote*) | RF(*over*) | RF(*under*) | *k*-NN | SVM | ANN | Tree |
|---|---|---|---|---|---|---|---|---|---|---|
| Breast Cancer | **0.0015** | 0.0311 | 0.0427 | 0.0264 | 0.0166 | 0.0422 | 0.0851 | 0.0486 | 0.3625 | 0.0723 |
| Credit Card | **0.0005** | ***0.0005*** | 0.0010 | 0.0046 | 0.0006 | 0.0083 | 0.1395 | 0.0071 | 0.0016 | 0.0014 |
| Drug Classification | **0.0038** | 0.0461 | 0.0624 | 0.1875 | 0.0296 | 0.0650 | 0.3542 | 0.1771 | 0.4998 | 0.0697 |
| Kc2 | **0.0093** | ***0.0099*** | 0.1743 | 0.0213 | 0.0213 | 0.2060 | 0.1579 | 0.9943 | 0.1925 | 0.1735 |
| Eeg eye | **0.0035** | ***0*** | 0.0063 | 0.0942 | 0.0521 | 0.0872 | 0.0553 | 0.9827 | 0.1094 | 0.2947 |
| Glass Classification | 0.0914 | 0.0974 | 0.1330 | **0.0405** | 0.0710 | 0.1518 | 0.2194 | 0.9839 | 0.2879 | 0.1993 |
| Pc4 | **0.0384** | ***0.0213*** | 0.0945 | 0.0411 | 0.0421 | 0.1519 | 0.1392 | 0.1006 | 0.9555 | 0.1120 |
| Madelon | **0.0222** | ***0.0218*** | 0.2675 | 0.3459 | 0.1397 | 0.3525 | 0.0416 | 0.0420 | 0.0435 | 0.2260 |
| Turing binary | **0.0102** | ***0.0214*** | 0.1321 | 0.3173 | 0.0437 | 0.1882 | 0.1392 | 0.1541 | 0.0271 | 0.1303 |
| KDD | **0.0099** | 0.0101 | 0.0355 | 0.1394 | 0.0098 | 0.1191 | 0.0198 | 0.0226 | 0.9804 | 0.0185 |
| Liver disorder | **0.0400** | ***0.0453*** | 0.0864 | 0.0469 | 0.0512 | 0.1153 | 0.0862 | 0.0915 | 0.7661 | 0.1126 |
| Wine | **0.0227** | ***0.0173*** | 0.0796 | 0.1029 | 0.0680 | 0.0324 | 0.1605 | 0.0305 | 0.2574 | 0.1235 |
| Soy bean | **0.0093** | ***0.0100*** | 0.0364 | 0.0706 | 0.0308 | 0.1229 | 0.0416 | 0.0420 | 0.0428 | 0.0402 |
| Ionosphere | **0.0429** | ***0.0172*** | 0.1168 | 0.2830 | 0.0985 | 0.1531 | 0.1107 | 0.0898 | 0.8913 | 0.1093 |
| Room Occupancy | **0** | ***0*** | 0.0002 | 0.0034 | 0.0003 | 0.0020 | 0.0001 | 0.00001 | 0.0211 | 0.0002 |
| Harth | **0.0121** | ***0.0119*** | 0.0438 | 0.2698 | 0.0379 | 0.1214 | 0.0253 | 0.0682 | 0.9318 | 0.0303 |
| Rocket League | **0.0334** | ***0.0331*** | 0.1032 | 0.4525 | 0.0914 | 0.1158 | 0.0622 | 0.0616 | 0.0614 | 0.0613 |
| Sirtuin6 | **0.0516** | ***0.0516*** | 0.2433 | 0.2680 | 0.1200 | 0.3275 | 0.0700 | 0.0965 | 0.6519 | 0.0847 |
| Toxicity | **0.0339** | ***0.0352*** | 0.1241 | 0.3828 | 0.1392 | 0.2750 | 0.0645 | 0.0570 | 0.0694 | 0.0692 |
| Dry bean | **0.0030** | ***0.0035*** | 0.0102 | 0.0500 | 0.0075 | 0.0134 | 0.0301 | 0.0103 | 0.0295 | 0.0091 |

 doi:10.20944/preprints202407.1684.v1



**Table 3.** The proposed methods, i.e., OTEC(oob), OTEC(sub), and other state-of-the-art methods, in terms of precision.

| Dataset | OTEC(*oob*) | *OTEC(sub)* | OTE | RF(*smote*) | RF(*over*) | RF(*under*) | *k*-NN | SVM | ANN | Tree |
|---|---|---|---|---|---|---|---|---|---|---|
| Breast Cancer | **1.0000** | *0.9629* | 0.9364 | 0.9670 | 0.9795 | 0.9517 | 0.9696 | 0.9353 | 0.7200 | 0.9029 |
| Credit Card | **0.9993** | ***0.9992*** | 0.6503 | 0.9923 | 0.9241 | 0.9505 | 0.9860 | 0.9889 | 0.6123 | 0.7284 |
| Drug Classification | **1.0000** | *0.9378* | 0.8655 | 0.8277 | 0.9714 | 0.9361 | 0.6401 | 0.8328 | 0.2006 | 0.9031 |
| Kc2 | 0.9267 | *0.8698* | 0.4658 | 0.9572 | **0.9996** | 0.6335 | 0.4017 | 0.1900 | 0.1407 | 0.5908 |
| Eeg eye | **0.9929** | ***1*** | 0.7578 | 0.9216 | 0.9576 | 0.9049 | 0.9327 | 0.0700 | 0.6646 | 0.7257 |
| Glass Classification | 0.9153 | *0.9022* | 0.8050 | **0.9739** | 0.8984 | 0.8166 | 0.6440 | 0.1300 | 0.2665 | 0.7204 |
| Pc4 | **0.9981** | ***0.9971*** | 0.9755 | 0.9904 | 0.9885 | 0.9059 | 0.8811 | 0.9207 | 0 | 0.9287 |
| Madelon | **0.9554** | ***0.9563*** | 0.4063 | 0.6586 | 0.8614 | 0.6449 | 0.1750 | 0.0192 | 0 | 0.7171 |
| Turing binary | **0.9796** | ***0.9799*** | 0.0162 | 0.6648 | 0.9757 | 0.4800 | 0.2519 | 0.2176 | 0.0541 | 0.0140 |
| KDD | **1** | ***1*** | 0.9987 | 0.8649 | 0.9896 | 0.2506 | 0.9802 | 0.9794 | 0 | 0.9830 |
| Liver disorder | **0.9860** | *0.9830* | 0.9835 | 0.9713 | 0.9835 | 0.9134 | 0.9206 | 0.9135 | 0.1604 | 0.9365 |
| Wine | **0.9584** | ***0.9720*** | 0.8302 | 0.9111 | 0.9187 | 0.9218 | 0.7678 | 0.9296 | 0.0041 | 0.7702 |
| Soy bean | **0.9888** | ***0.9864*** | 0.4845 | 0.9309 | 0.6332 | 0.5975 | 0.0286 | 0.0288 | 0.0096 | 0.3968 |
| Ionosphere | **0.9733** | *0.6763* | 0.9292 | 0.7239 | 0.9589 | 0.9209 | 0.8954 | 0.9137 | 0 | 0.9397 |
| Room Occupancy | **1** | ***1*** | 0.9971 | 0.9962 | 0.9971 | 0.9936 | 0.9943 | 1 | 0 | 0.9948 |
| Harth | **0.9980** | ***0.9976*** | 0.9750 | 0.7336 | 0.9828 | 0.9190 | 0.9800 | 0.9317 | 0 | 0.9749 |
| Rocket League | **0.9336** | ***0.9362*** | 0.0759 | 0.5431 | 0.3916 | 0.0890 | 0.1240 | 0 | 0 | 0 |
| Sirtuin6 | **0.9554** | ***0.9589*** | 0.6373 | 0.7638 | 0.9509 | 0.7683 | 0.9200 | 0.9235 | 0 | 0.9182 |
| Toxicity | **0.9323** | ***0.9314*** | 0.0386 | 0.5924 | 0.4527 | 0.1049 | 0.0100 | 0 | 0 | 0 |
| Dry bean | **0.9961** | ***0.9952*** | 0.8161 | 0.9503 | 0.9639 | 0.9702 | 0.0353 | 0.9283 | 0 | 0.8880 |

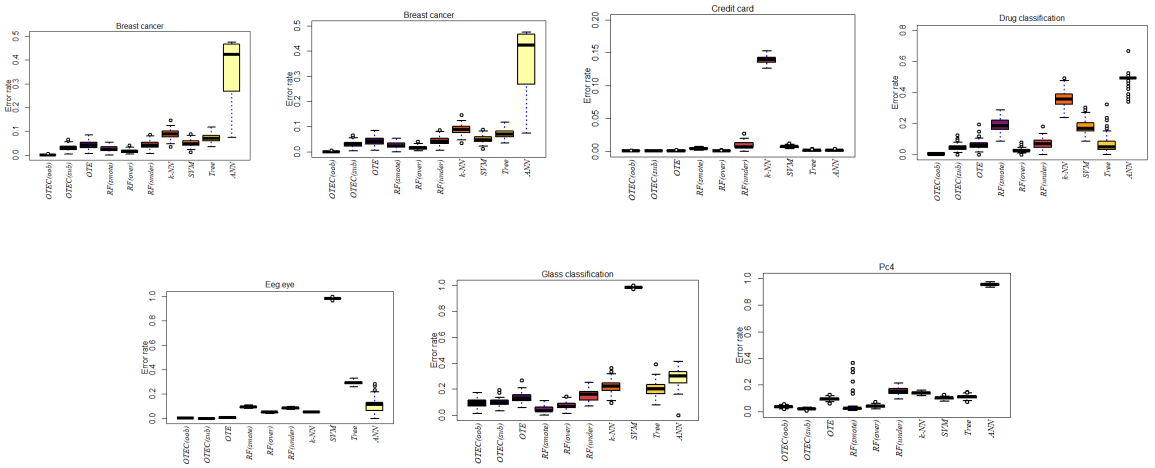
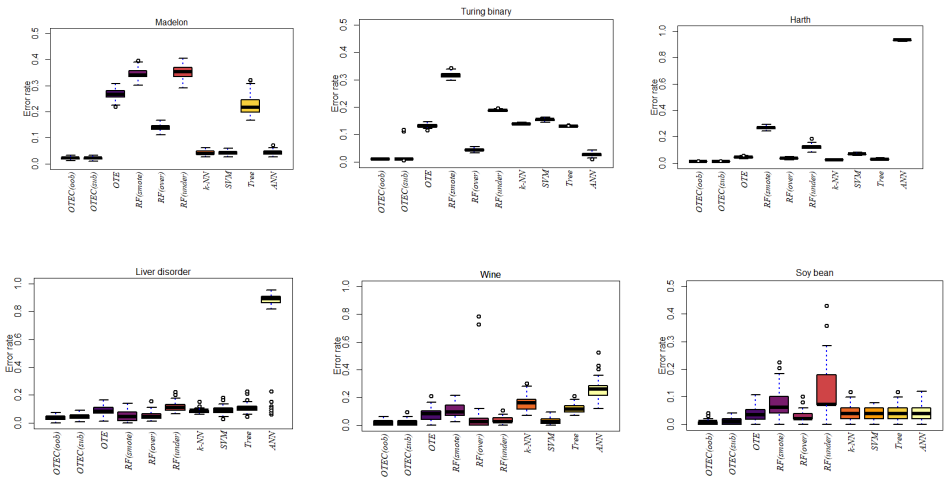



**Figure 1.** Box-plots of OTEC(oob), OTEC(sub) with other state-of-the-art method in term of classification error rate on various datasets.



Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2024 doi:10.20944/preprints202407.1684.v1



**Figure 2.** Box-plots of OTEC(oob), OTEC(sub) with other state-of-the-art method in term of classification error rate on various datasets.

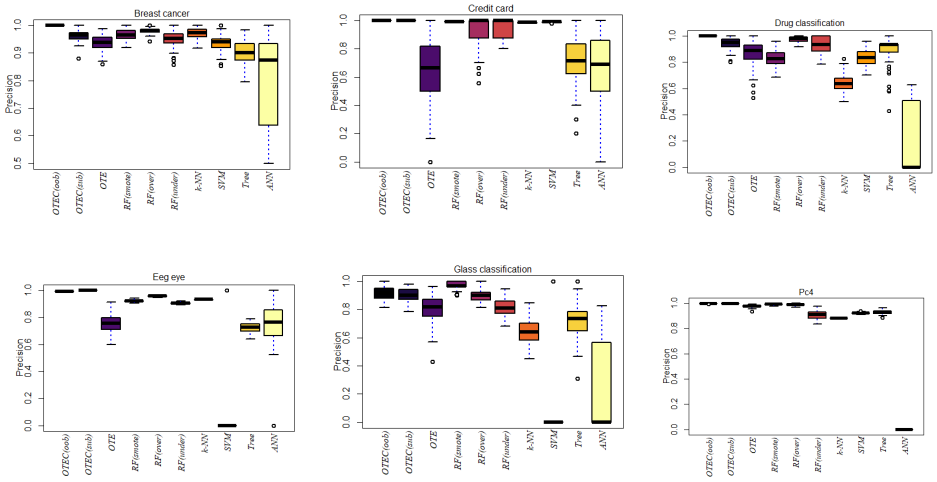



**Figure 3.** Box-plots of OTEC(oob), OTEC(sub) with other state-o-the-art method in term of precision on various datasets.

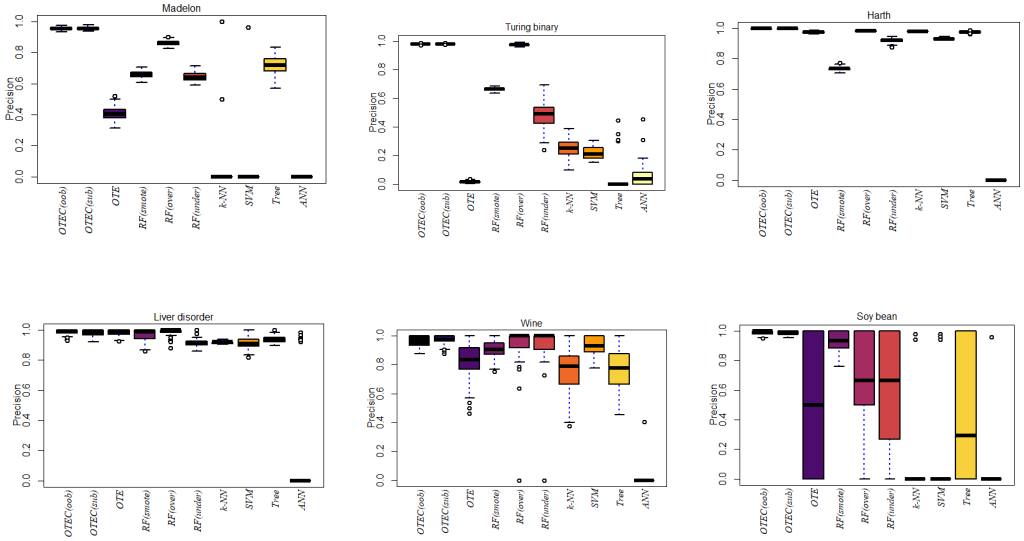



**Figure 4.** Box-plots of OTEC(oob), OTEC(sub) with other state-o-the-art method in term of precision on various datasets.

The parameter W, which represents the percentage of the best trees selected for the final ensemble, is crucial for the OTEC(oob) and OTEC(sub) methods. The diverse values of W indicate the different behaviors of the method. This study assessed the impact of W = (10%, 30%, 50%, 70%, 90%) of the total number of trees on the classification method. As shown in Figure 5, subsets of the total number of trees ranging from 10% to 90% were used to calculate the classification error rate. For each dataset, Figure 5 shows that an increase in the number of trees used by the ensemble decreased the OTEC(oob) classification error rate. Similar to OTEC(oob), Figure 6 shows that the classification error rate of OTEC(sub) decreased with an increase in the number of trees in the ensemble. In Figures 5–6, the x-axis represents the percentage of the best trees selected, while the y-axis represents classification error rate results.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2024 doi:10.20944/preprints202407.1684.v1



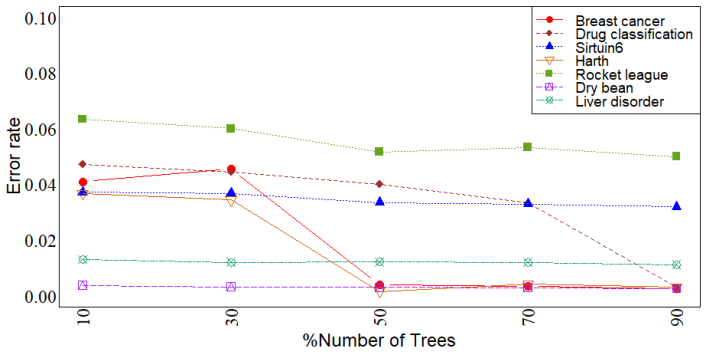



**Figure 5.** Multi-line plot shows the percentage of the best trees in the ensemble with the classification error rate of the proposed method, OTEC (oob).

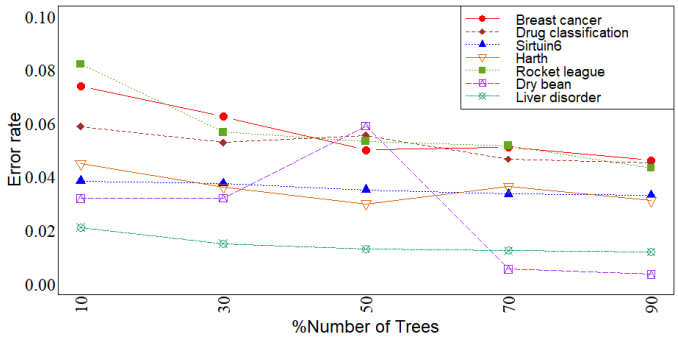



**Figure 6.** Multi-line plot shows the percentage of the best trees in the ensemble with the classification error rate of the proposed method, OTEC (sub). .

## 4. Simulation

In this section, we present two simulation scenarios for the proposed method. The first scenario is intended to show the circumstances in which the proposed method is useful, whereas the second scenario shows a data-generating setting where the proposed method might not be suitable. In total, 10,000 instances across 19 variables were synthetically generated. Each of these variables generates observations as a multivariate normal distribution with different means and variances, whereas the other generates observations with a multinomial distribution with four categories. To generate observations from a binary response, the first imbalance ratio specifies the imbalance in the observations. The binary response, $\mathbf{Y} = \mathbf{B}(\mathbf{p})$ , is generated using a logit-type function given the variables $\mathbf{P}(\mathbf{y}|\mathbf{X})$:

$$\mathbf{p}(\mathbf{y_{imb}}|\mathbf{X}) = \frac{\exp(\mathbf{y_{imb}})}{\mathbf{1}+\exp(\mathbf{1}-\mathbf{y_{imb}})}.$$

The imbalance ratio of the number of instances in the majority class compared to the minority class observations $\mathbf{n^0}$ was 5.67, of which 8500 belonged to the majority class and 1500 to the minority class. The values of $\mathbf{y}$ used in Equation 2 are as follows:

$$\mathbf{y} = \mathbf{X\beta} + \epsilon$$

This setup was used to generate 10,000 observations. All the methods presented in this study were applied using the same experimental setup as that used for the benchmark datasets. The second model was constructed similarly. The class-wise distribution was 50/50, where a class ratio of 1.02 indicated that the data were balanced. The difference between the two models was that the former contained an imbalanced class distribution, whereas the latter did not. A total of 100 realizations were performed. The proposed methods, OTEC(oob) and OTEC(sub), outperformed the other rivals in the first scenario. In the second scenario, OTEC(oob) and OTEC(sub) did not perform well because there was no class imbalance problem. Bar plots of the proposed method on the simulated datasets are

 doi:10.20944/preprints202407.1684.v1



shown in Figures 7–8. The results demonstrate that the proposed methods are appropriate in the presence of a severe class imbalance problem in the data.

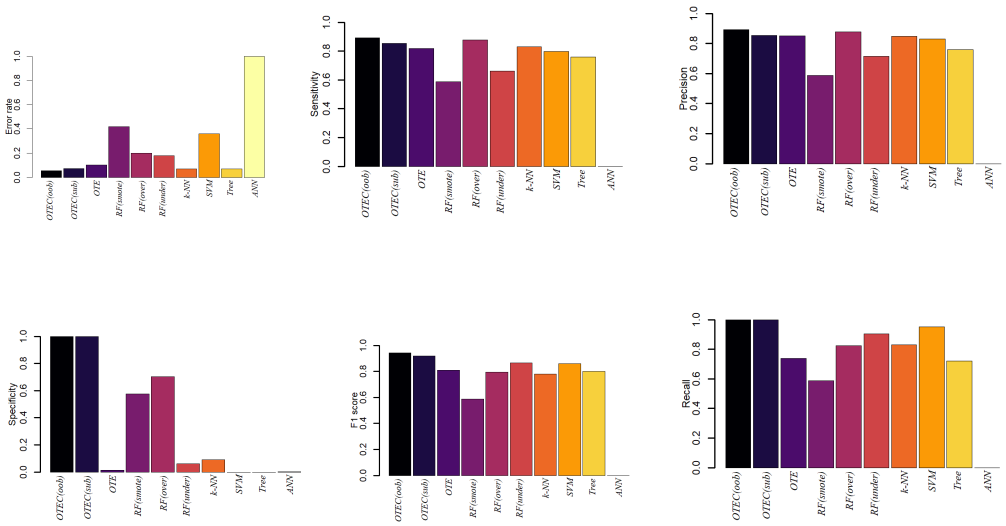



**Figure 7.** Bar plots of the proposed method, i.e., OTEC(oob), OTEC(sub) comparing with other state-of-the-art methods in terms of classification error rate, sensitivity, specificity, precision, recall, and F1 score on the imbalanced simulated dataset.

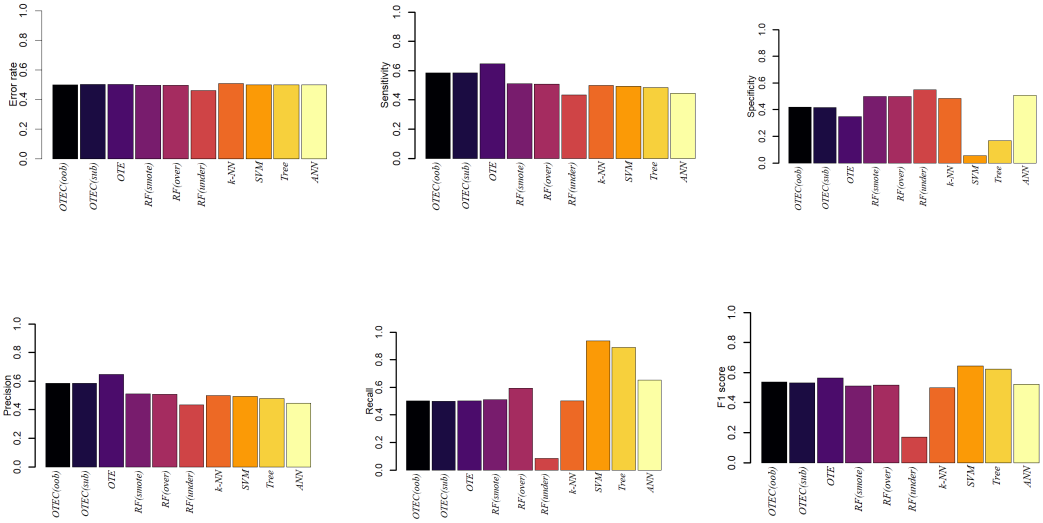



**Figure 8.** Bar plots of the proposed method, i.e., OTEC(oob), OTEC(sub) comparing with other methods in terms of classification error rate, sensitivity, specificity, precision, recall, and F1 score on the balanced simulated dataset.

## 5. Conclusions

The uneven distribution of observations into two classes, where one of the classes significantly outperforms the other, adversely affects the performance prediction of machine-learning methods. Therefore, balancing the data by synthetically generating additional observations for minority classes, a class that is usually of high interest, along with model selection for a decision-tree ensemble, has resulted in improved prediction performance. The proposed ensemble algorithm, along with synthetic data generation for data balancing, effectively solved the key challenge of extremely imbalanced classification problems.

 doi:10.20944/preprints202407.1684.v1



The two methods proposed in this study address the imbalanced data problem, as shown by the analysis of several benchmark and simulated datasets. The proposed methods outperformed conventional machine-learning tools, such as optimal tree ensemble, random forest with SMOTE, random forest with under-sampling, random forest with oversampling, k-NN, SVM, classification tree, and artificial neural networks, in terms of error rate, sensitivity, specificity, precision, recall, and F1-score. With the help of the proposed method, the performance of machine-learning models on extremely imbalanced binary classification problems can be significantly improved, producing more dependable and robust classification results in practical applications.

For future work in the direction of this paper, one may consider the use of SMOTE and ADASYN (Adaptive Synthetic Minority Oversampling Technique) for data augmentation methods customized to the dataset. Using these methods, the performance of the model can be improved by adding artificial data points for the minority class. Using the active learning strategy, the model may be trained to automatically search for new data points, particularly from the minority class. This will allow the underrepresented class most informative data to be the model's key focus. Feature engineering methods that can generate new features that are more informative for the underrepresented class in conjunction with the proposed method might improve the performance further. Feature selection could also be used to determine which features are most relevant for classification in an unbalanced context. Other methods, such as cost-sensitive learning, could be combined with the regularized tree forest. Cost-sensitive learning enables the model to concentrate its performance on underrepresented data by providing greater weight to minority class misclassifications.

**Supplementary Materials:** The following supporting information can be downloaded at the website of this paper posted on Preprints.org. Table A1, Figures A1–A14.

**Author Contributions:** Conceptualization, S.S.; methodology, S.S. and S.G.; software, S.G.; validation, S.S. and S.G.; formal analysis, S.G. and S.S.; investigation, S.S. and S.G.; resources, S.S.; data curation, S.S., and S.G.; writing—original draft preparation, S.S., and S.G.; writing—review and editing, S.G. and S.S.; visualization, S.S. and S.G.; supervision, S.S.; project administration, S.S.; funding acquisition, S.S. Both authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the START-UP Research, grant number 12B025, College of Business and Economics at the United Arab Emirates University.

**Data Availability Statement:** The data supporting the findings of this study are available upon request. Interested researchers may contact the corresponding author to obtain access to the data for further analysis and validation.

**Conflicts of Interest:** The authors declare no conflicts of interest. The authors have no relevant financial or non-financial interests to disclose.

**Acknowledgments:** The authors gratefully acknowledge funding support from the United Arab Emirates University, which made this research possible.

## References

1. Weiss, G. M. (2004). Mining with rarity: a unifying framework. ACM Sigkdd Explorations Newsletter, 6(1), 7-19.
2. Parmar, J., Chouhan, S., Raychoudhury, V., & Rathore, S. (2023). Open-world machine learning: applications, challenges, and opportunities. ACM Computing Surveys, 55(10), 1-37.
3. Wang, L., Han, M., Li, X., Zhang, N., & Cheng, H. (2021). Review of classification methods on unbalanced data sets. IEEE Access, 9, 64606-64628.
4. de Giorgio, A., Cola, G., & Wang, L. (2023). Systematic review of class imbalance problems in manufacturing. Journal of Manufacturing Systems, 71, 620-644.
5. Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. Handling imbalanced datasets: A review, gests international transactions on computer science and engineering 30 (2006) 25–36. Synthetic Oversampling of Instances Using Clustering.
6. Visa, S., & Ralescu, A. (2005, April). Issues in mining imbalanced data sets-a review paper. In Proceedings of the sixteen midwest artificial intelligence and cognitive science conference (Vol. 2005, pp. 67-73). sn.

 doi:10.20944/preprints202407.1684.v1




7. Monard, M. C., & Batista, G. E. A. P. A. (2002). Learning with skewed class distributions. Advances in Logic, Artificial Intelligence and Robotics, 85, 173-180.
8. Prince, M., & Prathap, P. J. (2023). An imbalanced dataset and class overlapping classification model for big data. Comput. Syst. Sci. Eng., 44(2), 1009-1024.
9. Kotsiantis, S. B., & Pintelas, P. E. (2003). Mixture of expert agents for handling imbalanced data sets. Annals of Mathematics, Computing & Teleinformatics, 1(1), 46-55.
10. Improving identication of di-cult small classes by balancing class distribution. In Articial Intelligence in Medicine: 8th Conference on Articial Intelligence in Medicine in Europe, AIME 2001 Cascais, Portugal, July 14, 2001, Proceedings 8, pages 6366. Springer, 2001.
11. Han, H., Wang, W. Y., & Mao, B. H. (2005, August). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878-887). Berlin, Heidelberg: Springer Berlin Heidelberg.
12. Han, H., Wang, L., Wen, M., & Wang, W. Y. (2006). Oversampling Algorithm Based on Preliminary Classification in Imbalanced Data Sets Learning. Journal of computer allocations (in Chinese), 26(8), 1894-1897.
13. Kubat, M., & Matwin, S. (1997, July). Addressing the curse of imbalanced training sets: one-sided selection. In Icml (Vol. 97, No. 1, p. 179).
14. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
15. Weiss, G. M. (2003). The effect of small disjuncts and class distribution on decision tree learning. Rutgers The State University of New Jersey, School of Graduate Studies.
16. Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. Journal of artificial intelligence research, 19, 315-354.
17. Jo, T., & Japkowicz, N. Class imbalances versus small disjuncts. ACM SIGKDD Explor. Newsl. 6 (1), 40–49 (2004).
18. Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter, 6(1), 20-29.
19. Leevy, J. L., Hancock, J., & Khoshgoftaar, T. M. (2023). Comparative analysis of binary and one-class classification techniques for credit card fraud data. Journal of Big Data, 10(1), 118.
20. Wongvorachan, T., He, S., & Bulut, O. (2023). A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. Information, 14(1), 54.
21. Chaabane, I., Guermazi, R., & Hammami, M. (2020). Enhancing techniques for learning decision trees from imbalanced data. Advances in Data Analysis and Classification, 14, 677-745.
22. Ksieniewicz, P. (2018, November). Undersampled majority class ensemble for highly imbalanced binary classification. In Second International Workshop on Learning with Imbalanced Domains: Theory and Applications (pp. 82-94). PMLR.
23. Du, H., Zhang, Y., Zhang, L., & Chen, Y. C. (2023). Selective ensemble learning algorithm for imbalanced dataset. Computer Science and Information Systems, (00), 23-23.
24. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.
25. Sebbanü, M., NockO, R., Chauchat, J., & Rakotomalala, R. (2000). Impact of learning set quality and size on decision tree performances. IJCSS, 1(1), 85.
26. Khan, Z., Gul, A., Mahmoud, O., Miftahuddin, M., Perperoglou, A., Adler, W., & Lausen, B. (2016). An ensemble of optimal trees for class membership probability estimation. In Analysis of Large and Complex Data (pp. 395-409). Springer International Publishing.
27. Khan, Z., Gul, A., Perperoglou, A., Miftahuddin, M., Mahmoud, O., Adler, W., & Lausen, B. (2020). Ensemble of optimal trees, random forest and random projection ensemble classification. Advances in Data Analysis and Classification, 14, 97-116.
28. Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. Journal of the American Statistical Association, 92(438), 548-560.
29. Adler, W., & Lausen, B. (2009). Bootstrap estimated true and false positive rates and ROC curve. Computational statistics & data analysis, 53(3), 718-729.
30. Johnson, R. W. (2001). An introduction to the bootstrap. Teaching statistics, 23(2), 49-54.