Machine Learning (2024) 113:4547–4578
https://doi.org/10.1007/s10994-023-06323-y

Hellinger distance decision trees for PU learning in imbalanced data sets

Carlos Ortega Vázquez[1] · Seppe vanden Broucke[1,2] · Jochen De Weerdt[1]

Received: 6 June 2022 / Revised: 30 December 2022 / Accepted: 20 February 2023 /
Published online: 28 March 2023

Abstract

Learning from positive and unlabeled data, or PU learning, is the setting in which a binary classifier can only train from positive and unlabeled instances, the latter containing both positive as well as negative instances. Many PU applications, e.g., fraud detection, are also characterized by class imbalance, which creates a challenging setting. Not only are fewer minority class examples compared to the case where all labels are known, there is also only a small fraction of unlabeled observations that would actually be positive. Despite the relevance of the topic, only a few studies have considered a class imbalance setting in PU learning. In this paper, we propose a novel technique that can directly handle imbalanced PU data, named the PU Hellinger Decision Tree (PU-HDT). Our technique exploits the class prior to estimate the counts of positives and negatives in every node in the tree. Moreover, the Hellinger distance is used instead of more conventional splitting criteria because it has been shown to be class-imbalance insensitive. This simple yet effective adaptation allows PU-HDT to perform well in highly imbalanced PU data sets. We also introduce PU Stratified Hellinger Random Forest (PU-SHRF), which uses PU-HDT as its base learner and integrates a stratified bootstrap sampling. Our empirical analysis shows that PU-SHRF substantially outperforms state-of-the-art PU learning methods for imbalanced data sets in most experimental settings.

Keywords PU Learning · Weakly supervised learning · Imbalanced classification · Ensemble learning

Editors: Nuno Moniz, Paula Branco, Luis Torgo, Nathalie Japkowicz, Michal Wozniak, Shuo Wang.

✉ Carlos Ortega Vázquez
carloseduardo.ortegavazquez@kuleuven.be

Seppe vanden Broucke
seppe.vandenbroucke@ugent.be

Jochen De Weerdt
jochen.deweerdt@kuleuven.be

[1] Research Centre for Information Systems Engineering, Faculty of Economics and Business, KU Leuven, Leuven, Belgium

[2] Department of Business Informatics and Operations Management, Faculty of Economics and Business Administration, Ghent University, Ghent, Belgium

1 Introduction

Despite the ubiquity of supervised learning in practice, many real-world applications, including fraud detection (Stripling et al., 2018; Li et al., 2014), text classification (Yarowsky, 1995; Liu et al., 2002), and medical diagnosis (Claesen et al., 2015; Chen et al., 2020), suffer from inaccurate or incomplete label information. Moreover, these applications are often also characterized by a high class imbalance. These applications relate to two areas of research: positive and unlabeled (PU) learning and imbalanced learning. In these applications, the underrepresented class is the class of interest (i.e., positive class): in fraud detection, the fraudulent cases represent less than one percent of the total transactions (Van Belle et al., 2022); despite a less severe class imbalance in medical diagnosis, the false negatives (i.e., undetected tumor) are more important than false positives. The underrepresentation of the minority class can become more challenging with the issue of incomplete label information: in the medical data, only positive information is often reported (i.e., diagnosed disease); in fraud detection, only a few fraudsters are identified whereas most of them go unnoticed. Thus, some applications can benefit from the connection between PU learning and imbalanced learning. PU learning assumes that labeled examples are positive, but unlabeled examples can belong to either the positive or negative class. Imbalanced learning aims to propose methods that handle settings in which the class distribution is significantly unequal. Accordingly, in this paper, we focus on the problem of learning from imbalanced data sets in which the negative class and a proportion of positives remain unlabeled.

PU learning has increasingly gained popularity in recent years, as demonstrated by the uptake in method development (Bekker & Davis, 2020). One approach that was first used in text classification identifies reliable negatives and learns from positives and the resulting reliable negatives (Yarowsky, 1995; Liu et al., 2002). Another approach assumes that all unlabeled examples are negative and applies standard classifiers (Lee & Liu, 2003; Mordelet & Vert, 2014). A last approach, with more recent developments, utilizes the class prior (i.e., positive class ratio) in existing algorithms to enable PU learning (Denis et al., 2005; Elkan & Noto, 2008; Li et al., 2014; Du Plessis et al., 2015; Kiryo et al., 2017). Other works have explored non-standard settings in PU learning motivated by domain applications. For example, a common assumption in PU learning is that the labeled examples are a random subset of the positive examples; however, this assumption is often violated in practice (He et al., 2018; Bekker et al., 2019).

Although most modern PU methods perform successfully in several benchmark data sets (Du Plessis et al., 2015; Kiryo et al., 2017; Chen et al., 2020), it remains unclear how well they perform in highly imbalanced data sets. Imbalanced PU classification poses new challenges that have not been sufficiently addressed. In this specific setting, the fact that only a few positive instances are known to the learner creates more severe class imbalance. A suitable PU method for an imbalanced setting should be able to exploit the small number of labeled positives and still learn from unlabeled instances. Only a few works have focused on imbalanced PU classification. Two works have proposed PU learning via optimizing an adaptation of the area under the receiver operating characteristic curve (ROC-AUC) for the semisupervised setting (Sakai et al., 2018; Xie & Li, 2018). However, optimizing the ROC-AUC does not guarantee optimization of more relevant metrics for imbalanced classification, such as the area under the precision-recall curve (Davis & Goadrich, 2006). A second approach, Cost-Sensitive Positive and Unlabeled learning (CSPU) (Chen et al., 2021), introduces the use of misclassification costs to address class imbalance. While being conceptually appealing, CSPU's requirement to have misclassification costs available is not easily met given that in several domains these costs are difficult to determine. Lastly, the imbalanced nonnegative PU learning method relies on oversampling to balance the PU data (Su et al., 2021). Nonetheless, oversampling might cause unnecessary overfitting, and the oversampling rate might need to be tuned as an extra hyper-parameter. Accordingly, we observe a gap in the literature for a technique that can perform well in highly imbalanced PU data without requiring resampling or misclassification costs.

Therefore, in this work, we introduce a novel tree-based technique that is designed to learn from imbalanced PU data, denoted as the PU Hellinger Decision Tree (PU-HDT). PU-HDT does not need to modify the imbalanced data distribution. Similar to other class-prior-based methods (Kiryo et al., 2017; Su et al., 2021; Du Plessis et al., 2015), PU-HDT exploits the class prior (i.e., the proportion of positive examples) to enable PU learning. At each node, the true positives are estimated from unlabeled instances rather than assuming that all unlabeled instances are negatives. Instead of using a traditional splitting criterion exhibiting demonstrated inferiority towards imbalanced data sets (e.g., Gini and entropy), PU-HDT uses the Hellinger distance (Cieslak & Chawla, 2008). The Hellinger distance has shown robustness to extreme degrees of class imbalance in previous studies (Cieslak et al., 2012; Lyon et al., 2014; Dal Pozzolo et al., 2014). These two improvements enable PU-HDT to handle highly imbalanced data sets effectively. The performance of PU-HDT can be further improved using an ensemble. We show that a modified random forest with PU-HDT as its base learner outperforms state-of-the-art PU learning methods under different experimental settings with class imbalance.

The remainder of this paper is organized as follows: Sect. 2 discusses different methods found in the imbalanced learning and PU learning literature. Section 3 introduces the PU-HDT algorithm and explains its inner working. Additionally, an ensemble method that uses PU-HDT as the base learner is presented. Section 4 describes the experimental setup, and Sect. 5 discusses the results. Section 6 provides general conclusions and implications based on the empirical analysis. Finally, we outline some possibilities for further research.

2 Related work

In this section, we provide an overview of related work in the imbalanced and PU learning domains.

2.1 Imbalanced learning

In numerous domain applications of binary classification, including medical diagnosis, churn prediction and fraud detection, the class of interest (i.e., minority class) is particularly rare. This constitutes a challenge for standard classifiers as most conventional algorithms are biased towards the majority class. Specifically, minority class examples are misclassified more often when compared to those from the majority class. Thus, several techniques have been proposed to address class imbalance. These methods can be divided into four main categories (Fernández et al., 2018): data-level methods, algorithm-level methods, cost-sensitive learning, and ensemble-based approaches, with the latter two being more sophisticated.

Data-level methods balance the class distribution by relying on a resampling strategy. An advantage of resampling is that the end-user can choose a standard classifier of preference. However, data-level approaches are sensitive to the specific settings: in the presence of outliers, sampling methods excessively distort the data distribution, which results in worse performance (Baesens et al., 2021). A popular data-level method is the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002). SMOTE creates new minority instances close to other minority examples via interpolation. However, SMOTE resamples the minority class without considering the density of the data, which can create further overlap between classes. Consequently, several works have proposed extensions of SMOTE that aim to overcome this problem: some well-known extensions include MWMOTE (Barua et al., 2012), Borderline-SMOTE (Han et al., 2005), and Adaptive Synthetic Sampling (ADASYN) (He et al., 2008). ADASYN adaptively generates new minority instances according to the class distribution and creates more minority examples when few minority are present in the neighborhood. In this paper, we consider ADASYN as the default data-level solution for class imbalance.

Algorithm-level methods modify existing classifiers to improve the predictive performance on the minority class. Several algorithm-level methods have been proposed based on popular classifiers, including support vector machines (SVM) (Gonzalez-Abril et al., 2014), nearest-neighbor methods (Cano et al., 2013; Liu & Chawla, 2011), and decision trees (Cieslak & Chawla, 2008; Liu et al., 2010; Sardari et al., 2017). In this work, we extend algorithmic methods based on decision trees. In particular, we focus on the splitting criterion, which is the main element that can be improved for the imbalanced setting. Decision trees such as C4.5 (Quinlan, 1993) and CART (Breiman et al., 1984) utilize splitting functions that are sensitive to highly skewed distributions as both Information Gain and Gini Index are biased towards the majority class. Hellinger Decision Tree (HDT) (Cieslak & Chawla, 2008) and Class Confidence Proportion Decision Tree (Liu et al., 2010) rely on skew-insensitive splitting functions. For instance, HDT has been shown experimentally to outperform other decision trees such as C4.5 (Quinlan, 1993) and CART (Breiman et al., 1984). Other works have used cost-sensitive learning to adapt the splitting criterion (Bahnsen et al., 2015; Vadera, 2010). Moreover, decision tree methods have been used in an ensemble setup (Chen et al., 2004; Cieslak et al., 2012; O'Brien & Ishwaran, 2019; Zelenkov, 2019).

Among the algorithm-level methods previously presented, HDT is one of the most popular in the literature. Motivated by HDT, several extensions have been proposed to handle different tasks under class imbalance: data streams (Grzyb et al., 2021; Lyon et al., 2014), multilabel classification (Daniels & Metaxas, 2017), and multiclass classification (Hoens et al., 2012). Other works have used the Hellinger distance to propose their own methods that aim to outperform the HDT (Akash et al., 2019; Su & Cao, 2019). Despite the popularity of HDT in other domains, it is not yet explored in weakly supervised learning. In this paper, we focus on a special case of weakly supervised learning: PU learning. Our technique represents an extension of HDT that can effectively handle PU data in the imbalanced setting.

2.2 PU learning

PU learning is a setting related to weakly supervised learning (Zhou, 2018), in which only positive and unlabeled examples exist. Different assumptions can be made regarding the observation of labeled positive examples or the underlying labeling mechanism. Most PU learning methods are built on the selected completely at random (SCAR) assumption. The SCAR assumption states that the positively labeled examples are a randomly selected subset of the set of positives. SCAR implies that supervised techniques can be used for PU learning because the ranking order of predictions is preserved as if the true label was known (Elkan & Noto, 2008). Selected at random (SAR) (Bekker et al., 2019) is a more realistic assumption regarding the labeling mechanism. For SAR, the probability of a positive observation being labeled depends on its features or attributes. The latter assumption is much more sensible for common PU learning applications, including recommendation systems, medical diagnosis, and fraud detection. One specific type of SAR is the probabilistic gap (PG), which assumes that the positive examples that more closely resemble negatives are less likely to be labeled (He et al., 2018). For instance, a fraudster is more likely to go unnoticed if their behavior mimics a normal profile.

PU learning methods can be divided into three categories: two-step techniques, biased learning, and class-prior-based methods (Bekker & Davis, 2020). Two-step techniques first identify the instances that are most likely to be negatives among the unlabeled; then, a model learns from the newly labeled negative and positive examples. A semisupervised technique can also be used to exploit the remaining unlabeled data (Liu et al., 2003). Early two-step techniques come from the text classification literature (Liu et al., 2002; Yu & Li, 2007; Li & Liu, 2003), in which models such as Naïve Bayes (NB) and SVM are often used. For instance, S-EM (Liu et al., 2002) first introduces labeled instances as “spies” to identify unreliable negatives and then applies semi-supervised NB to predict the labels of the unreliable negatives. A two-step approach works well as long as separability exists between the negative and positive classes (Bekker & Davis, 2020). Biased learning techniques consider unlabeled instances as negatives with label noise (Lee & Liu, 2003; Claesen et al., 2015; Mordelet & Vert, 2014). Most biased learning techniques place a high misclassification cost on false positives by either introducing asymmetric penalization (Liu et al., 2003) or by using bagging (Mordelet & Vert, 2014; Claesen et al., 2015). For instance, PU Bagging selects all labeled examples and takes bootstrap samples of the unlabeled examples. One advantage of PU Bagging is that it enables the user to choose any base learner: SVM is used in Mordelet and Vert (2014) but other models such as a decision tree can be selected as well. The class-prior-based methods utilize the class prior information to either preprocess the data or modify the algorithm (Elkan & Noto, 2008; Du Plessis et al., 2015; Plessis et al., 2017; Bekker et al., 2019). Among them, some techniques are based on the empirical minimization framework, which modifies the loss function to incorporate the class prior. Accordingly, well-known algorithms have been adapted for PU learning, e.g., logistic regression (Bekker et al., 2019; Du Plessis et al., 2015) and neural networks (Kiryo et al., 2017), which can be considered the state-of-the-art in the field.

Although the PU learning literature has developed a plethora of methods, most of the works have focused on a balanced setting. Likewise, no study has focused on the evaluation of well-known PU learning methods in highly imbalanced settings. Compared to imbalanced classification with complete label information, PU classification under class imbalance suffers from severe underrepresentation of the positive class. In such a scenario, the bias towards the majority class worsens for most of PU methods. A few studies have proposed methods to handle class imbalance. These PU methods for imbalanced setting have used cost-sensitive learning (Chen et al., 2021), algorithm-level approaches (Sakai et al., 2018; Xie & Li, 2018), and data-level approaches (Su et al., 2021). Motivated by previous work on the adaption to PU learning of the risk minimization framework (Kiryo et al., 2017; Du Plessis et al., 2015), CSPU introduces class-dependent costs to improve the performance of PU classification under class imbalance. However, this particular setting requires complete information of the misclassification costs, which might not be available; in some applications, such as credit scoring or fraud detection, the misclassification costs are better represented at the instance level (Bahnsen et al., 2015; Zelenkov, 2019). Based on the semisupervised learning literature, the risk minimization framework can be adapted to optimize a semisupervised variant of the ROC-AUC (Sakai et al., 2018; Xie & Li, 2018). Nevertheless, optimizing the ROC-AUC does not automatically lead to an optimization of relevant metrics for the imbalanced setting, such as the F-score (Chen et al., 2021) and the area under the precision-recall curve (Davis & Goadrich, 2006). Unlike the CSPU, imbalanced nonnegative PU learning (imbalanced nnPU) (Su et al., 2021) does not require misclassification information as it relies on oversampling to balance the class distribution. Oversampling, however, might create harmful overfitting because it creates exact copies of the positive labeled instances (Fernández et al., 2018): in order to avoid overfitting, the oversampling rate can be tuned as an additional hyperparameter. Our work contributes to the literature by introducing a Decision Tree technique that can natively handle imbalanced data without requiring complete information of misclassification costs or a resampling method. To the best of our knowledge, this is the first work that integrates the effectiveness of Hellinger Decision Trees for imbalanced classification into PU learning.

3 PU hellinger decision tree (PU-HDT)

In this section, we introduce PU-HDT. Moreover, we illustrate its workings on a half-moons data set and expand the technique towards a PU Stratified Hellinger Random Forest.

3.1 The PU-HDT algorithm

The Hellinger Decision Tree (HDT) exploits the Hellinger distance to improve the splitting mechanism in imbalanced settings. The goal of the Hellinger distance is to capture

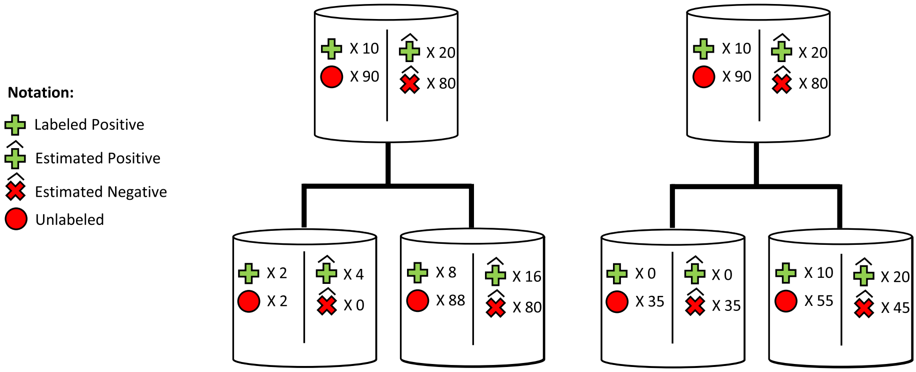


Fig. 1 Difference between Gini Index and Hellinger distance in node splitting. Gini Index is better in the left tree (0.166) compared to the right tree (0.169); PU Hellinger distance is better in the right tree (0.707) than in the left tree (0.459)

the divergence between the positive and negative class distribution without being dominated by the class imbalance. Equation (1) further illustrates the robustness to class imbalance. The Hellinger distance can be calculated in a parent node i as follows:

$$HD_i = \sqrt{\left(\sqrt{\frac{N_{left_i}}{N_i}} - \sqrt{\frac{P_{left_i}}{P_i}}\right)^2 + \left(\sqrt{\frac{N_{right_i}}{N_i}} - \sqrt{\frac{P_{right_i}}{P_i}}\right)^2}, \tag{1}$$

where N_i and P_i are the counts of negatives and positives in the parent node i, N_{left_i} and P_{left_i} are the counts of the examples that fall into the left child node, and N_{right_i} and P_{right_i} the ones that fall into the right child node. Thus, there is no influence of the proportion of the majority class. In comparison, the Gini Index, as used in, e.g., CART, depends on p_j, the proportion of class j: $I_G = 1 - \sum_{j=1}^{2} p_j^2$. With the Hellinger distance, the HDT can create better splits during tree construction in imbalanced data sets; however, HDT is not directly applicable to PU data sets.

Therefore, we adapt the HDT (Cieslak & Chawla, 2008) to PU learning by exploiting the class prior, represented as α. Under the SCAR assumption, the class prior enables estimation of the label frequency c, the proportion of positive examples that are labeled in the data. Equation (2) illustrates the previous statement:

$$\begin{aligned} Pr(l=1) &= Pr(l=1 \mid y=1)Pr(y=1) + Pr(l=1 \mid y=0)Pr(y=0) \\ &= Pr(l=1 \mid y=1)Pr(y=1) \\ &= c\,\alpha, \end{aligned} \tag{2}$$

where l is the observed labeled and y is the true label. The proportion of labeled positives $Pr(l=1)$ can easily be estimated from the training data. In the PU setting, only the labeled instances can belong to the positive class, which implies that the conditional probability $Pr(l = 1 \mid y = 0) = 0$. The label frequency c enables the estimation of the counts of positives and negatives from PU data. In a given node i, the estimated count of positives $\hat{P}_i$ can be calculated as follows:

$$
\begin{aligned}
\hat{P}_i &= \min\left\{\frac{L_i}{c}, T_i\right\} \\
\hat{P}_i &= \min\left\{L_i \frac{\alpha}{Pr(l = 1)}, T_i\right\},
\end{aligned}
\tag{3}
$$

where L_i is the count of labeled positives and T_i is the count of total instances in node i. Equation (3) indicates that, in a given node, the number of positives cannot exceed the total number of examples. The estimated count of negatives $\hat{N}_i$ can be computed from the difference between the total count of instances and the number of positives: $\hat{N}_i = T_i - \hat{P}_i$. This estimation is similar to the one found in POSC4.5 (Denis et al., 2005), but it is used to modify entropy as the splitting criterion. Moreover, Eq. (3) emphasizes the importance of the class prior for adapting the HDT to PU learning: this means that PU-HDT falls into the category of class-prior-based methods. Despite the relevance of the class prior in PU-HDT, it is often unavailable in PU learning, so we require domain knowledge or methods for class prior estimation (Du Plessis & Sugiyama, 2014; Bekker & Davis, 2018; Plessis et al., 2017; Elkan & Noto, 2008; Ramaswamy et al., 2016). Figure 1 illustrates the difference between Gini Index and Hellinger Distance in node splitting. The left tree is split according to the Gini Index whereas the right tree follows the Hellinger distance with the PU adaptation.

Algorithm 1 outlines how a PU-HDT is built based on the Hellinger Distance with the estimated counts of positive and negative instances. PU-HDT follows the same binary tree construction as other decision trees, such as CART (Breiman et al., 1984), C4.5 (Quinlan, 1993), and of course, HDT (Cieslak & Chawla, 2008). Algorithm 1 can limit the size of the tree to avoid overfitting if it is used as a stand-alone classifier. In the case of using PU-HDT in a bagging ensemble or Random Forest, the tree can fully grow, as overfitting is no longer an issue (Breiman, 2001). Algorithm 1 first creates a tree node n; given its recursive aspect, if the tree node n reaches the maximum height h, then the algorithm stops. After the creation of the tree node n, a random selection of features f_{sel} can be optionally obtained for the tree node n. However, a standard decision tree does not implement any random feature selection: F_X and f_{sel} are the same set. Then, the optimal split value $x^*_{f_{max}}$ is found through a search within the set of features f_{sel}; the optimal split value is the one that maximizes the PU Hellinger distance. Afterward, the training instances, shown as features and label (X, L), are divided into two subsets $(X_{left}, L_{left}; X_{right}, L_{right})$ depending on the position of the instance in $x_{f_{max}}$ with regards to the optimal split value $x^*_{f_{max}}$. Finally, for each subset, a new tree node (*n.left* and *n.right*) is created to iterate the procedure again. Figure 1 visualizes a scenario showing the advantage of the (PU-)Hellinger distance over the Gini Index. On the one hand, the Gini Index prefers a node split (left tree) that concentrates most of the positive and negative class in one single child node. On the another hand, the (PU-)Hellinger distance indicates a preference for a node split (right tree) that concentrates all the positives in one child node.

Algorithm 1: PU-HDT(X, L, α, h, ϕ)

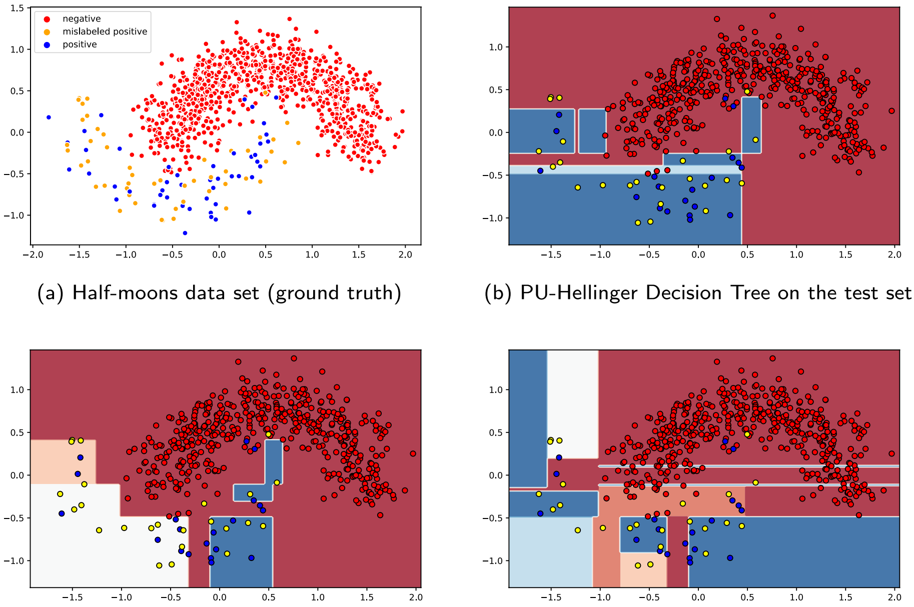
```
Input: $X$ - data, $L$ - labels, $\alpha$ - class prior, $h$ - depth, $\phi$ - number of
        considered features at each split
Output: PU-HDT - Hellinger Decision Tree for PU learning

create a tree node $n$
Initialize $e \leftarrow 0$
if $e > h$ then
    return
else
    $e \mathrel{+}= 1$
end
let $F_X$ be the set of features from input $X$
$f_{sel} \leftarrow RandomSample(F_X, \phi)$
$f_{max} \leftarrow \text{argmax}_{f \in f_{sel}} PUHellingerDistance(X, f)$
let $x^*_{f_{max}}$ be the optimal split value in feature $f_{max}$
split $X$ into $X_{left}(x_{f_{max}} < x^*_{f_{max}})$ and $X_{right}(x_{f_{max}} \geq x^*_{f_{max}})$
split $L$ into $L_{left}(x_{f_{max}} < x^*_{f_{max}})$ and $L_{right}(x_{f_{max}} \geq x^*_{f_{max}})$
$n.left \leftarrow$ PU-HDT$(X_{left}, L_{left}, h, \phi)$
$n.right \leftarrow$ PU-HDT$(X_{right}, L_{right}, h, \phi)$

Function PUHellingerDistance($X$, $f$):
    Initialize $BestHellingerDistance \leftarrow -1$
    let $V_f$ be the set of values of feature $f$
    for each value $v \in V_f$ do
        $w \leftarrow V_f \setminus v$
        estimate $\hat{P}_{f,v}$, $\hat{N}_{f,v}$, $\hat{P}_{f,w}$, and $\hat{N}_{f,w}$ according to Eq. (3)
        $\hat{P} \leftarrow \hat{P}_{f,v} + \hat{P}_{f,w}$
        $\hat{N} \leftarrow \hat{N}_{f,v} + \hat{N}_{f,w}$
        $HD \leftarrow \left(\sqrt{\hat{N}_{f,v}/\hat{N}} - \sqrt{\hat{P}_{f,v}/\hat{P}}\right)^2 + \left(\sqrt{\hat{N}_{f,w}/\hat{N}} - \sqrt{\hat{P}_{f,w}/\hat{P}}\right)^2$
        if $HD > BestHellingerDistance$ then
            $BestHellingerDistance \leftarrow HD$
        end
    end
return $\sqrt{BestHellingerDistance}$
```

3.2 An illustrative example: half-moons data set

As an illustrative example, we consider the popular "half-moons" data set that consists of two-dimensional points generated from two interleaved half circles. Figure 2 shows a class-imbalanced variant of the data set in which the class prior $\alpha = 5\%$: the upper



(a) Half-moons data set (ground truth)

(b) PU-Hellinger Decision Tree on the test set

(c) Hellinger Decision Tree on the test set

(d) CART on the test set

Fig. 2 Comparison of decision boundaries of decision trees, including PU-Hellinger decision tree, to illustrate the challenge of learning from positive and unlabeled data in an imbalanced setting. The red and orange dots represent the negative and hidden positives whereas the blue ones refer to the positive class. The areas with darker color (blue or red) points out more certainty regarding the classification into the positive (blue) or negative class (red). The lighter color, consequently, implies a higher uncertainty of the classifier

half-moon corresponds to the negative class, whereas the lower half-moon corresponds to the positive class. Moreover, half of the positives are mislabeled (i.e., unlabeled in the PU setting). The only information on the positive class is the labeled positives (blue dots), while the rest remains unlabeled (orange and red dots). Three techniques are compared: PU-HDT, HDT, and CART. The CART decision tree cannot learn from PU data, so only a few unlabeled positives fall into the blue regions: the positive distribution is not well represented by the decision boundary of the CART decision tree. The HDT performs better because of the built-in insensitivity to the imbalanced setting; however, the algorithm cannot learn from positive and unlabeled data. In the region where unlabeled positives dominate (lower left), the HDT provides predictions with high uncertainty. Lastly, the PU-HDT can learn from PU data and is suitable for the imbalanced setting. Most of the unlabeled positives fall into the blue regions as the technique considers that some positives are unlabeled. We see a clearer representation of the positive distribution that follows a half-moon shape. We argue that the class prior and Hellinger distance are essential and complementary in our method for imbalanced PU classification. One complicating factor in imbalanced classification is data complexity (Fernández et al., 2018). In a data set that shows high complexity, the few positives cannot represent well enough the minority (positive) class. Techniques that deal with imbalanced data are more robust to the underrepresentation of the minority class: HDT already provides an advantage over CART in Fig. 2. However, the PU setting poses another complicating factor that the Hellinger Distance cannot solve by itself: some of the unlabeled (i.e., negative) instances are positive. This is the reason that HDT cannot classify correctly most of the unlabeled positives as such. The class prior allows us to calculate the proportion of the unlabeled positives that need to be considered in the decision tree. Both Hellinger distance and the class prior can be successfully utilized in PU-HDT to enable PU learning in imbalanced data sets.

3.3 PU stratified Hellinger random forest

Ensemble learning generally improves the performance of a single learner by training several base learners and combining their output. A well-known ensemble method is Random Forest (Breiman, 2001), which extends bagging by incorporating randomized feature selection, with the base learner being a decision tree. Thus, we can use the PU-HDT as the base learner in a Random Forest. Moreover, we can further modify the ensemble algorithm based on the insights from PU Bagging (Mordelet & Vert, 2014): PU Bagging is naturally suitable for imbalanced learning because each bootstrap sample consists of all the available labeled positives and a subsample of the unlabeled data so that a balanced training set can be achieved. Although PU-HDT can handle imbalanced data, there is a practical reason to obtain bootstrap samples from the unlabeled data: the capability of PU-HDT to learn from PU data can be hindered by a bootstrap sample that contains a sparser representation of the positive distribution. For example, in Fig. 2b, the PU-HDT fails to learn the complete true positive distribution (ideally a blue half-moon shape) because there is a small region on the left that does not contain any labeled positives that can be exploited to estimate the true positives. Therefore, we propose the PU Stratified Hellinger Random Forest (PU-SHRF), ensuring that all labeled positives are represented in each bootstrap sample. Algorithm 2 outlines how PU-SHRF is designed based on PU-HDT and the stratified bootstrap sampling. Algorithm 2 represents the random forest setup for PU-HDT. Unlike a stand-alone decision tree, Algorithm 2 considers random feature selection when initializing a tree node: ϕ corresponds to the squared root of the number of features $\mid F_X \mid$; this is the default option in `scikit-learn`. Then, a bootstrap sample (X'_U, L'_U) is obtained from the unlabeled data. The size of the bootstrap sample is defined by K_U. The labeled positive instances (X'_{lab}, L'_{lab}) are added to the bootstrap sample (X'_U, L'_U) to provide the training data (X', L') to the PU-HDT. The PU-HDT will be added to the initialized Forest. Algorithm 2 stops when the number t of trees is fitted and added to the Forest. Compared to standard Random Forest, PU-SHRF requires two extra hyperparameters: the size of a stratified bootstrap sample K_U and the class prior α.

```
Algorithm 2: PU-SHRF($X$, $L$, $t$, $K_U$, $\alpha$)
Input: $X$ - training data, $L$ - labels, $t$ - number of trees, $K_U$ - size of
       bootstrap sample from unlabeled data, $\alpha$ - class prior
Output: Forest - set of $t$ PU Hellinger trees
Initialize Forest
let $\mid F_X \mid$ be the number of features from input $X$
$\phi \leftarrow \sqrt{\mid F_X \mid}$
for $i \leftarrow 1$ to $t$ do
    $X'_U, L'_U \leftarrow BootstrapSamp(X_{unl}, L_{unl}, K_U)$
    $X' \leftarrow X'_U \cup X_{lab}$
    $L' \leftarrow L'_U \cup L_{lab}$
    Forest $\leftarrow$ Forest $\cup$ PU-HDT$(X', L', \alpha, \phi)$
end
```

4 Experimental setup

The main goal of the experimental evaluation is to demonstrate the classification performance benefit of PU-HDT when applied to imbalanced PU data, i.e., data sets in which only a small percentage of positive observations is present and only a small percentage of the unlabeled observations would actually be a positive observation if the label were known. Therefore, we compare PU-HDT, PU-HRF, and PU-SHRF with 12 competitor methods in which eight methods come from the PU literature; the techniques are detailed detailed in Sect. 4.2. We include the Hellinger Decision Tree (HDT), Hellinger Random Forest (HRF) and the Stratified Hellinger Random Forest (SHRF). The evaluation utilizes nineteen benchmark data sets. We use two evaluation metrics that are commonly used in imbalanced learning (Cieslak & Chawla, 2008; Davis & Goadrich, 2006; Fernández et al., 2018), namely, the area under the precision-recall curve (PR-AUC) and the F1-score, representing the harmonic mean between precision and recall. In contrast to PR-AUC, the F1-score depends on a threshold that might disadvantage techniques that do not provide calibrated scores, for instance, tree-based methods. Thus, the threshold is optimized according to a validation set to maximize the F1-score for all techniques for each experimental setting. Furthermore, hypothesis testing is applied to statistically validate the empirical results, following the recommendation of Demšar (2006). First, the Iman-Davenport test is applied to determine whether all methods perform the same, as expressed in the null hypothesis. Then, the Holm's post hoc test is used to compare the best performing model with the other techniques. The source code for our techniques and the experimental setup are publicly available on GitHub.[1]

[1] A software implementation of our two-step method is available at https://github.com/CarlosOrtegaV/PU_Hellinger_Trees.

Table 1 Summary of data sets

Data set	Examples	Features	$P(y = 1)$ (%)
Car Good (CGO)	1728	6	3.99
Churn Chile (CCH)	5263	42	5.00
Forest Cover (FCO)	286,048	10	0.96
Fraud Car Insurance (FCI)	15,420	30	5.98
Fraud Card Credit (FCC)	282,982	29	0.16
KDDCup land vs portsweep (KDD)	1061	41	1.97
Churn Korean (CKO)	2221	8	5.00
Mammography (MAM)	11,183	6	2.32
Pendigits (PEN)	6870	16	2.27
Pizza Cutter 1 (PCU)	661	37	7.87
Poker 8 vs 6 (POK)	1477	10	1.15
Satellite (SAT)	5000	36	1.48
Shuttle (SHU)	49,097	9	7.15
Thyroid (THY)	7200	21	7.41
Yeast 6 (YEA)	1484	8	2.35
TV Churn (TVC)	9379	46	4.79
Speech (SPE)	3686	400	2.35
MNIST (MNI)	7603	100	9.21
Fraud IEEE (FIE)	590,540	251	3.50

4.1 Data

Nineteen data sets, summarized in Table 1, covering different applications such as churn prediction, fraud detection and image recognition, are considered. The churn data sets stem from the telecommunications industry, the credit card fraud data set is provided by Wordline and ULB (MLG, 2018) and Fraud IEEE on Kaggle, and the car insurance fraud data set is provided by Oracle (Oracle, 2015). Pizza Cutter 1, and Satellite are OpenML data sets (Vanschoren et al., 2013). Forest Cover, Speech, MNIST, Shuttle, Pendigits and Mammography are found in ODDS (Shebuti, 2016). Poker 8 vs 6, Car Good, KDD Cup land-vs-portsweep data sets are available on the KEEL repository (Alcalá-Fernandez et al., 2011). Finally, the Thyroid data set is found in UCI repository (Dua & Graff, 2019). The selection of the data sets is driven by our goal to compare techniques in a highly imbalanced setting. In previous works (Grzyb et al., 2021; Akash et al., 2019; Su et al., 2021), the class imbalance in benchmark data sets is not always extreme: several data sets show an imbalance ratio below 10. However, we are particularly interested in a more extreme setting. Thus, we select some of the most imbalanced data sets on the aforementioned public repositories.

To create the PU data, we flip completely at random (i.e., SCAR) some positives into unlabeled observations in each of the data sets. In the experiments, the number of labeled positives is determined by the flip ratio, which is the proportion of positives to be unlabeled. Three values of the flip ratio are considered: 25, 50, and 75%. For each experimental setting, we perform 20 repetitions of a holdout validation that splits the data into a training set (70%) and test set (30%) with a different random seed. The large

Table 2 Experimental hyperparameters

Technique	Setting
PU-SHRF	$K_U = U_{training}$, trees = 100, $\alpha = \alpha_{groundtruth}$
PU-HRF	Trees = 100, $\alpha = \alpha_{groundtruth}$
ADASYN + random forest	Neighbors = 5, sampling strategy = balanced ratio, trees = 100
PU-HDT	Max depth = 5, $\alpha = \alpha_{groundtruth}$
HDT	Max depth = 5
uPU	Base learner = XGBoost
nnPU	Base learner = XGBoost, $\alpha = \alpha_{groundtruth}$
Imbalanced uPU	Base learner = XGBoost, $\alpha = \alpha_{groundtruth}$
PU bagging	Hyperparameters recommended by authors
Rank pruning	Hyperparameters recommended by authors
PU weighted logistic regression	Hyperparameters recommended by authors
Elkan–Noto's method	Hyperparameters recommended by authors
Spy-EM	Hyperparameters recommended by authors
HDT	Max depth = 5
HRF	Trees = 100
SHRF	$K_U = U_{training}$, trees = 100

number of repetitions is needed as training from imbalanced PU data might lead to unstable performance (Mordelet & Vert, 2014). For the data sets that contain more than 10,000 examples, we sample 10,000 observations without replacement to be used in each repetition to limit the computation time of the experiments. In total, there are 1140 settings (19 datasets × 20 repetitions × 3 flip ratios) in the experiments where the class prior estimate equals the ground truth. For the sensitivity analysis, in Sect. 5.4, there are 1900 settings (19 datasets × 20 repetitions × 5 class prior estimates).

4.2 Techniques

We compare our PU Hellinger-based techniques with eight well-known PU learning techniques: imbalanced nonnegative PU learning (imbalanced nnPU) (Su et al., 2021), nonnegative PU learning (nnPU) (Kiryo et al., 2017), unbiased PU learning (uPU) (Du Plessis et al., 2015), PU Bagging (Mordelet & Vert, 2014), Rank Pruning (Northcutt et al., 2017), PU Weighted Logistic Regression (Lee & Liu, 2003), Elkan-Noto's method, that is, the preprocessing method using label probabilities (Elkan & Noto, 2008), and Spy-EM (Liu et al., 2002). We also include four non-PU baselines: Random Forest (Breiman, 2001) combined with ADASYN (He et al., 2008), HDT (Cieslak & Chawla, 2008), and its ensemble-based HRF and SHRF.

Table 2 summarizes the hyperparameter configuration of the methods in the experimental setup. PU learning methods in our experimental setup require hyperparameters that need to be specified by the end-user. However, the authors have suggested values for most of the

Table 3 Average F1-score (%) with optimal threshold, average rank in parenthesis and standard deviation (SD) at different flip ratios

Model	F1-score (rank) ± SD		
	Flip Ratio:		
	25%	50%	75%
PU-HDT	55.6 (9.6) ± 6.3	52.8 (9.3) ± 7.7	49.4 (8.2) ± 9.3
PU-HRF	**67.6 (4.1) ± 5.4**	**64.5 (4.2) ± 5.8**	57.4 (5.8) ± 6.7
PU-SHRF	**67.3 (4.2) ± 5.4**	**64.2 (4.2) ± 5.5**	**58.0 (5.1) ± 6.6**
Imbalanced nnPU	64.5 (5.6) ± 5.0	60.8 (6.3) ± 7.1	54.3 (7.4) ± 7.5
nnPU	63.9 (6.0) ± 5.3	60.4 (6.6) ± 6.9	52.1 (8.2) ± 8.0
uPU	63.1 (7.2) ± 5.1	59.3 (7.8) ± 6.4	52.4 (7.9) ± 6.7
Ranking pruning	55.4 (9.3) ± 6.6	51.9 (9.1) ± 7.9	47.7 (8.2) ± 8.4
PU bagging	57.4 (8.8) ± 5.7	57.3 (7.9) ± 5.6	57.0 (5.8) ± 6.2
Elkan–Noto's method	36.0 (12.5) ± 12.1	29.3 (13.0) ± 16.6	9.4 (14.2) ± 11.0
PU W. logistic regression	53.4 (9.8) ± 5.6	50.4 (9.4) ± 7.4	45.5 (8.4) ± 8.0
Spy-EM	29.6 (13.8) ± 4.9	28.6 (13.4) ± 5.9	29.2 (12.1) ± 7.5
ADASYN+RF	66.1 (5.9) ± 5.9	62.5 (6.1) ± 6.0	56.1 (6.1) ± 6.9
HDT	55.6 (9.5) ± 6.7	52.6 (9.3) ± 7.3	46.0 (9.1) ± 9.9
HRF	63.4 (6.8) ± 6.6	60.5 (6.5) ± 7.0	54.3 (6.6) ± 7.8
SHRF	62.7 (7.0) ± 6.3	59.4 (7.0) ± 7.0	53.7 (6.8) ± 7.1
Iman–Davenport test	$p < 0.01$	$p < 0.01$	$p < 0.01$

Best-performing model is indicated by bold and underlined in each column

Values in bold indicate that the best-performing model does not outperform the classifier in the model column at the 5% significance level

hyper-parameters that have shown good results in previous experiments. For the PU Hellinger-based techniques and unbiased PU learning, along with its extensions, the class prior is an essential hyperparameter, as it enables the labeled positives to be weighted to enable PU learning. In this work, we assume that the class prior is known at training time. In practice, the class prior can be estimated using several methods from the literature (Du Plessis & Sugiyama, 2014; Bekker & Davis, 2018; Plessis et al., 2017; Elkan & Noto, 2008; Ramaswamy et al., 2016). Another important hyperparameter for PU-SHRF is K_U, which is the number of unlabeled examples in each bootstrap. PU Bagging also uses K_U; however, it subsamples the unlabeled examples to balance the training data and avoid sample contamination by hidden positives. PU-SHRF can increase the number of unlabeled examples in the bootstrap samples because it can naturally address class imbalance and PU data. Thus, we opt for $K_U = U_{training}$ as the default value, which leads to stratified bootstrap samples for each PU Hellinger tree in the ensemble. For the PU-HDT and HDT, the max depth of the tree is set to 5 to avoid overfitting. Rank pruning, PU-weighted logistic regression, Elkan-Noto's method and Spy-EM follow the hyperparameters recommended by the authors. The rest of the hyperparameters are set to the default values in `scikit-learn`.

Table 4 Average PR-AUC (%) with average rank in parenthesis and standard deviation (SD) at different flip ratios

Model	PR-AUC (rank) ± SD		
	Ratio:		
	25%	50%	75%
PU-HDT	48.0 (11.3) ± 7.0	44.1 (11.2) ± 7.6	39.4 (10.7) ± 9.1
PU-HRF	**66.0 (4.4) ± 6.0**	**62.8 (4.7) ± 6.4**	54.1 (6.5) ± 7.2
PU-SHRF	**65.4 (4.6) ± 5.9**	**62.3 (4.7) ± 6.1**	**55.0 (5.5) ± 7.3**
Imbalanced nnPU	63.0 (5.9) ± 5.5	59.9 (6.0) ± 7.1	**54.2 (5.6) ± 7.7**
nnPU	62.6 (6.18) ± 5.8	59.7 (6.1) ± 7.0	53.2 (5.9) ± 7.6
uPU	60.5 (7.5) ± 6.6	55.5 (8.3) ± 7.9	46.4 (8.9) ± 7.8
Ranking pruning	52.0 (9.6) ± 6.7	49.1 (9.4) ± 7.1	44.6 (8.7) ± 8.9
PU bagging	49.5 (10.6) ± 6.0	50.4 (9.2) ± 6.2	50.4 (7.5) ± 6.6
Elkan–Noto's method	49.8 (10.4) ± 7.9	40.7 (11.5) ± 12.1	23.9 (12.6) ± 14.3
PU W. logistic regression	50.2 (10.2) ± 5.5	47.2 (9.9) ± 7.3	42.3 (9.2) ± 8.4
Spy-EM	32.1 (13.23) ± 6.3	30.8 (12.8) ± 8.6	30.2 (11.4) ± 9.3
ADASYN+RF	64.7 (6.1) ± 6.7	60.6 (5.9) ± 7.1	53.2 (6.2) ± 7.7
HDT	47.7 (11.4) ± 7.5	43.5 (11.4) ± 7.6	36.5 (11.3) ± 8.8
HRF	**66.3 (4.3) ± 6.0**	**63.0 (4.4) ± 6.2**	**55.8 (5.1) ± 7.4**
SHRF	**65.7 (4.3) ± 6.0**	**62.6 (4.5) ± 5.9**	**55.8 (4.9) ± 7.4**
Iman–Davenport test	$p < 0.01$	$p < 0.01$	$p < 0.01$

Best-performing model is indicated by bold and underlined in each column

Values in bold indicate that the best-performing model does not outperform the classifier in the model column at the 5% significance level

5 Results and discussion

5.1 Analysis on aggregated results

The average and standard deviation of the F1-score and PR-AUC are shown in Tables 3 and 4. The average rank is included in parenthesis. We also report on the average F1-score and PR-AUC per data set in the "Appendix". Additionally, results on the ROC-AUC are presented in Table 5. The bold and underlined value indicates the best performing model for a given flip ratio. Based on the average rank of each technique's metrics over all data sets, the Iman-Davenport test (Demšar, 2006) rejects the null hypothesis that all methods perform equally ($p < 0.01$). Furthermore, we apply Holm's post hoc test to identify the sources of the differences in performance. The best-performing model is used as the control classifier in the pairwise comparison with all other models.

PU-SHRF and PU-HRF outperform all other techniques in terms of the F1-score with the optimized threshold. Furthermore, we can further validate with the Holm's post hoc test that PU-SHRF and PU-HRF statistically outperform the rest of techniques at the 5% significance level. The uPU, nnPU and imbalanced nnPU, considered to be state-of-the-art in the literature, generally perform better than earlier PU methods. Moreover, we observe the benefit of using imbalanced nnPU, designed for imbalanced data sets, over nnPU: imbalanced nnPU performs better than nnPU at every flip ratio. PU Bagging remains as a competitive alternative at the highest flip ratio. A possible explanation is that PU Bagging naturally handles the class imbalance because each of the bootstrap samples consists of a balanced subset of the training data. The fact that techniques for imbalanced classification such as ADASYN + Random Forest and HRF outperform most of the PU methods emphasizes the weakness of conventional PU methods for imbalanced classification.

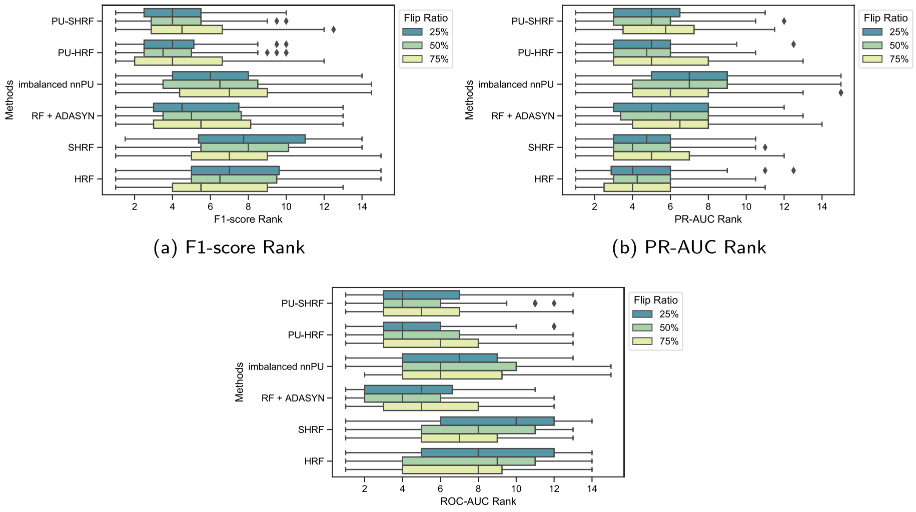
In terms of PR-AUC, HRF stands out as the best technique at every flip ratio. However, PU-SHRF is not outperformed with statistical significance at 5%. Similarly, PU-HRF is not outperformed with statistical significance at 5% when the flip ratio is either 25 or 50%. The other competitor PU method that is not statistically outperformed is imbalanced nnPU at 75%. Despite the good performance in terms of F1-score and ROC-AUC, PU Bagging performs poorly with respect to PR-AUC. This might suggest that the balanced bootstrap improves recall at the expense of precision. Furthermore, the use of a traditional data-level method for handling class imbalance together with a standard classifier, for instance ADASYN + Random Forest, is a good alternative that outperforms most older PU techniques. Based on the results in Table 4, we can observe that techniques that natively handle class imbalance perform especially well in terms of PR-AUC.

The results per data set also give more insights about the techniques' performance. Table 6, 8, and 10 present the average F1-score per data set at different flip ratios. Table 7, 9, and 11 contain the average PR-AUC for each data set at different flip ratios. The best performing technique remains consistent at low and medium flip ratios in most data sets: in the PEN data set, PU-SHRF outperforms all other techniques in F1-score, except for the setting with the high flip ratio where PU bagging dominates. This might explain why PU bagging becomes a competitive alternative to our methods in the aggregate results (Table 3) at the high flip ratio. Moreover, PU-HRF and PU-SHRF show strong performance in both F1-score and PR-AUC compared to other PU methods: in FCC, the most imbalanced data set, PU-HRF and PU-SHRF substantially outperform the PU techniques.

From the empirical analysis, we can derive some general insights that highlight the advantages of PU-SHRF and PU-HRF. PU classification under high class imbalance poses a challenge to most PU methods. Despite not being able to learn from PU data, a resampling strategy might be sufficient to outperform most PU methods. The PU methods that perform well in imbalanced data sets are those that have integrated a specialized mechanism that diminishes the bias towards the majority class: imbalanced nnPU incorporates oversampling in the risk minimization, whereas PU Bagging exploits balanced bootstrap sampling. However, each of these strategies achieves either better recall (i.e.,ROC-AUC or F1-score) or better precision (i.e., PR-AUC). We also observe that HRF and SHRF are competitive techniques for imbalanced learning, particularly in precision related metrics such as PR-AUC. Our PU methods (PU-SHRF and PU-HRF) allows to reach a state-of-the-art performance because it combines a better mechanism to retrieve unlabeled positives and the robustness of Hellinger distance for imbalanced learning. Furthermore, a tailored bootstrap sampling that guarantees that the positive instances are always considered can help to improve the performance under a high flip ratio.

5.2 Analysis on the five top imbalanced data sets

We present more granular results for data sets with the highest class imbalance. We are particularly interested in the scenario where the underrepresentation of the minority class is severe: the scenario relates to applications such as fraud detection and medical diagnosis. Besides our PU Methods, we consider HRF, SHRF, ADASYN+RF and imbalanced nnPU



(a) F1-score Rank

(b) PR-AUC Rank

(c) ROC-AUC Rank

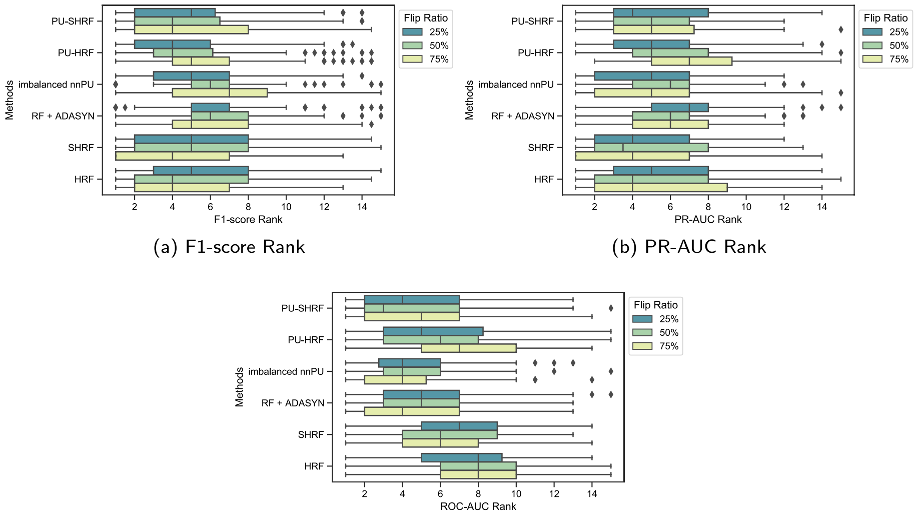
Fig. 3 Comparison of F1-score with optimal threshold, PR-AUC, and ROC-AUC ranks across different flip ratios in the top imbalanced data sets. Each experimental setting is repeated 20 times

in the analysis based on the performance shown in Sect. 5.1. Furthermore, we select the five data sets that display of the highest class imbalance: FCC, FCO, POK, SAT, and KDD. Figure 3 visualizes the rank of each metric to emphasize the relative performance between methods.

In Fig. 3a, we can observe that our PU methods present a better median rank than other methods across all flip ratios. Furthermore, we can observe that our PU methods show less variability in the F1-score ranks compared to the rest of the competitors. In contrast to the aggregated results, RF+ADASYN seems to dominate imbalanced nnPU for the high dimensional data sets. In Fig. 3b, SHRF and HRF maintain a slight advantage over our PU methods. PU-SHRF and PU-HRF present a better median rank than RF+ADASYN and imbalanced nnPU. The overall strong performance in PR-AUC of the Hellinger-based methods in the top imbalanced data sets provides evidence in line with previous studies regarding the effectiveness of the Hellinger distance on extremely imbalanced data sets (Cieslak & Chawla, 2008; Cieslak et al., 2012). Lastly, in Fig. 3c, PU-SHRF, PU-HRF, and RF+ADASYN perform similarly and outperform the other methods. At low and medium flip ratio, our PU methods outperform the other competitors (except for RF+ADASYN) in median rank.

5.3 Analysis on the top five high dimensional data sets

Similar to the previous subsection, we present more in-detail results for data sets with the highest number of features. Given that our PU methods are based on the random forest algorithm, a high number of irrelevant features might negatively affect the methods' performance. The same methods in Sect. 5.2 are used in the analysis. Furthermore, we select the following data sets that contain the highest number of features: SPE, FIE, MNI, TVC, and CCH. Figure 4 also visualizes the rank of F1-score, ROC-AUC, and PR-AUC.



(a) F1-score Rank

(b) PR-AUC Rank

(c) ROC-AUC Rank

Fig. 4 Comparison of F1-score with optimal threshold, PR-AUC, and ROC-AUC ranks across different flip ratios in the top high dimensional data sets. Each experimental setting is repeated 20 times

In Fig. 4a, our PU methods perform slightly better than imbalanced nnPU and RF+ADASYN. However, we observe a large overlap between our methods with SHRF and HRF in F1-score and PR-AUC. Moreover, PU-SHRF and HRF present smaller variability in ranks than SHRF and HRF. In Fig. 4b, SHRF and HRF maintain a slight advantage over our PU methods. PU-SHRF shows a better median rank than RF+ADASYN and imbalanced nnPU across the flip ratios. However, RF+ADASYN obtains the smallest interquartile range (IQR) whereas SHRF and HRF suffer from the largest IQR. In Fig. 4c, PU-SHRF and PU-HRF outperform SHRF and HRF, accordingly. Moreover, PU-SHRF reaches the best median rank at low and medium flip ratio. Nevertheless, imbalanced nnPU and RF+ADASYN dominate at high flip ratio. It is important to notice that imbalanced nnPU does not outperform the rest of the methods despite using XGBoost as its classifier. In the positive–negative setting, boosting is more robust to high dimensional data sets than bagging-based techniques. However, under the PU setting, bagging often shows a better ensemble strategy as it is more robust to overfitting (Frénay & Verleysen, 2013).

5.4 Sensitivity analysis on class prior estimate

The accurate estimation of the class prior is important for most of the state-of-the-art PU methods (Bekker & Davis, 2020). Our tree-based methods exploit the class prior to incorporate unlabeled positives into the node split. On the one hand, an underestimated class prior can lead to insufficient retrieval of unlabeled positives for the learning task. On the other hand, an overestimation of the class prior can consistently create more false positives because an excessive number of unlabeled instances are regarded as positives. Several studies have proposed techniques to provide an estimate of the class prior (Bekker & Davis, 2018; Plessis et al., 2017; Ramaswamy et al., 2016) under the SCAR assumption. In this work, we opt for a sensitivity analysis to measure the change in performance when the

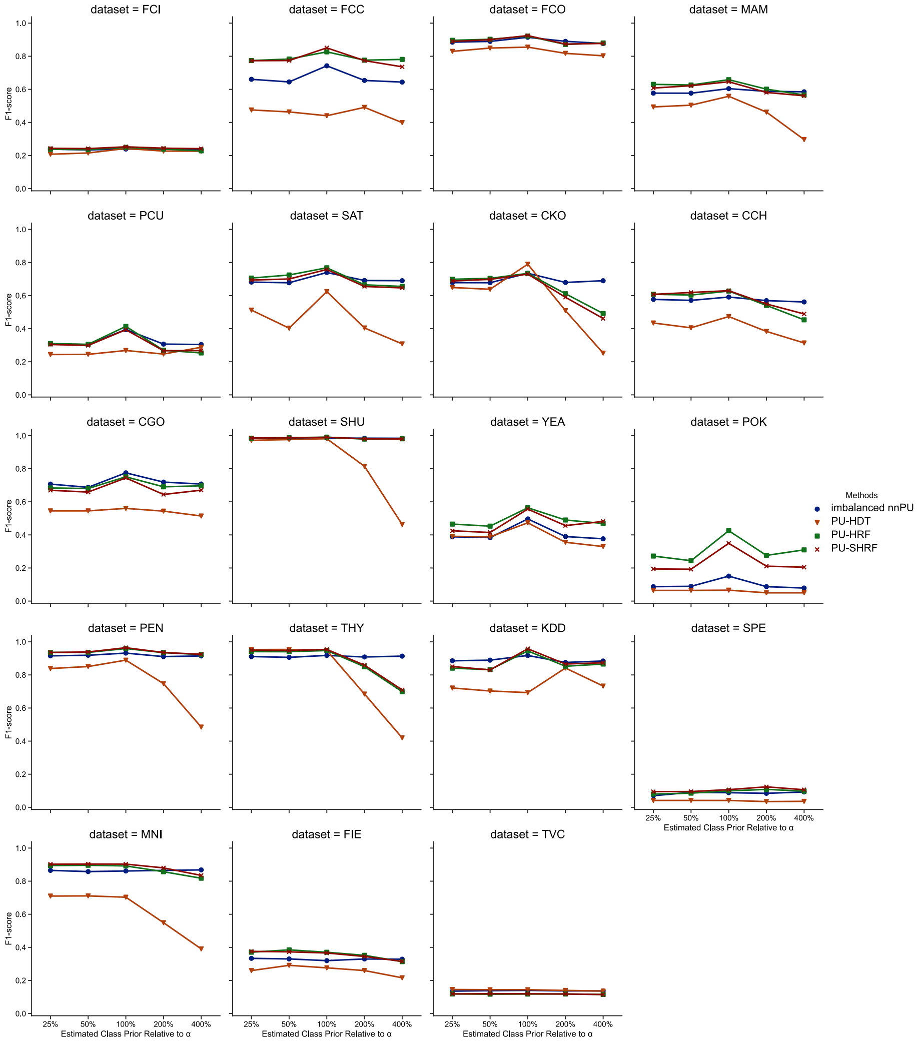


Fig. 5 Average F1-score (%) with optimal threshold per data set at 50% flip ratio across different class prior estimates. Each experimental setting is repeated 20 times

class prior is wrongly estimated. The sensitivity analysis focuses on the SCAR setting with a 50% flip ratio to represent a scenario in which half of the positives are mislabeled. Similar previous subsections, we select imbalanced nnPU, PU-HDT, PU-HRF and PU-SHRF for the sensitivity analysis. We utilize different values of the class prior estimate based on the ground-truth class prior α: the sensitivity analysis includes five values for the class prior estimate $\hat{\alpha} \in \{0.25\alpha, 0.50\alpha, \alpha, 2\alpha, 4\alpha\}$. We perform the analysis per data set with 20 repetitions for each experimental setting.

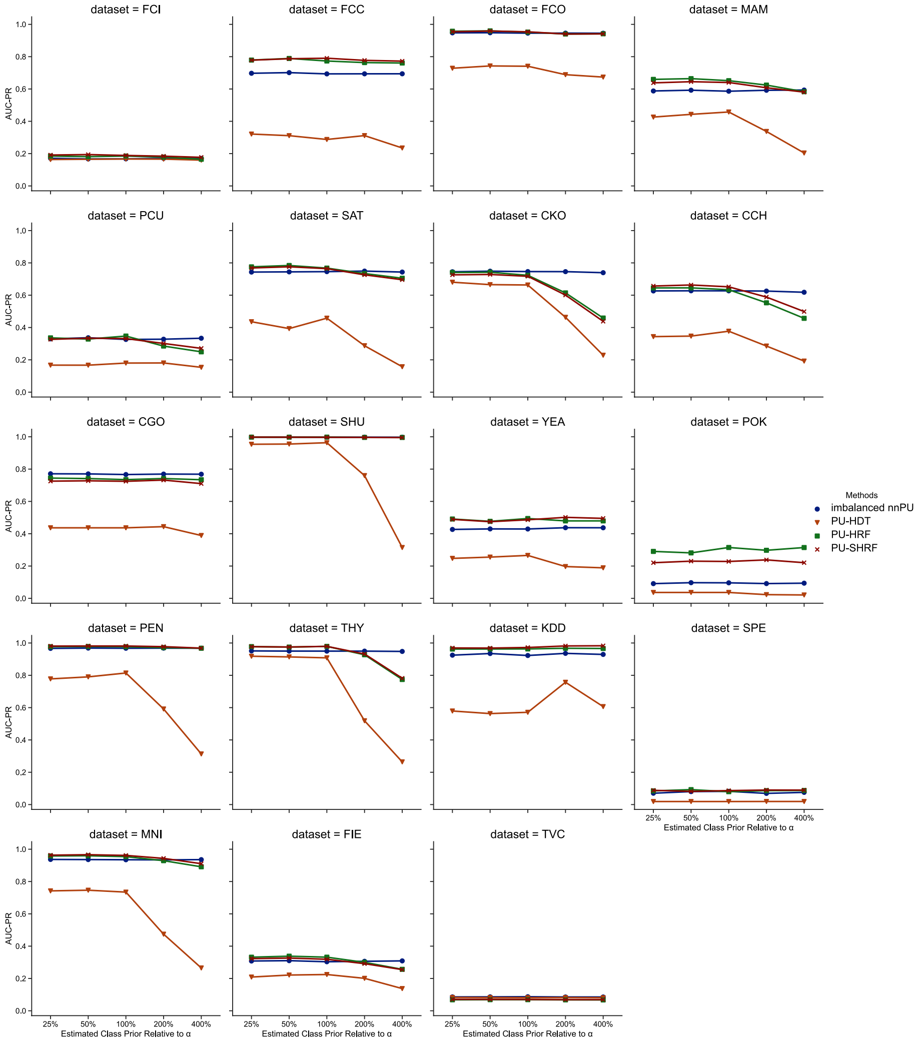


Fig. 6 Average PR-AUC (%) per data set at 50% flip ratio across different class prior estimates. Each experimental setting is repeated 20 times

Figure 5 and Fig. 6 present the results of the sensitivity analysis. Among the selected PU methods, imbalanced nnPU seems to be the most robust to a wrong class prior estimate. The robustness of (imbalanced) nnPU in our experiments confirms previous results (Kiryo et al., 2017): due to the modified PU loss function that prevents a negative empirical risk, nnPU offers stronger robustness compared with uPU (Du Plessis et al., 2015). In Fig. 5 on F1-score, in some data sets (e.g., FCC, MAM, PCU, CGO, YEA, POK, KDD), our algorithms present a “hat" pattern with a peak in performance when the estimated class prior is the ground truth. The pattern is more extreme in PU-HDT compared to PU-HRF and PU-SHRF. Thus, our ensemble-based methods should be preferred over PU-HDT due to stronger robustness to a wrong class prior estimate. Despite an average lower F1-score, imbalanced nnPU shows a smoother “hat" pattern which relates to a smaller drop in performance when the class prior is wrongly estimated. The overestimation of the class prior can be more harmful to our PU methods than the imbalanced nnPU.

In Fig. 6 on PR-AUC, our PU methods shows in some data sets a stronger weakness to overestimation than underestimation: this pattern is more noticeable in PCU, CKO, CCH, and THY. For PU-HDT, the drop in performance is larger than the ensemble-based PU methods when the class prior is overestimated. Intuitively, overestimating the class prior leads to more false positives in the method’s learning.

6 Conclusions

In this paper, we introduce a novel PU learning technique to handle highly imbalanced data sets: PU Hellinger Decision Tree (PU-HDT). PU-HDT utilizes the Hellinger distance as the splitting criterion, which shows robustness to extreme class imbalance. Furthermore, PU-HDT can learn from PU data by means of the estimation of positives from unlabeled instances at each node of the tree. Unlike other PU methods for imbalanced learning, the PU-HDT does not entail additional misclassification costs or require a resampling strategy. By using PU-HDT as the base learner, we propose the PU Hellinger Stratified Random Forest (PU-SHRF). The empirical analysis suggests that PU-SHRF generally outperforms all well-known PU methods under all experimental settings. Moreover, we emphasize the weakness of most PU methods to the imbalanced setting: techniques for imbalanced learning can outperform state-of-the-art PU methods without an adaptation for imbalanced data sets. Statistical hypothesis testing is applied to further validate the empirical findings. Furthermore, we perform a sensitivity analysis regarding the class prior estimate. We show that PU-SHRF and PU-HRF are more robust to PU-HDT to a wrong class prior estimate. Nevertheless, it is important that the class prior is sufficiently well estimated.

There are several possible research directions for future work. In this work, we assume that labeled positives represent a random subset of the positive class: this scenario refers to the selected completely at random (SCAR) assumption. However, the SCAR assumption does not hold in most real-world applications. Thus, we could extend the current work to accommodate more realistic assumptions. Another interesting line of work relates to imbalanced data streams. Previous works have already exploited HDTs in imbalanced data streams. To the best of our knowledge, no work has yet explored PU learning in imbalanced data streams.

Appendix 1

See Tables 5, 6, 7, 8, 9, 10, 11.

Table 5 Average ROC-AUC (%) with average rank in parenthesis and standard deviation (SD) at different flip ratios

Model	ROC-AUC (Rank) ± SD		
	Flip Ratio:		
	25%	50%	75%
PU-HDT	82.8 (11.2) ± 4.5	79.8 (11.5) ± 4.8	76.4 (10.7) ± 5.8
PU-HRF	**90.2 (5.0) ± 3.1**	**89.2 (5.3) ± 3.4**	84.9 (7.2) ± 3.7
PU-SHRF	**90.2 (4.5) ± 3.0**	**89.3 (4.5) ± 3.1**	**85.7 (5.9) ± 3.7**
imbalanced nnPU	**89.3 (5.5) ± 3.1**	88.0 (5.5) ± 3.9	**85.7 (5.5) ± 4.5**
nnPU	**89.3 (5.9) ± 2.9**	88.1 (5.8) ± 3.8	84.7 (6.0) ± 5.3
uPU	89.1 (7.5) ± 3.3	86.7 (8.5) ± 4.1	82.7 (8.5) ± 5.3
Ranking Pruning	84.9 (9.0) ± 3.7	83.9 (8.8) ± 4.1	80.2 (8.5) ± 5.4
PU Bagging	88.3 (7.9) ± 3.4	88.5 (6.8) ± 2.9	**87.1 (5.1) ± 3.6**
Elkan-Noto's Method	81.1 (10.9) ± 6.5	76.3 (11.6) ± 10.8	65.5 (12.1) ± 17.5
PU W. Logistic Regression	84.8 (8.9) ± 3.6	83.0 (8.8) ± 4.6	79.5 (8.6) ± 5.6
Spy-EM	79.3 (11.7) ± 5.4	78.6 (10.9) ± 7.3	75.5 (9.8) ± 7.7
ADASYN+RF	**90.7 (5.3) ± 2.9**	**89.8 (5.0) ± 3.0**	**85.8 (5.6) ± 3.9**
HDT	82.8 (11.1) ± 4.6	79.6 (11.5) ± 4.9	74.3 (11.3) ± 6.0
HRF	89.4 (7.9) ± 3.4	88.2 (8.0) ± 3.6	84.5 (8.0) ± 4.2
SHRF	89.2 (7.8) ± 3.4	88.3 (7.4) ± 3.3	85.0 (7.3) ± 4.0
Iman-Davenport test	$p < 0.01$	$p < 0.01$	$p < 0.01$

Best-performing model is indicated by bold and underlined in each column

Values in bold indicate that the best-performing model does not outperform the classifier in the model column at the 5% significance level

Table 6 Average F1-score (%) with optimal threshold over holdout 20 repetitions at 25% flip ratio

Model	CGO	CCH	FCO	FCI	FCC	KDD	CKO	MAM	PEN	PCU	POK	SAT	SHU	THY	YEA	TVC	SPE	MNI	FIE
PU-HDT	56.1	46.4	88.9	25.6	57.3	87.8	**83.4**	55.3	91.1	26.9	6.1	61.8	98.8	95.7	51.6	**16.0**	4.4	72.1	31.6
PU-HRF	78.4	66.0	94.6	26.8	**85.0**	99.2	81.3	**68.1**	**98.4**	**44.7**	50.1	77.7	**99.5**	96.6	58.4	12.1	14.8	91.2	41.4
PU-SHRF	79.9	**67.0**	**94.8**	26.6	84.9	99.2	80.2	67.3	**98.4**	44.5	44.8	**78.3**	**99.5**	**96.8**	57.5	12.0	14.4	91.7	40.0
imbalanced nnPU	**81.7**	65.3	94.3	25.7	80.0	98.1	82.2	65.2	97.1	41.1	20.0	74.2	99.2	94.3	52.2	14.0	12.1	90.5	38.0
nnPU	79.7	65.7	93.9	25.7	80.1	97.7	81.5	65.7	96.9	42.6	14.8	74.9	99.1	94.9	49.3	14.2	10.3	90.7	37.1
uPU	79.0	61.0	91.7	26.6	78.0	96.8	82.3	65.9	90.9	37.6	28.1	70.2	98.0	91.5	50.6	14.3	12.0	84.9	40.0
Ranking Pruning	66.4	43.9	92.8	26.7	61.6	95.9	22.6	63.1	92.7	35.1	18.1	69.9	98.1	80.5	56.0	10.8	13.0	81.8	22.8
PU Bagging	59.4	46.3	90.8	22.5	64.1	**100.0**	29.9	44.3	93.9	41.4	36.2	75.6	97.9	88.5	**59.9**	11.5	**22.8**	83.1	22.5
Elkan-Noto's Method	29.9	26.1	92.6	14.1	24.9	97.7	10.2	12.8	93.8	28.0	0.0	67.4	80.9	15.3	30.8	5.6	8.7	35.3	9.0
PU Weighted LR	68.1	43.3	91.9	**27.0**	42.1	91.8	22.6	59.4	89.2	38.9	6.2	73.2	97.9	82.9	51.3	11.3	13.4	82.7	22.0
Spy-EM	54.2	13.0	48.5	14.3	4.8	90.7	14.9	27.2	65.1	26.2	5.2	6.3	93.9	16.3	8.1	9.1	3.8	51.4	9.3
ADASYN+RF	79.0	62.7	93.6	25.1	80.3	98.9	69.4	62.9	98.0	43.7	**62.3**	72.6	99.0	96.5	55.6	12.0	15.3	89.1	39.0
HDT	56.1	46.3	88.3	25.4	52.5	87.9	82.9	57.8	90.5	26.3	6.1	63.8	99.0	96.4	51.4	15.8	4.4	73.3	32.5
HRF	75.2	64.2	93.1	25.3	79.0	93.2	78.4	64.6	96.0	32.6	29.9	72.7	99.2	95.8	48.9	11.9	11.9	91.4	**41.6**
SHRF	74.7	64.5	93.0	25.4	75.5	91.9	76.5	64.5	96.2	36.5	19.1	70.7	99.1	96.0	48.9	12.0	14.8	**92.0**	40.5

Best-performing model is indicated by bold in each column

Table 7 Average PR-AUC (%) over holdout 20 repetitions at 25% flip ratio

Model	CGO	CCH	FCO	FCI	FCC	KDD	CKO	MAM	PEN	PCU	POK	SAT	SHU	THY	YEA	TVC	SPE	MNI	FIE
PU-HDT	43.4	40.6	80.1	18.0	39.8	79.2	74.3	47.9	87.1	17.8	3.2	45.8	97.7	92.2	31.9	8.4	2.1	76.6	26.6
PU-HRF	78.6	69.6	97.7	20.8	77.6	99.7	80.0	**69.0**	99.4	38.0	39.7	80.2	99.9	**98.8**	**51.0**	7.1	12.0	97.0	38.5
PU-SHRF	77.8	71.5	97.9	21.0	77.1	99.8	78.2	68.1	99.5	38.4	31.2	80.7	99.9	**98.8**	49.1	7.1	13.1	97.3	36.3
Imbalanced nnPU	79.7	70.2	96.6	19.0	74.8	98.3	80.0	64.2	98.6	36.1	12.4	76.6	99.9	97.0	41.7	**8.9**	10.0	96.4	36.5
nnPU	75.9	70.9	96.5	19.1	74.5	97.6	79.5	64.6	98.8	36.9	9.7	75.7	99.9	97.1	41.2	**8.9**	9.2	96.6	36.2
uPU	76.4	63.3	91.5	21.0	72.3	97.2	77.0	63.0	92.5	30.8	19.1	69.9	99.2	90.3	40.7	8.8	11.3	86.8	38.1
Ranking Pruning	52.2	39.2	96.0	17.7	52.3	98.9	15.4	60.2	96.0	28.4	11.8	68.3	97.1	84.2	47.0	6.8	9.2	89.9	17.6
PU Bagging	42.4	34.0	87.0	12.4	49.5	**100.0**	16.4	27.0	92.4	28.7	26.3	72.4	96.9	87.1	44.0	6.8	15.3	87.1	15.2
Elkan-Noto’s Method	49.3	37.5	95.2	9.0	37.1	98.9	6.9	54.3	95.8	30.5	13.5	71.8	97.1	87.3	36.5	5.3	12.9	89.1	18.9
PU Weighted LR	54.1	37.2	96.1	17.5	34.5	95.4	15.6	53.5	92.5	33.1	1.9	75.5	97.0	85.7	40.5	6.7	10.7	89.6	16.6
Spy-EM	55.2	9.8	40.4	11.7	3.4	84.5	8.3	29.3	76.4	16.2	2.7	6.1	95.6	43.0	49.5	6.8	2.5	63.1	6.0
ADASYN+RF	**79.9**	64.2	97.3	17.6	72.0	99.5	67.2	60.9	**99.6**	36.4	**57.2**	75.4	99.9	98.6	47.3	7.3	**16.3**	95.4	36.4
HDT	43.4	38.4	80.4	17.7	37.1	79.5	73.6	48.6	83.0	17.8	3.2	46.7	97.9	93.4	30.4	8.4	2.1	78.0	26.0
HRF	78.9	70.6	**98.2**	20.6	**79.6**	99.4	**80.9**	68.8	99.4	39.4	40.8	**81.0**	**100.0**	98.4	49.5	7.1	10.8	97.1	**38.6**
SHRF	78.5	**71.8**	98.0	**21.1**	78.1	99.9	78.8	68.1	99.4	**40.2**	32.4	80.3	**100.0**	98.4	49.1	7.2	13.6	**97.5**	36.8

Best-performing model is indicated by bold in each column

Table 8 Average F1-score (%) with optimal threshold over holdout 20 repetitions at 50% flip ratio

Model	CGO	CCH	FCO	FCI	FCC	KDD	CKO	MAM	PEN	PCU	POK	SAT	SHU	THY	YEA	TVC	SPE	MNI	FIE
PU-HDT	56.0	47.3	85.5	24.3	44.0	69.3	79.0	55.8	88.9	26.8	6.6	62.4	98.1	94.8	47.3	14.4	4.1	70.3	27.6
PU-HRF	75.1	62.7	92.2	24.9	82.7	94.5	73.4	**65.8**	95.9	**41.4**	42.5	**76.8**	**99.1**	94.8	56.4	11.8	9.9	89.2	37.0
PU-SHRF	74.4	**63.0**	92.5	25.3	**85.1**	95.8	73.0	64.6	**96.5**	39.4	34.9	75.8	99.0	**95.4**	55.6	11.9	10.7	**90.3**	36.6
imbalanced nnPU	**77.5**	59.1	91.4	23.9	74.2	91.8	73.5	60.4	93.2	39.5	15.1	73.9	98.5	91.8	49.6	14.0	8.8	86.2	32.0
nnPU	75.0	59.4	91.4	23.4	74.0	91.8	71.7	60.7	93.1	39.4	15.0	74.8	98.1	92.1	49.1	**14.5**	5.9	87.1	31.9
uPU	72.1	52.2	87.2	23.9	71.3	93.6	75.5	62.7	84.6	35.7	35.3	64.3	96.5	87.4	45.7	13.9	7.5	80.6	36.0
Ranking Pruning	65.4	45.2	90.7	25.5	57.1	79.9	20.0	62.0	89.8	32.7	12.4	59.9	97.7	74.6	56.4	10.6	12.1	77.8	17.1
PU Bagging	55.7	46.3	91.2	22.6	68.9	**100.0**	34.4	46.6	92.7	40.2	38.7	75.0	97.6	86.7	**59.3**	11.7	**20.1**	82.3	18.9
Elkan-Noto's Method	23.2	20.3	69.2	12.9	23.5	79.1	9.4	15.5	69.7	23.7	2.9	60.9	56.4	24.2	15.0	5.1	11.3	30.0	4.6
PU Weighted LR	68.6	42.5	**92.6**	**25.7**	32.6	74.0	21.1	57.9	87.9	36.3	7.4	61.9	97.5	76.8	51.7	10.8	15.5	80.2	16.8
Spy-EM	54.5	12.2	48.9	18.8	5.2	82.3	14.5	29.3	66.1	24.1	6.4	7.8	94.0	15.7	8.6	9.3	4.0	32.9	8.4
ADASYN+RF	74.8	57.8	90.8	24.7	82.2	96.3	58.9	58.3	96.0	38.4	**46.8**	68.6	97.9	95.0	56.8	11.5	10.2	87.1	35.7
HDT	56.0	45.5	85.1	24.3	46.4	70.3	**80.2**	55.4	87.8	26.3	6.6	56.2	97.7	**95.4**	46.6	14.4	4.2	71.1	29.1
HRF	68.0	61.6	90.3	23.8	78.3	83.2	71.5	62.6	93.6	30.5	26.3	72.4	98.6	94.0	46.3	11.8	8.4	89.6	**38.5**
SHRF	65.5	60.4	89.0	24.2	78.7	82.9	68.8	61.9	93.9	30.0	17.9	71.9	98.6	94.6	42.4	11.9	8.6	90.1	37.5

Best-performing model is indicated by bold in each column

Table 9 Average PR-AUC (%) over holdout 20 repetitions at 50% flip ratio

Model	CGO	CCH	FCO	FCI	FCC	KDD	CKO	MAM	PEN	PCU	POK	SAT	SHU	THY	YEA	TVC	SPE	MNI	FIE
PU-HDT	43.7	37.7	74.1	16.6	28.8	57.1	66.3	45.8	81.4	18.0	3.6	45.8	96.3	90.8	26.6	8.1	1.9	73.4	22.4
PU-HRF	73.5	63.4	95.2	18.5	77.3	96.3	72.3	65.1	97.7	**34.7**	31.5	76.8	**99.8**	97.9	49.4	6.9	8.0	95.4	33.2
PU-SHRF	72.5	65.2	95.4	18.9	79.0	97.2	71.7	64.0	98.1	33.1	22.8	76.4	99.7	97.9	48.6	7.0	8.6	96.2	31.9
imbalanced nnPU	**76.6**	62.7	94.6	16.8	69.3	92.3	74.6	58.6	96.7	32.6	9.6	74.6	99.7	95.0	42.9	8.7	8.2	93.5	30.3
nnPU	73.4	63.6	94.5	16.9	71.2	91.2	72.3	58.9	96.8	33.5	9.0	74.7	99.7	95.7	42.7	**8.8**	7.0	93.9	30.2
uPU	70.4	49.7	85.4	17.9	64.8	93.8	73.0	57.0	85.6	25.5	24.4	63.4	98.1	85.0	34.7	8.1	6.7	79.8	30.7
Ranking Pruning	54.5	40.7	91.7	16.7	48.7	89.7	12.6	61.0	91.4	23.4	6.6	55.8	96.0	79.5	46.5	6.5	11.3	88.2	12.2
PU Bagging	38.7	37.4	88.9	12.5	56.6	**100.0**	19.8	29.5	93.1	29.2	27.9	73.6	96.5	88.4	44.0	6.6	13.8	87.7	13.1
Elkan-Noto's Method	36.6	23.6	89.6	8.1	30.8	85.5	6.5	24.4	84.7	25.7	9.6	60.1	92.0	61.3	23.3	5.2	**14.2**	80.9	11.8
PU Weighted LR	54.5	37.9	95.6	16.4	23.6	81.1	13.6	52.0	90.2	28.0	3.6	61.8	96.1	81.2	43.2	6.4	11.0	87.3	12.5
Spy-EM	55.8	10.5	41.9	13.5	3.8	73.0	8.7	31.0	69.4	17.2	4.2	11.7	95.3	39.2	**53.0**	5.5	2.6	44.3	5.2
ADASYN+RF	75.1	57.4	94.7	16.4	76.8	98.1	52.3	55.5	**98.6**	31.0	**35.8**	70.1	99.6	**98.1**	48.1	7.2	11.2	94.0	32.1
HDT	43.7	34.7	74.3	16.5	31.1	56.3	66.8	44.3	79.0	16.7	3.6	39.3	95.5	91.4	25.6	8.2	1.9	74.6	22.1
HRF	74.3	65.3	**96.0**	18.1	78.9	96.0	**74.7**	**66.4**	97.8	32.8	28.6	**78.4**	**99.8**	97.5	48.5	6.8	8.0	95.9	**33.8**
SHRF	72.7	**66.3**	95.8	**19.1**	**79.1**	96.9	72.7	64.8	98.2	33.3	23.2	77.9	**99.8**	97.5	47.4	7.1	8.0	**96.5**	32.6

Best-performing model is indicated by bold in each column

Table 10 Average F1-score (%) with optimal threshold over holdout 20 repetitions at 75% flip ratio

Model	CGO	CCH	FCO	FCI	FCC	KDD	CKO	MAM	PEN	PCU	POK	SAT	SHU	THY	YEA	TVC	SPE	MNI	FIE
PU-HDT	56.9	39.3	78.1	23.3	36.2	69.8	74.7	52.1	86.6	24.1	11.1	57.4	97.2	89.9	34.0	12.6	4.5	67.4	23.8
PU-HRF	60.0	49.1	84.8	22.4	81.5	93.8	56.0	**57.8**	90.4	32.0	32.1	**70.6**	95.3	89.0	46.8	11.4	4.4	82.9	30.9
PU-SHRF	61.4	50.4	84.7	23.3	**82.2**	95.0	58.6	56.6	90.9	34.6	26.9	67.3	95.9	90.2	50.1	11.1	7.6	84.8	31.2
imbalanced nnPU	61.3	46.8	82.5	19.1	69.8	91.7	62.3	51.7	84.4	33.9	14.2	67.2	94.4	82.8	45.7	13.4	4.9	76.8	28.4
nnPU	63.1	44.6	80.6	18.9	62.1	89.1	58.0	51.7	80.6	31.9	16.7	65.1	89.9	80.4	37.2	**13.7**	3.6	74.6	28.5
uPU	58.6	42.3	81.5	21.5	58.9	90.0	65.8	54.3	77.1	30.9	18.5	57.7	94.5	83.5	35.7	12.9	6.1	76.7	29.5
Ranking Pruning	63.0	39.0	84.5	24.7	51.5	75.2	17.8	62.3	85.0	26.0	17.7	45.0	97.3	65.1	49.3	10.0	9.3	69.0	14.1
PU Bagging	56.2	41.3	**88.9**	23.3	77.0	**100.0**	39.9	51.8	**92.6**	**36.4**	**42.7**	**70.6**	**97.4**	80.9	**58.9**	11.4	**16.3**	78.3	18.7
Elkan-Noto's Method	3.6	6.6	7.5	3.4	15.5	30.1	0.5	1.6	28.8	10.3	1.4	27.0	14.0	14.2	2.6	0.1	4.4	4.6	1.5
PU Weighted LR	**65.8**	37.8	86.5	**25.3**	25.5	63.2	20.5	57.0	85.7	24.8	14.1	43.0	97.2	63.5	48.1	10.4	13.2	71.5	12.4
Spy-EM	52.5	10.1	46.4	17.9	5.4	80.2	13.0	30.8	77.5	23.4	7.4	26.4	95.4	15.0	15.3	9.1	4.0	18.9	6.9
ADASYN+RF	59.1	48.2	85.0	23.3	77.4	93.7	47.7	50.4	91.7	33.8	28.9	58.1	95.9	90.9	49.1	11.7	7.7	83.1	30.2
HDT	56.6	37.0	71.2	23.1	24.8	53.4	**75.3**	51.0	82.5	19.6	10.3	44.4	94.7	**93.3**	32.1	12.4	4.7	66.5	21.0
HRF	54.5	49.9	84.6	22.1	68.9	85.8	53.9	55.2	88.6	26.4	18.8	67.8	96.3	89.0	34.7	11.7	6.3	85.7	32.5
SHRF	53.7	**51.1**	85.2	21.9	66.6	86.2	54.4	53.0	86.8	26.4	12.6	64.7	96.4	89.9	33.6	11.4	7.2	**86.5**	**32.7**

Best-performing model is indicated by bold in each column

Table 11 Average PR-AUC (%) over holdout 20 repetitions at 75% flip ratio

Model	CGO	CCH	FCO	FCI	FCC	KDD	CKO	MAM	PEN	PCU	POK	SAT	SHU	THY	YEA	TVC	SPE	MNI	FIE
PU-HDT	42.3	26.0	62.6	14.4	23.8	58.3	60.5	38.8	76.5	15.0	7.5	38.2	94.5	82.0	17.7	7.3	1.9	66.2	15.9
PU-HRF	57.8	44.0	86.4	14.1	72.2	95.6	50.8	54.9	91.2	23.9	21.2	66.7	96.5	93.4	36.8	6.2	3.9	89.4	23.3
PU-SHRF	58.9	46.9	86.4	15.2	**74.8**	97.0	53.7	52.9	92.2	26.5	16.0	65.7	97.5	94.4	38.1	6.5	6.6	91.8	23.6
imbalanced nnPU	58.5	46.7	89.8	13.9	64.8	90.5	**63.1**	49.3	92.5	**29.5**	11.7	67.1	**99.1**	90.3	39.0	8.2	4.8	87.2	24.3
nnPU	**61.1**	47.3	**91.0**	14.5	59.6	88.2	56.6	46.4	93.4	28.8	11.8	64.7	98.7	90.3	34.0	**8.3**	5.4	86.9	24.1
uPU	50.3	32.4	77.8	14.3	57.1	89.0	59.9	41.8	74.8	19.0	12.9	48.8	95.9	79.3	25.5	7.8	3.3	71.7	20.6
Ranking Pruning	52.6	32.9	86.4	15.9	49.3	74.7	10.9	**60.2**	87.6	17.6	13.8	35.7	95.2	70.4	39.7	6.2	9.9	79.6	9.1
PU Bagging	39.2	35.2	88.7	13.2	65.1	**100.0**	25.8	34.4	94.5	24.9	**31.2**	64.8	96.7	85.0	45.2	6.6	9.7	85.9	11.9
Elkan-Noto’s Method	31.1	11.3	28.7	8.8	29.0	74.6	5.5	11.0	52.9	13.9	13.8	33.7	55.2	8.3	17.0	5.4	7.4	36.9	8.8
PU Weighted LR	53.3	31.1	88.5	**16.1**	19.1	69.7	11.8	51.0	88.0	17.9	12.0	39.5	95.2	68.8	36.0	6.1	**11.3**	79.3	9.4
Spy-EM	53.8	9.9	37.6	12.3	4.0	70.6	10.7	33.1	82.4	17.0	4.7	48.2	95.8	21.2	**46.7**	4.8	4.1	12.4	3.7
ADASYN+RF	55.2	42.4	85.9	14.4	73.0	95.2	43.2	45.7	**95.3**	23.2	20.6	57.0	97.7	**95.1**	38.0	6.6	6.6	89.9	25.1
HDT	41.8	24.0	55.7	14.5	15.8	42.8	60.6	36.6	70.6	11.6	7.3	28.4	90.5	87.7	16.6	7.3	1.9	64.5	16.2
HRF	57.1	48.5	90.6	14.8	73.1	94.9	56.7	56.3	93.3	27.3	21.9	**72.0**	97.7	93.6	34.4	6.3	4.0	91.9	**26.0**
SHRF	59.4	**50.1**	90.6	15.2	71.7	95.9	58.3	54.2	93.3	26.4	17.3	69.1	97.8	94.4	35.2	6.4	5.4	**93.5**	25.2

Best-performing model is indicated by bold in each column

Ackowledgements We gratefully acknowledge the financial support provided by KBC Groep NV for this research project.

Author Contributions Carlos Ortega Vazquez, Seppe vanden Broucke, and Jochen De Weerdt contributed to all parts of the study. The first draft of the manuscript was mostly written by Carlos Ortega Vazquez, and the subsequent editing and revision was performed by Seppe van Broucke and Jochen De Weerdt. All authors commented on previous versions. All authors read and approved the final manuscript.

Funding We received financial support provided by KBC Groep NV for this research project.

Data availability The preprocessing of data sets can be found at https://github.com/CarlosOrtegaV/PU_Hellinger_Trees. The original open-source data sets can be found in the original data repositories as described in Sect. 4.1.

Code availability Code used for this research is available at https://github.com/CarlosOrtegaV/PU_Hellinger_Trees.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Akash, P. S., Kadir, M. E., Ali, A. A., & Shoyaib, M. (2019). Inter-node Hellinger distance based decision tree. In *IJCAI* (pp. 1967–1973).

Alcalá-Fernandez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing, 17*.

Baesens, B., Höppner, S., Ortner, I., & Verdonck, T. (2021). robROSE: A robust approach for dealing with imbalanced data in fraud detection. *Statistical Methods & Applications, 30*, 841–861.

Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Example-dependent cost-sensitive decision trees. *Expert Systems with Applications, 42*(19), 6609–6619.

Barua, S., Islam, M. M., Yao, X., & Murase, K. (2012). MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on knowledge and data engineering, 26*(2), 405–425.

Bekker, J., & Davis, J. (2018). Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32).

Bekker, J., & Davis, J. (2020). Learning from positive and unlabeled data: A survey. *Machine Learning, 109*, 719–760.

Bekker, J., Robberechts, P., & Davis, J. (2019). Beyond the selected completely at random assumption for learning from positive and unlabeled data. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 71–85).

Breiman, L. (2001). Random forests Random forests. *Machine Learning, 45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.

Cano, A., Zafra, A., & Ventura, S. (2013). Weighted data gravitation classification for standard and imbalanced data Weighted data gravitation classification for standard and imbalanced data. *IEEE Transactions on Cybernetics, 43*(6), 1672–1687.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Chen, C., Liaw, A., Breiman, L., et al. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley, 110*(1–12), 24.

Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., & Wang, Z. (2020). Self-PU: Self boosted and calibrated positive-unlabeled training. In III, H. D. & Singh, A. (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 1510–1519). PMLR.

Chen, X., Gong, C., & Yang, J. (2021). Cost-sensitive positive and unlabeled learning. *Information Sciences, 558*, 229–245.

Cieslak, D. A., & Chawla, N. V. (2008). Learning decision trees for unbalanced data. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 241–256).

Cieslak, D. A., Hoens, T. R., Chawla, N. V., & Kegelmeyer, W. P. (2012). Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery, 24*(1), 136–158.

Claesen, M., De Smet, F., Suykens, J. A., & De Moor, B. (2015). A robust ensemble approach to learn from positive and unlabeled data using SVM base models. *Neurocomputing, 160*, 73–84.

Dal Pozzolo, A., Johnson, R., Caelen, O., Waterschoot, S., Chawla, N. V., & Bontempi, G. (2014). Using HDDT to avoid instances propagation in unbalanced and evolving data streams. In *2014 International joint conference on neural networks (IJCNN)* (pp. 588–594).

Daniels, Z. A., & Metaxas, D. N. (2017). Addressing imbalance in multi-label classification using structured hellinger forests. In *Thirty-first AAAI conference on artificial intelligence*.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on machine learning* (pp. 233–240).

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1–30.

Denis, F., Gilleron, R., & Letouzey, F. (2005). Learning from positive and unlabeled examples. *Theoretical Computer Science, 348*(1), 70–83.

Dua, D., & Graff, C. (2019). UCI machine learning repository.

Du Plessis, M., Niu, G., & Sugiyama, M. (2015). Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning* (pp. 1386–1394).

Du Plessis, M., Niu, G., & Sugiyama, M. (2015). Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning* (pp. 1386–1394).

Elkan, C., & Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 213–220).

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets* (Vol. 11). Springer.

Frénay, B., & Verleysen, M. (2013). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems, 25*(5), 845–869.

Gonzalez-Abril, L., Nunez, H., Angulo, C., & Velasco, F. (2014). GSVM: An SVM for handling imbalanced accuracy between classes inbi-classification problems. *Applied Soft Computing, 17*, 23–31.

Grzyb, J., Klikowski, J., & Woźniak, M. (2021). Hellinger distance weighted ensemble for imbalanced data stream classification. *Journal of Computational Science, 51*, 101314.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878–887).

He, F., Liu, T., Webb, G. I., & Tao, D. (2018). Instance-dependent PU learning by Bayesian optimal relabeling. arXiv preprint arXiv:1808.02180

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322–1328).

Hoens, T. R., Qian, Q., Chawla, N. V., & Zhou, Z.-H. (2012). Building decision trees for the multi-class imbalance problem. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 122–134).

Kiryo, R., Niu, G., Plessis, M. C.d., & Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. arXiv preprint arXiv:1703.00593

Lee, W. S., & Liu, B. (2003). Learning with positive and unlabeled examples using weighted logistic regression. In *ICML* (Vol. 3, pp. 448–455).

Li, H., Chen, Z., Liu, B., Wei, X., & Shao, J. (2014). Spotting fake reviews via collective positive-unlabeled learning. In *2014 IEEE international conference on data mining* (pp. 899–904).

Li, X., & Liu, B. (2003). Learning to classify texts using positive and unlabeled data. In *IJCAI* (Vol. 3, pp. 587–592).

Liu, B., Dai, Y., Li, X., Lee, W. S., & Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. In *Third IEEE international conference on data mining* (pp. 179–186).

Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2002). Partially supervised classification of text documents . In *ICML* (Vol. 2, pp. 387–394).

Liu, W., & Chawla, S. (2011). Class confidence weighted KNN algorithms for imbalanced data sets. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 345–356).

Liu, W., Chawla, S., Cieslak, D. A., & Chawla, N. V. (2010). A robust decision tree algorithm for imbalanced data sets. In *Proceedings of the 2010 SIAM international conference on data mining* (pp. 766–777).

Lyon, R. J., Brooke, J., Knowles, J. D., & Stappers, B. W. (2014). Hellinger distance trees for imbalanced streams. In *2014 22nd International conference on pattern recognition* (pp. 1969–1974).

MLG. (2018). Credit card fraud version 3. https://www.kaggle.com/mlg-ulb/creditcardfraud

Mordelet, F., & Vert, J.-P. (2014). A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters, 37*, 201–209.

Northcutt, C. G., Wu, T., & Chuang, I. L. (2017). Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels. In *Proceedings of the thirty-third conference on uncertainty in artificial intelligence, UAI 2017*. AUAI Press.

Oracle. (2015). Oracle database online documentation 12c. https://docs.oracle.com/database/121/

O'Brien, R., & Ishwaran, H. (2019). A random forests quantile classifier for class imbalanced data. *Pattern Recognition, 90*, 232–249.

Plessis, M. C. D., Niu, G., & Sugiyama, M. (2017). Class-prior estimation for learning from positive and unlabeled data. *Machine Learning, 106*(4), 463–492.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Ramaswamy, H. G., Scott, C., & Tewari, A. (2016). Mixture Proportion Estimation via Kernel Embeddings of Distributions. In *Proceedings of the 33nd international conference on machine learning, ICML 2016* (Vol. 48, pp. 2052–2060). JMLR.org.

Sakai, T., Niu, G., & Sugiyama, M. (2018). Semi-supervised AUC optimization based on positive-unlabeled learning. *Machine Learning, 107*(4), 767–794.

Sardari, S., Eftekhari, M., & Afsari, F. (2017). Hesitant fuzzy decision tree approach for highly imbalanced data classification. *Applied Soft Computing, 61*, 727–741.

Shebuti, R. (2016). Odds library. http://odds.cs.stonybrook.edu

Stripling, E., Baesens, B., Chizi, B., & vanden Broucke, S. (2018). Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud. *Decision Support Systems, 111*, 13–26.

Su, C., & Cao, J. (2019). Improving lazy decision tree for imbalanced classification by using skew-insensitive criteria. *Applied Intelligence, 49*(3), 1127–1145.

Su, G., Chen, W., & Xu, M. (2021). Positive-Unlabeled Learning from Imbalanced Data. In *International joint conferences on artificial intelligence IJCAI* (pp. 2995–3001). Montreal: ijcai.org.

Vadera, S. (2010). CSNL: A cost-sensitive non-linear decision tree algorithm. *ACM Transactions on Knowledge Discovery from Data (TKDD), 4*(2), 1–25.

Van Belle, R., Van Damme, C., Tytgat, H., & De Weerdt, J. (2022). Inductive graph representation learning for fraud detection. *Expert Systems with Applications, 193*, 116463.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). OpenML: Networked science in machine learning. *SIGKDD Explorations, 15*(2), 49–60. https://doi.org/10.1145/2641190.2641198

Xie, Z., & Li, M. (2018). Semi-supervised AUC optimization without guessing labels of unlabeled data. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32).

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics* (pp. 189–196).

Yu, S., & Li, C. (2007). Pe-puc: A graph based PU-learning approach for text classification. In *International workshop on machine learning and data mining in pattern recognition international workshop on machine learning and data mining in pattern recognition* (pp. 574–584).

Zelenkov, Y. (2019). Example-dependent cost-sensitive adaptive boosting. *Expert Systems with Applications, 135*, 71–82.

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review, 5*(1), 44–53.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.