electronics

MDPI

Article

KDE-Based Ensemble Learning for Imbalanced Data

Firuz Kamalov [1,*], Sherif Moussa [1] and Jorge Avante Reyes [2]

[1] Department of Electrical Engineering, Canadian University Dubai, Dubai 117781, United Arab Emirates
[2] Bosch AG, 70469 Stuttgart, Germany
* Correspondence: firuz@cud.ac.ae

Abstract: Imbalanced class distribution affects many applications in machine learning, including medical diagnostics, text classification, intrusion detection and many others. In this paper, we propose a novel ensemble classification method designed to deal with imbalanced data. The proposed method trains each tree in the ensemble using uniquely generated synthetically balanced data. The data balancing is carried out via kernel density estimation, which offers a natural and effective approach to generating new sample points. We show that the proposed method results in a lower variance of the model estimator. The proposed method is tested against benchmark classifiers on a range of simulated and real-life data. The results of experiments show that the proposed classifier significantly outperforms the benchmark methods.

Keywords: imbalanced data; kernel density estimate; ensemble method; data sampling

Citation: Kamalov, F.; Moussa, S.; Avante Reyes, J. KDE-Based Ensemble Learning for Imbalanced Data. *Electronics* **2022**, *11*, 2703. https://doi.org/10.3390/electronics11172703

Academic Editors: Maja Matetic, Xiaoshuan Zhang and Marija Brkić Bakarić

Received: 23 July 2022
Accepted: 23 August 2022
Published: 29 August 2022

1. Introduction

Imbalanced datasets stem from skewed a distribution of class labels in data. It is a frequent occurrence in several fields, including fraud detection, text categorization, medical diagnostics and many others [1,2]. In general, any field that involves rare events would produce an imbalanced class distribution. Traditional classification algorithms struggle to properly handle such data due to inherent bias in their algorithms. In particular, the majority of the current algorithms aim to minimize the total prediction error. Thus, a classifier will be more inclined to concentrate on the majority class labels as they make up a greater portion of total labels. As a result, the minority class labels receive less attention during training [3]. Meanwhile, the minority class labels are frequently of more significance than the majority class labels. For instance, in healthcare, it is more critical to diagnose patients with a disease even though they make up only a small fraction of all patients. In fraud detection, while only a small portion of transactions are fraudulent, they cause the most amount of damage. Therefore, correctly classifying the minority class instances is often of greater need than the majority class instances.

The most commonly used approach to deal with imbalanced data is through sampling. Sampling can be divided into two categories: undersampling and oversampling. In undersampling, a random fraction of the majority class is sampled to balance with the minority class. There exist various implementations of this approach, including Random Undersampling and NearMiss. In oversampling, the minority class size is increased to the level of the majority class. Popular oversampling methods include SMOTE, ADASYN and Random Oversampling. After balancing, the data are used to train a classifier. More recently, a new oversampling approach based on Kernel Density Estimation (KDE) has been proposed. KDE has been shown to be an effective method for sampling balanced data. The KDE sampling algorithm operates by approximating the intrinsic distribution of the minority points and using it to generate new minority samples. It provides a natural and efficient approach to equalizing class sizes. While most of the existing sampling methods are based on specific distribution models, KDE is a nonparametric approach that makes no model assumptions. Thus, it is less likely to suffer from the model selection bias.

Data resampling naturally lends itself to bootstrap aggregation. We can repeatedly sample the data to obtain a balanced training set and build an individual decision tree at each iteration. The resulting collection of trees is employed to make predictions on unseen data using the majority vote rule [4]. Bootstrap aggregating has been shown to be an effective classification algorithm for balanced data. In this paper, we apply bootstrap aggregation to imbalanced data using the KDE sampling technique. Concretely, we construct a collection of decision trees, where each tree is trained on a KDE-balanced set. Then new predictions are made via majority vote from the predictions of the individual decision trees.

We evaluate the effectiveness of the proposed KDE-based ensemble classifier on several synthetic and real-life datasets. As a benchmark, we employ single decision tree and random forest classifiers. The results of extensive experiments reveal that the proposed approach often performs better than the benchmark classifiers. We conclude that given the sound theoretical underpinnings of KDE theory and the results of the experiments, the proposed ensemble approach would be a valuable classification tool in the context of imbalanced data.

The proposed ensemble method offers three key contributions:

1. Each tree in the ensemble is trained using uniquely generated, synthetically balanced data. Consequently, the variance of the ensemble predictor is reduced (Equation (7)).
2. KDE sampling does not make any modeling assumptions regarding the distribution of the data. It can be applied in any scenario [5–7].
3. The numerical experiments demonstrate the effectiveness of the proposed approach over the standard benchmarks.

Our paper is organized as follows. Section 2 contains a brief overview of the relevant literature. In Section 3, we discuss the theoretical underpinnings of the KDE theory. Section 4 describes the proposed ensemble classification algorithm. In Section 5, we produce the results of the numerical experiments to evaluate the effectiveness of the proposed method. Section 6 concludes the paper.

2. Literature Review

There exist several approaches designed to address the issue of imbalanced class distribution. The most popular approach to dealing with skewed class distribution is based on equalizing the number of samples in each class. Resampling is arguably the most effective method of achieving a balanced dataset. Resampling techniques can be split into two categories: undersampling and oversampling. In undersampling, a subset of the majority class—of the same size as the minority class—is randomly selected. Random undersampling (RUS)—where a portion of the majority class is randomly selected with uniform probability—is the simplest undersampling technique. In NearMiss, a more sophisticated approach is used to sample from the majority class, where the points that are close to the border with the minority class are more likely to be chosen [8]. In oversampling, the new minority samples are generated based on the existing minority data. A widely used oversampling approach called SMOTE produces new points via uniform linear interpolation between the existing minority points [9,10]. An extension of the SMOTE algorithm called ADASYN generates new samples in a similar fashion as SMOTE, albeit with a greater focus on the minority points that lie in regions with a high level of majority samples [11]. In another extension, the authors introduce a new data augmentation method called H-SMOTE and apply it to few-shot classification problems [12,13]. A novel approach based on kernel density estimation is applied to generate new minority samples in [14]. The proposed method works by estimating the fundamental probability distribution of the minority points. Afterward, the estimated probability distribution function is used to create synthetic samples. Fusion methods that integrate several algorithms have also been employed to handle imbalanced data [15,16].

Ensemble methods combine a collection of individual classifiers into a single algorithm [17]. The main goal of the ensemble approach is to reduce the variance of the predictor

with minimal detriment to the bias. At the core of most of the ensemble approaches is the concept of bootstrap aggregating (bagging), where a number of decision tree classifiers are trained on different subsets of the original data.

Ensemble approaches have been applied in a range of applications. In [18], the authors use a two-step ensemble approach to identify COVID-19 patients based on cough sounds. The authors split each sound recording into segments and use them as inputs to shallow convolutional neural networks. Ensemble methods have been actively employed in designing intelligent intrusion detection systems. In particular, random forest has been shown to be an effective IDS, including for imbalanced data [19]. Stacking ensemble learning was used to combine multiple feature representations extracted via a graph neural network to diagnose autism spectrum disorder [20]. The authors in [21] investigate the effect of ensemble learning as a light approach to enhancing the out-of-distribution generalization of machine reading comprehension systems by combining the outputs of some pre-trained base models without retraining a big model. With the rise in popularity of deep learning models, several researchers have combined random forest with neural networks [22,23].

One of the simplest bagging approaches for imbalanced data is undersampling. The authors in [24] use evolutionary undersampling on the majority class to extend the under-bagging ensemble method. Their method proposes to construct base classifiers using new subsets of the majority class that are sampled using an evolutionary approach. The results showed that the proposed ensemble method performed adequately on highly imbalanced data. The authors in [25], introduced Roughly Balanced Bagging (RBB), where the number of samples in each class is determined differently. Concretely, in each bootstrap set, the number of minority instances equals the number of the original minority samples, while the size of the majority class is determined based on the negative binomial distribution. RBB is a popular method and has been used in various contexts [26]. Random class ratios in each bootstrap set were used to build an ensemble of classifiers in [27]. In [28], the authors proposed an ensemble method based on moving the threshold that preserves the natural class distribution of the data. Other approaches, including deep learning, have also been applied to deal with imbalanced data [29].

3. KDE Sampling

Nonparametric probability density estimation is a critical technique in statistical data analysis. It is employed to mathematically model the probability density function of a multivariate random variable given a random sample set.

The estimated density function can be employed to analyze various attributes of the random variable. Let $\{x_1, x_2, \ldots, x_n\}$ be an independent identically distributed (i.i.d.) sample selected from an unknown probability density distribution f. Then, the kernel density estimate (KDE) of f is given by the following equation

$$\tilde{f}(x) = \frac{1}{n} \sum_{j=1}^{n} K_h(x - x_j), \tag{1}$$

where h is the bandwidth parameter, K is the kernel function and $K_h(t) = \frac{1}{h} K(\frac{t}{h})$ is the similarity measure between a pair of points. The bandwidth parameter plays an important role in density estimation. As shown in Figure 1, a small value of h leads to low bias but high variance, while a large value of h leads to high bias but low variance.

The optimal value of the bandwidth is identified using cross-validation. Concretely, grid search is applied to determine the value of h that minimizes the mean integrated square error (MISE) of the sample:

$$\text{MISE}_n(h) = \frac{1}{n} \sum_{i=1}^{n} (\tilde{f}(x_i) - f(x_i))^2. \tag{2}$$

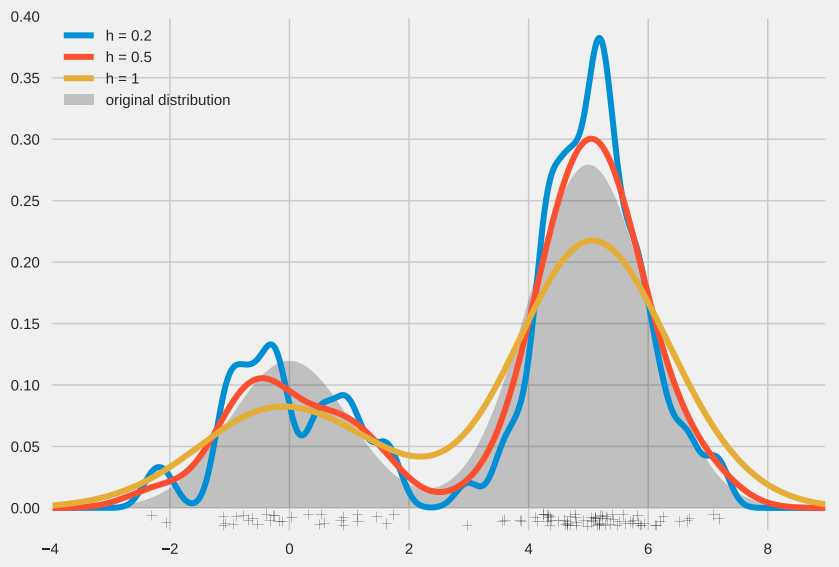

Figure 1. The grey curve represents the original (underlying) distribution of the data. The colored curves represent the KDE estimations of the original curve for different values of h.

Alternatively, the rule of thumb approach can also be used to estimate the optimal h. For instance, according to Scott's rule of thumb, the optimal value of h is given by:

$$h = n^{-\frac{1}{5}} \cdot s, \tag{3}$$

where s is the standard deviation of the sample.

The multivariate KDE is very similar to the one-dimensional approach described above. Let $\{x_1, x_2, \ldots, x_n\}$ be a d-dimensional random sample of vectors drawn from a density distribution f. Then the kernel density estimate is given by the following equation

$$\tilde{f}_H(x) = \frac{1}{n} \sum_{j=1}^{n} K_H(x - x_j), \tag{4}$$

where H is a $d \times d$ bandwidth matrix. The bandwidth matrix can be determined using the multivariate version of Scott's rule, among other approaches:

$$H = n^{-\frac{1}{d+4}} \cdot S, \tag{5}$$

where S is the covariance matrix. In our paper, we employ the multivariate normal density distribution as the kernel function:

$$K_H(x) = \frac{1}{\sqrt{(2\pi)^d |H|}} e^{-\frac{1}{2} x^T H^{-1} x}. \tag{6}$$

The KDE technique is a well-established method. It offers a flexible approach to modeling density distributions. It has been used in a variety of applications [14].

4. KDE Ensemble

In this section, we discuss the details of the proposed ensemble algorithm for imbalanced data. The algorithm consists of two main components: ensemble of decision trees and KDE-based sampling. In the ensemble approach, a collection of trained decision trees are used to make a prediction. The KDE technique is used to balance the training data for each individual tree.

Let m be the number of decision tree estimators used in the ensemble. For each decision tree T_i, we choose a subset X_i of the data. Since the original data is assumed to be imbalanced, the sample subset X_i will also be imbalanced. We apply KDE with the

Gaussian kernel and Scott's bandwidth to balance each training set X_i, $i = 1, 2, \ldots, m$. Then each decision tree T_i is trained on the balanced set $\tilde{X}_i$. The output of an ensemble classifier is calculated based on the mode of predictions of individual trees. The pseudocode for the proposed approach is presented in Algorithm 1 below.

Algorithm 1 Pseudocode for the proposed KDE ensemble classifier

X = original (imbalanced) data
m = number of decision trees
T_i = individual decision tree
for $i = 1, 2, \ldots, m$ **do**
    Choose a random subset X_i with replacement
    Balance X_i using KDE oversampling
    Train T_i on X_i
end for
Combine $\{T_i\}$ into a single classifier via the majority vote

The proposed classification algorithm is further illustrated in Figure 2. The original dataset is composed of the majority (blue) and minority (red) points. Random uniform selection with replacement is applied to construct a training subset X_i for each decision tree T_i. The KDE technique is used to balance each training set X_i. Concretely, we synthetically increase the number of minority samples to the level of the majority samples. The KDE method has the advantage of reducing the chances of overfitting the new sample points compared to other sampling methods.

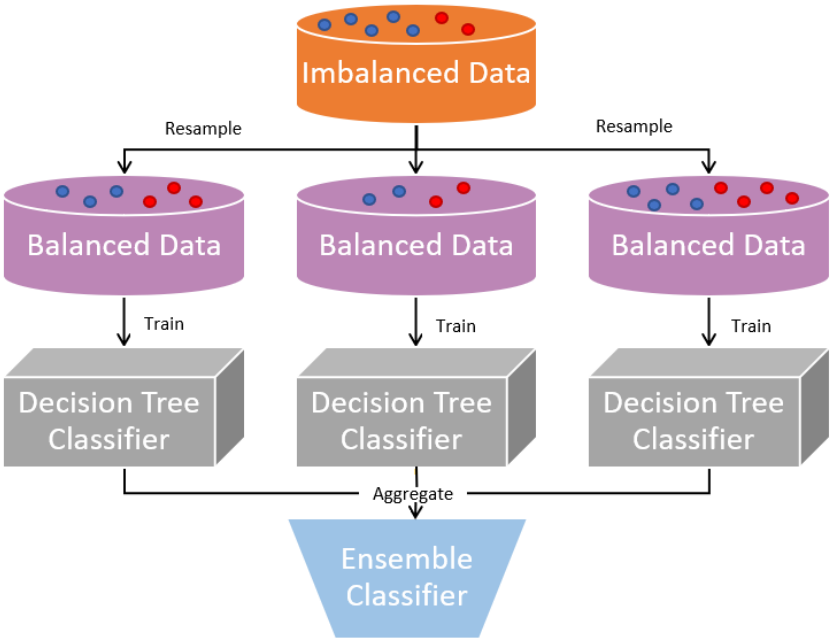

Figure 2. KDE-based ensemble classification algorithm. The minority points (red) are resampled using the KDE technique to balance the training sets.

There exists a variety of kernel functions that can be implemented in the estimator. The proposed algorithm utilizes the Gaussian kernel, where the bandwidth is given by Equation (3). An individual decision tree classifier T_i is trained on each training subset. Given a new test point, we pass it onto each decision tree. The predicted value of the ensemble is determined by the majority vote. The proposed bootstrapping process produces a better model performance as it reduces the variance of the model without increasing the bias.

KDE-based resampling leads to a reduction in the variance of the model estimator. Let $\{\phi_m\}_{k=1}^{M}$ be a collection of base classifiers, and $\Phi = \frac{1}{M}\Sigma\phi_k$ be the ensemble classifier. Then

$$\begin{aligned}(\Phi) &= \mathrm{Cov}(\frac{1}{M}\Sigma\phi_k, \frac{1}{M}\Sigma\phi_l) \\ &= \frac{1}{M^2}\Sigma_{k,l=1}\mathrm{Cov}(\phi_k, \phi_l) \\ &= \frac{1}{M}\Sigma_k\mathrm{Cov}(\phi_k, \phi_k) + \frac{1}{M^2 - M}\Sigma_{k\neq l}Cov(\phi_k, \phi_l) \\ &= \frac{1}{M}\Sigma_k(\phi_k) + \frac{1}{M^2 - M}\Sigma_{k\neq l}\mathrm{Cov}(\phi_k, \phi_l).\end{aligned} \tag{7}$$

It follows from Equation (7) that the ensemble variance depends on the variance of the individual base learners as well as their pairwise covariance. The variance of a base classifier hinges on the variance of underlying sampled data. The variance of the sampled data can be measured as the expected cumulative distance between two samples. Let $\{x_i\}_{i=1}^{n}$ and $\{x_i'\}_{j=1}^{n}$ be randomly selected samples. Let S and $\tilde{S}$ denote the cumulative distance between the original samples and their KDE-balanced counterparts, respectively. Then,

$$\begin{aligned}S &= \Sigma_{i,j}|x_i - x_j'| \\ &\geq \int\int(\tilde{f} - \tilde{f}')\,dx\,dx' \\ &= \tilde{S}\end{aligned} \tag{8}$$

It follows from Equation (8) that KDE-sampled data leads to a reduction in sample-to-sample variance. Thus, the proposed method leads to a lower variance in the model estimator.

The theoretical time complexity of the KDE ensemble method is similar to that of random forest. To simplify the discussion, we concentrate on the difference in time complexity of a single decision tree between KDE ensemble and random forest. Let X be a given dataset with n samples and dimension d. Let n_M be the number of the majority samples in X. Then, the time complexity of a single decision tree in random forest is given as $\mathcal{O}(n\log(n)d) + \mathcal{O}(nd)$, which asymptotically equals $\mathcal{O}(n\log(n)d)$. On the other hand, the time complexity of a single decision tree in the KDE ensemble is given by

$$\mathcal{O}(2n_M\log(2n_M)d), \tag{9}$$

where $\mathcal{O}(2n_M\log(2n_M))$ is the time complexity due to sorting the balanced data in one dimension.

5. Numerical Experiments

To evaluate the performance of the proposed ensemble method, we test it on several imbalanced datasets and compare the results against the benchmarks.

5.1. Experimental Design

The experiments are based on both synthetic and real-life datasets to test the effectiveness of the proposed algorithm. As a benchmark, we employ decision tree and random forest classifiers. Decision tree is a single basic classifier that repeatedly splits the data at different cutoff points of features according to the minimum Gini impurity value. Random forest is an ensemble classifier consisting of several individual decision trees. Each tree in random forest is constructed based on a randomly sampled subset of the original dataset. In addition, during the splitting process, the features considered for splitting are also randomly chosen. Random forest has the advantage of lower variance over a single decision tree, and it is less likely to overfit the data.

Each real-life dataset is split into training/testing sets, preserving the class ratios. The training set consists of 75% of the data, with the remaining dedicated to testing. After training and testing each classifier (KDE, DT, RF), we calculate the corresponding AUC and

F_1-score. To ensure the robustness of the results, we repeat the entire process five times. In other words, we use five different combinations of training and testing sets. We calculate the F_1-score and AUC of a classifier on each train/test combination and report the average results.

The traditional measures of performance, such as the error rate and accuracy, fail to properly capture the effectiveness of a classifier on imbalanced data. One of the most common classification metrics used with imbalanced data is Area Under Curve (AUC), which represents the likelihood of the classifier ranking a randomly chosen positive instance above a randomly chosen negative instance [30]. Another frequently employed used with imbalanced data is the F_1-score. The F_1-score combines precision and recall into a single balanced metric value. In particular, it is the harmonic mean of the precision and recall given by the equation

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{10}$$

To determine the statistical significance of our results, we apply hypothesis testing on two proportions. In particular, let $\hat{p}_1 = \frac{x_1}{n}$ and $\hat{p}_2 = \frac{x_2}{n}$ be the accuracies of classifier 1 and classifier 2, respectively, where n is the total number of samples and x_i is the number of correctly classified samples. Then, we obtain the Gaussian random variable

$$Z = \frac{\hat{p}_2 - \hat{p}_1}{\sqrt{2\hat{p}(1-\hat{p})/n}}, \tag{11}$$

where $\hat{p} = \frac{x_1+x_2}{2n}$. The test statistic Z can be used to check the null hypothesis $H_0 : \hat{p}_1 = \hat{p}_1$.

All the numerical experiments are conducted in Python using the machine learning library *sklearn*.

5.2. Simulated Data Experiments

The synthetic data consists of 100 minority and 1000 majority class points. The majority data are generated randomly according to a two-dimensional standard normal distribution with its center at $(0, 0)$. The minority points are also generated according to a two-dimensional standard normal distribution but with its center at (0.5, 0.5). The data are illustrated in Figure 3.

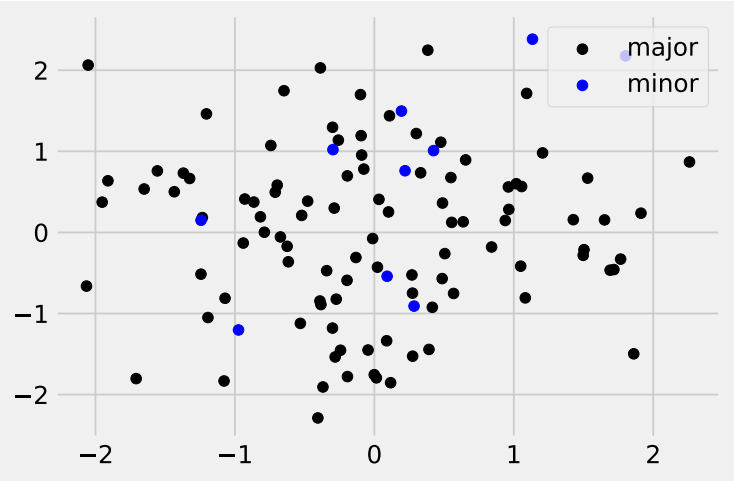

Figure 3. Distribution of points in the synthetically generated dataset.

We contrast the performance of the KDE-based ensemble to decision tree and random forest classifiers using the synthetic dataset. The results are presented in Table 1 below. As shown in Table 1, the KDE-based classifier significantly exceeds the benchmark classifiers both in terms of AUC and the F_1-score. The KDE classifier outscores DT and RF by 10% and 9%, respectively, based on the AUC. Similarly, the KDE classifier outscores DT and

RF by 16% and 20%, respectively, based on the F_1-score. The difference between the accuracy of the KDE ensemble and random forest is statistically significant, with a p-value of 1.30105×10^{-24}. The example of the synthetic dataset presented herein illustrates the effectiveness in balancing the training set. KDE has been demonstrated to be a potent balancing tool, and the current result further supports its use.

Table 1. Experimental results using the synthetic data described in Figure 3.

	KDE	DT	RF
AUC	0.585	0.485	0.495
F1	0.202	0.040	0.000

The second set of synthetic data again consists of 100 minority and 1000 majority class points. The majority data are generated randomly according to a three-dimensional standard normal distribution with its center at the origin. The minority points are also generated according to a three-dimensional standard normal distribution but with its center at (1.5, 1.5, 1.5). The data are illustrated in Figure 4.

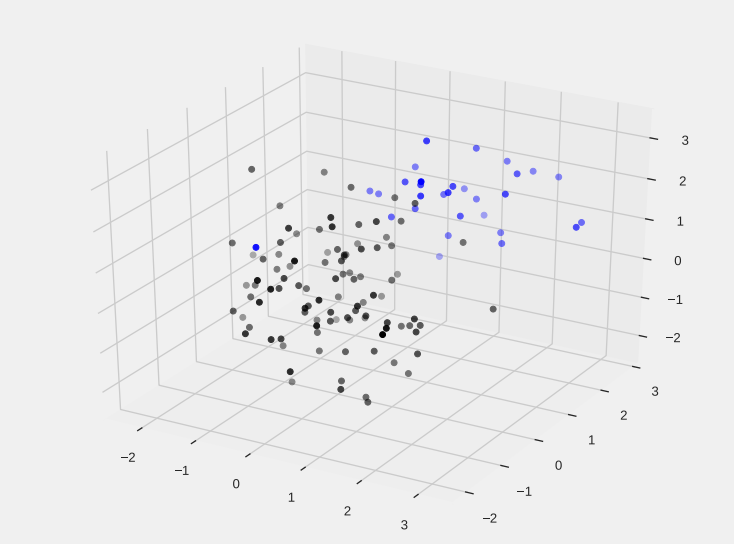

Figure 4. Distribution of points in the synthetically generated dataset.

As shown in Table 2, the KDE-based classifier significantly outperforms the benchmark classifiers in terms of AUC and partially the F_1-score. The difference between the accuracy of the KDE ensemble and random forest is statistically significant, with a p-value of 0.00063.

Table 2. Experimental results using the synthetic data described in Figure 4.

	KDE	DT	RF
AUC	0.833	0.696	0.808
F1	0.51	0.472	0.615

To further compare the performance of the proposed algorithm against the benchmarks, we consider the execution times. As shown in Table 3, the KDE-based ensemble is slower than an individual decision tree, but it is faster than random forest. Since the ensemble classifier trains multiple trees, it is not surprising that it is slower than an individual decision tree. It is more appropriate to compare the KDE ensemble to random forest, in which case the former displays faster execution. In particular, the KDE ensemble is significantly faster on both datasets than random forest.

Table 3. Execution times (in s) on the synthetic datasets.

	KDE	DT	RF
Data 1	0.3062	0.0048	0.6764
Data 2	0.1265	0.0041	0.5208

5.3. Real-Life Data Experiments

We use several real-life datasets to evaluate the effectiveness of the proposed ensemble classifier. There are a total of nine datasets that are divided into three subgroups. All the datasets used in our experiments are available from the UCI repository [31,32]. The descriptions of the datasets are provided in Table 4. We use a range of different datasets to ensure that we achieve comprehensive results. The datasets are selected from various fields, including biology, astronomy, social science and medical diagnostics. The class ratios in the datasets ranges between 9.7:1 and 26:1, while the number of samples ranges between 72 and 11,183 and the number of features ranges between 6 and 294.

Table 4. Details of the experimental datasets.

Name	Ratio	#S	#F
abalone	9.7:1	4177	10
yeast_ml8	13:1	2417	103
scene	13:1	2407	294
libras_move	14:1	360	90
ozone_level	34:1	2536	72
mammography	42:1	11,183	6
satimage	9.3:1	6435	36
yeast_me2	28:1	1484	8
wine_quality	26:1	4898	11

We begin our real-life data experiments with three datasets *abalone*, *yeast_ml18* and *scene*. As shown in Figure 5, the KDE-based ensemble approach outperforms the single decision tree and random forest classifiers on all three datasets. In particular, the KDE-based method yields significantly better results on the *abalone* and *yeast_ml18* datasets, both in terms of AUC and F_1-score. On the *scene* dataset, the KDE-based method is level with the decision tree classifier but is significantly better than the random forest classifier. The values of AUC are all above 0.5, indicating a nontrivial classification. The difference between the accuracy of the KDE ensemble and random forest is statistically significant for all three datasets considered in Figure 5, with p-values 0.0004, 1.8421×10^{-16} and 0.0031 for *abalone*, *yeast_ml8* and *scene*, respectively. We conclude that, overall, the KDE-based ensemble outperforms the benchmark classifiers on the initial set of data.

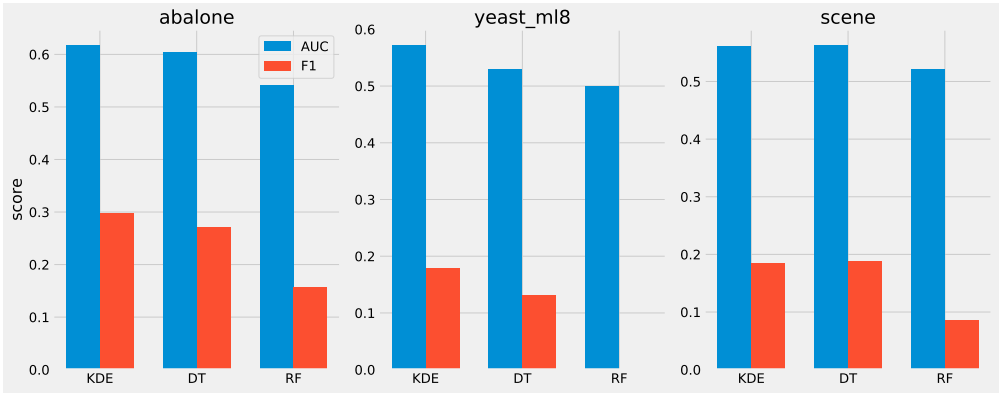

Figure 5. Classifier performances as measured by AUC and the F_1-score.

The results of the next group of tested datasets—*libras_move*, *ozone_level* and *mammography*—are presented in Figure 6. As shown in the figure, the KDE-based approach yields robust results on the tested data. The KDE-based ensemble outperforms the benchmark classifiers in terms of AUC on all three tested datasets. In particular, we obtain significantly superior results on *libras_move* and *mammography* datasets with the values of AUC at or above 0.8. The difference between the accuracy of the KDE ensemble and random forest is statistically significant for all three datasets considered in Figure 6, with p-values $0.0015, 1.8730 \times 10^{-7}$ and 2.5391×10^{-13} for *libras_move*, *ozone_level* and *mammography*, respectively. The F_1-score results are somewhat weaker. The proposed classifier yields the second best F_1-score on the first two datasets and the last score on the last dataset. Given the superior performance based on AUC and moderate performance based on the F_1-score, we surmise that the KDE-based ensemble generally outperforms the benchmark classifiers on the second set of data.

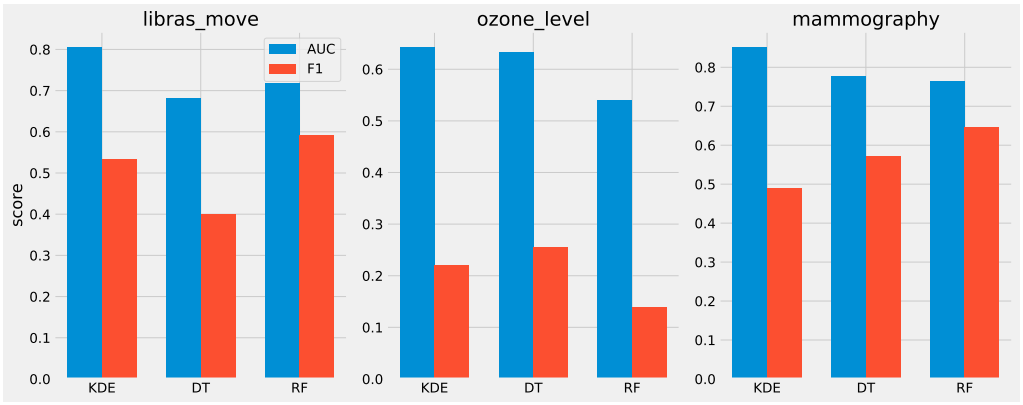

Figure 6. Classifier performances as measured by AUC and the F_1-score.

The results of the KDE-based ensemble on the third set of data are mixed. The results of the experiments are shown in Figure 7. On the one hand, the KDE-based approach produces the highest AUC on the *satimage* and *wine_quality* datasets and the second highest AUC on the *yeast_me2* dataset. It also outperforms the random forest classifier on the *yeast_me2* dataset. Meanwhile, the proposed classifier produces the lowest F_1-scores on the *satimage* and *wine_quality* datasets. The difference between the accuracy of the KDE ensemble and random forest is statistically significant for all three datasets considered in Figure 7, with p-values $1.2246 \times 10^{-16}, 2.8011 \times 10^{-5}$ and 3.8585×10^{-14} for *satimage*, *yeast_me2* and *wine_quality*, respectively. Thus, we conclude that, overall, the effectiveness of the KDE-based ensemble on the third set of data is on par with the benchmark classifiers.

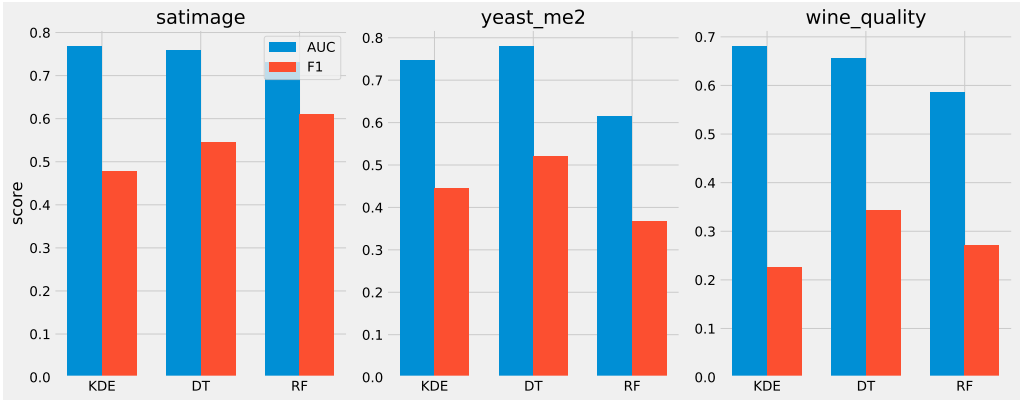

Figure 7. Classifier performances as measured by AUC and the F_1-score.

To further analyze the performance of the proposed method, we consider the execution times on real-life data. As shown in Table 5, the KDE ensemble and random forest produce

different execution times depending on the data. While the KDE ensemble is faster on *abalone*, *wine_quality* and *mammography* datasets, random forest is faster on *satimage* and *yeast_m18* datasets.

Table 5. Execution times (in s) on the real-life datasets.

	Satimage	Abalone	Yeast_ml8	Wine_Quality	Mammography
DT	0.0978	0.0457	0.5114	0.0320	0.0266
RF	2.5724	1.5662	2.5942	0.7085	0.7862
KDE	3.9242	0.3840	5.8890	0.6219	0.5519

5.4. Discussion

The proposed KDE ensemble method performed well against random forest on synthetic data. It achieved better accuracy and faster execution times on both datasets. The performance of the KDE ensemble on synthetic data indicates that it is well suited for cases where the data are close to independent and identically distributed (i.i.d). It is not surprising given the theoretical underpinnings of KDE. Although the i.i.d. assumption may not hold completely true in real life, many datasets are reasonably close to i.i.d. for KDE to perform accurately.

The KDE ensemble method performed relatively well on real-life data. It outperformed random forest on the majority of the tested datasets, including *abalone*, *yiest_ml8*, *scene*, *ozone_level* and *yiest_me2*. On the other hand, the KDE ensemble showed lower accuracy on four datasets: *libras_move*, *satimage*, *wine_quality* and *mammography*. The execution times of the KDE ensemble are similar to random forest.

We note that in some cases, the F_1-scores are relatively low for both the KDE ensemble and random forest. In general, given an imbalanced test set, the overall recall rate is expected to be low. Since the F_1-score is calculated based on the precision and recall rates (Equation (10)), it will lead to a lower score.

The analysis of the performance reveals that KDE does not perform well on data with an extreme imbalance ratio. However, we believe that it is a common problem for most sampling-based classifiers. Artificially creating too many minority samples leads to distortion of the data. In general, we expect the proposed method to perform very well if the data are close to i.i.d. and the imbalance ratio is less than 13:1.

6. Conclusions

In this work, we presented a novel ensemble classifier aimed at handling imbalanced datasets. Our approach is based on using the KDE-based sampling technique. The proposed method balances the training data for each tree in the ensemble using density estimates of the minority class. KDE offers a smooth and effective approach to balancing data by estimating the underlying distribution of the data.

To evaluate the effectiveness of the proposed method, we employed both synthetic and real-life data. The results from the experiments on the synthetic data showed a significant advantage of the KDE-based ensemble over the benchmark classifiers. We further tested the proposed method on a set of nine real-life datasets. The KDE-based ensemble consistently outperformed the benchmark classifiers in terms of AUC. The results of F_1-score were mixed. The proposed method produced high F_1-scores on the first portion of datasets and poorer results on the second half of the tested datasets. The moderate performance based on the F_1-scores shows that the proposed approach is not perfect. Nevertheless, the overall results establish the proposed method as a competitive alternative to the existing classification algorithms for dealing with imbalanced data.

The proposed method has low computational complexity. The efficiency of the method stems from the underlying use of decision trees, which are known as one of the fastest classification algorithms. We believe that given a strong performance on the tested datasets

together with its simple implementation, the proposed method offers a useful tool for dealing with imbalanced data.

The KDE ensemble approach showed promising results for dealing with imbalanced data. As a future avenue for research, the proposed method can be applied to multi-class imbalanced data. In addition, KDE can be applied to other ensemble approaches that do not employ decision trees as base classifiers. A key assumption in constructing KDE is the identical distribution of the sample points, which raises the question of the out-of-distribution effectiveness of KDE as another avenue for future research.

Author Contributions: Conceptualization, F.K.; Data curation, J.A.R.; Formal analysis, F.K. and S.M.; Funding acquisition, S.M.; Investigation, J.A.R.; Methodology, F.K. and J.A.R.; Project administration, S.M.; Software, J.A.R.; Supervision, F.K.; Writing—original draft, F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://archive.ics.uci.edu/.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kamalov, F. Forecasting significant stock price changes using neural networks. *Neural. Comput. Appl.* **2020**, *32*, 1–13. [CrossRef]
2. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
3. Thabtah, F.; Hammoud, S.; Kamalov, F.; Gonsalves, A. Data imbalance in classification: Experimental evaluation. *Inf. Sci.* **2020**, *513*, 429–441. [CrossRef]
4. Sun, Z.; Song, Q.; Zhu, X.; Sun, H.; Xu, B.; Zhou, Y. A novel ensemble method for classifying imbalanced data. *Pattern Recognit.* **2015**, *48*, 1623–1637. [CrossRef]
5. Kim, J.; Scott, C.D. Robust kernel density estimation. *J. Mach. Learn. Res.* **2012**, 13, 2529–2565.
6. Gramacki, A. *Nonparametric Kernel Density Estimation and Its Computational Aspects*; Springer International Publishing: Cham, Switzerland, 2018; Volume 37.
7. Weglarczyk, S. Kernel density estimation and its application. In *ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2018; Volume 23, p. 00037.
8. Mani, I.; Zhang, I. kNN approach to unbalanced data distributions: A case study involving information extraction. In Proceedings of Workshop on Learning from Imbalanced Datasets, Washington, DC, USA, 30 August 2003; Volume 126.
9. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
10. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
11. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 1322–1328.
12. Chao, X.; Zhang, L. Few-shot imbalanced classification based on data augmentation. *Multimed. Syst.* **2021**, 1–9. [CrossRef]
13. Yang, J.; Guo, X.; Li, Y.; Marinello, F.; Ercisli, S.; Zhang, Z. A survey of few-shot learning in smart agriculture: Developments, applications, and challenges. *Plant Methods* **2022**, *18*, 1–12. [CrossRef]
14. Kamalov, F. Kernel density estimation based sampling for imbalanced class distribution. *Inf. Sci.* **2020**, *512*, 1192–1201. [CrossRef]
15. Yang, P.; Liu, W.; Zhou, B.B.; Chawla, S.; Zomaya, A.Y. Ensemble-based wrapper methods for feature selection and class imbalance learning. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Gold Coast, Australia, 14–17 April 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 544–555.
16. Yijing, L.; Haixiang, G.; Xiao, L.; Yanan, L.; Jinling, L. Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. *Knowl.-Based Syst.* **2016**, *94*, 88–104. [CrossRef]
17. Yildirim, P.; Birant, U.K.; Birant, D. EBOC: Ensemble-based ordinal classification in transportation. *J. Adv. Transp.* **2019**, *2019*, 7482138. [CrossRef]
18. Mohammed, E.A.; Keyhani, M.; Sanati-Nezhad, A.; Hejazi, S.H.; Far, B.H. An ensemble learning approach to digital corona virus preliminary screening from cough sounds. *Sci. Rep.* **2021**, *11*, 15404. [CrossRef] [PubMed]
19. Tama, B.A.; Lim, S. Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Comput. Sci. Rev.* **2021**, *39*, 100357. [CrossRef]
20. Wang, Y.; Liu, J.; Xiang, Y.; Wang, J.; Chen, Q.; Chong, J. MAGE: Automatic diagnosis of autism spectrum disorders using multi-atlas graph convolutional networks and ensemble learning. *Neurocomputing* **2022**, *469*, 346–353. [CrossRef]

21. Baradaran, R.; Amirkhani, H. Ensemble learning-based approach for improving generalization capability of machine reading comprehension systems. *Neurocomputing* **2021**, *466*, 229–242. [CrossRef]
22. Malebary, S.J.; Hashmi, A. Automated breast mass classification system using deep learning and ensemble learning in digital mammogram. *IEEE Access* **2021**, *9*, 55312–55328. [CrossRef]
23. Yang, R.; Zheng, K.; Wu, B.; Wu, C.; Wang, X. Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning. *Sensors* **2021**, *21*, 8281. [CrossRef]
24. Galar, M.; Fernández, A.; Barrenechea, E.; Herrera, F. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognit.* **2013**, *46*, 3460–3471. [CrossRef]
25. Hido, S.; Kashima, H.; Takahashi, Y. Roughly balanced bagging for imbalanced data. Statistical Analysis and Data Mining. *ASA Data Sci. J.* **2009**, *2*, 412–426.
26. Lango, M.; Stefanowski, J. Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data. *J. Intell. Inf. Syst.* **2018**, *50*, 97–127. [CrossRef]
27. Diez-Pastor, J.F.; Rodríguez, J.J.; García-Osorio, C.; Kuncheva, L.I. Random balance: Ensembles of variable priors classifiers for imbalanced data. *Knowl.-Based Syst.* **2015**, *85*, 96–111. [CrossRef]
28. Collell, G.; Prelec, D.; Patil, K.R. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing* **2018**, *275*, 330–340. [CrossRef] [PubMed]
29. Elakkiya, R.; Jain, D.K.; Kotecha, K.; Pandya, S.; Reddy, S.S.; Rajalakshmi, E.; Subramaniyaswamy, V. Hybrid Deep Neural Network for Handling Data Imbalance in Precursor MicroRNA. *Front. Public Health* **2021**, *9*, 1410.
30. Fawcett, T. An introduction to ROC analysis. *Pattern Recogn. Lett.* **2006**, *27*, 861–874. [CrossRef]
31. Dua, D.; Graff, C. *UCI Machine Learning Repository*; Irvine, C.A., Ed.; University of California, School of Information and Computer Science: Berkely, CA, USA, 2019. Available online: http://archive.ics.uci.edu/ml (accessed on 12 June 2022).
32. Kamalov, F.; Denisov, D. Gamma distribution-based sampling for imbalanced data. *Knowl.-Based Syst.* **2020**, *207*, 106368. [CrossRef]