Learning Ensembles in the Presence of Imbalanced Classes

Amal Saadallah[1], Nico Piatkowski[1], Felix Finkeldey[2], Petra Wiederkehr[2] and Katharina Morik[1]
[1]*Artificial Intelligence Group, Department of Computer Science, TU Dortmund, Germany*
[2]*Virtual Machining Group, Department of Computer Science, TU Dortmund, Germany*

Keywords: Class Imbalance, Ensemble, Classification.

Abstract: Class imbalance occurs when data classes are not equally represented. Generally, it occurs when some classes represent rare events, while the other classes represent the counterpart of these events. Rare events, especially those that may have a negative impact, often require informed decision-making in a timely manner. However, class imbalance is known to induce a learning bias towards majority classes which implies a poor detection of minority classes. Thus, we propose a new ensemble method to handle class imbalance explicitly at training time. In contrast to existing ensemble methods for class imbalance that use either data driven or randomized approaches for their constructions, our method exploits both directions. On the one hand, ensemble members are built from randomized subsets of training data. On the other hand, we construct different scenarios of class imbalance for the unknown test data. An ensemble is built for each resulting scenario by combining random sampling with the estimation of the relative importance of specific loss functions. Final predictions are generated by a weighted average of each ensemble prediction. As opposed to existing methods, our approach does not try to fix imbalanced data sets. Instead, we show how imbalanced data sets can make classification *easier*, due to a limited range of true class frequencies. Our procedure promotes diversity among the ensemble members and is not sensitive to specific parameter settings. An experimental demonstration shows, that our new method outperforms or is on par with state-of-the-art ensembles and class imbalance techniques.

1 INTRODUCTION

In many real-word situations, rare events and unusual behaviors, such as process failure or instability in machine engineering, rare diseases and bank frauds, are usually represented with imbalanced data observations. In other words, one or more classes, usually the ones that represent such events, are underrepresented in the data set. This issue, known to the Data Mining community as the class imbalance problem, makes the detection of rare events a challenging task (Galar et al., 2012; Haixiang et al., 2017).

In the case of binary classification with class imbalance ratio of 1%, a trivial solution that always predicts the majority class, will achieve an accuracy of $\approx 99\%$; though the accuracy seems high, the solution is meaningless since not a single instance of the minority class is classified correctly. Thus, in our work, we make the implicit assumption that the minority class has a higher cost than the majority class (Zhou, 2012).

Several machine learning approaches have been proposed over the past decades to handle the class imbalance problem, most of which are based on resampling techniques, cost sensitive learning, and ensemble methods (Galar et al., 2012; Haixiang et al., 2017).

In this paper, we address the problem of class imbalance via an ensemble method. In general, ensemble methods train multiple classifiers and combine their predictions to solve the same classification task[1]. Ensembles are known to deliver a higher quality than each single ensemble member (Hansen and Salamon, 1990) and they provide state-of-the-art results in various real-world tasks (Galar et al., 2012).

Ensemble methods are not designed to work with imbalanced data sets since they do not take class imbalance into account during the ensemble construction. However, they have shown successful results when applied to this task through the combination of various processing techniques at a data-level (e.g. random sampling, feature selection, cost-sensitive learning methods) with different ensemble methods (Galar et al., 2012; Haixiang et al., 2017).

[1]While ensembles may indeed be applied to regression problems as well, we focus in this work on classification tasks.

Saadallah, A., Piatkowski, N., Finkeldey, F., Wiederkehr, P. and Morik, K.
Learning Ensembles in the Presence of Imbalanced Classes.
DOI: 10.5220/0007681508660873
In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, pages 866-873
ISBN: 978-989-758-351-3

Nevertheless, to the best of our knowledge, none of these techniques allow to include explicit (estimated) knowledge about the class imbalance in the distribution.

In this work, we explain how to address class imbalance directly by a new ensemble construction that exploits both data-driven and randomized approaches. On the one hand, ensembles members are built from randomized subsets of the training data. On the other hand, we use the class ratio from the training data to construct different scenarios of class imbalance in the unknown test set. The random sampling is then carried out for each scenario. This procedure promotes diversity among the ensemble members and subdivides the ensemble into multiple weighted stages. As opposed to existing methods, our approach does not try to fix imbalanced data sets. Instead, we show how imbalanced data sets can make classification *easier*, due to a limited range of true class frequencies.

2 REVIEW ON EXISTING TECHNIQUES FOR HANDLING IMBALANCED DATA SETS

Several works have been proposed in literature to address the class imbalance problem over the last decades. Resampling methods and cost-sensitive learning are the two main strategies that have been employed for imbalanced learning.

Resampling strategies allow to rebalance the data set in order to mitigate the effect of the bias of machine learning algorithms towards majority classes which results in poor generalization and unacceptable error rates on minority classes (Japkowicz, 2000; Chawla et al., 2002). Resampling methods are adaptable preprocessing techniques as they are independent of the selected classifier. They fall into three main families with regards to the method used for balancing the class distribution:

Over-sampling methods: aim at balancing the class distribution through the creation of new minority class samples, either by random duplication (Japkowicz, 2000) or synthetic creation. SMOTE (synthetic minority oversampling technique) (Chawla et al., 2002) is a method which artificially generates synthetic instances from minority class by inserting samples with random linear interpolation between a minority sample and its k nearest neighbours. Many approaches based on SMOTE were introduced in the literature (Gao et al., 2012; Zhang and Li, 2014). Gao et al. (Gao et al., 2012) employed a Parzen-window kernel function to estimate the probability density function of the minority class, from which synthetic instances are generated as additional training data to rebalance the class distribution. Zhang et al. (Zhang and Li, 2014) introduced RWO-sampling which is a random walk oversampling method to balance the class distribution by generating synthetic instances via random walks through the training data.

Under-sampling methods: aim at balancing the class distribution through the random elimination of samples from majority classes. Batista et al. (Batista et al., 2000) proposed a more sophisticated undersampling technique by classifying samples from the majority class into three main categories safe, borderline and noise. Only safe majority class instances and all minority class instances are considered for training the classifier.

Hybrid methods: are a combination of over-sampling and under-sampling methods (Peng and Yao, 2010; Cateni et al., 2014). AdaOUBoost (Peng and Yao, 2010) adaptively oversamples minority class instances and undersamples majority class ones to build different classifiers. The classifiers are combined according to their accuracy to create the final prediction. Cateni et al. (Cateni et al., 2014) address the class imbalance problem by combining oversampling and a similarity-based under-sampling techniques.

Solving the class imbalance problem is also possible through algorithmic modifications of existing machine learning classifiers or ensemble constructions. Several works reported in (Haixiang et al., 2017) employ various modifications to existing learning methods, e.g. support vector machines, k-nearest neighbors, or neural networks. Modifications can be introduced by enhancing the discriminatory power of the classifiers using kernel transformation to increase the separability of the original training space (Gao et al., 2016). They can also be formulated by converting the loss functions to penalize errors made in the classification of minority samples stronger (Kim et al., 2016).

Combining classifiers in ensemble frameworks is also another common approach in the class imbalance literature (Galar et al., 2012; Haixiang et al., 2017). Within ensemble-based classifiers, we can distinguish four main families. The first family includes resampling-based ensembles. An ensemble of classifiers is created after training base classifiers on balanced data sets obtained with a resampling technique (Sun et al., 2015; Tian et al., 2011). In the second family, the ensemble is built based on boosting after applying some data resampling strategy. This approach adds a bias toward the minority class to the weight distribution used to train the next classifier at each iteration. Most of the proposed methods inside

this family are based on the first applicable boosting algorithm, Adaboost, proposed by Freund and Schapire (Freund et al., 1996). Many extensions of Adaboost were employed in the context of class imbalance such as Adaboost.M1 and Adaboost.M2 (Freund and Schapire, 1997; Schapire and Singer, 1999). SMOTEBoost (Chawla et al., 2003) is a combination of SMOTE and AdaBoost.M2, where synthetic instances are introduced before the reweighting step in AdaBoost.M2. The synthetic instances have the same weight as the instances in the original dataset, while the weights attributed to the originals are updated according to a pseudo-loss function. RUSBoost (Seiffert et al., 2010) is another example of boosting combined with random sampling and oppositely to SMOTEBoost, it uses random under-sampling to eliminate instances from the majority class in each iteration. Within the third family, we find Bagging-based ensembles, which are known for their simplicity and good generalization ability (Galar et al., 2012). The key factor in these methods is the combination of resampling for class imbalance with the collection of each bootstrap replica in order to obtain a useful classifier in each iteration while maintaining the diversity aspect. Many approaches have been developed using bagging ensembles (Galar et al., 2012), such as UnderBagging (Barandela et al., 2003) and OverBagging (Galar et al., 2012). SMOTEBagging (Wang and Yao, 2009) is also an example of bagging ensembles, where minority class instances are created through a combination of over-sampling and SMOTE and the set of majority class instances is bootstrapped in each iteration in order to form a more diverse ensemble.

3 GENERAL METHODOLOGY

Here, input data is denoted as a multi-set $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{1 \le i \le N}$ which consists of $|\mathcal{D}| = N$ data points $x^{(i)}$ and their corresponding label $y^{(i)}$. We assume that each pair $(x^{(i)}, y^{(i)})$ is an independent realization of a random variable (X, Y) which follows some arbitrary but fixed probability measure $\mathbb{P}$. Each random variable has its own state space, which is denoted by a calligraphic version of the random variable's letter, e.g., $\mathcal{X}$ denotes the state space of X. In this paper, we will use $\mathcal{X} = \mathbb{R}^d$ for some $d \in \mathbb{N}$ and binary class labels, e.g., $\mathcal{Y} = \{0, 1\}$. However, our method works with any state spaces which are supported by the underlying classifiers. Generic realizations of random variables are denoted by lowercase boldface letters, like x. The symbol $\mathbb{1}_{\{\text{expression}\}}$ is the indicator function that evaluates to 1 if and only if the expression is true. We denote empirical estimates by a tilde symbol, e.g., $\tilde{\mathbb{E}}[X]$ is the expected value of X estimated from data, and $\mathbb{E}[X]$ is the true expectation. To simplify notation, we use $\mathbb{P}(Y = y \mid X = x) = \mathbb{P}(y \mid x)$ whenever the corresponding random variables can be inferred from the context.

3.1 Motivation

To understand the intuition behind our approach, suppose we learn a model M from a data set $\mathcal{D}$ to solve a classification task. Choosing $M = \mathbb{P}$ results in the *Bayes optimal classification*: $\hat{y} = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(y \mid x)$, where $\hat{y}$ is the predicted class label. $\mathbb{P}$ is unknown, but we may indeed try to learn it from data, resulting in the estimate $\tilde{\mathbb{P}}(y \mid x) = \tilde{\mathbb{P}}(y,x)/\tilde{\mathbb{P}}(x) \propto \tilde{\mathbb{P}}(y,x)$. However, since our data set $\mathcal{D}$ is finite, there will always be a discrepancy between the true $\mathbb{P}$ and its empirical counterpart $\tilde{\mathbb{P}}$. Due to this deviation from the true distribution our classifier is likely to behave erroneous. If we would know the estimation error $\varepsilon(y,x) = \mathbb{P}(y,x) - \tilde{\mathbb{P}}(y,x)$ for each *possible* data point $(y,x) \in \mathcal{Y} \times \mathcal{X}$, we could correct our model.

3.2 Class Frequency Correction for Binary Classification

Now, assume $\mathcal{Y} = \{0, 1\}$ and consider that we estimate $\tilde{\mathbb{P}}$ from an arbitrary but fixed data set $\mathcal{D}$. Without any prior knowledge about ε, we must distinguish the following three cases (it suffices to consider only one class due to symmetry of the binary classification task): (1) $\mathbb{P}(Y = 1) > \tilde{\mathbb{P}}(Y = 1)$, (2) $\mathbb{P}(Y = 1) \approx \tilde{\mathbb{P}}(Y = 1)$, and (3) $\mathbb{P}(Y = 1) < \tilde{\mathbb{P}}(Y = 1)$. To prepare our classifier for these situations, we construct three new data sets $\mathcal{D}^{>}_{\text{Val}}, \mathcal{D}^{\approx}_{\text{Val}}, \mathcal{D}^{<}_{\text{Val}}$—one for each of the above scenarios. In the first case, the class $y = 1$ is actually more likely than the data suggests, hence $\varepsilon > 0$ and we should create a subsample $\mathcal{D}^{>}_{\text{Val}}$ from $\mathcal{D}$, such that $\tilde{\mathbb{P}}^{>}(Y = 1) = \tilde{\mathbb{P}}(Y = 1) + \varepsilon$, where $\tilde{\mathbb{P}}^{>}$ is the class distribution in $\mathcal{D}^{>}_{\text{Val}}$—we refer to this process as *rebalancing*. The third case is symmetrical: we have $\varepsilon < 0$ and we shall subsample $\mathcal{D}^{<}_{\text{Val}}$ from $\mathcal{D}$. Finally, for the case (2), we subsample the set $\mathcal{D}^{\approx}_{\text{Val}}$ via stratified proportionate allocation—the class distribution will approximately match the class distribution in $\mathcal{D}$ and $\varepsilon \approx 0$.

The general procedure works as follows: Split all data which is available for training into a training set and a validation set. Then, subsample the validation set to generate three specific sets $\mathcal{D}^{>}_{\text{Val}}, \mathcal{D}^{\approx}_{\text{Val}}, \mathcal{D}^{<}_{\text{Val}}$. For each case $\mu \in \{>, \approx, <\}$, we learn a model m^{μ} on the training data and use the data in the validation set $\mathcal{D}^{\mu}_{\text{Val}}$ to refine the learning process. This procedure yields three families of models, namely $M^{>}, M^{\approx}$ and $M^{<}$.

M^μ may contain one or many models m^μ depending on the chosen optimization procedure e.g., repeating the subsampling process multiple times or optimizing different quality metrics q. The resulting models will later be combined to form an ensemble, but first, we shed some light on how to choose the *probability offset* $\alpha = |\varepsilon|$, which is required for our method.

3.3 Class Imbalance and the Probability Offset

At a first glance, choosing the probability offset α might seem infeasible. How should one even guess which $\alpha \in (0;1)$ is appropriate? Surprisingly, it turns out that class imbalance *simplifies* this problem! In fact, the stronger the class imbalance, the tighter is the range for reasonable probability offsets. Let us formalize this result.

Lemma 1 (Probability of Probability Offsets)**.** *Let $\mathcal{D}$ be a data set with empirical class distribution $\tilde{\mathbb{P}}(y)$. We denote the class ratio w.r.t. the minority class by $r = \min\left\{\frac{\mathbb{P}(Y=0)}{\mathbb{P}(Y=1)}, \frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)}\right\}$. There exists c, c' such that for all $\alpha > 0$, the probability of the event*

$$|\tilde{\mathbb{P}}(Y=0) - \mathbb{P}(Y=0)| \geq \alpha$$

is upper bounded by

$$\exp\left(-\frac{\alpha^2 c}{r+\alpha c'} + \ln 2\right) .$$

Proof. For a sequence of independent random variables $Z_1, Z_2, \ldots, Z_N$ with zero-mean and $|Z_i| \leq K$, we know from the Bernstein inequality (cf. Theorem 7.30 in (Foucart and Rauhut, 2013)) that

$$\mathbb{P}\left(\left|\sum_{i=1}^{N} Z_i\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\frac{\varepsilon^2}{2}}{\sum_{i=1}^{N}\mathbb{V}[Z_i] + \frac{K\varepsilon}{3}}\right)$$

Consider the random data set $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{1\leq i\leq N}$. Using the above inequality with $\varepsilon = |\mathcal{D}|\alpha$, $Z_i = (\mathbb{1}_{\{Y^{(i)}=1\}} - \mathbb{E}[\mathbb{1}_{\{Y^{(i)}=1\}}])$ and observing that $\mathbb{E}[\mathbb{1}_{\{Y^{(i)}=1\}}] = \mathbb{P}(Y=1)$ yields

$$\begin{aligned}&\mathbb{P}\left(\left|\tilde{\mathbb{P}}(Y=1) - \mathbb{P}(Y=1)\right| \geq \alpha\right)\\ \leq\ & 2\exp\left(-\frac{(\alpha N)^2/2}{\sum_{i=1}^{N}\mathbb{V}[Z_i] + K(\alpha N)/3}\right) ,\end{aligned}$$

where the inequality on the left hand side is divided by $|\mathcal{D}| = N$, and $\mathbb{V}[Z_i]$ is the variance of Z_i. Since $\mathbb{V}[R+c] = \mathbb{V}[R]$ for any random variable R, we have $\mathbb{V}[Z_i] = \mathbb{V}[\mathbb{1}_{\{Y^{(i)}=1\}}]$. Since $\mathbb{1}_{\{Y^{(i)}=1\}}$ is a Bernoulli random variable, its variance is $\mathbb{P}(Y=0)\mathbb{P}(Y=1)$ and thus $\mathbb{V}[Z_i] = \mathbb{P}(Y=0)\mathbb{P}(Y=1)$. By construction, r is an upperbound on the variance, e.g., $\mathbb{V}[Z_i] < r$. Finally, noticing that $|Z_i| < 1$ proves the lemma with $c = N/2$ and $c' = 1/3$. ■

The lemma tells us that the probability that α is large decreases exponentially fast as a function α, damped by the class imbalance which is measured by r. Let us consider as an example the case where, $\mathbb{P}(Y=1) = 0.9$, i.e., 10% of all data points belong to the minority class. By applying Lemma 1 to this scenario while keeping all other quantities fixed, we see that a deviation of more than 5% happens in at most 0.011285% of all data sets. A deviation of at least 2.5% occurs for at most 14.615% of all data sets. Thus, we see that large α values (say, $> 5\%$) are very unlikely in the presence of class imbalance.

3.4 The Ensemble Framework

The procedure that we described at the beginning of this section yields three diverse types of classifiers $M^>, M^\approx$ and $M^<$, each optimized with respect to a different validation set. The resulting classifiers are combined into an ensemble which is prepared to handle different scenarios of class discrepancy and different quality metrics q. The ensemble weights can be considered as unobserved random variables and apply expectation-maximization to estimate them (Soltanmohammadi et al., 2015). However, such techniques assume the existence of a context and require the classifier accuracy to be Lipschitz continuous. In contrast, we propose to interpret $\mu \in \{>, \approx, <\}$ and q as random variables. Let $m(x) = \hat{y}$ be the prediction of a single model, the outcome of the ensemble is:

$$\mathbb{E}[m(x)] = \tag{1}$$

$$\sum_{\mu\in\{>,\approx,<\}} \mathbb{P}(\mu) \sum_{q\in Q} \mathbb{P}(q \mid \mu) \sum_{t=1}^{T} \mathbb{P}(t \mid \mu, q) m^{\mu}_{q,t}(x) ,$$

where Q is a set of quality metrics, e.g., $Q = \{\text{precision}, \text{recall}\}$, and T is the number of models that we have generated for each combination of μ and q—our experimental results will show that small values of T suffice. The values of $\mathbb{P}(\mu)$, $\mathbb{P}(q \mid \mu)$, and $\mathbb{P}(t \mid \mu, q)$ control how the specific models $m^\mu_{q,t}$ influence the final outcome. The probability $\mathbb{P}(\mu)$ expresses our knowledge on how the class distribution in the training set differs from the true one. According to Lemma 1, all directions are equally likely—our experiments validate this. The probabilities $\mathbb{P}(q \mid \mu)$ can be interpreted as the *importance of the quality measure q* for the actual prediction task. As such, the importance depends on $\mathcal{D}$, Q, or even on μ, and cannot be derived in general. Finally, $\mathbb{P}(t \mid \mu, q)$ are the classical (normalized) ensemble weights. Since each weak model depends on μ, q, and t, the outcome may be interpreted as a mixture of mixtures of experts.

4 EXPERIMENTAL DEMONSTRATION

We conduct an experimental evaluation to assess the behavior of our proposed method. Our experiments are driven by the following questions: (**Q1**) Do the empirical results confirm our theoretical findings about the choice of α to design the class imbalance scenarios? (**Q2**) How does our ensemble construction perform, compared to state-of-the-art methods? To this end, we consider six case studies.

4.1 Case Studies

The first four sets, namely, "Pima", "Yeast", "Glass" and "Haberman" are benchmark data sets, which are publicly available on the UCI repository (Lichman, 2013). The "Welding" and "Milling" data sets describe real-word industrial processes, where resulting manufactured pieces have to be classified into $\{\text{ok}, \text{not-ok}\}$, given a set of processes features.

Table 1: Summary description of the imbalanced data sets used in the experimental study.

Data-set	#Ex.	#Atts.	(min% : maj%)
Pima	768	8	(34.90 : 65.10)
Yeast	1484	8	(10.99 : 89.01)
Glass	214	10	(07.98 : 92.02)
Haberman	306	3	(26.56 : 73.44)
Welding	809	5	(09.00 : 91.00)
Milling	5700	4	(11.00 : 89.00)

Table 1 summarizes the properties of the selected data-sets, namely: number of examples (#Ex.), number of attributes (#Atts.), class name (minority and majority Class(min;maj)), the percentage of examples of each class.

4.2 Experimental Setup

All experiments are cross-validated and conducted using the `R` software. The data sets of the six case studies are split into a training, a validation and a test set for each cross validation fold. 60 % of the data was used for training, while the remaining 40% are equally split between validation and testing. The baseline classifier in this work is Decision Tree (DT) (Breiman et al., 1984). Our framework is compared to three baseline models, namely Decision Tree (Breiman et al., 1984), Random Forest (RF) (Breiman, 2001), and Gradient Boosting Machine (GBM) (Friedman, 2001). The parameters of both GBM and RF are tuned using grid search procedures over a set of input parameters. In addition to the plain baseline methods, our method is also compared to various state-of-the-art methods for handling class imbalance. The name of the methods and the corresponding literature references are detailed in Table 2. An additional ensemble of DT combined with random undersampling is also added for comparison and denoted by "Ens". The number of trees in the ensemble is set to 6 (= maximum number of trees used in our proposed method).

Our Method. Three validation sets $\mathcal{D}^{>}_{\text{Val}}, \mathcal{D}^{\approx}_{\text{Val}}, \mathcal{D}^{<}_{\text{Val}}$ are designed from the validation set to represent each of the three possible scenarios of class imbalance. We investigate two sets of quality metrics Q: $\{\text{precision}, \text{recall}\}$ and $\{\text{F1-score}\}$. For simplicity, the number of trees T per quality measure is set to 1 in these experiments which implies $\mathbb{P}(t \mid \mu, q) = 1$. So for each scenario, we have one tree for each component of Q. For example, for the first setting of Q: $\{\text{precision}, \text{recall}\}$, we have two trees one aiming to maximize the recall and the other aiming at maximizing the precision. In the first setting , $\mathbb{P}(\text{recall} \mid \mu)$ is set to $\tilde{\mathbb{P}}(Y = 1)$ with $\mathbb{P}(\text{precision} \mid \mu) = 1 - \mathbb{P}(\text{recall} \mid \mu)$. The rationale behind this choice is to interpret the frequency of the minority class as a proxy for the importance of the recall, where we estimate the frequencies on the training data. Since in a realistic application of our method, each of the three cases in $\{>, \approx, <\}$ is equally likely, we set $\mathbb{P}(\mu) = 1/3$, which is a generic choice in the absence of any prior knowledge about class proportion in the test set. Predictions are performed by thresholding the output of the ensemble (Eq. 1) at $1/2$.

4.3 Results

Table 2 encloses descriptive statistics of the experimental results on the F1-score for both majority and minority classes per method/case study.

4.3.1 Impact of α

Experiments on varying α for the six data sets confirms our theoretical findings in Lemma 1, which indicates that large α values are unlikely. In fact, the maximum alpha value depends on the particular data set. E.g., on the glass data, the frequency of the minority class is 7.98%, thus, we cannot set α above this value. In contrast, the frequency of the minority class of the Pima data is 34.9%, which implies that we can drive α up to 30% without problems. On "Yeast", "Haberman", and "Milling", the AUC is almost constant w.r.t. to α. On "Welding", we can observe a clear decrease in AUC for an increasing α. On

Table 2: Experimental Results for our six case studies. Abbreviations: **DT** = decision tree (Breiman et al., 1984), **GBM** = gradient boosting machine (Friedman, 2001), **RF** = random forest (Breiman, 2001), **DT/GBM/RF+U** = DT/GBM/RF+under-sampling (Japkowicz, 2000), **DT/GBM/RF+O** = DT/GBM/RF+over-sampling (Japkowicz, 2000) **DT/GBM/RF+S** = DT/GBM/RF+SMOTE (Chawla et al., 2002), **DT/RF+UBA** = DT/RF+UnderBagging (Barandela et al., 2003), **DT/RF+AD** = DT/RF+Adaboost.m2 (Schapire and Singer, 1999), **DT/RF+SBA** = DT/RF+SMOTEBagging (Wang and Yao, 2009), **DT/RF+SBO** = DT/RF+SMOTEBoost (Chawla et al., 2003), **DT/RF+RUS** = DT/RF+RUSBoost (Seiffert et al., 2010), **PRO1** = our Ensemble using precision and recall, **PRO2** = our Ensemble using F1-score, Min. = F1 for minority class, Maj. = F1 for majority class.

Method	Welding		Pima		Glass		Haberman		Milling		Yeast	
	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.	Min.	Maj.
DT	20.34 ±13.43	93.18 ±1.9	54.97 ±6.26	81.19 ±3.91	6.67 ±11.55	93.52 ±2.36	29.47 ±15.95	80.51 ±6.76	18.94 ±10.32	87.69 ±3.01	67.59 ±11.64	97.1 ±0.56
GBM	11.57 ±4.02	94.98 ±0.59	59.22 ±3.3	83.09 ±3.6	0.0 ±0.0	93.97 ±1.71	32.96 ±13.97	81.53 ±2.58	12.78 ±12.51	90.42 ±3.57	56.15 ±18.43	96.78 ±0.99
RF	12.83 ±9.95	**95.53** ±**0.87**	62.4 ±2.21	**85.02** ±**1.79**	6.67 ±11.55	93.89 ±2.47	21.65 ±19.89	77.85 ±9.82	5.13 ±8.88	90.85 ±2.02	67.46 ±11.05	97.58 ±0.29
DT+U	24.17 ±12.51	88.43 ±6.22	56.81 ±2.42	76.77 ±2.76	13.47 ±11.84	89.55 ±6.46	28.59 ±17.9	71.62 ±10.95	26.86 ±4.32	83.75 ±7.75	63.66 ±23.14	94.9 ±3.37
GBM+U	24.6 ±7.73	93.69 ±1.84	62.6 ±1.86	80.56 ±3.07	6.06 ±10.5	92.37 ±2.15	37.33 ±5.33	74.26 ±6.6	18.91 ±6.27	81.51 ±6.33	72.57 ±8.89	97.06 ±0.88
RF+U	23.29 ±10.33	94.61 ±1.94	62.96 ±4.87	79.91 ±2.82	5.56 ±9.62	91.53 ±3.8	28.38 ±24.6	78.45 ±9.81	26.19 ±2.06	85.65 ±5.13	71.33 ±20.06	96.65 ±2.4
DT+O	18.72 ±8.94	90.02 ±1.66	59.6 ±5.09	72.32 ±3.98	14.07 ±12.24	93.08 ±2.45	35.47 ±12.99	72.27 ±2.63	22.33 ±11.54	74.55 ±2.65	65.54 ±18.32	95.93 ±1.95
GBM+O	15.43 ±11.22	94.75 ±0.66	59.04 ±2.96	80.28 ±3.19	0.0 ±0.0	93.22 ±1.75	35.81 ±13.26	78.37 ±2.14	23.02 ±3.44	79.62 ±1.82	64.23 ±10.46	96.65 ±1.03
RF+O	13.69 ±6.03	94.83 ±1.08	64.77 ±4.59	82.82 ±2.25	0.0 ±0.0	93.97 ±1.71	22.8 ±20.35	**82.88** ±**1.89**	31.11 ±10.18	85.52 ±3.21	66.72 ±17.23	97.15 ±0.78
DT+S	19.27 ±3.74	87.86 ±1.37	36.07 ±5.38	81.78 ±2.02	18.14 ±3.31	0.0 ±0.0	34.02 ±1.4	56.45 ±20.48	21.67 ±20.21	85.15 ±3.38	77.01 ±7.38	97.48 ±0.65
GBM+S	11.85 ±6.29	94.77 ±1.36	57.53 ±5.04	82.41 ±0.62	13.01 ±12.28	60.58 ±28.81	40.49 ±9.53	49.84 ±25.36	6.67 ±11.55	90.51 ±1.26	61.63 ±21.63	96.8 ±1.35
RF+S	0.0 ±0.0	95.22 ±0.83	57.33 ±1.32	81.57 ±5.51	21.9 ±7.19	57.8 ±9.21	24.2 ±15.89	68.55 ±10.76	8.33 ±14.43	**90.87** ±**2.91**	73.0 ±13.48	97.53 ±0.62
Ens	14.45 ±1.84	8.0 ±2.88	21.23 ±36.78	81.22 ±2.66	0.0 ±0.0	93.97 ±1.71	43.89 ±4.85	80.47 ±5.2	14.56 ±14.29	45.07 ±36.02	71.05 ±13.6	96.85 ±0.67
DT+UBA	25.02 ±4.46	91.87 ±2.28	**68.32** ±**3.58**	82.31 ±2.32	0.0 ±0.0	94.73 ±1.11	42.21 ±9.33	69.27 ±6.4	26.57 ±5.47	79.28 ±4.87	57.95 ±19.71	93.46 ±2.38
DT+AD	7.39 ±9.22	94.79 ±0.93	64.07 ±0.28	84.64 ±1.36	7.41 ±12.83	93.17 ±1.36	40.61 ±9.2	79.62 ±6.9	14.7 ±15.43	87.02 ±3.43	65.62 ±10.21	97.01 ±0.66
DT+SBA	22.11 ±2.54	71.87 ±5.22	65.28 ±3.61	84.3 ±1.5	29.76 ±15.02	90.86 ±1.59	42.05 ±7.2	77.87 ±5.95	29.67 ±6.98	58.0 ±3.82	77.6 ±6.49	97.6 ±0.53
DT+RUS	15.17 ±10.5	93.67 ±1.48	63.25 ±2.76	78.8 ±1.99	7.14 ±12.37	63.52 ±55.02	38.76 ±10.58	57.84 ±29.71	22.65 ±12.99	83.34 ±1.35	64.5 ±14.72	95.2 ±2.25
DT+SBO	16.47 ±2.49	0.0 ±0.0	62.23 ±3.3	74.8 ±1.39	14.39 ±12.92	89.47 ±4.23	42.39 ±8.75	77.56 ±5.83	22.65 ±12.99	83.34 ±1.35	68.57 ±11.06	97.09 ±0.57
RF+UBA	12.63 ±18.91	95.47 ±1.1	65.2 ±4.6	78.65 ±1.67	23.06 ±4.12	59.93 ±17.86	41.01 ±6.66	61.44 ±24.54	23.53 ±11.76	86.3 ±2.12	60.41 ±20.0	93.9 ±2.74
RF+AD	5.9 ±6.84	95.41 ±0.93	61.62 ±0.8	84.36 ±1.89	8.33 ±14.43	**94.31** ±**1.32**	32.8 ±18.72	80.71 ±4.31	14.7 ±15.43	87.02 ±3.43	69.37 ±20.63	96.41 ±1.58
RF+SBA	24.62 ±4.67	76.19 ±4.18	62.75 ±4.03	84.64 ±1.06	13.46 ±12.61	91.48 ±3.58	27.51 ±24.04	81.02 ±5.74	28.33 ±7.11	49.95 ±14.79	71.59 ±14.17	97.3 ±1.04
RF+RUS	8.66 ±11.21	95.41 ±1.14	67.27 ±3.55	81.41 ±0.76	21.8 ±3.43	83.04 ±6.79	40.25 ±7.21	64.79 ±18.8	16.73 ±14.58	86.54 ±1.66	64.01 ±17.16	95.23 ±1.65
RF+SBO	16.47 ±2.49	0.0 ±0.0	63.15 ±6.92	78.23 ±3.84	21.67 ±20.21	90.59 ±2.51	35.5 ±19.59	80.2 ±5.67	14.86 ±12.93	84.86 ±1.73	69.91 ±12.65	97.21 ±0.98
PRO1	**38.65** ±**4.09**	94.17 ±1.17	67.82 ±2.07	83.55 ±1.99	**26.19** ±**2.06**	93.86 ±0.66	**46.32** ±**4.17**	78.79 ±6.43	**42.21** ±**14.73**	87.92 ±6.56	**83.37** ±**10.0**	**98.5** ±**0.71**
PRO2	**37.84** ±**5.77**	94.18 ±1.19	67.93 ±1.35	84.08 ±3.54	**26.46** ±**3.67**	93.85 ±1.34	**47.47** ±**6.14**	78.89 ±5.9	**42.77** ±**14.46**	86.81 ±8.47	**82.37** ±**18.89**	**98.61** ±**0.96**

"Glass" and "Pima", changing α induces some jitter on the AUC. However, there is neither an increasing nor a decreasing trend—all results have the same order of magnitude. We hence conclude that small α values suffice and that our method is not overly sensitive w.r.t. specific choices of α. Hence, the answer to question **Q1** is in the affirmative, and fix $\alpha = 3\%$ for the other experiments, i.e., no optimization over α is performed.

4.3.2 Classification Performance

The results of our large comparison of classification methods is shown in Table 2. In most real-world applications, the F1-score of the minority class ("Min.") is of exceptional importance. Not surprisingly, in almost all cases, the baseline methods DT, RF and GBM are outperformed by methods for imbalanced data. There is no clear winner among the state-of-the-art undersampling, oversampling, or SMOTE techniques. It depends on the particular data set which strategy delivers the best F1-score for the minority class. It is thus especially remarkable that our method achieves the best and second best F_1-score—depending on the choice of quality measures Q—for the minority class on five of six data sets. Only decision tree UnderBagging achieves a better minority F1-score on the "Pima" data. Nevertheless, note that our proposed methods exhibits a lower standard error. Thus, a "typical" run of our method might outperform DT+UBA. In terms of majority ("Maj.") F1-score, our method is outperformed in 5 of 6 cases. However, in two cases, the best result is delivered by a plain random forest without any technique to handle class imbalance—the corresponding minority F1-score is thus far from optimal. The other three cases are led by RF + oversampling, RF + SMOTE, and RF + Adaboost—all three are variants of the random forest. However, the decline majority F1-score of our proposed method is in four of five cases below 3 percent. Assuming that F1-score on the minority class is the most important measure, we answer **Q2** in the affirmative.

5 CONCLUSION

In this paper, we presented a new ensemble method for classification in the presence of imbalanced classes. We started by reviewing the state-of-the-art in the area of classification with imbalanced classes. Real data sets are always finite which leads to a corruption of empirical class frequencies. Moreover, we proved that these corruptions are unlikely to be large in case of class imbalance and proposed a method to correct for small corruptions. The correction is performed by generating specialized validation sets which correspond to different scenarios. Each validation set may then be used to induce an ensemble of classifiers. We discussed how the classifiers for different scenarios can be combined into an ensemble and proposed different choices for the ensemble weights. In an experimental demonstration, we validated our theoretical findings, and showed that our method outperforms several state-of-the-art methods in terms of F_1-score. Since our insights about class imbalance and erroneous empirical class frequencies are completely new, our work may serve as the basis for multiple new research directions.

REFERENCES

Barandela, R., Valdovinos, R., and Sánchez, J. (2003). New applications of ensembles of classifiers. *Pattern Analysis & Applications*, 6(3):245–256.

Batista, G. E., Carvalho, A. C., and Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In *Proc. of MICAI*, pages 315–325. Springer.

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

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

Cateni, S., Colla, V., and Vannucci, M. (2014). A method for resampling imbalanced datasets in binary classification tasks for real-world problems. *Neurocomputing*, 135:32 – 41.

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

Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. In *Proc. of the 7th PKDD*, pages 107–119.

Foucart, S. and Rauhut, H. (2013). *A Mathematical Introduction to Compressive Sensing*. Springer New York.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Proc. of ICML 1996*, volume 96, pages 148–156.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Sys-*

tems, Man, and Cybernetics, Part C (Applications and Reviews), 42(4):463–484.

Gao, M., Hong, X., Chen, S., and Harris, C. J. (2012). Probability density function estimation based oversampling for imbalanced two-class problems. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE.

Gao, X., Chen, Z., Tang, S., Zhang, Y., and Li, J. (2016). Adaptive weighted imbalance learning with application to abnormal activity recognition. *Neurocomputing*, 173:1927–1935.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.

Japkowicz, N. (2000). The class imbalance problem: Significance and strategies. In *Proc. of the Int. Conf. on Artificial Intelligence*.

Kim, S., Kim, H., and Namkoong, Y. (2016). Ordinal classification of imbalanced data with application in emergency and disaster information services. *IEEE Intelligent Systems*, 31(5):50–56.

Lichman, M. (2013). UCI machine learning repository.

Peng, Y. and Yao, J. (2010). Adaouboost: adaptive oversampling and under-sampling to boost the concept learning in large scale imbalanced data sets. In *Proceedings of the international conference on Multimedia information retrieval*, pages 111–118. ACM.

Schapire, R. E. and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., and Napolitano, A. (2010). Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197.

Soltanmohammadi, E., Naraghi-Pour, M., and van der Schaar, M. (2015). Context-based unsupervised data fusion for decision making. In *Proc. of the 32nd ICML*, pages 2076–2084.

Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., and Zhou, Y. (2015). A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, 48(5):1623–1637.

Tian, J., Gu, H., and Liu, W. (2011). Imbalanced classification using support vector machine ensemble. *Neural Computing and Applications*, 20(2):203–209.

Wang, S. and Yao, X. (2009). Diversity analysis on imbalanced data sets by using ensemble models. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 324–331. IEEE.

Zhang, H. and Li, M. (2014). Rwo-sampling: A random walk over-sampling approach to imbalanced data classification. *Information Fusion*, 20:99–116.

Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Machine Learning & Pattern Recognition. CRC Press.