Received: 12 August 2020 | Revised: 1 June 2021 | Accepted: 30 June 2021 | Published on: 2 August 2021

DOI: 10.1002/sam.11538

WILEY

RESEARCH ARTICLE

Imbalanced classification: A paradigm-based review

Yang Feng[1] | Min Zhou[2] | Xin Tong[3]

[1] Department of Biostatistics, School of Global Public Health, New York University, New York, New York USA

[2] Division of Science and Technology, Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, China

[3] Department of Data Sciences and Operations, Marshall School of Business, University of Southern California, Los Angeles, California USA

Correspondence

Xin Tong, Department of Data Sciences and Operations, Marshall School of Business, University of Southern California, Los Angeles, CA 90089, USA.
Email: xint@marshall.usc.edu

Funding information

National Science Foundation, Grant/Award Number: DMS-1554804; US National Institutes of Health, Grant/Award Number: R01 GM120507

Abstract

A common issue for classification in scientific research and industry is the existence of imbalanced classes. When sample sizes of different classes are imbalanced in training data, naively implementing a classification method often leads to unsatisfactory prediction results on test data. Multiple resampling techniques have been proposed to address the class imbalance issues. Yet, there is no general guidance on when to use each technique. In this article, we provide a paradigm-based review of the common resampling techniques for binary classification under imbalanced class sizes. The paradigms we consider include the classical paradigm that minimizes the overall classification error, the cost-sensitive learning paradigm that minimizes a cost-adjusted weighted type I and type II errors, and the Neyman–Pearson paradigm that minimizes the type II error subject to a type I error constraint. Under each paradigm, we investigate the combination of the resampling techniques and a few state-of-the-art classification methods. For each pair of resampling techniques and classification methods, we use simulation studies and a real dataset on credit card fraud to study the performance under different evaluation metrics. From these extensive numerical experiments, we demonstrate under each classification paradigm, the complex dynamics among resampling techniques, base classification methods, evaluation metrics, and imbalance ratios. We also summarize a few takeaway messages regarding the choices of resampling techniques and base classification methods, which could be helpful for practitioners.

KEYWORDS

binary classification, classical classification (CC) paradigm, cost-sensitive (CS) learning paradigm, imbalance ratio, imbalanced data, Neyman–Pearson (NP) paradigm, resampling methods

1 | INTRODUCTION

Classification is a widely studied type of supervised learning problem with extensive applications. A myriad of classification methods (e.g., logistic regression [LR], support vector machines [SVMs], random forest [RF], neural networks [NNs], boosting), which we refer to as the *base classification methods* in this paper, have been developed to deal with different distributions of data [32]. However, in the case where the classes are of different sizes (i.e., the imbalanced classification scenario), naively applying the existing methods could lead to undesirable results. Some prominent applications include defect detection [4], medical diagnosis [16], fraud detection [66], spam email

Yang Feng and Min Zhou contributed equally to this work.

filtering [68], text categorization [71], oil spills detection in satellite radar images [33], and land use classification [49]. To address the class size imbalance scenario, there has been extensive research on developing different methods [58, 40, 25]. Some popular tools include resampling techniques [1, 3, 40], direct methods [37, 38, 48, 57, 72], postprocessing methods [11], as well as different combinations of these tools. The most common and understandable class of approaches is resampling techniques. However, there lacks a consensus about when and how to use them.

In this work, we aim to provide some guidelines on using resampling techniques for imbalanced binary classification. We first disentangle the general claims of undesirability in classification results under imbalanced classes, via listing a few common paradigms and evaluation metrics. To decide which resampling technique to use, we need to be clear on the paradigms as well as the preferred evaluation metrics. Sometimes, the chosen paradigm and the evaluation metric are not compatible, which makes the problem unsolvable by any technique. When they are, we will show that the optimal resampling technique depends on both the paradigm and the base classification method.

There are different degrees of data imbalance. We characterize this degree by the *imbalance ratio* (IR) [23], which is the ratio of the sample size of the majority class and that of the minority class. In real applications, IR can range from 1 to more than 1000. For instance, a rare disease occurs only in 0.1% of the human population [5]. We will show that different IRs might demand different combinations of resampling techniques and base classification methods.

This review conducts extensive simulation experiments as well as a real dataset on credit card fraud to concretely illustrate the dynamics among data distributions, IR, base classification methods, and resampling techniques. This is the *first* time that such dynamics are explicitly examined. To the best of our knowledge, this is also the first time that a review paper uses running simulation examples to demonstrate the advantages and disadvantages of the reviewed methods. Through simulation and real data analysis, we give practitioners a look into the complicated nature of the imbalanced data problem in classification, even if we narrow our search to the resampling techniques only. For important applications where data distributions can be approximately simulated, practitioners are encouraged to mimic our simulation studies and properly evaluate the combinations of resampling techniques and base classification methods. In the end, we summarize a few takeaway messages regarding the choices of resampling techniques and base classification methods, which could be helpful for practitioners.

The rest of the review is organized as follows. In Section 2, we describe three classification paradigms and discuss their corresponding objectives. Then, we introduce a matrix of classification algorithms as pairs of resampling techniques and the base classification methods in Section 3. Section 4 provides a list of commonly used evaluation metrics for imbalanced classification. In Sections 5 and 6, we conduct a systematic simulation study and a real data analysis to evaluate the performance of different combinations of resampling techniques and base classification methods, under different paradigms, data distributions, and IRs, in terms of various evaluation metrics. We conclude the review with a short discussion in Section 7.

2 | THREE CLASSIFICATION PARADIGMS

In this section, we review three classification paradigms that are defined by different objective functions. Concretely, we consider the classical classification (CC) paradigm that minimizes the overall classification error (Section 2.1), the cost-sensitive (CS) learning paradigm that minimizes the cost-adjusted weighted type I and type II errors (Section 2.2), and the Neyman–Pearson (NP) paradigm that minimizes the type II error subject to a type I error constraint (Section 2.3).

Assume $X \in \mathcal{X} \subset \boldsymbol{R}^d$ is a random vector of d features, and $Y \in \{0, 1\}$ is the class label. Let $\mathbb{P}(Y = 0) = \pi_0$ and $\mathbb{P}(Y = 1) = \pi_1 = 1 - \pi_0$. Throughout the article, we label the minority class as 0 and the majority class as 1 (i.e., $\pi_0 \leq \pi_1$). Also, for language consistency, we call class 0 the negative class and class 1 the positive class. Please note that the minority class might be referred to as "positive" in medical applications.

2.1 | CC paradigm

A classifier is defined as $\phi : \mathcal{X} \to \{0, 1\}$, which is a mapping from the feature space to the label space. The overall classification error (risk) is naturally defined as $R(\phi) = \mathbb{E}[\mathbb{I}(\phi(X) \neq Y)] = \mathbb{P}(\phi(X) \neq Y)$, where $\mathbb{I}(\cdot)$ is the indicator function. In binary classification, most existing classification methods focus on the minimization of the overall classification error (risk) [27, 30]. In this article, this paradigm is referred to as *CC paradigm*. Under this paradigm, the *CC oracle* ϕ^* is a classifier that minimizes the population risk; that is,

$$\phi^* = \arg\min_{\phi} R(\phi).$$

It is well known that $\phi^* = \mathbb{I}(\eta(x) > 1/2)$, where $\eta(x) = \mathbb{E}(Y|X = x)$ is the regression function [31]. In practice, we construct a classifier $\widehat{\phi}$ based on finite sample $\{(X_i, Y_i), \ i = 1, \ldots, n\}$ using some classification method.

Popular the CC paradigm is, it may not be the ideal choice when the class sizes are imbalanced. By the *law of total probability*, we decompose the overall classification error as a weighted sum of type I and II errors, that is,

$$R(\phi) = \pi_0 R_0(\phi) + \pi_1 R_1(\phi),$$

where $R_0(\phi) = \mathbb{P}(\phi(X) \neq Y | Y = 0)$ denotes the (population) type I error (the conditional probability of misclassifying a class 0 observation as class 1); and $R_1(\phi) = \mathbb{P}(\phi(X) \neq Y | Y = 1)$ denotes the (population) type II error (the conditional probability of misclassifying a class 1 observation as class 0). However, in many practical applications, we may want to treat type I and II errors differently under two common scenarios. One is the *asymmetric error importance* scenario. In this scenario, making one type of error (e.g., type I error) is more serious than making the other type of error (e.g., type II error). For instance, in severe disease diagnosis, misclassifying a diseased patient as healthy could lead to missing the optimal treatment window while misclassifying a healthy patient as diseased can lead to patient anxiety and incur additional medical costs. The other is the *imbalanced class proportion* scenario. Under this scenario, π_0 is much smaller than π_1, and minimizing the overall classification error could sometimes result in a larger type I error. For applications that fit these two scenarios, the overall classification error may not be the optimal choice to serve the users' purpose, either as an optimization criterion or as an evaluation metric. Next, we will introduce two other paradigms that have been used the address the asymmetric error importance and imbalanced class proportion issues.

2.2 | CS learning paradigm

In the asymmetric error importance and imbalanced class proportion scenarios introduced at the end of Section 2.1, the cost of type I error is usually higher than that of type II error. For example, in spam email filtering, the cost of misclassifying a regular email as spam is much higher than the cost of misclassifying spam as a regular email. A popular approach to incorporate different costs for these two types of errors is the *CS learning* paradigm [21, 69]. Let $C(\phi(X), Y)$ being the cost function for classifier ϕ at observation pair (X, Y). Let $C_0 = C(1, 0)$ and $C_1 = C(0, 1)$ being the costs of type I and II errors, respectively. For the correct classification result, we have $C(0, 0) = C(1, 1) = 0$. Then, CS learning minimizes the expected misclassification cost [34]:

$$\begin{aligned} R_c(\phi) &= \mathbb{E}C(\phi(X), Y) \\ &= C_0\mathbb{P}(\phi(X) = 1, Y = 0) + C_1\mathbb{P}(\phi(X) = 0, Y = 1) \\ &= C_0\mathbb{P}(\phi(X) = 1 | Y = 0)\mathbb{P}(Y = 0) + C_1\mathbb{P}(\phi(X) \\ &= 0 | Y = 1)\mathbb{P}(Y = 1) \\ &= C_0\pi_0 R_0(\phi) + C_1\pi_1 R_1(\phi). \end{aligned}$$

There are primarily two types of approaches in the literature on CS learning. The first type is called *direct methods*, which builds a CS learning classifier by incorporating the different misclassification costs into the training process of the base classification method. For instance, there has been much work on CS decision tree (DT) [6, 38, 63], CS boosting [39, 57, 65], CS SVM [47], and CS NN [72]. The second type is usually referred to as *postprocessing methods*, in such a way that we adjust the decision threshold with the base classification algorithm unmodified. An example of this can be found in [20]. Some additional references on CS learning include [25, 40, 41, 64, 70, 73].

In this review, we focus on the *postprocessing methods* as it combines well with any existing base classification algorithm without the need to change its internal mechanism, which is also better understood among practitioners. In addition, it serves the purpose of making an informative comparison among different learning paradigms across different classification methods. On the population level, with the knowledge of C_0 and C_1, the *CS oracle* is

$$\phi_c^* = \arg\min_{\phi} R_c(\phi) = \mathbb{I}\left(\eta(x) > \frac{C_0}{C_0 + C_1}\right),$$

which reduces to the CC oracle ϕ^* when $C_0 = C_1$.

Although CS learning has its merits on the control of asymmetric errors, its drawback is also apparent because it is sometimes difficult or immoral to assign the values of costs C_0 and C_1. In most applications, including the severe disease classification, these costs are unknown and cannot be easily provided by experts. One way to extricate from this dilemma is to set the majority class misclassification cost $C_1 = 1$ and the minority class misclassification cost $C_0 = \pi_1/\pi_0$ [11].

2.3 | NP paradigm

Besides requiring the knowledge of costs for different misclassification errors, the CS learning paradigm does not provide an explicit probabilistic control on type I error under a pre-specified level. Even if the practitioner tunes the empirical type I error equal to the pre-specified level,

TABLE 1 Three types of classification paradigms in binary classification

Paradigm	Objective	Oracle classifier
Classical	Minimize the overall classification error	$\phi^* = \arg\min_{\phi} R(\phi)$
Cost-sensitive	Minimize the expected misclassification cost	$\phi_c^* = \arg\min_{\phi} R_c(\phi)$
Neyman–Pearson	Minimize type II error while controlling type I error under α	$\phi_\alpha^* = \arg\min_{\phi:R_0(\phi)\leq\alpha} R_1(\phi)$

the population-level type I error still has a non-trivial chance of exceeding this level [61, 62]. To deal with this issue, another emerging statistical framework to control asymmetric error is called *NP paradigm* [9, 50, 60–62], which aims to minimize type II error $R_1(\phi)$ while controlling type I error $R_0(\phi)$ under a desirable level. The corresponding *NP oracle* is

$$\phi_\alpha^* = \arg\min_{\phi:R_0(\phi)\leq\alpha} R_1(\phi),$$

where α is a targeted upper bound for type I error. It can be shown that $\phi_\alpha^*(\cdot) = \mathbb{I}(\eta(\cdot) > D_\alpha^*)$ for some properly chosen D_α^*. Unlike $1/2$ or $C_0/(C_0 + C_1)$, D_α^* is not known unless one has access to the distribution information. Tong et al. [61] proposed an umbrella algorithm for NP classification, which adapts existing scoring-type classification methods (e.g., LR, SVMs, RF) by choosing an order-statistics based thresholding level so that the resulting classifier has type I error bounded from above by α with high probability. This thresholding mechanism, along with the thresholds $1/2$ and $C_0/(C_0 + C_1)$ for CC and CS paradigms respectively, will be systematically studied in combination with several state-of-the-art base classification methods in numerical studies.

2.4 | A summary of three classification paradigms

For readers' convenience, we summarize the three classification paradigms with their corresponding objectives and oracle classifiers in Table 1.

3 | A MATRIX OF ALGORITHMS FOR IMBALANCED CLASSIFICATION

In this section, we introduce a matrix of algorithms for imbalanced classification, which consists of combinations of resampling techniques and base classification methods.

To fix idea, assume among the n observation pairs $\{(X_i, Y_i),\ i = 1, \dots, n\}$, there are n_0 observations with $Y_i = 0$ (the minority class) and n_1 observations with $Y_i = 1$ (the majority class). Then, the IR $= n_1/n_0$.

3.1 | Resampling techniques

To address the imbalanced classification problem under one of the three classification paradigms described in Section 2, resampling techniques are often used to create a new training dataset by balancing the number of data points in the minority and majority classes in order to alleviate the effect of class size imbalance in the process of classification. López et al. [40] pointed out that about one third of their reviewed papers have used resampling techniques. They are usually divided into three categories: undersampling, oversampling, and hybrid methods.

The undersampling methods directly discard a subset of observations of the majority class. It includes two main versions: the *cluster-based undersampling* and *random undersampling* [25, 35, 59, 67]. In the *cluster-based undersampling*, a clustering algorithm is applied to cluster the majority class such that the number of clusters is equal to that of the data points in the minority class (i.e., n_0 clusters), and then one point is randomly selected from each cluster. Nevertheless, the clustering process could be quite slow when n_1 is large. *Random undersampling* is a simpler and more efficient approach, which randomly eliminates the data points from the majority class to make it of size n_0. By undersampling, the processed training dataset is a combination of n_0 randomly chosen data points from the majority class and all (n_0) data points from the minority class. However, undersampling may lead to loss of information as a large portion of the data from the majority class is discarded.

The oversampling method, on the other hand, increases the number of data points in the minority class from n_0 to n_1 while keeping the observations from the majority class intact. The leading two approaches are *random oversampling* and *SMOTE* [5, 22, 26, 28, 44]. *Random oversampling*, as a counterpart of *random undersampling*, is perhaps the most straightforward approach to duplicate the data points of minority class randomly. One version of the approach samples $n_1 - n_0$ observations with replacement from the minority class and add them to the new training set. The approach *SMOTE* is the acronym for the "synthetic minority over-sampling technique" proposed by Chawla et al. [13]. It generates $n_1 - n_0$ new synthetic data points for the minority class by interpolating pairs of k-nearest neighbors (KNNs). We review

Algorithm 1. SMOTE [13]

```
For any data point of minority class
 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})^\top$, the multiple $N = \mathrm{IR} - 1$,
 number of nearest neighbors $K$
Step 1: Find the $K$ nearest neighbor points of $X_i$ in
        the minority class: $X_{i_1}, \dots, X_{i_k}$.
Step 2:
  for do $j = 1 : N$
     randomly choose one of the $K$ nearest
     neighbor points: $X_{i_j} = (X_{i_j 1}, X_{i_j 2}, \dots, X_{i_j d})^\top$.
     generate a random number $r_s \sim Unif[0, 1]$.
     generate the synthetic data point for the
     minority class as $X_j^* = (X_{j1}^*, X_{j2}^*, \dots, X_{jd}^*)^\top$,
     where $X_{js}^* = X_{is} + r_s * (X_{i_j s} - X_{is})$,
     $s = 1, \dots, d$.
  end for
Return $X_1^*, \dots, X_N^*$ as new synthetic data points.
```

the details of SMOTE in Algorithm 1. Compared with undersampling, oversampling methods usually require longer training time and could cause over-fitting. A popular extension of SMOTE is the borderline-SMOTE (BLSMOTE) [26], which only oversamples the minority observations near the borderline and the essential step to generate the data point is similar to the SMOTE algorithm in Algorithm 1 (see [26] for a detailed description for BLSMOTE).

The hybrid method is just a combination of undersampling and oversampling methods [10, 12, 19, 53]. It simultaneously decreases the number of data points from the majority class and increases the number of data points from the minority class to n_h, where the above-described undersampling and oversampling methods can be used. The hybrid method could serve as an option that balances the goodness of fit, computational cost as well as robustness of the classifier.

3.2 | Classification methods

Using any of the resampling methods, we will arrive at a new training dataset that has balanced classes. Naturally, we can apply any existing base classification method on this new dataset coupled with one of the paradigms described in Section 2.

Many classification methods have been extensively studied. The well-known ones include DT [54], KNNs [2], linear discriminant analysis [42], LR [45], naïve Bayes [51], NN [52], RF [8], SVM [17], and XGBoost (XGB) [14], among others.

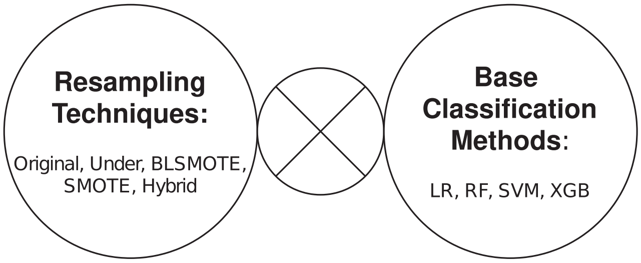
Resampling Techniques:
Original, Under, BLSMOTE, SMOTE, Hybrid
Base Classification Methods:
LR, RF, SVM, XGB

FIGURE 1 A summary of the matrix of algorithms

To learn more about these methods, we refer the readers to a review of classification methods [32] and a book on statistical learning [27].

3.3 | A summary of the matrix of algorithms

In numerical studies, we consider a matrix of classification algorithms shown in Figure 1, as combinations of resampling techniques described in Section 3.1 and four (out of many) state-of-the-art classification methods described in Section 3.2.

In Figure 1, "Original" refers to *no resampling*, "Under" refers to *random undersampling*, and "Hybrid" refers to *a hybrid of random undersampling and SMOTE*. Note here we chose random undersampling, SMOTE, and BLSMOTE as representatives of undersampling and oversampling methods due to their popularity among practitioners. The readers can easily study other types of resampling technique and classification method combinations by adapting the companion code from this review.

In the numerical studies, we will conduct a comparative study on those 20 combinations described in Figure 1 under each of the three paradigms introduced in Section 2 with the IR varying from 1 to 128, in terms of different evaluation metrics which will be introduced in the next section. A flowchart demonstrating our imbalanced classification system can be found in Figure 2.

4 | EVALUATION METRICS

In this section, we will review several popular evaluation metrics to compare the performance of different classification algorithms.

For a given classifier, suppose that it classifies the ith observation X_i to $\widehat{Y}_i$ (Y_i denotes the true label). Then, the classification results can be summarized into the four terms: *true positives* $\mathrm{TP} = \sum_{i=1}^n \mathbb{I}(Y_i = 1, \widehat{Y}_i = 1)$, *false positives* $\mathrm{FP} = \sum_{i=1}^n \mathbb{I}(Y_i = 0, \widehat{Y}_i = 1)$, *false negatives* $\mathrm{FN} = \sum_{i=1}^n \mathbb{I}(Y_i = 1, \widehat{Y}_i = 0)$, and *true negatives* $\mathrm{TN} = \sum_{i=1}^n \mathbb{I}(Y_i =$

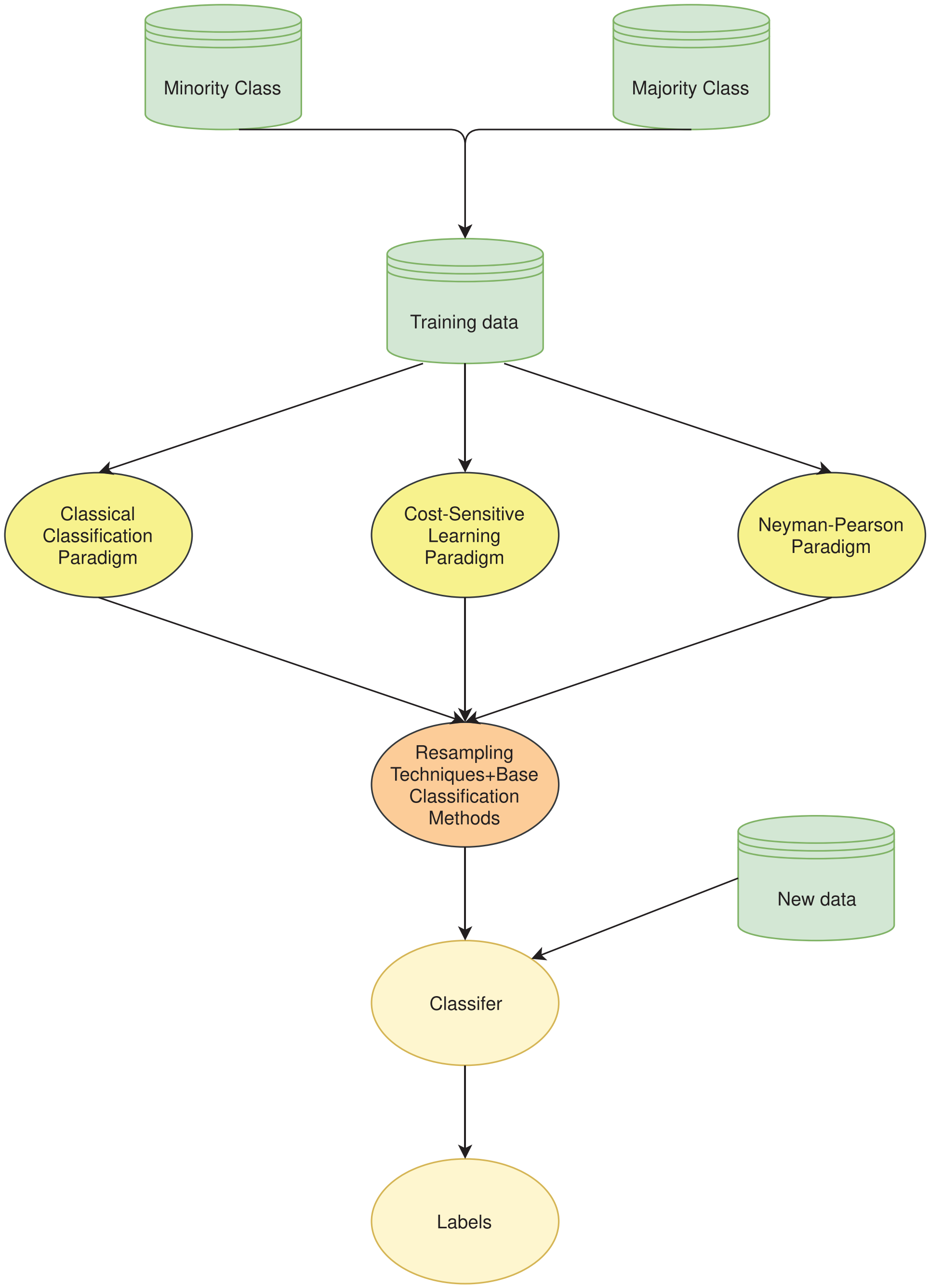
Minority Class
Majority Class
Training data
Classical Classification Paradigm
Cost-Sensitive Learning Paradigm
Neyman-Pearson Paradigm
Resampling Techniques+Base Classification Methods
New data
Classifer
Labels

FIGURE 2 A flow chart for imbalanced classification with a paradigm-oriented view

TABLE 2 Confusion matrix for a two-class problem

	Predicted class 0	Predicted class 1
True class 0	TN	FP
True class 1	FN	TP

$0, \widehat{Y}_i = 0$). These four terms are usually summarized in the so-called *confusion matrix* (Table 2).

Note that in Table 2, the class 0 is being regarded as the "negative class." In practice, sometimes we may need to set class 0 as the "positive class."

Then, the *empirical risk* can be denoted as

$$\widehat{R} = \widehat{\pi}_0 \widehat{R}_0 + \widehat{\pi}_1 \widehat{R}_1 = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}},$$

where $\widehat{\pi}_0 = (\mathrm{TN} + \mathrm{FP})/(\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN})$, $\widehat{\pi}_1 = (\mathrm{FN} + \mathrm{TP})/(\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN})$ are the empirical proportions of class 0 and 1; $\widehat{R}_0$ and $\widehat{R}_1$ are the empirical type I and II errors, respectively, that is,

$$\widehat{R}_0 = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}, \quad \widehat{R}_1 = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}.$$

TABLE 3 Various evaluation metrics

Metric	Formula
Risk	$(\mathrm{FP}+\mathrm{FN})/(\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN})$
Type I error ($\widehat{R}_0$)	$\mathrm{FP}/(\mathrm{TN}+\mathrm{FP})$
Type II error ($\widehat{R}_1$)	$\mathrm{FN}/(\mathrm{FN}+\mathrm{TP})$
Cost	$C_0\widehat{\pi}_0\widehat{R}_0+C_1\widehat{\pi}_1\widehat{R}_1$
F-score (class 0)	$2/(\mathrm{Precision}_0^{-1}+\mathrm{Recall}_0^{-1})$
F-score (class 1)	$2/(\mathrm{Precision}_1^{-1}+\mathrm{Recall}_1^{-1})$
ROC-AUC	The area under the ROC curve
PR-AUC (class 0)	The area under the PR curve when class 0 is negative
PR-AUC (class 1)	The area under the PR curve when class 0 is positive

Similarly, for given costs C_0 and C_1, the *empirical misclassification cost* is expressed as

$$\widehat{R}_c=C_0\widehat{\pi}_0\widehat{R}_0+C_1\widehat{\pi}_1\widehat{R}_1.$$

Another popular synthetic metric in the imbalanced classification literature is the *F*-score (also F_1-score or *F*-measure, [7]) for class 0, which is the harmonic mean of Precision and Recall: where $\mathrm{Precision}_0=\mathrm{TN}/(\mathrm{TN}+\mathrm{FN})$ and $\mathrm{Recall}_0=\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$. Similarly, we can also define *F*-score for class 1 as

$$F\text{-score (class 1)}=\frac{2}{\mathrm{Precision}_1^{-1}+\mathrm{Recall}_1^{-1}},$$

where $\mathrm{Precision}_1=\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and $\mathrm{Recall}_1=\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$. Here, we set *F*-score (class 0) or *F*-score (class 1) to 0 if the corresponding precision or recall is undefined or equal to 0.

When the parameter in a classification method (e.g., the threshold of scoring functions) is varied, we usually get different trade-offs between type I and type II errors. A popular tool to visualize these trade-offs is the receiver operating characteristic (ROC) curve [7, 29]. The area under the ROC curve (ROC-AUC) provides an aggregated measure for the method's performance. ROC-AUC has been used extensively to compare the performance of different classification methods. However, when the data are highly imbalanced, the ROC curves

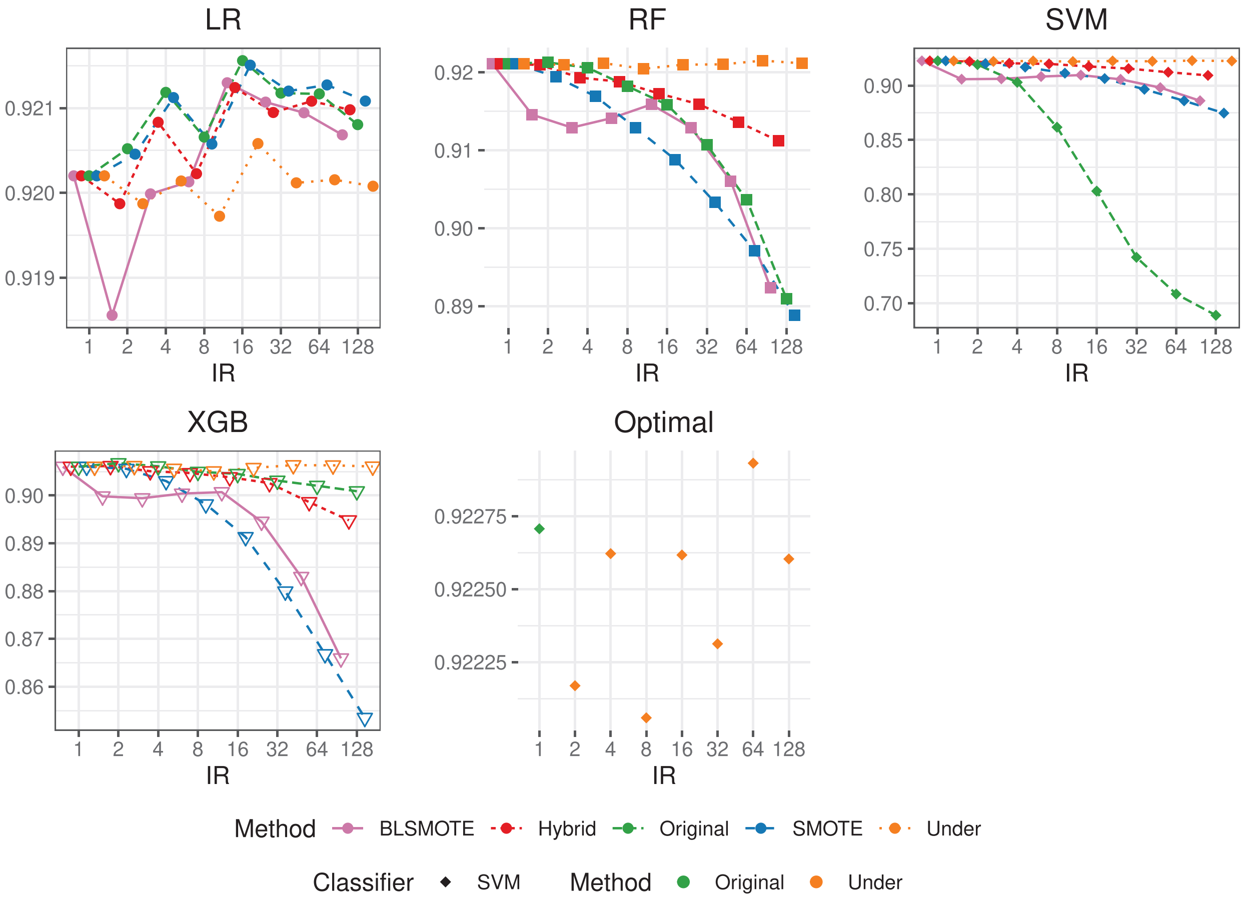

FIGURE 3 ROC-AUC of different methods in Example 1(a). The minimum and maximum of standard error: LR(0.0003, 0.0005), RF(0.0004, 0.0007), SVM(0.0003, 0.0029), XGB(0.0005, 0.0013)

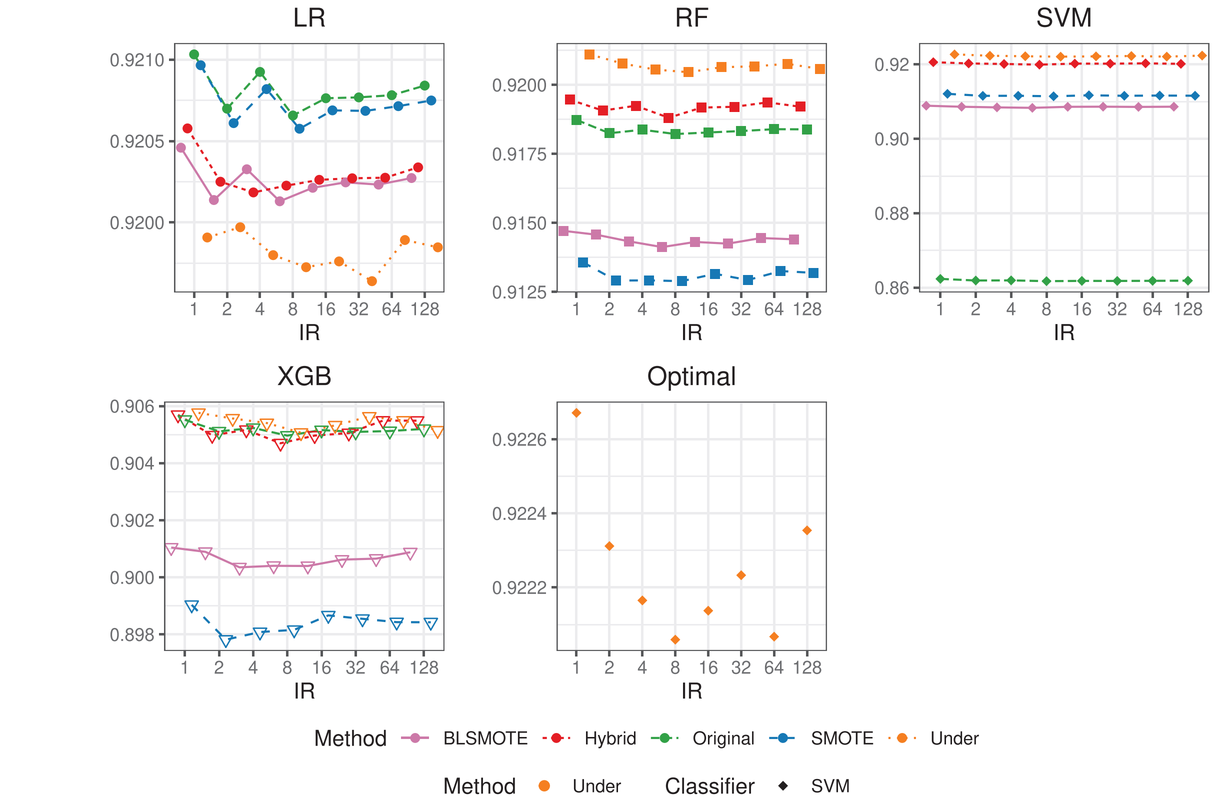

FIGURE 4 ROC-AUC of different methods in Example 1(b). The minimum and maximum of standard error: LR(0.0003, 0.0005), RF(0.0003, 0.0006), SVM(0.0003, 0.0010), XGB(0.0005, 0.0008)

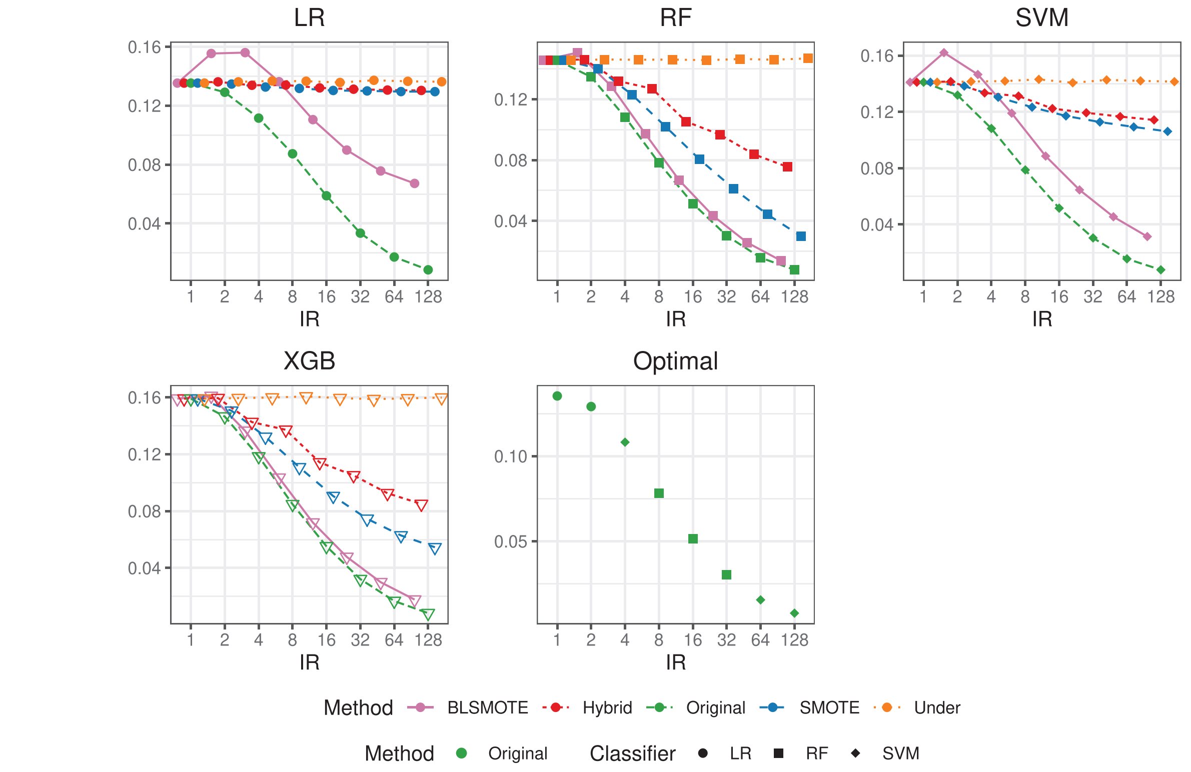

FIGURE 5 Risk of different methods under CC paradigm in Example 1(a). The minimum and maximum of standard error: LR(0, 0.0011), RF(0,0.0014), SVM(0,0.0012), XGB(0, 0.0014)

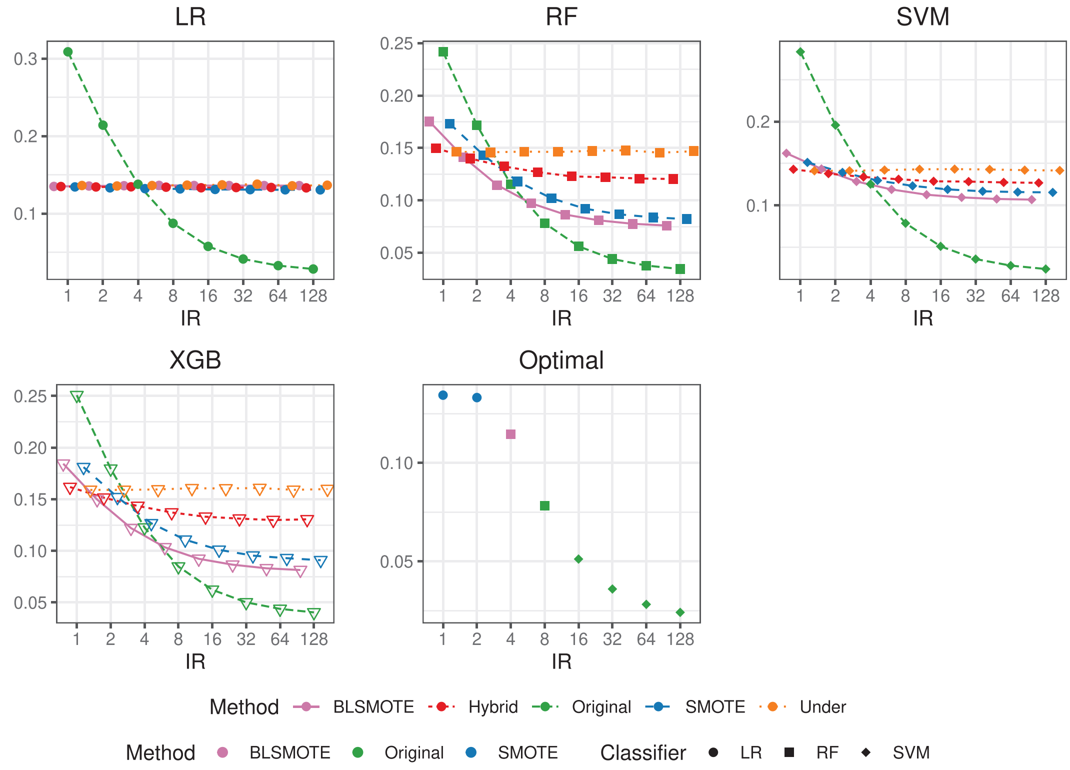

FIGURE 6 Risk of different methods under CC paradigm in Example 1(b). The minimum and maximum of standard error: LR(0.0001, 0.0018), RF(0.0002, 0.0016), SVM(0.0001, 0.0016), XGB(0.0002, 0.0016)

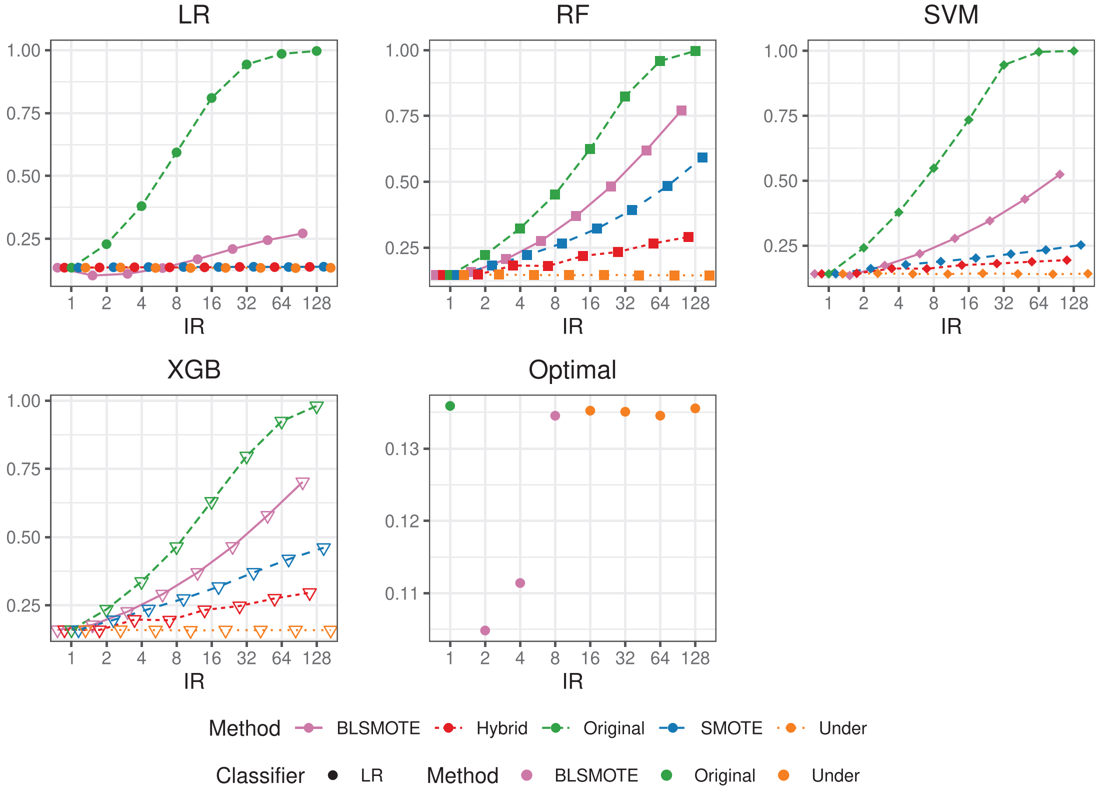

FIGURE 7 Type I error of different methods under CC paradigm in Example 1(a). The minimum and maximum of standard error: LR(0, 0.0037), RF(0, 0.0032), SVM(0, 0.0034), XGB(0, 0.0027)

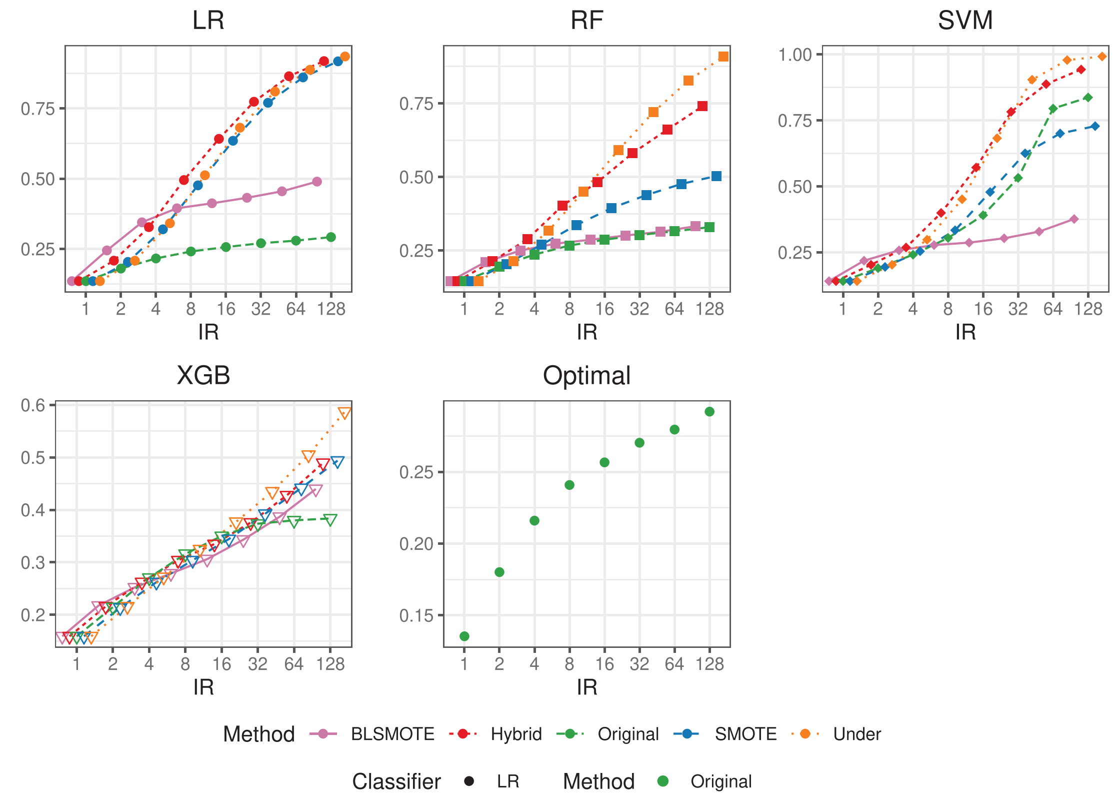

FIGURE 8 Cost of different methods under CS learning paradigm in Example 1(a). The minimum and maximum of standard error: LR(0.0006, 0.0066), RF(0.0007, 0.0068), SVM(0.0004, 0.0113), XGB(0.0004, 0.0066)

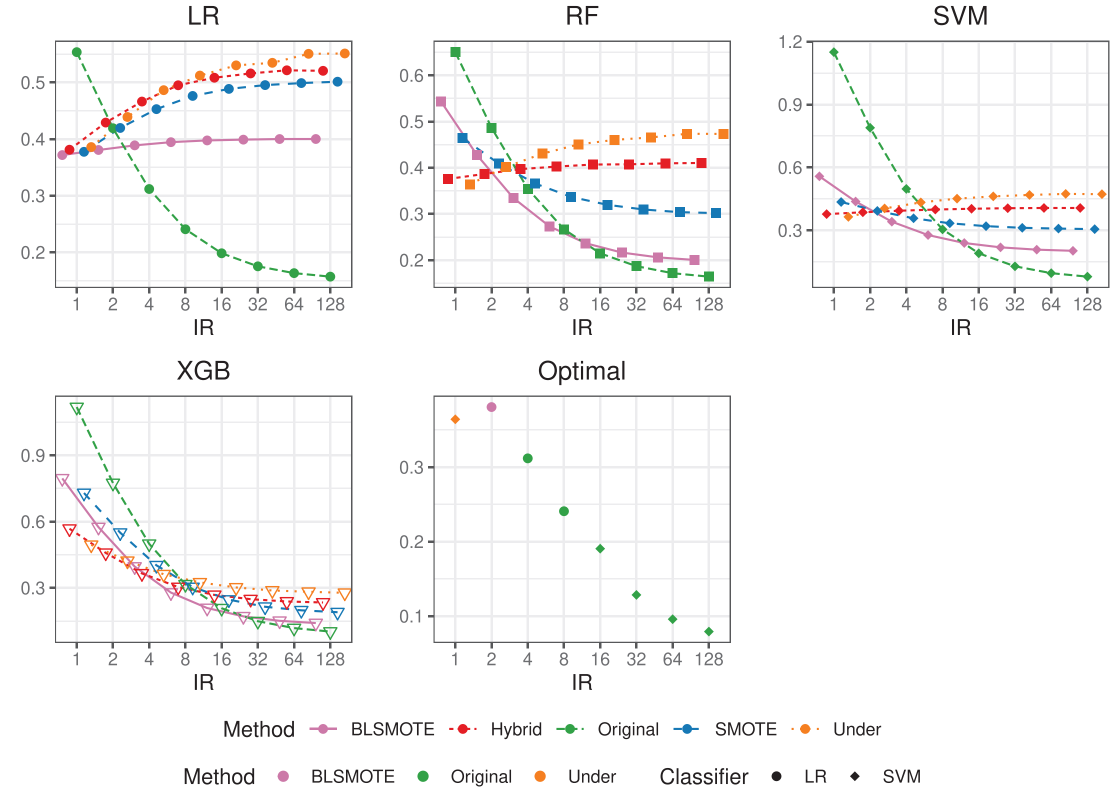

FIGURE 9 Cost of different methods under CS learning paradigm in Example 1(b). The minimum and maximum of standard error: LR(0.0006, 0.0061), RF(0.0008, 0.0054), SVM(0.0003, 0.0075), XGB(0.0005, 0.0065)

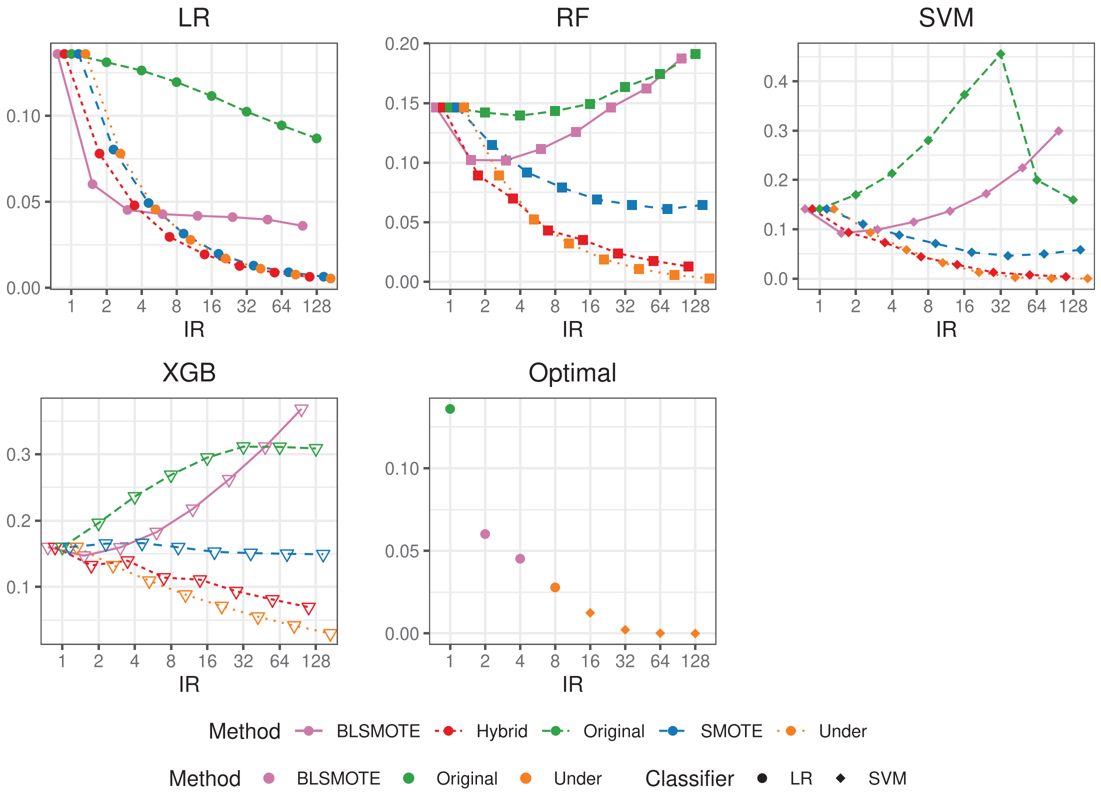

FIGURE 10 Type I error of different methods under CS learning paradigm in Example 1(a). The minimum and maximum of standard error: LR(0.0002, 0.0014), RF(0.0002, 0.0017), SVM(0, 0.0078), XGB(0.0007, 0.0019)

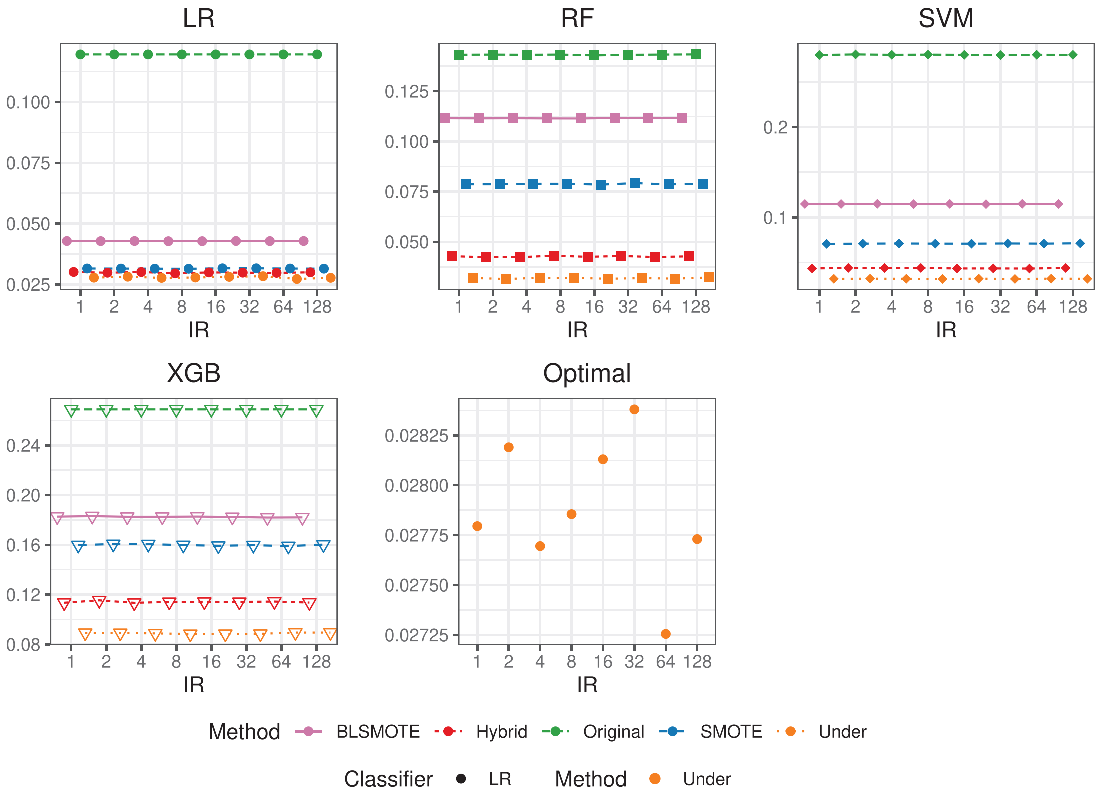

FIGURE 11 Type I error of different methods under CS learning paradigm in Example 1(b). The minimum and maximum of standard error: LR(0.0006, 0.0010), RF(0.0006, 0.0015), SVM(0.0008, 0.0019), XGB(0.0010, 0.0017)

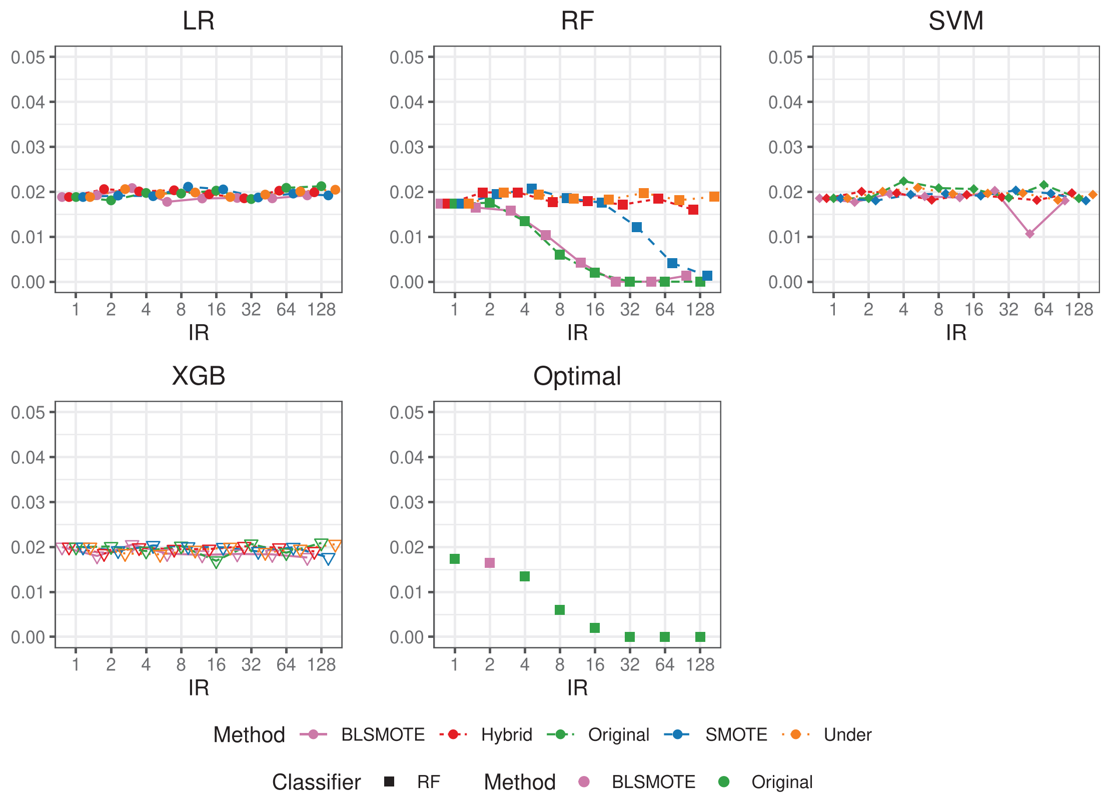

FIGURE 12 Type I error of different methods under NP paradigm in Example 1(a). The minimum and maximum of standard error: LR(0.0010, 0.0014), RF(0, 0.0015), SVM(0.0009, 0.0015), XGB(0.0009, 0.0013)

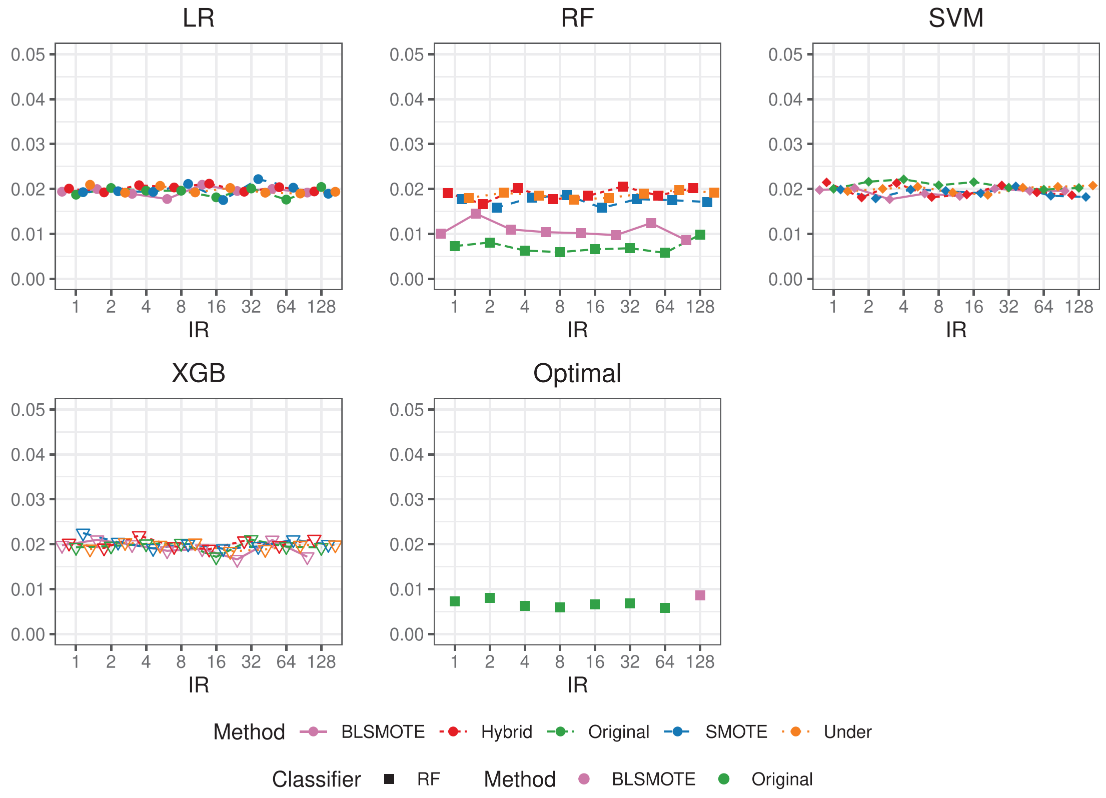

FIGURE 13 Type I error of different methods under NP paradigm in Example 1(b). The minimum and maximum of standard error: LR(0.0009, 0.0014), RF(0.0010, 0.0017), SVM(0.0009, 0.0015), XGB(0.0009, 0.0014)

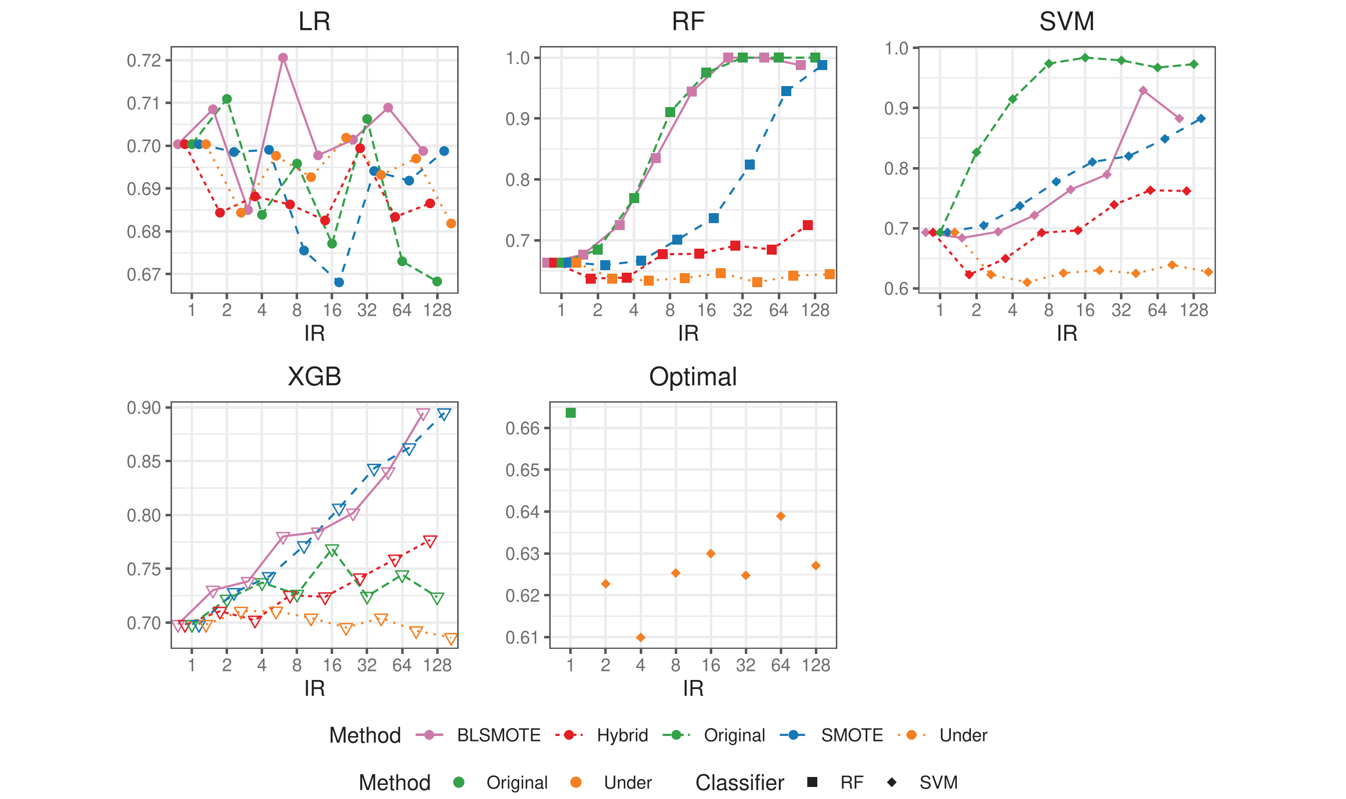

FIGURE 14 Type II error of different methods under NP paradigm in Example 1(a). The minimum and maximum of standard error: LR(0.0142, 0.0182), RF(0, 0.0219), SVM(0.0015, 0.0149), XGB(0.0081, 0.0140)

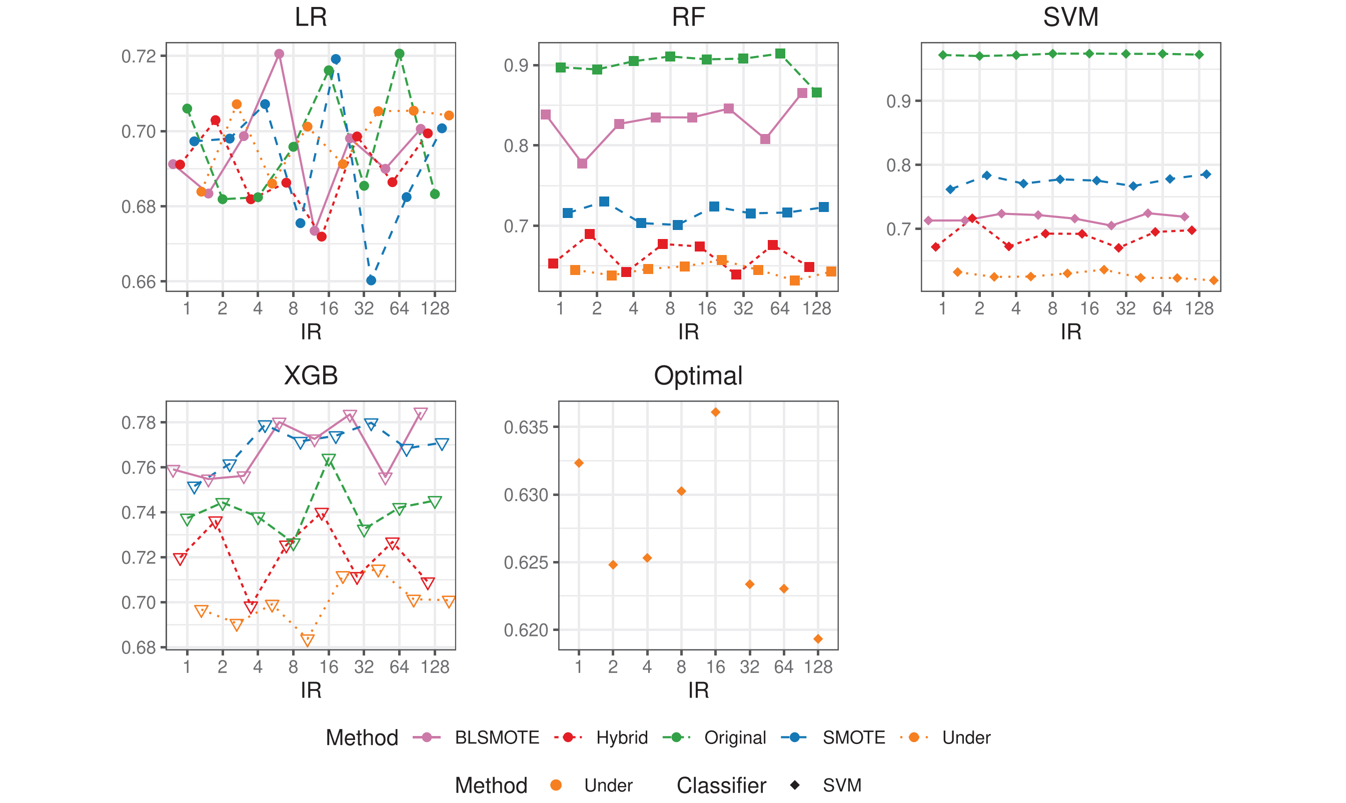

FIGURE 15 Type II error of different methods under NP paradigm in Example 1(b). The minimum and maximum of standard error: LR(0.0137, 0.0174), RF(0.0126, 0.0232), SVM(0.0023, 0.0153), XGB(0.0097, 0.0136)

TABLE 4 The frequency of winning methods in Example 1(a)

	Method								
Metric	**LR**	**RF**	**SVM**	**XGB**	**BLSMOTE**	**Hybrid**	**Original**	**SMOTE**	**Under**
AUC	0	0	**8**	0	0	0	1	0	**7**
PR-AUC(0)	0	**5**	3	0	1	0	**6**	0	1
PR-AUC(1)	0	0	**8**	0	0	0	1	0	**7**
CC-Type I	**8**	0	0	0	3	0	1	0	**4**
CS-Type I	4	0	**4**	0	2	0	1	0	**5**
NP-Type I	0	**8**	0	0	1	0	**7**	0	0
Type I	12	8	4	0	6	0	9	0	9
CC-Type II	1	0	**7**	0	0	0	**8**	0	0
CS-Type II	1	0	**4**	3	1	0	**7**	0	0
NP-Type II	0	1	**7**	0	0	0	1	0	**7**
Type II	2	1	18	3	1	0	16	0	7
CS-Cost	**8**	0	0	0	0	0	**8**	0	0
CC-Risk	2	**3**	**3**	0	0	0	**8**	0	0
NP-Risk	0	1	**7**	0	0	0	1	0	**7**
Risk	2	4	10	0	0	0	9	0	7
CC-F-score(0)	3	**4**	0	1	**5**	0	1	2	0
CS-F-score(0)	2	0	**4**	2	1	0	**7**	0	0
NP-F-score(0)	0	2	**6**	0	0	0	1	0	**7**
F-score(0)	5	6	10	3	6	0	9	2	7
CC-F-score(1)	2	1	**5**	0	0	0	**8**	0	0
CS-F-score(1)	1	0	**4**	3	1	0	**7**	0	0
NP-F-score(1)	0	1	**7**	0	0	0	1	0	**7**
F-score(1)	3	2	16	3	1	0	16	0	7
Total	32	26	77	9	15	0	75	2	52

can present an overly optimistic view of classifiers' performance [18]. Precision–Recall (PR) curves and their AUCs (PR-AUC) have been advocated as an alternative metric when dealing with imbalanced data [24, 55]. Note that we also have two versions of PR-AUC, depending on which class we call "positive": PR-AUC (class 0) and PR-AUC (class 1).

Now, we summarize all of the metrics discussed in Table 3.

5 | SIMULATION

In this section, we conduct extensive simulation studies to compare the matrix of 20 combinations of classification methods and resampling approaches introduced in Section 3 under each of the three classification paradigms described in Section 2 when the IR varies, using evaluation metrics reviewed in Section 4.

5.1 | Data generation process

We consider the following two examples with different data generation mechanisms.

Example 1. The conditional distributions for each class are multivariate t_4 distributions with a common covariance matrix but different mean vectors. Concretely,

$$\begin{aligned} &\text{Class } 0 : X|(Y=0) \sim t_4(\mu^0, \Sigma), \\ &\text{Class } 1 : X|(Y=1) \sim t_4(\mu^1, \Sigma), \end{aligned}$$

where $\mu^0 = (0, 0, 0, 0, 0)^\top$, $\mu^1 = (2, 2, 2, 0, 0)^\top$, and

$$\Sigma = \begin{pmatrix} 1 & 0.5 & 0.25 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 & 0 \\ 0.25 & 0.5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

TABLE 5 The frequency of winning methods in Example 1(b)

	Method								
Metric	**LR**	**RF**	**SVM**	**XGB**	**BLSMOTE**	**Hybrid**	**Original**	**SMOTE**	**Under**
AUC	0	0	**8**	0	0	0	0	0	**8**
PR-AUC(0)	0	**5**	3	0	0	0	**8**	0	0
PR-AUC(1)	0	0	**8**	0	0	0	0	0	**8**
CC-Type I	**8**	0	0	0	**7**	0	0	0	1
CS-Type I	**8**	0	0	0	0	0	0	0	**8**
NP-Type I	0	**8**	0	0	1	0	**7**	0	0
Type I	16	8	0	0	8	0	7	0	9
CC-Type II	2	2	**4**	0	1	0	**5**	2	0
CS-Type II	0	0	**8**	0	0	0	**8**	0	0
NP-Type II	0	0	**8**	0	0	0	0	0	**8**
Type II	2	2	20	0	1	0	13	2	8
CS-Cost	3	0	**5**	0	1	0	**6**	0	1
CC-Risk	2	2	**4**	0	1	0	**5**	2	0
NP-Risk	0	0	**8**	0	0	0	0	0	**8**
Risk	2	2	12	0	1	0	5	2	8
CC-F-score(0)	3	2	**3**	0	2	0	**4**	2	0
CS-F-score(0)	2	0	**6**	0	0	0	**8**	0	0
NP-F-score(0)	0	0	**8**	0	0	0	0	0	**8**
F-score(0)	5	2	17	0	2	0	12	2	8
CC-F-score(1)	1	2	**5**	0	1	0	**6**	1	0
CS-F-score(1)	1	0	**7**	0	0	0	**8**	0	0
NP-F-score(1)	0	0	**8**	0	0	0	0	0	**8**
F-score(1)	2	2	20	0	1	0	14	1	8
Total	30	21	93	0	14	0	65	7	58

a. To have a precise control on the IR, we explicitly generate $n_0 = 300$ observations from the minority class (class 0) and n_1 observations from the majority class, where $\text{IR} = n_1/n_0$ is a pre-specified value varying in $\{2^i,\ i = 0, 1, \ldots, 7\}$. This leads to a training sample $\{(X_i, Y_i),\ i = 1, \ldots, n\}$ where $n = n_0 + n_1$. Following the same mechanism, we also generate a test sample with size m consisting of $m_0 = 2000$ and $m_1 = m_0 \times \text{IR}$ observations from class 0 and 1, respectively. This generation mechanism guarantees the same IR for both training and test samples.

b. To observe the influence of different IR for test samples, we fix $\text{IR}_{\text{train}} = 8$ for training samples and vary IR_{test} in $\{2^i, i = 0, 1, \ldots, 7\}$ for test samples. The parameters n_0 and m_0 are 300 and 2000 respectively; and $n_1 = 300 \times 8 = 2400$, $m_1 = m_0 \times \text{IR}_{\text{test}}$.

Example 2. The conditional distributions for each class are multivariate Gaussian versus a mixture of multivariate Gaussian. Concretely,

$$\text{Class } 0: \quad X|(Y = 1) \sim \mathcal{N}\left(\frac{1}{2}(\mu^0 + \mu^1), \Sigma\right), \tag{1}$$

$$\text{Class } 1: \quad X|(Y = 0) \sim \frac{1}{2}\mathcal{N}(\mu^0, \Sigma) + \frac{1}{2}\mathcal{N}(\mu^1, \Sigma), \tag{2}$$

where μ^0, μ^1, and Σ are the same as Example 1. The remaining data generation mechanisms are the same as in Example 1. As a result, we also have Example 2(a) with the same training and testing IR and 2(b) where we fix the training IR and vary the testing IR.

5.2 | Implementation details

Regarding the resampling methods, we consider the following four options.

TABLE 6 The frequency of winning methods in Example 2(a)

	Method								
Metric	**LR**	**RF**	**SVM**	**XGB**	**BLSMOTE**	**Hybrid**	**Original**	**SMOTE**	**Under**
AUC	0	0	**8**	0	0	0	1	0	**7**
PR-AUC(0)	0	0	**8**	0	0	0	1	0	**7**
PR-AUC(1)	0	0	**8**	0	0	0	1	0	**7**
CC-Type I	0	0	**8**	0	0	0	1	0	**7**
CS-Type I	**6.1**	0	1.9	0	0.9	1.4	1	**2.4**	2.3
NP-Type I	3	**5**	0	0	2	0	**5**	0	1
Type I	9.1	5	9.9	0	2.9	1.4	7	2.4	10.3
CC-Type II	**7**	0	1	0	0	0	**8**	0	0
CS-Type II	0	0	3	**5**	1	0	**7**	0	0
NP-Type II	0	5	**3**	0	0	2	1	0	**5**
Type II	7	5	7	5	1	2	16	0	5
CS-Cost	0	**3**	**4**	1	**6**	0	1	1	0
CC-Risk	**7**	0	1	0	0	0	**8**	0	0
NP-Risk	0	4	**4**	0	0	2	1	0	**5**
Risk	7	4	5	0	0	2	9	0	5
CC-*F*-score(0)	0	0	**8**	0	0	0	1	0	**7**
CS-*F*-score(0)	0	0	**7**	1	**5**	0	1	1	1
NP-*F*-score(0)	0	5	**3**	0	0	2	1	0	**5**
F-score(0)	0	5	18	1	5	2	3	1	13
CC-*F*-score(1)	**7**	0	1	0	0	0	**8**	0	0
CS-*F*-score(1)	0	0	3	**5**	1	0	**7**	0	0
NP-*F*-score(1)	0	5	**3**	0	0	2	1	0	**5**
F-score(1)	7	5	7	5	1	2	16	0	5
Total	30.1	27	74.9	12	15.9	9.4	55	4.4	59.3

- No resampling (Original): we use the training dataset as it is without any modification.
- Random undersampling (Under): we keep all the n_0 observations in the minority class and randomly sample n_0 observations without replacement from the majority class. Then, we have a balanced dataset in which each class is of size n_0.
- Oversampling (SMOTE, BLSMOTE): we keep all the n_1 observations in the majority class. We use SMOTE and BLSMOTE (R Package **smotefamily**, v1.3.1 [56]) to generate new synthetic data for the minority class until the new training set is balanced. Then, we have a balanced dataset in which each class is of size n_1. Following the default choice in **smotefamily**, we set the number of nearest neighbors $K = 5$ in the oversampling process.
- Hybrid methods (Hybrid): we conduct a combination of random undersampling and SMOTE with the final training set consists of n_h minority and majority observations with $n_h = \lfloor \sqrt{n_0 * n_1}/n_0 \rfloor * n_0$ where $\lfloor \cdot \rfloor$ is the floor function.

Regarding the base classification methods, we apply the following R packages or functions with their default parameters.

- Logistic regression (`glm` function in base R).
- Random forest (R Package **randomForest**, v4.6.14 [36]).
- Support vector machine (R Package **e1071**, v1.7.2 [43]).
- XGBoost (R Package **xgboost**, v0.90.0.2 [15]).

Regarding the classification paradigms, some specifics are listed below.

- CS learning paradigm: we specify the cost $C_0 = \text{IR}$ and $C_1 = 1$.

TABLE 7 The frequency of winning methods in Example 2(b)

	Method								
Metric	**LR**	**RF**	**SVM**	**XGB**	**BLSMOTE**	**Hybrid**	**Original**	**SMOTE**	**Under**
AUC	0	0	**8**	0	0	0	0	0	**8**
PR-AUC(0)	0	0	**8**	0	0	0	0	0	**8**
PR-AUC(1)	0	0	**8**	0	0	0	0	0	**8**
CC-Type I	0	0	**8**	0	0	0	0	0	**8**
CS-Type I	**8**	0	0	0	**2**	**2**	0	**2**	**2**
NP-Type I	0	**8**	0	0	2	0	**6**	0	0
Type I	8	8	8	0	4	2	6	2	10
CC-Type II	**8**	0	0	0	0	0	**8**	0	0
CS-Type II	0	0	0	**8**	0	0	**8**	0	0
NP-Type II	0	**6**	2	0	0	**3**	0	0	**5**
Type II	8	6	2	8	0	3	16	0	5
CS-Cost	1	1	2	**4**	1	0	**4**	1	2
CC-Risk	**7**	0	1	0	0	0	**7**	0	1
NP-Risk	0	**6**	2	0	0	**4**	0	0	4
Risk	7	6	3	0	0	4	7	0	5
CC-*F*-score(0)	0	0	**8**	0	3	1	0	0	**4**
CS-*F*-score(0)	0	1	**4**	3	**4**	0	3	1	0
NP-*F*-score(0)	0	**6**	2	0	0	**4**	0	0	4
F-score(0)	0	7	14	3	7	5	3	1	8
CC-*F*-score(1)	**8**	0	0	0	0	0	**8**	0	0
CS-*F*-score(1)	0	0	0	**8**	0	0	**8**	0	0
NP-*F*-score(1)	0	**6**	2	0	0	**3**	0	0	**5**
F-score(1)	8	6	2	8	0	3	16	0	5
Total	32	34	55	23	12	17	52	4	59

- NP paradigm: we use the NP umbrella algorithm as implemented in R package **nproc** v2.1.4, and set $\alpha = 0.05$ and the tolerance level $\delta = 0.05$.

Denote by $|S|$ the cardinality of a set S. Let $O = \{\text{CC}, \text{CS}, \text{NP}\}$, $T = \{$Original, Under, SMOTE, BLSMOTE, Hybrid$\}$, $C = \{\text{LR}, \text{RF}, \text{SVM}, \text{XGB}\}$, and $B = \{2^i, \ i = 0,1,2, \ldots, 7\}$. Hence, there are $|O| \times |T| \times |C| \times |B|$ (480) classification systems studied in this paper for a given imbalanced classification problem.

For each ensemble system, we evaluate the performance of different classifiers in terms of the following metrics reviewed in Section 4: overall classification error (Risk), type I error, type II error, expected misclassification cost (Cost), *F*-score (class 0), and *F*-score (class 1). When the threshold varies for each classification method, we also report the area under ROC curve (ROC-AUC) and the area under PR curve (PR-AUC [class 0] and PR-AUC [class 1]).

We repeat the experiment 100 times and report the average performance in terms of mean, standard error, and winning methods for each metric and classification paradigm combination. The results are summarized in Figures 3–15 as well as in Tables 4–7.

5.3 | Results and interpretations

For each figure, we present the results of classification methods under each IR in the first four panels, while the last panel shows the optimal combination of resampling technique and base classification method under each IR.

Next, we provide some interpretations and insights from the figures and tables under each classification paradigm.

For Example 1(a), where we vary the training and testing IR at the same time, we present the ROC-AUC in

TABLE 8 The frequency of winning methods when IR of test data varies in credit fraud detection

	Method								
Metric	**LR**	**RF**	**SVM**	**XGB**	**BLSMOTE**	**Hybrid**	**Original**	**SMOTE**	**Under**
AUC	0	**8**	0	0	0	2	0	**6**	0
PR-AUC(0)	0	**8**	0	0	0	0	0	**8**	0
PR-AUC(1)	0	**8**	0	0	0	**8**	0	0	0
CC-Type I	**7**	0	0	1	0	0	0	0	**8**
CS-Type I	0	0	**8**	0	0	0	0	0	**8**
NP-Type I	0	**8**	0	0	0	0	**8**	0	0
Type I	7	8	8	1	0	0	8	0	16
CC-Type II	0	0	**4**	**4**	0	0	**8**	0	0
CS-Type II	0	0	0	**8**	0	0	**8**	0	0
NP-Type II	0	**6**	2	0	0	**6**	0	0	2
Type II	0	6	6	12	0	6	16	0	2
CS-Cost	0	1	0	**7**	0	2	**3**	3	0
CC-Risk	1	**3**	0	4	0	3	1	**4**	0
NP-Risk	0	**6**	2	0	0	**6**	0	0	2
Risk	1	9	2	4	0	9	1	4	2
CC-*F*-score(0)	1	**3**	0	**4**	0	**4**	1	**3**	0
CS-*F*-score(0)	0	0	0	**8**	0	0	**6**	2	0
NP-*F*-score(0)	0	**8**	0	0	0	**8**	0	0	0
F-score(0)	1	11	0	12	0	12	7	5	0
CC-*F*-score(1)	1	**4**	0	3	0	3	1	**4**	0
CS-*F*-score(1)	0	0	0	**8**	0	0	**6**	2	0
NP-*F*-score(1)	0	**6**	2	0	0	**6**	0	0	2
F-score(1)	1	10	2	11	0	9	7	6	2
Total	10	69	18	47	0	48	42	32	22

Figure 3 as an overall measure of classification methods without the need to specify the classification paradigm. First of all, LR is surprisingly stable for all resampling techniques across all IRs. Another study on the robustness of LR for imbalanced data can be found in Owen [46]. Then, from the panels corresponding to RF, SVM, and XGB, we suggest that it is essential to apply specific resampling techniques to keep the ROC-AUC at a high value when IR increases. For Example 1(b) where we fix the training IR and vary the testing IR, the ROC-AUC in Figure 4 is more robust across the board. In addition, we report the range of the standard errors for each base classification method in the captions of Figures 3 and 4, and they are all very small. Thus, the standard error does not affect the determination of the optimal combination. We omit the plots of ROC-AUC for Example 2 as they look similar.

5.3.1 | CC paradigm

We first focus on analyzing the results for Example 1. Figures 5 and 6 exhibit the risk of different methods. We observe that the empirical risk of all classifiers without resampling is smaller than that with any resampling technique in most cases, and decreases as IR increases. This is in line with our intuition that if the risk is the primary measure of interest, we would be better off not applying any resampling techniques. In addition, we observe that only undersampling leads to a stable risk when the IR increases for all four base classification methods considered. Finally, the resampling techniques can make risk more stable across all IRs in Figure 6.

As mentioned in Section 2, minimizing the risk with imbalanced data could lead to large type I errors, demonstrated clearly in Figure 7. By using the resampling

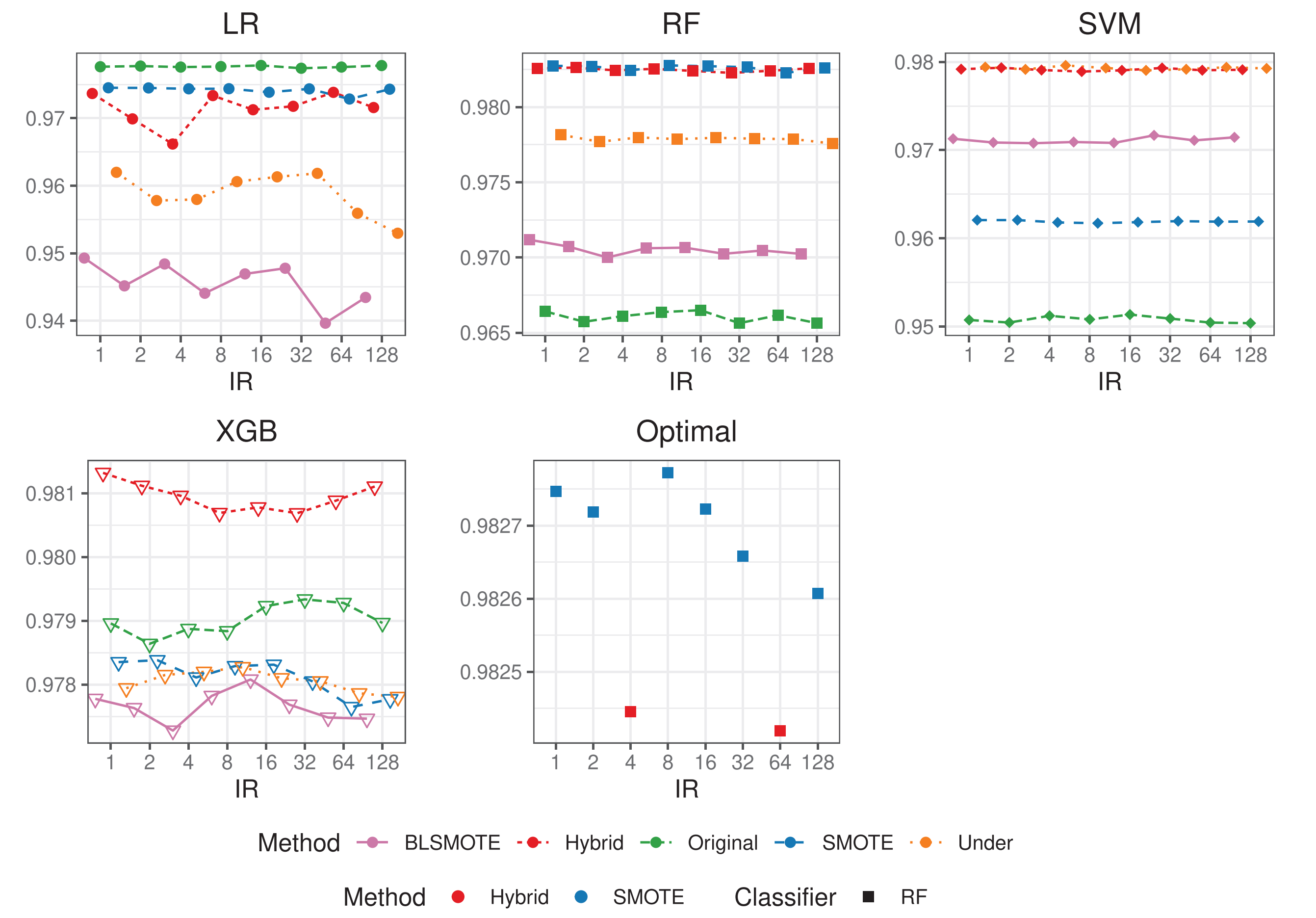

FIGURE 16 ROC-AUC of different methods in real data. The minimum and maximum of standard error: LR(0.0005, 0.0089), RF(0.0005, 0.0009), SVM(0.0005, 0.0014), XGB(0.0005, 0.0007)

techniques, however, we can have much better control over type I error as IR increases. In particular, undersampling works well for all four classification methods. Lastly, we note that the optimal choices when IR > 1 all involve resampling techniques.

The figures for Example 2 convey a similar message as in Example 1 that we do not need any resampling if the goal is to minimize the risk. On the other hand, applying certain resampling techniques is critical to bring down the type I error and increase the ROC-AUC value. Again, we omit these figures to save space.

5.3.2 | CS learning paradigm

When we are in the CS learning paradigm, the objective is to minimize the expected total misclassification cost. We again first look at the results from Example 1. Naturally, we would like to see the impact of the resampling techniques on different classification methods in terms of empirical cost, which is summarized in Figures 8 and 9. From the figures, we observe that no resampling leads to the smallest cost in most cases. When IR is large, BLSMOTE leads to the smallest cost for SVM.

Now, we look at the results for type I error in Figures 10 and 11, where we discover that all classification methods benefit significantly from resampling techniques with undersampling being the best choice for most scenarios.

5.3.3 | NP paradigm

The NP paradigm aims to minimize type II error while controlling type I error under a target level α. In the current implementation, we set $\alpha = 0.05$. From Figures 12 and 13, we observe that the type I errors are well controlled under α throughout all IRs for all base classification methods in Example 1(a) and (b).

When we look at Figure 14, the benefits that resampling techniques can bring are apparent in most cases. Undersampling or hybrid resampling leads to a type II error well under control. Moreover, type II error is

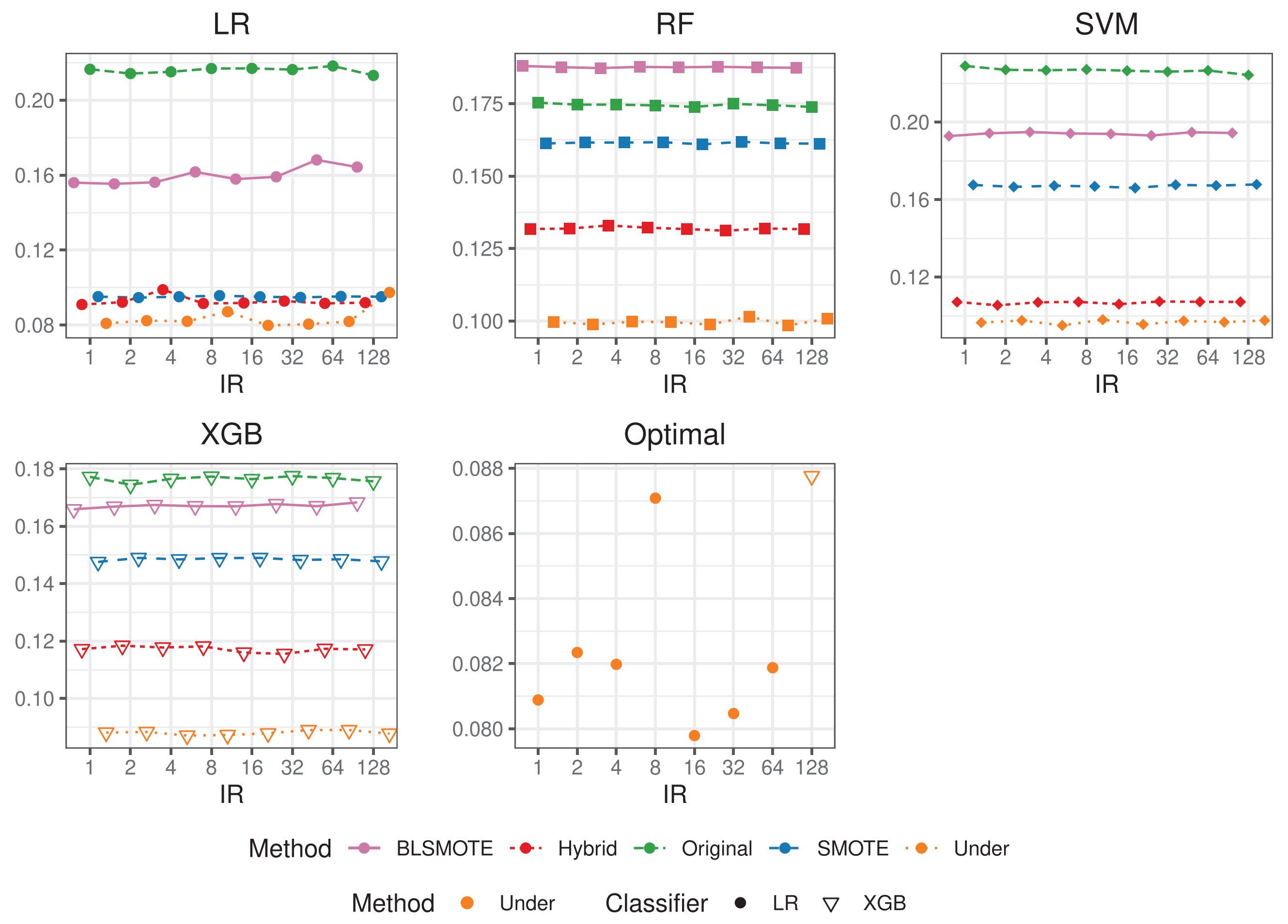

FIGURE 17 Type I error of different methods under CC paradigm in real data. The minimum and maximum of standard error: LR(0.0020, 0.0086), RF(0.0019, 0.0024), SVM(0.0019, 0.0036), XGB(0.0018, 0.0023)

more robust when different IRs are selected for the test dataset.

For Example 2, we have the same conclusion that resampling techniques can help to reduce type II error with the type I error well controlled under α.

5.3.4 | Summary

In addition to the plots, we summarize in Tables 4–7 the winning frequency of resampling techniques and classification methods in terms of each evaluation metric of all IRs in Examples 1(a), (b) and 2(a), and 2(b), respectively. The number in each cell of tables represents the winning frequency for each base classification method or each resampling technique for the given metric. The numbers in bold represent the most frequent winning combination of resampling techniques and classification methods. Clearly, the optimal choices differ for different evaluation metrics, IRs, and data generation mechanisms. From these tables and the above figures, we can draw the following conclusions:

a. All the classifiers can control the type I error under a certain level α under the NP paradigm (see Figures 12 and 13).

b. For most base classification methods, ROC-AUC can usually benefit from resampling techniques, whether or not the test class proportion is at the same level of imbalance as the training set (see Figures 3 and 4).

c. Resampling techniques, in general, bring down the type I error regardless of the classification paradigm (see Figures 7 and 11).

d. The optimal combination of base classification method and resampling technique should be interpreted together with both the paradigm and evaluation metric. For example, in Table 4, the combination “LR + Under” leads to the minimal type I error under the CC paradigm.

e. When the training class proportion is fixed and IR varies for the test dataset, the results are robust in most cases (see Figures 6, 11, and 15).

6 | REAL DATA-CREDIT CARD FRAUD DETECTION

The credit card transaction data are available at http://kaggle.com/mlg-ulb/creditcardfraud. It includes credit card transactions made in September 2013 by European cardholders. In particular, it contains transactions that occurred in 2 days, where we have 492 frauds out of 284,807 transactions. Therefore, this dataset is highly imbalanced with an IR about 578 (284,315/492). Due to confidentiality issues, the website does not provide the original features and more background information about this dataset. Features V1, V2, … , V28 are the principal components obtained with PCA. The only features which have not been transformed with PCA are "Time" and "Amount." Feature "Time" contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature "Amount" is the transaction amount. They are scaled to zero mean and unit variance. Using the feature "Class," we redefine "0" as the fraud class (class 0) and "1" as the no-fraud class (class 1). We use the features V1, V2, … , V28, Time and Amount as predictor variables for the classification methods.

We specify the IR as 128 for training dataset and extract a subsample from this large dataset. In particular, we randomly sample $n_0 = 300$ data points from class 0 (fraud) and $n_1 = n_0 * \text{IR} = 38{,}400$ from class 1 (no-fraud). This procedure creates our training dataset. The test dataset contains a random sample of $m_0 = 192$ for class 0 and $m_1 = m_0 * \text{IR}_{\text{test}}$ for class 1 from the remaining data, where IR_{test} varies in $\{2^i,\ i = 0, 1, \dots, 7\}$. This splitting mechanism implies that IR will be different for the training and test datasets.

The remaining implementation details are the same as in Section 5.2. We still repeat the experiment 100 times and report the average performance and frequency of winning methods by the mean for each metric and classification paradigm combination. The frequency of winning methods were summarized in Table 8 and report Figures 16 and 17 and omit the other figures since they convey similar information to that in Section 5.3.

From Figures 16 and 17, resampling techniques are in general beneficial for the metrics in most cases. In addition, most of the results are robust when the test IR increases. This is consistent with the simulation results. Table 8 shows that the combination "RF + Hybrid" has the top performance. Note that this appears to be different from the choices implied by Tables 4-7, which again show that the best performing method highly depends on the data generation process. This actually agrees with our understanding of SVM versus RF in that RF may be more effective than SVM in a more complex scenario. Moreover, the optimal methods depend on the learning paradigm and evaluation metrics. For example, if our objective is to minimize the overall risk under the CC paradigm, "RF + SMOTE" is the best choice in Table 8; if our objective is to minimize the type II error while controlling the type I error under a specific level, "RF + Hybrid" performs the best. Therefore, there is no universal best combination for the imbalanced classification problem.

7 | DISCUSSION

In this paper, we review the imbalanced classification with a paradigm-based view. In addition to the few take-away messages we offered in the simulation section, the main message from the review is that there is no single best approach to imbalanced classification. The optimal choice for resampling techniques and base classification methods highly depends on the classification paradigms, evaluation metric, as well as the severity of imbalancedness (IR).

Admittedly, we only considered a selective list of resampling techniques and base classification methods. There are many other combinations that are worth further consideration. In addition, we presented results from two simulated data generation processes as well as a real dataset, which could be unrepresentative for specific applications. We suggest practitioners adapt our analysis process for evaluating different choices for imbalanced classification to align with their data generation mechanism.

Furthermore, in our numerical experiments, all base classification methods were applied using the corresponding R-packages with their default parameters. Note that although we did not tune the parameters due to the already-extensive simulation settings, it is well known that parameter tuning could further improve the performance of a classifier in certain situation. For example, the parameter k in SMOTE [13] can be selected via cross-validation. We leave a systematic study of the impact of parameter tuning on imbalanced classification as a future research topic.

Lastly, we focused on binary classification throughout the review. We expect similar interpretations and conclusions from multi-class imbalanced classification.

ACKNOWLEDGMENT

This work is partially supported by NSF CAREER Grant DMS-1554804.

DATA AVAILABILITY STATEMENT

All data involved in the paper are either from simulation or from the public domain. The data and code will be available for the public upon publication.

ORCID

Xin Tong https://orcid.org/0000-0001-8534-3827

REFERENCES

1. F. Alahmari, *A comparison of resampling techniques for medical data using machine learning*, J. Inf. Knowl. Manag. 19(01) (2020), 2040016.
2. N. S. Altman, *An introduction to kernel and nearest-neighbor nonparametric regression*, Am. Stat. 46(3) (1992), 175–185.
3. M. Anis, M. Ali, S. A. Mirza, and M. M. Munir, *Analysis of resampling techniques on predictive performance of credit card classification*, Mod. Appl. Sci. 14(7) (2020), 92.
4. N. P. Arnqvist, B. Ngendangenzwa, E. Lindahl, L. Nilsson, and J. Yu, *Efficient surface finish defect detection using reduced rank spline smoothers and probabilistic classifiers*, Econom. Stat. 18 (2021), 89–105.
5. C. L. Beaulieu, J. Majewski, J. Schwartzentruber, M. E. Samuels, B. A. Fernandez, F. P. Bernier, M. Brudno, B. Knoppers, J. Marcadier, D. Dyment, S. Adam, D. E. Bulman, S. J. M. Jones, D. Avard, M. T. Nguyen, F. Rousseau, C. Marshall, R. F. Wintle, Y. Shen, S. W. Scherer, FORGE Canada Consortium, J. M. Friedman, J. L. Michaud, and K. M. Boycott, *FORGE Canada consortium: Outcomes of a 2-year national rare-disease gene-discovery project*, Am. J. Hum. Genet. 94(6) (2014), 809–817.
6. J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. E. Brodley, *Pruning decision trees with misclassification costs*, European Conference on Machine Learning, Springer, 1998, pp. 131–136.
7. A. P. Bradley, *The use of the area under the roc curve in the evaluation of machine learning algorithms*, Pattern Recogn. 30(7) (1997), 1145–1159.
8. L. Breiman, *Random forests*, Mach. Learn. 45(1) (2001), 5–32.
9. A. Cannon, J. Howse, D. Hush, and C. Scovel, Learning with the Neyman–Pearson and min–max criteria, *Los Alamos National Laboratory, Tech. Rep. LA-UR*, 02-2951, 2002.
10. P. Cao, J. Yang, W. Li, D. Zhao, and O. Zaiane, *Ensemble-based hybrid probabilistic sampling for imbalanced data learning in lung nodule cad*, Comput. Med. Imaging Graph. 38(3) (2014), 137–150.
11. C. L. Castro and A. P. Braga, *Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data*, IEEE Trans. Neural Netw. Learn. Syst. 24(6) (2013), 888–899.
12. S. Cateni, V. Colla, and M. Vannucci, *A method for resampling imbalanced datasets in binary classification tasks for real-world problems*, Neurocomputing 135 (2014), 32–41.
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *SMOTE: Synthetic minority over-sampling technique*, J. Artif. Intell. Res. 16 (2002), 321–357.
14. T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.
15. T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, and Y. Li, *xgboost: Extreme gradient boosting*. R package version 0.90.0.2, 2019, available at https://CRAN.R-project.org/package=xgboost.
16. Y.-S. Chen, *An empirical study of a hybrid imbalanced-class DT-RST classification procedure to elucidate therapeutic effects in uremia patients*, Med. Biol. Eng. Comput. 54(6) (2016), 983–1001.
17. C. Cortes and V. Vapnik, *Support-vector networks*, Mach. Learn. 20(3) (1995), 273–297.
18. J. Davis and M. Goadrich, *The relationship between precision–recall and ROC curves*, Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 233–240.
19. J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, and L. I. Kuncheva, *Random balance: Ensembles of variable priors classifiers for imbalanced data*, Knowl. Based Syst. 85 (2015), 96–111.
20. P. Domingos, *MetaCost: A general method for making classifiers cost-sensitive*, KDD 99 (1999), 155–164.
21. C. Elkan, *The foundations of cost-sensitive learning*, International Joint Conference on Artificial Intelligence, vol. 17, Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.
22. V. García, J. S. Sánchez, R. Martín-Félez, and R. A. Mollineda, *Surrounding neighborhood-based smote for learning from imbalanced data sets*, Prog. Artif. Intell. 1(4) (2012), 347–362.
23. V. García, J. S. Sánchez, and R. A. Mollineda, *On the effectiveness of preprocessing methods when dealing with different levels of class imbalance*, Knowl. Based Syst. 25(1) (2012), 13–21.
24. M. Goadrich, L. Oliphant, and J. Shavlik, *Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction*, International Conference on Inductive Logic Programming, Springer, 2004, pp. 98–115.
25. H. Guo, Y. Li, J. Shang, M. Gu, Y. Huang, and B. Gong, *Learning from class-imbalanced data: Review of methods and applications*, Expert Syst. Appl. 73 (2017), 220–239.
26. H. Han, W.-Y. Wang, and B.-H. Mao, *Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning*, in *International Conference on Intelligent Computing*, Springer, Hefei, China, 2005, 878–887.
27. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, Springer Science & Business Media, New York, NY, 2009.
28. H. He, Y. Bai, E. A. Garcia, and S. Li, *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*, 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.
29. J. Huang and C. X. Ling, *Using AUC and accuracy in evaluating learning algorithms*, IEEE Trans. Knowl. Data Eng. 17(3) (2005), 299–310.
30. G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, Vol 112, Springer, New York, NY, 2013.
31. V. Koltchinskii, *Oracle inequalities in empirical risk minimization and sparse recovery problems: Ecole d'Eté de Probabilités de*

Saint-Flour XXXVIII-2008, Vol 2033, Springer Science & Business Media, Berlin, Germany, 2011.

32. S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, *Supervised machine learning: A review of classification techniques*, Emerg. Artif. Intell. Appl. Comput. Eng. 160 (2007), 3–24.
33. M. Kubat, R. C. Holte, and S. Matwin, *Machine learning for the detection of oil spills in satellite radar images*, Mach. Learn. 30(2) (1998), 195–215.
34. M. Kuhn and K. Johnson, *Applied predictive modeling*, Vol 26, Springer, New York, NY, 2013.
35. N. S. Kumar, K. N. Rao, A. Govardhan, K. S. Reddy, and A. M. Mahmood, *Undersampled* k*-means approach for handling imbalanced distributed data*, Prog. Artif. Intell. 3(1) (2014), 29–38.
36. A. Liaw and M. Wiener, *Classification and regression by randomforest*, R News 2(3) (2002), 18–22 https://CRAN.R-project.org/doc/Rnews/
37. Y. Lin, Y. Lee, and G. Wahba, *Support vector machines for classification in nonstandard situations*, Mach. Learn. 46(1) (2002), 191–202.
38. C. X. Ling, Q. Yang, J. Wang, and S. Zhang, *Decision trees with minimal costs*, Proceedings of the Twenty-First International Conference on Machine Learning, ACM, 2004, p. 69.
39. V. López, S. Del Río, J. M. Benítez, and F. Herrera, *Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data*, Fuzzy Sets Syst. 258 (2015), 5–38.
40. V. López, A. Fernández, S. García, V. Palade, and F. Herrera, *An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*, Inf. Sci. 250 (2013), 113–141.
41. V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera, *Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics*, Expert Syst. Appl. 39(7) (2012), 6585–6608.
42. G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*, Vol 544, John Wiley & Sons, 2004.
43. D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, *e1071: Misc functions of the Department of Statistics, Probability Theory Group (formerly: E1071), TU Wien*. R package version 1.7-2, 2019, available at https://CRAN.R-project.org/package=e1071.
44. I. Nekooeimehr and S. K. Lai-Yuen, *Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets*, Expert Syst. Appl. 46 (2016), 405–416.
45. J. A. Nelder and R. W. Wedderburn, *Generalized linear models*, J. R. Stat. Soc. Ser. A: Gen. 135(3) (1972), 370–384.
46. A. B. Owen, *Infinitely imbalanced logistic regression*, J. Mach. Learn. Res. 8(4) (2007), 761–773.
47. X. Qiao and Y. Liu, *Adaptive weighted learning for unbalanced multicategory classification*, Biometrics 65(1) (2009), 159–168.
48. X. Qiao, H. H. Zhang, Y. Liu, M. J. Todd, and J. S. Marron, *Weighted distance weighted discrimination and its asymptotic properties*, J. Am. Stat. Assoc. 105(489) (2010), 401–414.
49. B. Ranneby and J. Yu, *Nonparametric and probabilistic classification using NN-balls with environmental and remote sensing applications*, in *Advances in Directional and Linear Statistics*, Springer, Berlin, Germany, 2011, 201–216.
50. P. Rigollet and X. Tong, *Neyman–Pearson classification, convexity and stochastic constraints*, J. Mach. Learn. Res. 12 (2011), 2831–2855.
51. I. Rish, *An empirical study of the naive Bayes classifier*, IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, 2001, pp. 41–46.
52. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning internal representations by error propagation*, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
53. J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, *SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering*, Inf. Sci. 291 (2015), 184–203.
54. S. R. Safavian and D. Landgrebe, *A survey of decision tree classifier methodology*, IEEE Trans. Syst. Man Cybern. 21(3) (1991), 660–674.
55. P. Singla and P. Domingos, *Discriminative training of Markov logic networks*, AAAI 5 (2005), 868–873.
56. W. Siriseriwan: smotefamily: A collection of oversampling techniques for class imbalance problem based on SMOTE. *R package version 1.3.1*, 2019, available at URL https://CRAN.R-project.org/package=smotefamily.
57. Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, *Cost-sensitive boosting for classification of imbalanced data*, Pattern Recogn. 40(12) (2007), 3358–3378.
58. Y. Sun, A. K. Wong, and M. S. Kamel, *Classification of imbalanced data: A review*, Int. J. Pattern Recognit. Artif. Intell. 23(04) (2009), 687–719.
59. Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, *A novel ensemble method for classifying imbalanced data*, Pattern Recogn. 48(5) (2015), 1623–1637.
60. X. Tong, *A plug-in approach to Neyman–Pearson classification*, J. Mach. Learn. Res. 14(1) (2013), 3011–3040.
61. X. Tong, Y. Feng, and J. J. Li, *Neyman–Pearson classification algorithms and NP receiver operating characteristics*, Sci. Adv. 4(2) (2018), eaao1659.
62. X. Tong, Y. Feng, and A. Zhao, *A survey on Neyman–Pearson classification and suggestions for future research*, Wiley Interdiscip. Rev.: Comput. Stat. 8(2) (2016), 64–81.
63. P. D. Turney, *Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm*, J. Artif. Intell. Res. 2 (1994), 369–409.
64. T. Voigt, R. Fried, M. Backes, and W. Rhode, *Threshold optimization for classification in imbalanced data in a problem of gamma-ray astronomy*, Adv. Data Anal. Classification 8(2) (2014), 195–216.
65. B. X. Wang and N. Japkowicz, *Boosting support vector machines for imbalanced data sets*, Knowl. Inf. Syst. 25(1) (2010), 1–20.
66. W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, *Effective detection of sophisticated online banking fraud on extremely imbalanced data*, World Wide Web 16(4) (2013), 449–475.
67. S.-J. Yen and Y.-S. Lee, *Cluster-based under-sampling approaches for imbalanced data distributions*, Expert Syst. Appl. 36(3) (2009), 5718–5727.
68. S. Youn and D. McLeod, *A comparative study for email classification*, in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, Springer, Dordrecht, The Netherlands, 2007, 387–391.

69. B. Zadrozny, J. Langford, and N. Abe, *Cost-sensitive learning by cost-proportionate example weighting*, ICDM 3 (2003), 435.
70. C. Zhang, W. Gao, J. Song, and J. Jiang, *An imbalanced data classification algorithm of improved autoencoder neural network*, 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), IEEE, 2016, pp. 95–99.
71. Z. Zheng, X. Wu, and R. Srihari, *Feature selection for text categorization on imbalanced data*, ACM SIGKDD Explor. Newsl. 6(1) (2004), 80–89.
72. Z.-H. Zhou and X.-Y. Liu, *Training cost-sensitive neural networks with methods addressing the class imbalance problem*, IEEE Trans. Knowl. Data Eng. 18(1) (2005), 63–77.
73. Q. Zou, S. Xie, Z. Lin, M. Wu, and Y. Ju, *Finding the best classification threshold in imbalanced classification*, Big Data Res. 5 (2016), 2–8.

How to cite this article: Y. Feng, M. Zhou, and X. Tong, *Imbalanced classification: A paradigm-based review*, Stat. Anal. Data Min.: ASA Data Sci. J. **14** (2021), 383–406. https://doi.org/10.1002/sam.11538