Information Sciences 657 (2024) 119956

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Cost-sensitive thresholding over a two-dimensional decision region for fraud detection

Jorge C-Rella [a,b,*], Ricardo Cao [a], Juan M. Vilar [a]

[a] Department of Mathematics, Research Group MODES, CITIC, University of A Coruña, A Coruña, Spain
[b] Department of Risks, ABANCA Financial Services, A Coruña, Spain

ARTICLE INFO

Keywords:
Cost-sensitive classification
Instance-dependent classification
Thresholding
Fraud detection
Risk analysis
Decision region

ABSTRACT

Credit fraud poses a challenging task in terms of detection. It can result in significant losses depending on the amount, so a cost-sensitive perspective needs to be taken. Classical approaches focus on estimating the probability of fraud and selecting a decision threshold, but they often fail to consider the transaction amount or account for the cumulative losses incurred within the sample. Consequently, these approaches can result in sub-optimal strategies. A new thresholding approach is proposed, based on the construction of a two-dimensional decision space with an estimated probability and the credit amount. This expansion allows more freedom for the optimal classification rule search, which is performed with a new algorithm. The proposed method generalizes previous approaches, so an improvement is consistently achieved. In addition, it allows a restricted search. This is shown in a study of two real data sets, comparing the results obtained by a wide range of classifiers.

1. Introduction

Fraud is a significant and growing risk for financial institutions [1,2], defined as an operation that intentionally leads to the total loss of the financed credit. Consequently, all operations must undergo initial fraud screening before following the standard risk analysis and granting procedures. This incurs in operational costs and limitations that cannot be overlooked [3–5], for which a percentage of operations to analyze (POA) restriction of ideally less than 5% and no more than 10% is imposed in practical applications. Detecting fraud in the context of consumer finance presents several challenges. Fraudsters adapt to risk policies and often modify their information, leading to class overlap [6] and making it difficult to identify patterns when training supervised models. Moreover, fraudulent cases are scarce, resulting in an extraordinarily imbalanced problem [3]. Lastly, regulation constraints limit the use of complex models, such as neural networks, in consumer finance [7,1,8], as all the decisions that are made must be explainable to the regulator and the client.

Classification models are commonly trained in terms of statistical performance measures that do not take into account the actual business objective, which is to minimize financial losses due to fraud. Several authors have pointed out that decision making based only on an estimated probability has a worse performance in problems where not all error types have the same weight [1,9–11]. Note that it would be preferable to detect a 10,000 € fraud than five 1,000 € frauds. Consequently, fraud should be acknowledged

* Corresponding author.
E-mail addresses: jorge.crella@udc.es (J. C-Rella), ricardo.cao@udc.es (R. Cao), juan.vilar@udc.es (J.M. Vilar).

https://doi.org/10.1016/j.ins.2023.119956
Received 14 March 2023; Received in revised form 18 October 2023; Accepted 25 November 2023

Available online 29 November 2023

Table 1
Amount-dependent cost matrix.

	$\hat{y}_i = 0$	$\hat{y}_i = 1$
$y_i = 0$	$C_i^{TN} = 0$	$C_i^{FP} = a\xi_i + b$
$y_i = 1$	$C_i^{FN} = \xi_i$	$C_i^{TP} = b$

as an *instance-dependent cost-sensitive* problem [12]. This creates an additional degree of complexity, as costs depend not only on the class but also on characteristics of the instance (the loan amount).

The literature, although extensive, lacks state-of-the-art references due to the absence of publicly available datasets and consequent comparisons. To overcome this, a wide range of models and methods is introduced and tested in this paper. Among cost-insensitive approaches, undersampling and oversampling techniques [3,5,13,1], although appropriate given the problem imbalance, imply bias or an increase in variance respectively [14,5,15]. Classification techniques as support vector machines (SVM) [2], data envelopment analysis methods [8] or fuzzy models [16,17] do not consider the loan amount in the decision making, which is an important drawback. In addition, given the difficulties presented, more complex models are unlikely to achieve a better classification than state-of-the-art models [7,18,9,5,19]. Regarding cost-sensitive approaches, there are two distinguished philosophies. The first one, known as *predict-and-optimize*, consists on construct a classifier considering a cost-sensitive objective function, tuning the estimated probabilities [20,6,21,4] or using weighted versions of logistic regression [20,12] or boosting algorithms [9,12] among others. The second one, known as *predict-then-optimize* or *thresholding*, is to focus on the decision making. A predictive model is trained with the aim of maximizing accuracy and then decision-making is optimized minimizing losses [3,22,13,12,10,4,23]. The drawback of these techniques is that classification rules are individual, overlooking aggregated losses.

The method introduced in this paper belongs to the latter thresholding philosophy, which [10,13] found to be the more effective. It is constructed a novel decision space using the variables on which losses depend on: an estimated fraud probability and the credit amount. In this expanded space there is more freedom for the optimal decision rule search, which is estimated with a new proposed algorithm. It includes and expands all previous thresholding approaches, so an improvement is obtained. Furthermore, the restricted search is possible, something that any previous approach solves [4,11]. As the algorithm works on a given cost specification and an estimated probability, it can be generalized to any other cost-sensitive problem as churn prediction, credit risk, medical diagnosis or logistic planning.

The rest of the paper is as follows. Next section presents the cost-sensitive classification problem to be addressed. Section 3 introduces state-of-the-art approaches for the fraud probability estimation. In Section 4, the available thresholding strategies are listed, emphasizing their drawbacks and motivating the proposed methodology, which is explained in Section 5. Finally, Section 6 studies the performance of the different combinations of classifiers and thresholding strategies over two real data sets. One was provided by a collaborator financial company (Bank data set) and the other one is a wide-used open fraud data set (Credit Card data set). Conclusions and future extensions are included in Section 7.

2. Cost-sensitive classification

Cost-sensitive classification address the prediction of a binary dependent variable $Y \in \{0, 1\}$ (0 indicating legitimate and 1 fraud) from a set of independent variables $\mathbf{X} = (X_1, \dots, X_p)$ taking into account costs of prediction error and potentially other costs. The objective is loss reduction, so model performance must be evaluated considering classification error costs, which depends on the estimated probability, $\hat{p}(\mathbf{x})$, of the conditional probability, $p(\mathbf{x}) = P(Y = 1 \mid \mathbf{X} = \mathbf{x})$, the credit amount, ξ, and the thresholding strategy. When only the estimated probability is considered for decision-making, prediction is defined by a cut-off point t as $\hat{Y} = \mathbb{I}(\hat{p}(\mathbf{x}) > t)$.

Classical approaches consider a cost matrix based on the true class Y and the predicted class $\hat{Y}$. These approaches assume that all errors of the same type have equal costs [13,9], which means a clear overlook of information [20,22,13,24]. To address this issue we consider a *loss function* constructed from an instance-dependent cost matrix as the one defined in Table 1, generalized from [3,22,13,11]. Costs are assumed independent of the covariable vector $\mathbf{X}$ in line with [24] and are motivated by the specific fraud problem raised by our collaborator financial entity.

In Table 1, C_i^{FN} encloses the cost of an undetected fraud, i.e. the total credit amount ξ_i. The lost benefit when classifying a legitimate client as a fraudster is summarized in C_i^{FP}. It incorporates the proportion of clients who forgo financing for doubting them, a_1, and the mean gain per operation, $a_2\xi_i$, with $a = a_1 a_2$. The fixed cost of investigating the operation, b, is included both in C_i^{FP} and C_i^{TP}, i.e. whenever $\hat{y}_i = 1$. Gains could be introduced, but they do not appear as there is only the possibility of loss when dealing with fraud. From Table 1, the loss function is defined as:

$$\ell(\hat{y}_i, \xi_i, y_i) = (1 - y_i)(1 - \hat{y}_i)C_i^{TN} + (1 - y_i)\hat{y}_i C_i^{FP} + y_i(1 - \hat{y}_i)C_i^{FN} + y_i\hat{y}_i C_i^{TP} \tag{1}$$

In this paper the performance metric considered is *savings*, with an spread use in the literature [3,20,22]. For a sample $(\hat{y}_i, \xi_i, y_i)_{i=1}^n$ it is expressed as:

$$\text{Savings} = 1 - \frac{\sum_{i=1}^n \ell(\hat{y}_i, \xi_i, y_i)}{\sum_{i=1}^n y_i \xi_i} \tag{2}$$

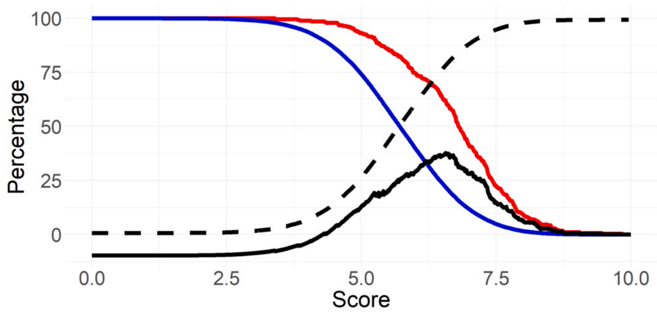

Fig. 1. Accuracy (dashed black), recall (red), POA (blue) and savings (solid black) considering different cut-off decision points over the score.

where the denominator is the total loss faced if no preventive action is taken in order to have a base reference [11]. The objective is minimization of $\sum_{i=1}^{n} \ell(\hat{y}_i, \xi_i, y_i)$, equivalently maximization of (2), along possible classifiers.

In order to show the importance of considering a cost-sensitive metric, for the Bank data set, a logistic model is fitted. Its POA $= \sum_{i=1}^{n} \hat{y}_i / n$, accuracy, recall and savings (2) are shown in Fig. 1 for different decision thresholds over the estimated probability, where it can be seen the nonlinear relationship between them. "Score" is referred as the escalation of the estimated fraud probability between 0 and 10, for the sake of confidentiality, and metrics are represented as percentages. This notation is followed throughout the paper. It can be seen that detecting more frauds not necessarily leads to an increase in savings (2) due to analysis costs. Also, the extreme class imbalance biases accuracy, as the greatest is achieved labeling all operations as legitimate. Consequently, the fraud detection problem should be addressed from a cost sensitive perspective, both conceptually and in order to obtain better practical results.

3. Classification methods

Probability estimation methods are presented in this section, so the thresholding strategies introduced in next section can be applied afterwards. There are introduced different non and cost-sensitive approaches, so in the practical application it could be tested if the latter help in the posterior thresholding, in line with [10]. Boosting approaches are introduced so as to obtain a measure of the predictive capacity that can be reached by considering a more complicated model. Other complex methods as bagging and SVM are not considered since, based on simplicity, flexibility and performance, boosting has shown to have a better behavior in practical applications [2,9]. In addition, they cannot be used in practice due to the interpretability restraints aforementioned, so just boosting seems to be enough as a benchmark.

3.1. Logistic regression

In practice, fraud detection is often addressed as a mere classification problem. Logistic regression is the *de facto* model in credit risk, modeling the probability as:

$$P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = p_\theta(\mathbf{x}_i) = \frac{1}{1 + \exp^{-(\beta_0 + \beta' \mathbf{x})}} \tag{3}$$

The problem becomes estimating the parameter $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta})$ that maximizes the log-likelihood function in equation (4) below with $w_i = 1/n$ for $i = 1, \cdots, n$. It assigns the same weight to both classification error types, which is not the case in many real applications as fraud detection.

3.2. Weighted logistic regression

In order to enhance and adapt logistic regression to the cost-sensitive setting, weights are introduced in the log-likelihood function [20,6,25]:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{n} w_i \left[y_i \log(p_\theta(\mathbf{x}_i)) + (1 - y_i) \log(1 - p_\theta(\mathbf{x}_i)) \right] \tag{4}$$

where $p_\theta(\mathbf{x})$ is as defined in (3) and w_i is the weight of the i-th instance. The probability modeling is the same as in equation (3), but it is considered another objective function. This affects the model parameter estimation, and therefore the classification. Weights are introduced either for balancing the data set [25] or for putting more emphasis on an operation depending on its amount [20], which is expected to improve classification error costs.

3.3. Boosting algorithms

Boosting [26] is an ensemble learning algorithm which outputs the estimated probability as a sum of T weak classifiers, whose performance is slightly better than random guessing. In this paper, shallow binary decision trees are considered as weak classifiers. Given a sample $(y_i, \mathbf{x}_i)_{i=1}^n$, a tree m_1 with Q terminal leafs is fitted depending on $\mathbf{x}$. This tree has associated a weight vector $\omega^{(1)} \in \mathbb{R}^Q$, which assigns the weight $\omega_q^{(1)}$ to each terminal node $q \in \{1, \ldots, Q\}$. Next, a tree m_2 with weights $\omega^{(2)}$ is fitted in order to improve the previous prediction. This is iterated T times and the estimated conditional fraud probability is computed as:

$$\hat{p}(\mathbf{x}) = \sum_{t=1}^{T} m_t(\mathbf{x}) \tag{5}$$

Adaboost and XGBoost are two of the most outstanding boosting algorithms, whose main difference is the regularization function present in XGBoost. They are introduced below.

3.3.1. AdaBoost

In Adaboost [9,27], each weak classifier is trained over a new weighted data set which gives more weight to misclassified observations in previous models. Thus, on each iteration, focus is put on refining the classification of all points as the algorithm progresses. Given a sample $(y_i, \mathbf{x}_i)_{i=1}^n$, weights for each observation i are equally initialized $v_i^1 = 1/n$ and a weak classifier, m_1 fitted maximizing accuracy, acc_1. On each successive step t, m_t is fitted with updated weights $v_i^t \propto v_i^{t-1} e^{-y_i \alpha_t m_{t-1}(\mathbf{x}_i)}$, where $\alpha_t = \log(acc_{t-1}/(1 - acc_{t-1}))$.

3.3.2. XGBoost

Extreme Gradient Boosting (XGBoost) [28,9] is a boosting algorithm which trains the model considering the objective function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \ell(\hat{p}(\mathbf{x}_i), y_i) + \sum_{t=1}^{T} \Omega(m_t) \tag{6}$$

where ℓ is a differentiable convex objective function, as the log-likelihood in equation (4) with $w_i = 1/n$ for $i = 1, \ldots, n$, $\Omega(m_t) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2$ and $\lambda, \gamma \geq 0$ constants penalizing the model complexity. The regularization function aims to choose a model that uses simple functions by penalizing the complexity of each tree, which prevents over fitting. In each step, a tree is trained to improve the previous prediction. For a prediction, $\hat{p}(\mathbf{x}_i)^{(t-1)}$, of the i-th observation at the $(t-1)$-th iteration, m_t is trained to minimize:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell(\hat{p}(\mathbf{x}_i)^{(t-1)} + m_t(\mathbf{x}_i), y_i) + \Omega(m_t)$$

with a gradient descent algorithm. This process is iterated T times and the final prediction computed as in (5).

3.3.3. LightGBM

Light gradient-boosting machine (LightGBM) [29] is another boosting algorithm similar to XGBoost. The main difference lies in the way it constructs trees. XGBoost grows a tree level by level. Instead, LightGBM grows trees leaf-wise, so in each step it selects the leaf that is expected to improve more the objective function and the tree grows from this leaf. In addition, the split point for the decision trees is selected with a histogram-based approach, yielding a greater efficiency. It implements more refinements to run faster than previous approaches maintaining a high level of accuracy, which is the principal advantage of this algorithm.

3.4. Instance-dependent cost-sensitive logistic regression

In this approach, introduced in [12] as *cslogit*, the novelty is that the estimation of the parameters is carried out in a cost-sensitive manner, minimizing the average expected cost (AEC) of a given loss function (1):

$$AEC(\boldsymbol{\theta}; \mathbf{x}_i, \xi_i, y_i) = \frac{\sum_{i=1}^{n} \ell(p_{\boldsymbol{\theta}}(\mathbf{x}_i), \xi_i, Y_i)}{n} \tag{7}$$

where $p_{\boldsymbol{\theta}}(\mathbf{x}_i)$ is as in Equation (3). Then, $\boldsymbol{\theta}$ is estimated as the minimizer of (7), which can be found using a gradient descent algorithm. By incorporating the AEC into the model fitting, there is a higher probability of detecting high-amount frauds, leading to an improvement in terms of savings.

3.5. Instance-dependent cost-sensitive boosting

Introduced in [12], this method is constructed as the XGBoost model introduced in Section 3.3.2 but optimizing AEC, instead of the accuracy. In order to do this, $\ell(\mathbf{x}_i, \xi_i, Y_i) = AEC(\boldsymbol{\theta}; \mathbf{x}_i, \xi_i, Y_i)$ is considered in the objective function (6). As the AEC is introduced in the model fitting, it is more likely to rely on high-amount frauds and therefore obtain greater savings.

4. Previous thresholding approaches

All classifiers produce probability estimates. Thresholding selects a decision strategy based on misclassification cost, converting cost-insensitive learning algorithms into cost-sensitive ones [23,10]. Most of the approaches rely on the estimated probability, for which calibrated probabilities are needed [15]. This is an added degree of complexity, as these are not always easy to obtain, specially in an unstable setting as fraud detection. State-of-the-art approaches are introduced in this section and represented in Fig. 2. This figure shows their difference and shortfalls. For the sake of confidentiality, the logarithm of the amount, rescaled into the interval $[0, 10]$, is shown. This notation is followed throughout the paper.

4.1. Youden's J statistic

Youden's J statistic [30] is often used in dichotomous decision problems, taking its maximizer as an optimal classification threshold. It is defined as:

$$J = \text{recall} + \text{specifity} - 1 = \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sum_{i=1}^{n} y_i} + \frac{\sum_{i=1}^{n}(1-y_i)(1-\hat{y}_i)}{\sum_{i=1}^{n}(1-y_i)} - 1 \tag{8}$$

where $\hat{y}_i = \mathbb{I}(\hat{p}(\mathbf{x}_i) > t)$. The cut-off point, t, maximizing J (8) minimizes the false positive and false negative rates. Nevertheless, as shown in Section 2, this could be sub optimal when the objective is loss reduction.

4.2. Brute force threshold

In order to scrutinize the best strategy considering only the estimated fraud probability, an empirical exhaustive search is considered. A grid is constructed dividing the one-dimensional decision space in 1,000 equally spaced intervals. Savings obtained when considering each cut-off point t in the grid is computed. The one that produces the maximum is taken as classification threshold [23]. Since the resulting savings are computed for each cut-off, the restricted search can be implemented as well, considering only the thresholds satisfying the POA restriction.

4.3. Bayes minimum risk

Bayes minimum risk (BMR) approach [22,13] corresponds to the theoretical optimal cost-sensitive decision rule. Considering the exogenous variable, ξ, and an estimated probability, $\hat{p}(\mathbf{x})$, the *risk* of a data point is defined:

$$R(\hat{y}, \xi \mid \mathbf{x}) = \ell(\hat{y}, \xi, 0)(1 - \hat{p}(\mathbf{x})) + \ell(\hat{y}, \xi, 1)\hat{p}(\mathbf{x})$$

where $\hat{y} \in \{0, 1\}$ and ℓ is a loss function as in (1). Then, an operation is labeled as fraud if $R(1, \xi \mid \mathbf{x}) \leq R(0, \xi \mid \mathbf{x})$, i.e. if the risk of classifying it as a fraud is lower than as legitimate. This leads to the decision rule $\hat{y}_i = \mathbb{I}(\hat{p}(\mathbf{x}_i) > t_i)$ with:

$$t_i = \frac{C_i^{FP} - C_i^{TN}}{C_i^{FP} - C_i^{TN} + C_i^{FN} - C_i^{TP}} = \frac{a\xi_i + b}{(1+a)\xi_i} \tag{9}$$

Note that although this approach is theoretically optimal, it could not be the case when considering the aggregated sample results [12]. For example, for $b = 10\$$, $a_1 = 0.05$ and $a_2 = 0.08$, a data point with $\xi_i = 300\$$ will be analyzed for fraud if $\hat{p}(\mathbf{x}_i) \geq 0.037$. Suppose that there are 40 operations, of which one is a fraud, with $\hat{p}(\mathbf{x}_i) = 0.04$ and $\xi_i = 300\$$, likely to occur in an umbalanced problem as fraud detection. The aggregated costs would be $40 \cdot 10\$$ versus a 300\$ fraud. Thus, a global strategy is more likely to produce better practical results. Besides, the frontier defined by (9) does not allow any flexibility in order to adapt the decision region, so the method cannot be used in practice as it does not fulfill the imposed POA restrictions.

4.4. Fixed cost matrix

Classical approaches consider a fixed value in the four entries of the cost matrix introduced in Table 1 for all i. The optimal threshold in terms of missclassification costs is the same as the one defined in Equation (9), but with a fixed cut-off point for every instance [13]. Thus, it can be considered as the fixed-threshold version of the BMR approach. Considering the mean of the instance-dependent cost matrix as proposed in [10], the optimal decision cut-off point becomes $t = \frac{1}{n}\sum_{i=1}^{n} t_i$, whit t_i as in Equation (9).

5. Two-dimensional thresholding

In order to overcome the limitations of the thresholding approaches presented in Section 4, we propose expanding the decision space to a two-dimensional map generated by the estimated probability, $\hat{p}(\mathbf{x})$, and the loan amount, ξ. In this space a more flexible and effective decision region can be explored, which is performed with a new proposed algorithm.

For numerical optimization, given a sample $\{(\hat{p}(\mathbf{x}_i), \xi_i)\}_{i=1}^{n}$, a grid, $G(k)$, is defined depending on a k parameter which drives the search smoothness:

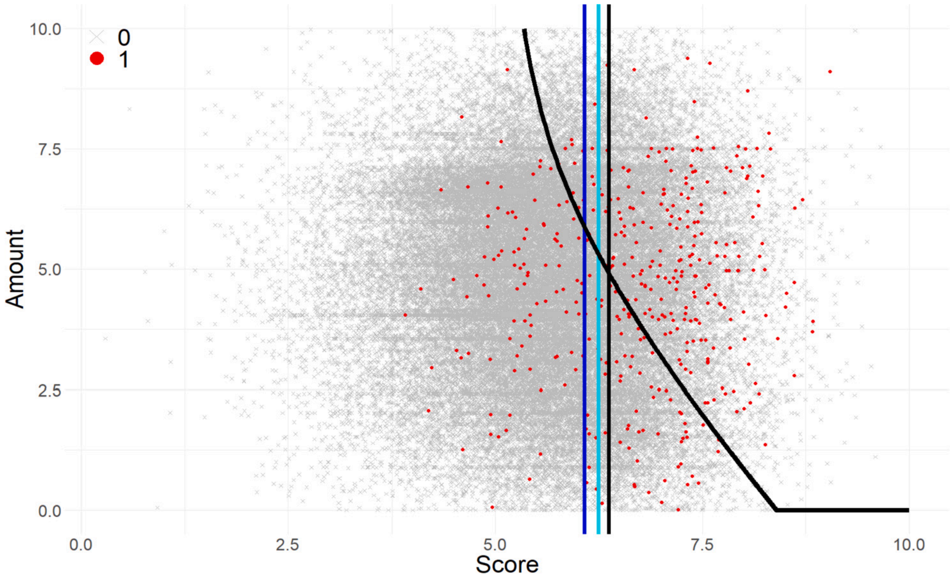

Fig. 2. State-of-the-art thresholding approaches: Youden's J Statistic (cyan), brute force (black), BMR (black parabola), fixed cost matrix threshold (blue).

$$G(k) = \left\{ \left(\hat{p}_{\min} + s\delta_1, \xi_{\min} + t\delta_2 \right) \right\}_{s,t=0}^{k-1} \tag{10}$$

where

$$\begin{aligned} \delta_1 &= (\hat{p}_{\max} - \hat{p}_{\min})/k \\ \delta_2 &= (\xi_{\max} - \xi_{\min})/k \\ \hat{p}_{\min} = \min\{\hat{p}(\mathbf{x}_i)\}_{i=1}^{n}&, \ \hat{p}_{\max} = \max\{\hat{p}(\mathbf{x}_i)\}_{i=1}^{n}, \\ \xi_{\min} = \min\{\xi_i\}_{i=1}^{n}&, \ \xi_{\max} = \max\{\xi_i\}_{i=1}^{n} \end{aligned} \tag{11}$$

The grid $G(k)$ could be based on the quantiles of orders $0/k, 1/k, \dots (k-1)/k$ of the empirical distribution in each of the dimensions. This can speed up the search process when the distributions are skewed, omitting regions of the space with low density.

When considering a two-dimensional decision space, a simple generalization is to take a threshold in each dimension. Hence the decision region consists of an upper right quadrant $Q_{\mathbf{r}} = \left\{ (\hat{p}, \xi) \in \mathbb{R}^2 \mid \hat{p} > r_1, \xi > r_2 \right\}$ defined by a point $\mathbf{r} = (r_1, r_2)$. Considering a set of points, $R = \left\{ \mathbf{r}_j \right\}_{j=1}^{m}$, a decision region is constructed as the union of the upper right quadrants $Q_{\mathbf{r}_j}$,

$$D(R) = \bigcup_{j=1}^{m} Q_{\mathbf{r}_j} \tag{12}$$

Given an instance $(\hat{p}_i, \xi_i)$ and a decision region $D(R)$ as (12), its labeling is defined as $\hat{y}_i^R = \mathbb{I}\left((\hat{p}_i, \xi_i) \in D(R) \right)$. For a sample $\mathcal{X} = \left\{ (\hat{p}_i, \xi_i, y_i) \right\}_{i=1}^{n}$, savings is defined as:

$$\mathcal{S}(R \mid \mathcal{X}) = 1 - \frac{\sum_{i=1}^{n} \ell(\hat{y}_i^R, \xi_i, y_i)}{\sum_{i=1}^{n} y_i \xi_i} \tag{13}$$

Algorithm 2-DDR(k) (2-dimensional decision region algorithm depending on the parameter k) is proposed for the optimal decision-making estimation. It starts with a decision region defined by the most northeast point of the grid $G(k)$, the one with highest estimated fraud probability and amount. In a recursively manner, each of the points of $G(k)$ surrounding the current decision region is added to the current region as in (12) and savings computed as in (13). The point whose inclusion produces the greatest savings increase is added. If there is no savings improvement with respect to the previous decision region, the next surrounding points of $G(k)$ are explored. The algorithm stops when the minimum in the data support is reached. An example of the first iterations is shown in Fig. 3. Starting with a preliminary decision region, savings is calculated considering the surrounding points of the grid. As no improvement is obtained, the next surrounding vertices are explored. This time, an improvement is obtained, so it is updated.

The algorithm works on a given estimated probability/score and loss function as the one introduced in Equation (1). Consequently, it can be generalized to any cost-sensitive problem just considering other loss function. The search is performed over all the space, so if the optimal decision has the shape of one of the thresholding proposals introduced in Section 4, it will be found except for some roughness depending on the k parameter. Thus, Algorithm 2-DDR(k) is expected to improve (or at least reproduce) state-of-the-art thresholding approaches in terms of savings. Calibrated probabilities are not needed as the proposed approach only rely on the points ordering, reducing the degree of complexity of the problem. In addition, it permits exploring restricted decision rules as well, just

Algorithm 2-DDR(k) Two-dimensional decision region algorithm.

Data $\mathcal{X} = \{(\hat{p}_i, \xi_i, y_i)\}_{i=1}^{n}$
Input k parameter
Compute
  Steps δ_1 and δ_2 as (11) and the grid $G(k)$ as defined in (10);
  $R := (\hat{p}_{\max}, \xi_{\max})$;
$F \leftarrow R$
while $\min(F) \neq (\hat{p}_{\min}, \xi_{\min})$ **do**
  $R_{old} \leftarrow R$
  $s \leftarrow \mathcal{S}(R_{old} \mid \mathcal{X})$, as defined in (13)
  $t \leftarrow 1$
  while $R = R_{old}$ **do**
    $F :=$ subset of $G(k)$ at distance $t\delta_1$ in the first dimension or $t\delta_2$ in the second dimension from $D(R)$
    $\mathbf{f}_m = \arg\max_{f \in F} \{\mathcal{S}(R \cup \{f\} \mid \mathcal{X})\}$
    if $\mathcal{S}(R \cup \{\mathbf{f}_m\} \mid \mathcal{X}) > s$ **then**
      $R \leftarrow R \cup \{\mathbf{f}_m\}$
    end if
    $t \leftarrow t + 1$
  end while
end while
Output A decision region defined as in (12)

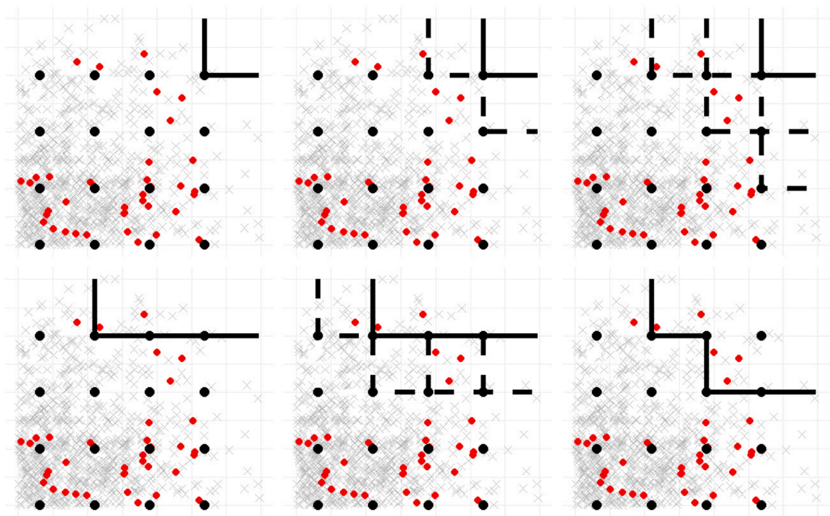

Fig. 3. Algorithm 2-DDR representation along the grid $G(k)$ (black dots), evaluated decision region (dashed lines) and the updated decision region (solid lines).

iterating until the POA restriction (e.g. 10% or 5%) is met. Thus, the optimal decision rule can be estimated in any cost-sensitive problem. Resulting regions for the Bank data set are plotted in Fig. 4. The algorithm, as it would be expected, focuses on high fraud probability and amount operations. Besides, it avoids areas with high legitimate points density, where the analysis cost do not compensate the fraudulent amount detected when considering aggregated costs. The strength of the algorithm is that this intuitive logic is developed automatically without any need of additional estimation neither parameter tuning.

Algorithm 2-DDR(k) depends only on a parameter, k, that determines how thorough the search is. A greater k implies a finer grid, giving more flexibility to the search, which results in greater savings. In practice we would suggest to start with a small value of k and try larger values until a plateau in savings in a validation set is reached. In our experience, this is achieved with k smaller than the number of cases. Regarding the computational complexity, it depends mainly on k^2. This makes the algorithm suitable for scalation to biggest data sets as it does not depend on the sample size.

Lastly, note that, by construction, the algorithm is stable and robust against outliers. Fig. 5 (left) shows the estimated frontiers for 5 simulated data from the same model, with $40,000$ instances and a 0.5% fraud percentage. It can be seen that there are slight differences depending on the sample, but the frontiers have an stable shape. Fig. 5 (right) shows the estimated decision region when considering four types of possible fraudulent outliers, situating 20 of each type in the score-amount space. Only frauds with high amount will be susceptible of influencing the algorithm. If they have a high estimated probability of being fraud as well, they are included in the first steps in the decision region and the same frontier is obtained. When having a low estimated probability, the same optimal decision region is obtained but including these outliers (as they imply a significant loss), which would not have any consequences in practice.

This proposal, Algorithm 2-DDR(k), belongs to the class of greedy algorithms. The algorithm iteratively makes an optimally local choice after another, reducing the problem into a smaller one. There is no a universal theoretical result that ensures optimality

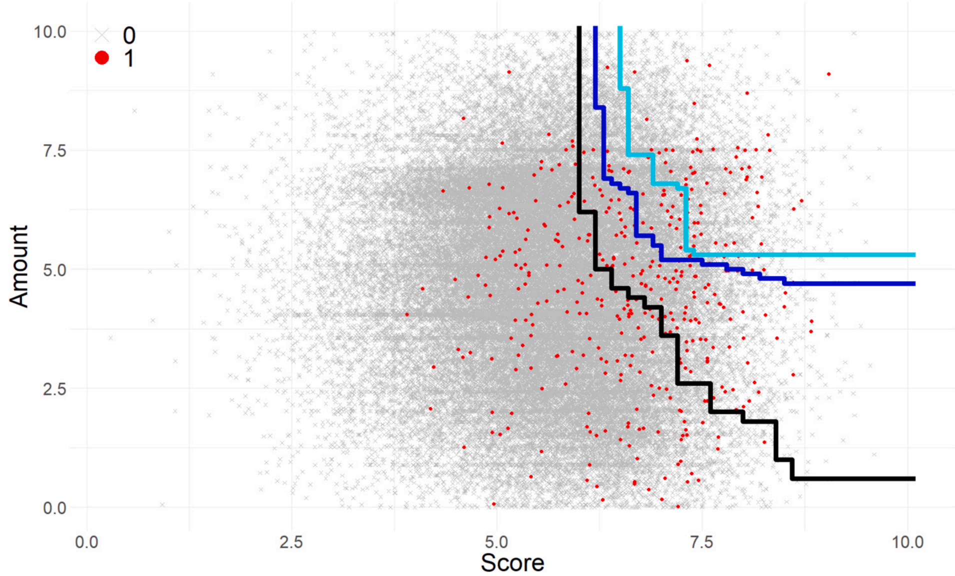

Fig. 4. Optimal decision frontiers in terms of savings estimated running Algorithm 2-DDR(k) with $k = 100$, fulfilling that POA $\leq 100\%$ (black), $\leq 10\%$ (blue), $\leq 5\%$ (cyan).

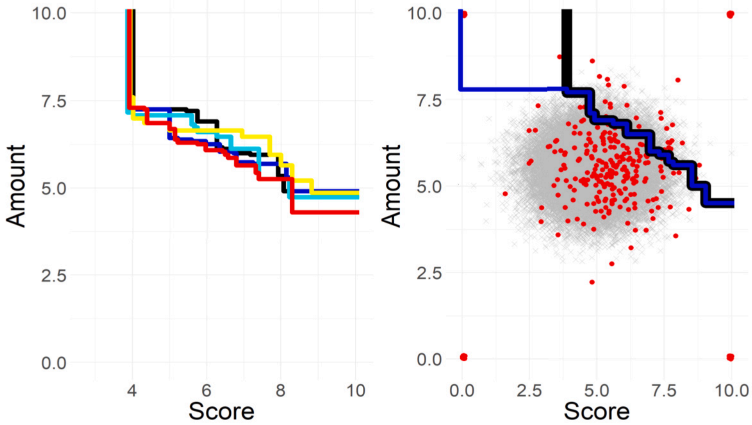

Fig. 5. Estimated decision frontiers running Algorithm 2-DDR over 5 simulated data from the same model (left) and over a sample with fraudulent outliers situated in the four corners of the graph (right), when not considering the outliers (black) and when considering the outliers (blue).

of greedy algorithms. Huffman trees for discrete/categorical variable simulation, Dijkstra's algorithm for shortest path finding and Kruskal algorithm for finding minimum spanning trees are examples of optimal greedy algorithms. Although, there exist other well-known examples of greedy algorithms (as the nearest unvisited city algorithm for the traveling salesman problem) that may perform much worse in pathological examples. Just to examine the practical performance of Algorithm 2-DDR(k), some experimental scenarios are considered in Section 6.2.

6. Experiments

Two real data sets are presented to evaluate the performance of all introduced approaches. Both consists in real fraud data sets, according to which they exhibit the difficulties presented in Section 1 as extreme imbalance and class overlap. To assess the classifiers performance independently from the thresholding strategy, we rely on threshold-independent metrics, namely AUC, Gini index, the Kolmogorov-Smirnov statistic (KS) and H-measure (H). The latter may be more informative given the high degree of class imbalance [10]. Expected savings (ES) $= 1 - \frac{\sum_{i=1}^{n} \ell(\hat{p}_i,\xi_i,Y_i)}{\sum_{i=1}^{n} Y_i\xi_i}$, inspired from (7), are summarized as well. To evaluate the impact of the threshold strategy, it is reported the accuracy (Acc), recall (Rec), specifity (Spec) and F-score (F) along with the objective function, savings (Sav) defined in (2).

The classifiers introduced in Section 3 are trained over the two data sets. These are logistic regression (LR), weighted logistic regression (WLR), Adaboost (AB), XGBoost (XGB), LightGBM (LGBM), cslogit (CSL) and csboost (CSB). Thresholding methods in-

Table 2
Summary of the second data set variables.

Variable	Type	IV
Commerce activity	Categorical	0.227
Client activity sector	Categorical	0.162
Housing situation	Categorical	0.152
Marital status	Categorical	0.136
Profession	Categorical	0.112
Region	Categorical	0.108
Commerce class	Categorical	0.032
Previous request indicator	Categorical	0.007
Commerce monthly amount	Continuous	0.007
Age	Continuous	0.004
Commerce mean default rate	Continuous	0.001
Loan term	Continuous	0.001

troduced in Section 4 and the algorithm proposed in Section 5 with $k = 25, 50, 100$ are applied over the estimated probabilities, previously calibrated following [13,15]. Thus, the thresholding approaches that rely on probabilities are not distorted. For each data set, a 5-fold cross validation is performed, stratified by the proportion of frauds and the amount following [12]. The mean results over the test sets are summarized in Section 6.2.

6.1. Data sets

Available at kaggle.com/mlg-ulb/creditcardfraud [5,12], the Credit Card data set consists of 284,807 credit card transactions made in two days, where there are 492 (0.17%) frauds. It consists in 28 variables resulting from a PCA along with a “Time” variable (seconds elapsed from the first use) and the “Amount” of each transaction. The “Class” variable indicates if an operation is legitimate (0) or fraudulent (1).

The Bank data set is a real data set of 210,180 loan requests lend by a collaborator financial entity, collected between January 2018 and December 2021 with a 0.67% fraud percentage. In order to preserve confidentiality, the number of registers is truncated and so the fraud proportion. The variables considered are summarized in Table 2 in terms of their information value (a measure of the relation between a variable and the odds ratio [31]). These are limited to information provided at the request time. There is no history of behavior or extensive databases to draw on, as is the case for credit cards, implying another handicap for modeling fraud. Only formalized requests are considered, because nothing can be assured about a non-formalized operation. Note that these are the operations of interest, and most difficult to detect, as they are the ones that passed all the filters and controls.

6.2. Results

Classifiers results obtained in the Credit Card data set over the test samples are summarized in Table 3. Prediction performance of all classifiers is outstanding in terms of classification as it can be seen in the high values of the AUC. This occurs thanks to the number of variables available and the discrimination power of these. As expected, the greatest ES is obtained with a cost-sensitive model (CSB). The flexibility offered by boosting combined with a cost-sensitive approach offers a model able to detect high-amount frauds, which leads to the highest ES. Table 5 summarizes thresholding approaches performance for the different classifiers. In this case the restricted search is not considered as it is a public data set without any imposed restriction. The first highlight is that detecting more frauds does not imply an increase in savings, as can be seen with JS. As cost are not considered in the decision making, the highest frauds are not detected and there is an increase in POA, leading to high analysis costs and consequently smaller savings. This is a clear example of the importance of the correct selection of operations to inspect in an amount-dependent problem. Csboost and LightGBM obtain the best metrics in the train set, but a much lower performance in the test set, due to a clear overfitting that could become dangerous in practice. This can be seen with all boosting approaches. In the test set, the best results in terms of savings are achieved with significantly smaller POA with the new proposed approach: Algorithm 2-DDR(k). It focuses on detecting high amount frauds, so it obtains the smaller POA with every classifier as well as the highest savings, for which is considered the outperforming approach. The smoothness effect of the parameter k is clear, with a direct relation with savings obtained in the train set and certain overfit in some classifiers. We highlight the cslogit approach, outperforming all other existing methods in the test set with an understandable model and low POA. Thus, for this dataset, cslogit with Algorithm 2-DDR(k) is the selected approach.

Regarding the Bank data set, classifiers results are summarized in Table 4. Cost-sensitive approaches outperform classical ones again in terms of ES, as expected. This setting is more difficult, which is reflected in the smaller values in Table 4 compared to the previous data set. Table 6 summarizes different thresholding strategies results. Csboost gives a very good performance, probably due to the scarcity of variables that make necessary a more complex modeling. Nevertheless, it seems to be falling into overfitting again, leading to improvable results in the test samples. Algorithm 2-DDR(k) outputs the best results under each classifier in terms of savings (2) again. It is also worth noting the fact that Algorithm 2-DDR(k) outperforms all classical approaches in terms of classification metrics except recall, making clear how the algorithm focus on the more profitable operations to analyze. As a consequence, POA tends to be significantly smaller, which is another advantage. Highest savings are obtained with Adaboost and Algorithm 2-DDR(k)

Table 3
Classification metrics for the different classifiers over the Credit Card data set.

Classifier	AUC	Gini	H	KS	ES
LR	97.56	95.13	87.13	88.99	44.12
WLR	96.81	93.62	85.45	87.73	42.80
AB	97.57	95.15	87.62	89.89	68.13
XGB	97.34	94.68	87.29	89.19	85.97
LGBM	97.62	95.23	82.63	86.87	49.50
CSL	93.85	87.70	81.13	83.54	67.51
CSB	95.45	90.91	82.22	86.05	92.49

Table 4
Classification metrics for the different classifiers over the Bank data set.

Classifier	AUC	Gini	H	KS	ES
LR	76.11	52.22	21.64	41.15	0.99
WLR	76.11	52.22	21.77	41.47	0.98
AB	77.06	54.12	22.63	42.48	1.14
XGB	77.13	54.27	23.59	42.75	2.56
LGBM	77.18	54.35	26.81	43.88	1.20
CSL	72.34	44.66	15.15	37.15	47.11
CSB	77.00	54.00	23.88	42.88	67.13

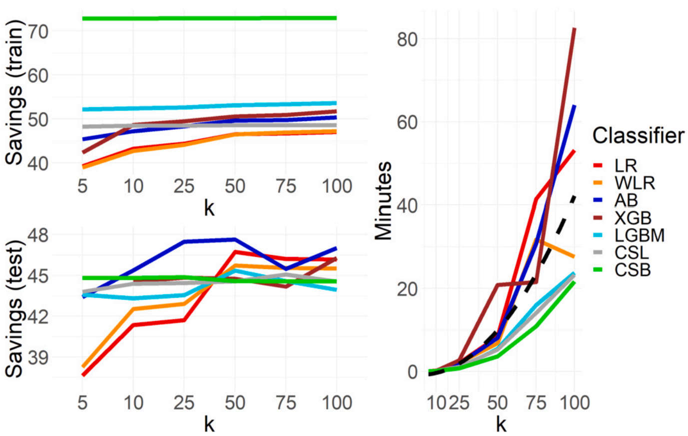

Fig. 6. Mean results summary in the Bank data set for the combination of all the classifiers introduced throughout the paper and the proposed algorithm with different values of k. Left graphs show the mean savings for the train and datasets respectively and the right graph the mean computational times in minutes, with a quadratic curve represented with a dashed black line superimposed.

in the test samples, but almost the same savings can be obtained considering logistic regression, a simpler model. This shows to what extent the results can be improved considering the proposed algorithm.

The restricted search is performed as it was required by our collaborator financial entity. POA of 10% and 5% are considered in Table 7. The results are parallel to the ones obtained in the unrestricted case. It is worth mentioning the 40.25% of savings achieved adjusting to the 10% POA restriction compared to the 47.65% of savings obtained with the best unrestricted approach. In this case, the best results are obtained with CSB. Thus, a complex model combined with Algorithm 2-DDR(k) produces satisfactory results, thanks to the flexibility of the model to discriminate frauds and the flexibility of Algorithm 2-DDR(k) to detect high-amount frauds. Likewise, similar results are obtained considering LightGBM, a complex cost-insensitive model, but with the proposed approach costs are taken into account in the decision-making. Nevertheless, taking into account the explainability restriction in the financial context, the finally selected model is the logistic regression along with Algorithm 2-DDR(k). Thus the bank can have a model that satisfies the interpretability restrains and its own workload limits obtaining a 33% reduction in losses due to fraud.

Undertaking a performance study for the proposed algorithm is fundamental in assessing its viability and practical utility. To better understand the effect of the parameter k and justify its choice, a sensitivity analysis of k is presented. The results obtained for the Bank data set are summarized in Fig. 6. Regarding the results in terms of savings, the plots at the left show the results obtained over the previous introduced data sets for different values of the parameter k. The upper-left plot in Fig. 6 shows the direct relation between k and savings in the train sets. This is due to the additional flexibility, but the performance clearly achieves a “horizontal asymptote” with all models. In the test set (bottom-left graph in Fig. 6) for the simpler models a larger value of k is needed, obtaining

Table 5
Mean results summary in the Credit Card data set for the combination of all the approaches introduced throughout the paper.

		TRAIN set						TEST set					
Classifier	Threshold	Sav	Acc	Rec	F	Spec	POA	Sav	Acc	Rec	F	Spec	POA
LR	JS	-5.43	98.33	**88.98**	99.15	98.34	1.80	-4.48	98.32	**89.25**	99.15	98.34	1.81
	BF	67.65	**99.93**	80.91	99.96	99.96	0.18	63.07	**99.91**	78.49	99.95	99.95	0.18
	CM	66.73	**99.93**	77.96	**99.97**	99.97	0.16	48.21	**99.91**	72.04	**99.96**	99.96	0.16
	BMR	68.80	99.86	38.98	99.93	99.96	0.10	65.98	99.85	36.56	99.92	99.95	0.11
	2-DDR(25)	70.78	99.88	40.05	99.94	**99.98**	0.09	**67.76**	99.87	35.48	99.93	**99.97**	**0.08**
	2-DDR(50)	71.16	99.88	39.78	99.94	**99.98**	**0.08**	66.12	99.86	34.41	99.93	**99.97**	**0.08**
	2-DDR(100)	**73.30**	99.86	42.20	99.93	99.96	0.11	63.94	99.83	36.56	99.91	99.93	0.13
WLR	JS	17.55	98.78	**87.63**	99.38	98.80	1.35	16.91	98.77	**88.17**	99.38	98.79	1.36
	BF	68.35	99.86	81.99	99.93	99.89	0.24	49.09	99.83	79.57	99.92	99.87	0.26
	CM	66.86	**99.92**	72.58	**99.96**	99.97	0.15	49.00	**99.91**	67.74	**99.95**	99.96	0.15
	BMR	72.52	99.85	39.78	99.93	99.95	0.11	**65.46**	99.83	36.56	99.91	99.93	0.13
	2-DDR(25)	70.68	99.88	38.17	99.94	**99.98**	**0.08**	51.58	99.86	32.26	99.93	**99.98**	**0.08**
	2-DDR(50)	72.42	99.87	38.71	99.94	99.97	0.09	50.34	99.84	32.26	99.92	99.95	0.10
	2-DDR(100)	**73.07**	99.86	38.98	99.93	99.96	0.11	62.66	99.83	33.33	99.91	99.94	0.12
AB	JS	-139.44	95.60	**93.82**	97.75	95.61	4.54	-116.50	95.72	**93.55**	97.81	95.73	4.42
	BF	64.07	**99.93**	73.12	**99.97**	**99.98**	0.14	55.78	**99.92**	72.04	**99.96**	99.96	0.16
	CM	62.68	99.92	77.42	99.96	99.95	0.17	55.36	**99.92**	75.27	**99.96**	99.96	0.17
	BMR	64.34	99.83	41.40	99.92	99.93	0.14	54.01	99.83	41.94	99.92	99.93	0.14
	2-DDR(25)	67.54	99.88	39.52	99.94	**99.98**	**0.08**	**57.63**	99.87	39.78	99.93	**99.97**	**0.10**
	2-DDR(50)	68.25	99.87	40.32	99.93	99.97	0.10	56.14	99.86	39.78	99.93	99.96	0.11
	2-DDR(100)	**68.44**	99.87	40.32	99.94	99.97	0.10	56.22	99.86	39.78	99.93	99.96	0.11
XGB	JS	9.79	98.56	**90.86**	99.28	98.58	1.57	4.33	98.58	**88.17**	99.28	98.60	1.54
	BF	71.26	**99.95**	84.95	**99.97**	99.97	0.17	63.78	**99.92**	82.80	**99.96**	99.94	0.19
	CM	71.24	**99.95**	85.22	**99.97**	99.97	0.17	63.78	**99.92**	82.80	**99.96**	99.94	0.19
	BMR	74.14	99.84	44.09	99.92	99.94	0.14	63.37	99.81	40.86	99.90	99.91	0.16
	2-DDR(25)	74.96	99.90	42.20	99.95	**99.99**	**0.08**	64.13	99.87	35.48	99.93	**99.98**	**0.08**
	2-DDR(50)	**76.80**	99.89	44.35	99.94	99.98	0.10	63.39	99.86	37.63	99.93	99.96	0.10
	2-DDR(100)	**76.80**	99.89	44.35	99.94	99.98	0.10	**65.25**	99.86	38.71	99.93	99.96	0.10
LGBM	JS	85.81	99.94	**96.77**	99.97	99.94	0.22	49.60	99.89	**77.42**	99.95	99.93	0.20
	BF	87.42	99.97	96.51	**99.99**	99.98	0.18	46.45	99.93	76.34	99.96	99.97	0.16
	CM	81.72	**99.98**	93.82	**99.99**	**99.99**	0.17	47.65	**99.95**	74.19	**99.97**	**99.99**	0.13
	BMR	89.91	99.90	46.77	99.95	**99.99**	**0.09**	**54.74**	99.87	32.26	99.94	**99.99**	**0.07**
	2-DDR(25)	90.99	99.91	49.19	99.96	**99.99**	**0.09**	50.04	99.88	34.41	99.94	**99.99**	**0.07**
	2-DDR(50)	90.99	99.91	49.19	99.96	**99.99**	**0.09**	49.95	99.88	33.33	99.94	**99.99**	**0.07**
	2-DDR(100)	**91.01**	99.91	49.73	99.96	**99.99**	**0.09**	50.04	99.88	34.41	99.94	**99.99**	**0.07**
CSL	JS	-1.28	98.48	**85.22**	99.23	98.50	1.64	1.94	98.51	**86.02**	99.25	98.53	1.61
	BF	69.40	**99.95**	79.84	**99.97**	99.98	0.15	64.94	**99.92**	76.34	**99.96**	99.96	0.16
	CM	69.23	99.94	80.65	**99.97**	99.97	0.16	66.48	**99.92**	77.42	**99.96**	99.96	0.17
	BMR	71.22	99.89	44.09	99.94	99.98	0.09	67.42	99.86	43.01	99.93	99.95	0.12
	2-DDR(25)	**72.23**	99.89	42.47	99.95	**99.99**	**0.08**	**68.82**	99.87	40.86	99.94	**99.97**	**0.10**
	2-DDR(50)	**72.23**	99.89	42.47	99.95	**99.99**	**0.08**	**68.82**	99.87	40.86	99.94	**99.97**	**0.10**
	2-DDR(100)	**72.23**	99.89	42.47	99.95	**99.99**	**0.08**	**68.82**	99.87	40.86	99.94	**99.97**	**0.10**
CSB	JS	32.35	98.96	**84.68**	99.48	98.98	1.15	34.89	98.89	**83.87**	99.44	98.92	1.22
	BF	91.39	99.89	48.39	**99.95**	99.98	0.10	62.37	99.86	45.16	99.93	99.95	0.13
	CM	92.29	**99.91**	46.77	**99.95**	99.99	**0.08**	**62.97**	**99.88**	40.86	**99.94**	99.97	0.09
	BMR	92.31	**99.91**	46.77	**99.95**	**100.00**	**0.08**	62.76	99.87	40.86	**99.94**	99.97	0.10
	2-DDR(25)	**92.36**	**99.91**	46.77	**99.95**	**100.00**	**0.08**	62.89	99.87	34.41	**99.94**	**99.98**	**0.07**
	2-DDR(50)	**92.36**	**99.91**	46.77	**99.95**	**100.00**	**0.08**	**62.97**	**99.88**	34.41	**99.94**	**99.98**	**0.07**
	2-DDR(100)	**92.36**	**99.91**	46.77	**99.95**	**100.00**	**0.08**	**62.97**	**99.88**	34.41	**99.94**	**99.98**	**0.07**

a peak mainly with $k = 50$ with all classifiers. This suggests that a larger value of k is not needed to obtain better results, as these are limited by the ranking provided by the considered classifier. The right plot in Fig. 6 shows the time (minutes) needed to run the algorithm for different values of k. It can be seen a quadratically correlation between the computational time and k. As a visual check, the curve $0.004x^2 - 0.798$ is represented with a dashed black line. Cost-sensitive classifiers and LightGBM were faster, probably due to the good ranking of high-amount frauds offered by these models.

7. Summary and conclusions

This work introduces a new cost-sensitive methodology for fraud detection to reduce aggregated losses, the main concern in any business. Algorithm 2-DDR(k) obtains the optimal classification rule given an estimated probability and a cost specification. In addition, it has the added advantage that any POA restriction can be considered. Thus, it can be generalized to any cost-sensitive setting,

Table 6
Mean results summary in the Bank data set for the combination of all the approaches introduced throughout the paper.

		TRAIN set						TEST set					
Classifier	Threshold	Sav	Acc	Rec	F	Spec	POA	Sav	Acc	Rec	F	Spec	POA
	JS	40.33	72.27	68.01	83.81	72.30	27.97	39.56	72.24	66.05	83.79	72.28	27.98
	BF	40.59	64.82	**74.57**	78.44	64.76	35.51	40.27	64.77	**74.85**	78.39	64.70	35.57
	CM	40.26	65.62	73.73	79.12	65.56	34.70	41.19	65.55	74.16	79.06	65.49	34.78
LR	BMR	45.21	**74.12**	59.36	**85.06**	**74.22**	**26.01**	45.19	**74.14**	58.98	**85.08**	**74.25**	**25.98**
	2-DDR(25)	44.36	72.37	57.53	83.86	72.47	27.73	41.70	72.31	55.85	83.81	72.43	27.77
	2-DDR(50)	46.52	73.23	58.14	84.47	73.33	26.88	**46.73**	73.25	58.17	84.49	73.36	26.86
	2-DDR(100)	**47.02**	73.37	58.81	84.56	73.47	26.75	46.18	73.36	58.63	84.56	73.47	26.75
	JS	40.82	71.49	68.85	83.27	71.50	28.77	40.66	71.51	67.78	83.29	71.54	28.73
	BF	40.54	66.73	72.94	79.89	66.69	33.58	40.96	66.71	72.76	79.87	66.66	33.61
	CM	40.20	65.42	**73.79**	78.97	65.37	34.90	41.01	65.33	**74.04**	78.91	65.27	34.99
WLR	BMR	45.50	**74.05**	59.62	**85.02**	**74.15**	**26.08**	44.51	**74.04**	59.09	**85.01**	**74.14**	**26.08**
	2-DDR(25)	44.11	72.13	57.85	83.69	72.23	27.98	42.91	72.07	56.54	83.65	72.18	28.02
	2-DDR(50)	46.49	73.67	57.65	84.76	73.79	26.43	**45.72**	73.59	56.66	84.70	73.71	26.50
	2-DDR(100)	**47.21**	73.95	58.08	84.95	74.06	26.16	45.51	73.92	57.01	84.93	74.04	26.18
	JS	46.09	69.63	73.70	81.96	69.60	30.69	43.46	69.72	70.68	82.03	69.71	30.57
	BF	46.14	66.82	**75.70**	79.99	66.76	33.53	44.14	66.92	**73.23**	80.06	66.88	33.40
	CM	46.21	67.88	74.69	80.75	67.84	32.45	44.50	67.91	72.65	80.77	67.87	32.40
AB	BMR	49.11	74.53	61.71	85.34	74.63	25.62	44.97	74.62	60.95	85.40	74.72	25.53
	2-DDR(25)	48.30	72.56	61.51	83.99	72.64	27.59	45.73	72.77	59.79	84.14	72.86	27.36
	2-DDR(50)	49.63	74.07	60.93	85.03	74.16	26.08	**47.65**	74.31	59.33	85.19	74.42	25.82
	2-DDR(100)	**50.36**	**75.00**	60.38	**85.64**	**75.10**	**25.14**	47.02	**75.16**	57.71	**85.75**	**75.28**	**24.94**
	JS	48.80	73.27	74.57	84.46	73.26	27.07	42.91	73.15	67.91	84.40	73.19	27.10
	BF	48.87	70.64	**76.90**	82.68	70.60	29.73	42.87	70.50	70.22	82.59	70.50	29.78
	CM	48.42	70.54	76.78	82.61	70.49	29.83	43.43	70.12	**70.91**	82.33	70.12	30.17
XGB	BMR	51.33	**76.33**	64.30	**86.51**	**76.42**	**23.86**	45.70	76.13	58.86	86.38	76.25	23.99
	2-DDR(25)	49.46	74.02	63.45	84.99	74.09	26.16	44.84	73.83	60.26	84.87	73.93	26.31
	2-DDR(50)	50.55	73.06	63.92	84.35	73.13	27.13	44.76	72.88	59.56	84.23	72.97	27.25
	2-DDR(100)	**51.75**	76.28	62.64	86.47	76.37	23.90	**46.29**	**76.17**	59.09	**86.41**	**76.29**	**23.95**
	JS	52.38	78.07	69.14	87.57	78.13	22.19	45.18	77.84	61.77	87.42	77.95	22.33
	BF	52.51	78.92	68.01	88.14	78.99	21.33	44.96	78.72	60.84	88.02	78.84	21.43
	CM	48.73	67.99	**77.01**	80.82	67.93	32.38	42.43	67.71	**70.69**	80.63	67.69	32.57
LGBM	BMR	44.36	70.54	63.22	82.64	70.59	29.64	39.82	70.35	56.32	82.52	70.45	29.74
	2-DDR(25)	52.61	84.67	57.59	91.66	84.86	15.43	43.54	84.43	47.85	91.53	84.68	15.54
	2-DDR(50)	53.11	84.58	56.75	91.60	84.77	15.52	**45.35**	84.34	46.70	91.47	84.60	15.61
	2-DDR(100)	**53.61**	**86.84**	55.65	**92.92**	**87.05**	**13.24**	43.93	**86.56**	44.72	**92.77**	**86.84**	**13.37**
	JS	48.18	71.50	66.74	83.28	71.53	28.73	44.01	71.47	63.84	83.27	71.53	28.71
	BF	48.31	71.91	66.10	83.56	71.95	28.31	44.25	71.89	63.26	83.55	71.95	28.29
	CM	47.33	70.39	**67.32**	82.52	70.41	29.85	43.47	70.34	**64.77**	82.49	70.38	29.86
CSL	BMR	47.27	70.69	66.65	82.72	70.72	29.54	43.94	70.66	64.31	82.71	70.71	29.53
	2-DDR(25)	48.49	73.92	59.07	84.93	74.03	26.20	44.44	73.98	56.77	84.97	74.10	26.11
	2-DDR(50)	48.60	74.19	58.43	**85.11**	74.30	**25.92**	**44.56**	**74.23**	55.97	**85.14**	**74.36**	**25.85**
	2-DDR(100)	**48.61**	**74.20**	58.49	**85.11**	**74.31**	**25.92**	44.55	**74.23**	56.55	85.13	74.35	25.86
	JS	72.62	85.22	75.99	91.97	85.28	15.14	44.63	85.16	50.87	91.95	85.40	14.85
	BF	72.64	**86.57**	73.90	**92.75**	**86.65**	**13.76**	44.68	**86.68**	47.85	**92.83**	**86.95**	**13.29**
	CM	66.80	76.03	**82.44**	86.29	75.99	24.41	44.61	75.84	**65.47**	86.19	75.91	24.37
CSB	BMR	66.78	77.36	79.80	87.15	77.33	23.05	44.17	77.22	63.15	87.08	77.31	22.97
	2-DDR(25)	72.87	86.00	74.57	92.42	86.08	14.34	**44.87**	86.05	48.67	92.47	86.31	13.93
	2-DDR(50)	72.90	86.26	74.28	92.58	86.34	14.07	44.59	86.40	47.97	92.67	86.66	13.57
	2-DDR(100)	**72.92**	86.11	74.43	92.49	86.19	14.22	44.57	86.18	48.32	92.54	86.44	13.80

with potential in other settings as churn prediction, credit risk, medical diagnosis or logistic planning. Regarding the computational time, empirical results show that it depends mainly on k quadratically, as it controls the search grid size, and not on the sample size, which makes this approach suitable for scalability to larger data sets. Moreover, it has been checked empirically that a larger value of k is not needed to obtain satisfactory and stable results. In fact, using a large k can become counterproductive leading to overfitting.

Previous thresholding approaches are contained in Algorithm 2-DDR(k) search, so it extends and improves previous thresholding approaches, without any need of further estimations. Thus, a consistent improvement was expected. This has been verified with the results over two real fraud data sets, summarized in Tables 5 and 6. Although some thresholding approaches outperform the proposed methodology in terms of classification, given a classifier they are always beaten by the new proposal in terms of the objective function, savings in (2). This illustrates the contrast between minimizing costs or classification error during training, indicating that these are

Table 7
Mean results summary in the Bank data set considering the 5% and 10% POA restriction for the combination of all the approaches introduced throughout the paper.

Classifier	Threshold	TRAIN set (5% POA)						TEST set (5% POA)					
		Sav	Acc	Rec	F	Spec	POA	Sav	Acc	Rec	F	Spec	POA
LR	BF	18.20	94.75	**23.34**	97.30	95.25	4.88	16.95	94.72	**21.78**	97.28	95.22	4.90
	2-DDR(25)	21.83	**95.17**	12.80	**97.52**	**95.73**	**4.33**	19.36	**95.13**	11.35	**97.50**	**95.70**	**4.35**
	2-DDR(50)	22.91	94.99	16.40	97.43	95.53	4.55	**22.23**	95.03	15.88	97.45	95.57	4.51
	2-DDR(100)	**24.90**	94.85	14.66	97.35	95.40	4.67	21.07	94.87	11.93	97.36	95.44	4.61
WLR	BF	18.24	94.77	**23.31**	97.31	95.26	4.86	16.84	94.74	**21.78**	97.30	95.25	4.87
	2-DDR(25)	21.32	**94.80**	13.79	**97.33**	**95.36**	**4.70**	20.99	**94.82**	13.21	**97.34**	**95.38**	**4.68**
	2-DDR(50)	25.56	94.60	15.73	97.22	95.14	4.93	22.45	94.60	13.67	97.22	95.15	4.91
	2-DDR(100)	**26.05**	94.75	14.37	97.30	95.31	4.76	**23.63**	94.75	12.52	97.31	95.32	4.73
AB	BF	20.36	94.79	**26.18**	97.32	95.27	4.88	19.29	94.84	**24.10**	97.34	95.32	4.81
	2-DDR(25)	24.52	**95.33**	14.86	**97.61**	**95.88**	**4.19**	22.01	**95.40**	13.09	**97.64**	**95.96**	**4.10**
	2-DDR(50)	26.85	95.00	17.01	97.43	95.54	4.55	24.64	94.99	15.29	97.43	95.54	4.53
	2-DDR(100)	**28.84**	94.64	17.07	97.24	95.17	4.91	**26.87**	94.62	15.52	97.24	95.17	4.90
XGB	BF	26.60	94.90	**33.38**	97.38	95.33	4.87	19.13	94.91	**26.07**	97.39	95.39	4.76
	2-DDR(25)	27.66	**95.97**	15.88	**97.94**	**96.52**	**3.56**	18.76	**95.91**	10.78	**97.91**	**96.50**	**3.56**
	2-DDR(50)	29.05	95.28	15.56	97.58	95.83	4.25	21.66	95.21	11.59	97.55	95.79	4.26
	2-DDR(100)	**32.04**	95.01	18.55	97.44	95.54	4.56	**25.74**	94.96	14.25	97.41	95.52	4.55
LGBM	BF	33.57	94.95	**39.62**	97.40	95.33	4.90	18.74	94.77	**27.11**	97.31	95.24	4.91
	2-DDR(25)	35.11	**95.74**	28.97	**97.81**	**96.19**	**3.98**	22.99	**95.61**	18.19	**97.75**	**96.14**	**3.95**
	2-DDR(50)	38.20	95.02	29.06	97.44	95.48	4.69	25.91	94.81	18.89	97.33	95.33	4.77
	2-DDR(100)	**40.48**	94.85	29.87	97.35	95.29	4.88	**29.95**	94.62	19.93	97.23	95.14	4.97
CSL	BF	0.00	**99.32**	0.00	**99.66**	**100.00**		0.00	**99.31**	0.00	**99.66**	**100.00**	
	2-DDR(25)	9.70	95.51	4.33	97.70	96.14	**3.86**	11.14	95.59	4.87	97.74	96.21	**3.80**
	2-DDR(50)	15.70	94.71	7.40	97.29	95.32	4.70	15.93	94.80	7.30	97.33	95.41	4.61
	2-DDR(100)	**16.89**	94.59	**7.87**	97.21	95.18	4.84	**16.49**	94.67	**7.65**	97.26	95.26	4.76
CSB	BF	42.24	**95.67**	**30.51**	**97.78**	**96.12**	**4.06**	21.16	**96.11**	**17.95**	**98.01**	**96.65**	**3.45**
	2-DDR(25)	48.81	95.11	25.84	97.49	95.58	4.56	27.62	95.64	14.94	97.77	96.19	3.88
	2-DDR(50)	50.12	94.94	27.11	97.40	95.41	4.75	29.97	95.45	16.91	97.67	95.99	4.09
	2-DDR(100)	**51.96**	94.76	29.09	97.31	95.22	4.95	**30.45**	95.27	17.61	97.58	95.81	4.28

Classifier	Threshold	TRAIN set (10% POA)						TEST set (10% POA)					
		Sav	Acc	Rec	F	Spec	POA	Sav	Acc	Rec	F	Spec	POA
LR	BF	28.82	90.07	**39.97**	94.76	90.41	9.80	27.72	90.11	**38.36**	94.78	90.46	9.73
	2-DDR(25)	30.46	**91.15**	19.39	**95.36**	**91.65**	**8.43**	30.23	**91.14**	18.54	**95.35**	**91.64**	**8.43**
	2-DDR(50)	32.59	90.04	21.45	94.75	90.52	9.57	27.63	89.99	17.85	94.73	90.49	9.57
	2-DDR(100)	**33.85**	90.55	23.92	95.03	91.01	9.09	**32.93**	90.64	22.83	95.09	91.11	8.99
WLR	BF	28.67	89.93	**39.97**	94.69	90.28	9.93	27.89	89.98	**38.70**	94.71	90.33	9.87
	2-DDR(25)	29.73	**91.72**	18.78	**95.67**	**92.22**	**7.85**	**30.32**	**91.75**	18.19	**95.69**	**92.25**	**7.82**
	2-DDR(50)	32.53	90.00	21.36	94.73	90.47	9.61	27.90	89.97	18.08	94.71	90.47	9.59
	2-DDR(100)	**33.31**	90.16	22.90	94.82	90.62	9.47	29.01	90.15	19.81	94.81	90.63	9.44
AB	BF	31.03	90.10	**39.71**	94.78	90.44	9.76	28.06	90.13	**36.73**	94.79	90.50	9.69
	2-DDR(25)	33.68	**91.35**	21.71	**95.47**	**91.82**	**8.27**	32.84	**91.43**	20.74	**95.51**	**91.91**	**8.17**
	2-DDR(50)	38.66	89.93	27.72	94.69	90.36	9.77	**37.13**	89.97	25.60	94.71	90.41	9.70
	2-DDR(100)	**38.89**	89.99	26.18	94.72	90.43	9.68	35.06	89.99	23.52	94.72	90.44	9.65
XGB	BF	35.51	90.31	**46.13**	94.89	90.62	9.64	26.65	90.30	**37.65**	94.89	90.66	9.53
	2-DDR(25)	37.48	**90.35**	24.59	**94.92**	**90.81**	**9.30**	32.79	**90.38**	21.67	**94.94**	**90.85**	**9.24**
	2-DDR(50)	40.09	90.23	27.02	94.85	90.66	9.46	31.71	90.29	21.79	94.89	90.76	9.33
	2-DDR(100)	**42.71**	89.86	29.67	94.65	90.27	9.86	**34.25**	89.91	24.22	94.68	90.36	9.74
LGBM	BF	44.74	90.17	**52.10**	94.81	90.43	9.86	33.05	89.99	**41.83**	94.71	90.32	9.90
	2-DDR(25)	44.20	**92.57**	35.99	**96.13**	**92.97**	**7.23**	33.17	**92.43**	26.54	**96.06**	**92.89**	**7.24**
	2-DDR(50)	47.79	90.89	39.27	95.21	91.24	8.97	38.71	90.71	30.13	95.11	91.13	9.02
	2-DDR(100)	**50.39**	90.03	43.25	94.74	90.35	9.88	**40.16**	89.80	33.60	94.61	90.19	9.97
CSL	BF	0.00	**99.32**	0.00	**99.66**	**100.00**		0.00	**99.31**	0.00	**99.66**	**100.00**	
	2-DDR(25)	28.54	91.04	16.46	95.30	91.55	**8.50**	25.63	91.08	14.83	95.32	91.60	**8.44**
	2-DDR(50)	30.19	90.12	18.43	94.80	90.61	9.45	27.84	90.15	16.92	94.81	90.65	9.40
	2-DDR(100)	**31.72**	89.81	**19.39**	94.62	90.29	9.77	**28.19**	89.83	**17.38**	94.64	90.33	9.72
CSB	BF	66.67	**90.45**	**59.51**	**94.96**	**90.66**	**9.68**	38.39	**90.97**	**37.42**	**95.26**	**91.33**	**8.86**
	2-DDR(25)	66.11	90.24	49.20	94.85	90.52	9.75	39.53	90.44	29.89	94.97	90.86	9.29
	2-DDR(50)	67.36	90.21	51.35	94.83	90.48	9.81	39.68	90.48	31.16	94.99	90.89	9.27
	2-DDR(100)	**67.80**	90.06	52.16	94.75	90.32	9.97	**40.25**	90.28	31.86	94.87	90.68	9.48

two different objectives. For each data set a different classifier was selected. Given the no-free-lunch theorem, there is never an overall winner and some experimentation will always be required to optimize performance.

Further extensions can be considered. More dimensions could be introduced, as an extra default probability dimension to optimize the credit admission strategy globally. More complex loss functions could be introduced as well. These are just some examples of possible extensions that can be made from the flexibility offered by the proposed method, which has shown a satisfactory performance in amount-dependent problems.

Funding

This research has been supported by MICINN Grant PID2020-113578RB-I00 and by the Xunta de Galicia (Grupos de Referencia Competitiva ED431C-2020-14 and Centro de Investigación del Sistema Universitario de Galicia ED431G 2019/01), all of them through the European Regional Development Fund (ERDF). The first author was financed by the Axencia Galega de Innovación Grant 14-IN606D-2021-2607768.

CRediT authorship contribution statement

Jorge C. Rella: Investigation, Conceptualization, Methodology, Software, Data curation, Writing-original draft preparation. **Ricardo Cao:** Validation, Methodology, Writing-reviewing and editing. **Juan M. Vilar:** Validation, Methodology, Writing-reviewing and editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ricardo Cao, Juan M. Vilar, Jorge C-Rella reports financial support was provided by Xunta de Galicia. Jorge C-Rella reports financial support was provided by Axencia Galega de Innovación. Jorge C-Rella reports a relationship with Abanca Corporacion Bancaria SA that includes: employment.

Data availability

The authors do not have permission to share data.

References

[1] Y. Lucas, J. Jurgovsky, Credit card fraud detection using machine learning: a survey, ArXiv, arXiv:2010.06479 [abs], 2020.

[2] S. Yusuf, D. Ekrem, Detecting credit card fraud by decision trees and support vector machines, in: IMECS 2011 - International MultiConference of Engineers and Computer Scientists 2011, vol. 1, 2011, pp. 442–447.

[3] D. Almhaithawi, A. Jafar, M. Aljnidi, Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk, SN Appl. Sci. 2 (9) (2020) 1574–1585, https://doi.org/10.1007/s42452-020-03375-w.

[4] T. Vanderschueren, B. Baesens, T. Verdonck, W. Verbeke, A new perspective on classification: optimally allocating limited resources to uncertain tasks, ArXiv, arXiv:2202.04369 [abs], 02 2022.

[5] A. Dal Pozzolo, O. Caelen, R.A. Johnson, G. Bontempi, Calibrating probability with undersampling for unbalanced classification, in: 2015 IEEE Symposium Series on Computational Intelligence, 2015, pp. 159–166.

[6] G. King, L. Zeng, Logistic regression in rare events data, Polit. Anal. 9 (2) (2002) 137–163, https://doi.org/10.1093/oxfordjournals.pan.a004868.

[7] E.I. Altman, G. Marco, F. Varetto, Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks (the Italian experience), J. Bank. Finance 18 (3) (1994) 502–529, https://doi.org/10.1016/0378-4266(94)90007-8.

[8] J.H. Min, Y.-C. Lee, A practical approach to credit scoring, Expert Syst. Appl. 35 (4) (2008) 1762–1770, https://doi.org/10.1016/j.eswa.2007.08.070.

[9] N. Nikolaou, N. Edakunni, M. Kull, P. Flach, G. Brown, Cost-sensitive boosting algorithms: do we really need them?, Mach. Learn. 104 (2016) 359–384, https://doi.org/10.1007/s10994-016-5572-x.

[10] T. Vanderschueren, T. Verdonck, B. Baesens, W. Verbeke, Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies, Inf. Sci. 594 (2022) 400–415, https://doi.org/10.1016/j.ins.2022.02.021.

[11] T. Verbraken, C. Bravo, R. Weber, B. Baesens, Development and application of consumer credit scoring models using profit-based classification measures, Eur. J. Oper. Res. 238 (2014) 505–513, https://doi.org/10.1016/j.ejor.2014.04.001.

[12] S. Höppner, B. Baesens, W. Verbeke, T. Verdonck, Instance-dependent cost-sensitive learning for detecting transfer fraud, Eur. J. Oper. Res. 297 (2021) 291–300, https://doi.org/10.1016/j.ejor.2021.05.028.

[13] C. Elkan, The foundations of cost-sensitive learning, in: International Joint Conference on Artificial Intelligence 2001, Lawrence Erlbaum Associates Ltd., 2001, pp. 973–978.

[14] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284, https://doi.org/10.1016/j.ejor.2021.05.028.

[15] A. Correa Bahnsen, A. Stojanovic, D. Aouada, B. Ottersten, Improving credit card fraud detection with calibrated probabilities, in: Proceedings of the 2014 SIAM International Conference on Data Mining, Pennsylvania, USA, 04 2014.

[16] R. Malhotra, D.K. Malhotra, Differentiating between good credits and bad credits using neuro-fuzzy systems, Eur. J. Oper. Res. 136 (1) (2002) 190–211, https://doi.org/10.1016/S0377-2217(01)00052-2.

[17] S. Piramuthu, Financial credit-risk evaluation with neural and neurofuzzy systems, Eur. J. Oper. Res. 112 (2) (1999) 310–321, https://doi.org/10.1016/S0377-2217(97)00398-6.

[18] V.S. Desai, J.N. Crook, G.A. Overstreet, A comparison of neural networks and linear scoring models in the credit union environment, Eur. J. Oper. Res. 95 (1) (1996) 24–37, https://doi.org/10.1016/0377-2217(95)00246-4.

[19] D. West, Neural network credit scoring models, Comput. Oper. Res. 27 (11) (2000) 1131–1152, https://doi.org/10.1016/S0305-0548(99)00149-5.

[20] A.C. Bahnsen, D. Aouada, B. Ottersten, Example-dependent cost-sensitive logistic regression for credit scoring, in: Proceedings - 2014 13th International Conference on Machine Learning and Applications, ICMLA 2014, 2014, pp. 263–269.

[21] W.-t. Yih, J. Goodman, G. Hulten, Learning at low false positive rates, in: The Third Conference on Email and Anti-Spam 2006, 2006.

[22] A.C. Bahnsen, A. Stojanovic, D. Aouada, B. Ottersten, Cost sensitive credit card fraud detection using Bayes minimum risk, in: Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013, vol. 1, 2013, pp. 333–338.

[23] V. Sheng, C. Ling, Thresholding for making classifiers cost sensitive, in: Proceedings of the National Conference on Artificial Intelligence, vol. 1, 2006, pp. 476–481.

[24] B. Zadrozny, C. Elkan, Learning and making decisions when costs and probabilities are both unknown, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2001, 2001, pp. 204–213.

[25] J. Pesántez-Narváez, M. Guillén, Penalized logistic regression to improve predictive capacity of rare events in surveys, J. Intell. Fuzzy Syst. 38 (2020) 1–11, https://doi.org/10.3233/JIFS-179641.

[26] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second edition, Springer Series in Statistics, 2009.

[27] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139, https://doi.org/10.1006/jcss.1997.1504.

[28] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[29] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 30 (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[30] W.J. Youden, Index for rating diagnostic tests, Cancer 3 (1) (1950) 32–35, https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.

[31] N. Siddiqi, Credit Risk Scorecards, Developing and Implementing Intelligent Credit Scoring, John Wiley & Sons, Inc., Hoboken, NJ, 2006.