Research Article

Example-dependent cost-sensitive credit cards fraud detection using SMOTE and Bayes minimum risk

Doaa Almhaithawi[1] · Assef Jafar[1] · Mohamad Aljnidi[1]

Received: 6 November 2019 / Accepted: 19 August 2020

Abstract
This paper presents fraud detection problem as one of the most common problems in secure banking research field, due to its importance in reducing the losses of banks and e-transactions companies. Our work will include: applying the common classification algorithms such as logistic regression (LR), random forest (RF), alongside with modern classifiers with state-of-the-art results as XGBoost (XG) and CatBoost (CB), testing the effect of the unbalanced data through comparing their results with and without balancing, then focusing on the savings measure to test the effect of cost-sensitive wrapping of Bayes minimum risk (BMR), we will concentrate on using F1-score, AUC and Savings measures after using the traditional measures duo to their suitability to our problem. The results show that CB has the best savings (0.7158) alone, (0.971) when using SMOTE and (0.9762) with SMOTE and BMR, while XG has the best savings (0.757) when using BMR without SMOTE.

Keywords Machine learning · Example-dependent cost-sensitive · Random forest (RF) · Extreme gradient boosting (XGBoost-XG) · CatBoost (CB) · Synthetic minority over-sampling technique (SMOTE) · Bayes minimum risk (BMR) · Fraud detection (FD)

Abbreviations

FD	Fraud detection
ML	Machine learning
LR	Logistic regression
RF	Random forest
GBM	Gradient boosting machine
XGBoost	Extreme gradient boosting
DT	Decision tree
ANN	Artificial neural networks
SVM	Support Vector machine
Acc	Accuracy
ROC	Receiving operating characteristic
AUC	Area under the ROC curve
Down	Under-sampling
Over	Over-sampling
SMOTE	Synthetic minority over-sampling technique
Cost	Wrapped with cost-sensitive
TP	True positive
FP	False positive
TN	True negative
FN	False negative

1 Introduction

From the beginning of the monetary transactions, the fraudsters have tried to gain money in multiple illegal ways, so using protection methods was a necessity.

The communications development and moving towards electronic monetary transactions make the fraud more common specially with the ease of exchanging experiences between the fraudsters and gaining access to the victim companies.

✉ Doaa Almhaithawi, doaa.almhethawi@hiast.edu.sy; Assef Jafar, assef.jafar@hiast.edu.sy; Mohamad Aljnidi, mohamed.aljnidi@hiast.edu.sy | [1]Higher Institute for Applied Sciences and Technology, HIAST, Damascus, Barzeh 31983, Syria.

Published online: 27 August 2020

The huge losses of banks and other financial institutions caused the increase of interest in research to prevent fraud and decrease its effects. However, methods and techniques could not be revealed to the public, because of the privacy imposed by the supporting companies of these researches, one reason is the high competition in the field, the other is to make sure that fraudsters cannot benefit from the results in improving their methods.

For the same reasons there was no standard dataset for research until 2015 when researchers published the fraud detection dataset [1].

Many researchers worked in this field and still, not only to solve a scientific problem but to help real companies and financial institutions to reduce their daily losses. However, some of them used statistics meanwhile others used machine learning approaches, supervised or unsupervised. In spite of the large number of researches which most of them are applied in real world systems, the researchers have never stopped improving their methods.

The contributions of this paper can be summarized as follow:

- Study the problem as a supervised binary classification and example-depended cost-sensitive.
- Use the boosting binary classifiers as they are highly recommended due to their good results and high performance.
- Use SMOTE [2] algorithm as a preprocessing balancing step.
- Compare the boosting classifiers with Logistic Regression and Random Forest.
- Suggest a combination of SMOTE with the best resulted binary classifier and an after training cost-sensitive wrapping (BMR [3]) to reach the best Savings.

In this paper, we will continue the introduction by introducing the FD problem with a brief description of the class-sensitive problems, focusing on example-dependent, and why our problem is considered one. Followed by a brief mention of the existing techniques commonly used to solve this problem in Sect. 2, machine learning in particular. Next, we will give the used dataset description with the measures we used to compare, after describing it in details in Sects. 3 and 4. The end will be the results from testing common classifiers (LR, RF, XGBoost [4] and CatBoost [5]) by the traditional measures and then a comparison between them and the cost-sensitive versions with and without SMOTE over-sampling by manually implemented Savings measure.

1.1 Fraud detection

Companies practically use human investigators to detect fraud. However, these investigators cannot process the huge amounts of daily transactions, along with the relations among them to detect patterns, so the institutions are developing smart systems to help them detect the suspicious transactions, which then are sent to investigators for wider investigation, resulting confirmation or denial of the fraud. That led to speed up the FD process while minimizing the cost.

The credit cards fraud detection problem is considered one of the most suitable problems to test the calculation intelligence algorithms [6]. There are many challenges in this case, one is the highly unbalanced classes, because of the small percentage of fraud transactions regarding the overall number of transactions, (no more than 0.1% [1]). Another is the concept drift, because of the tide relation with the human development through time, which can be explained mathematically as a changed distribution through time problem.

The main reason of the unbalanced data is the unbalanced distribution, it is not the data that is unbalanced, but the distribution which the data is drawn from [7].

To solve the unbalance problem in our data we can use data-level or algorithmic-level techniques. Data-level strategies are used as a pre-processing step to rebalance the dataset so the two classes have a relatively close number of samples, before any algorithm is applied and it can be grouped into five main categories as mentioned in [1]: sampling, ensemble, cost-based, distance-based and hybrid. Meanwhile the algorithmic-level, algorithms which are less effected by the unbalancing problem, they can be either specifically designed classifiers deal with unbalanced distribution or classifiers that minimize overall classification cost.

Our problem is binary classification because of the two classes, fraud/not fraud, as results. There are plenty of algorithms commonly used, and we will mention some in details in a later section.

1.2 Cost-sensitive problems

In most of the classification methods, there is no difference between the cost of the correctly classified and misclassified examples and they only focus on the accuracy. But in some real-world problems this is not enough. As in FD, to classify a transaction as fraud but in reality it is not (False

Positive), that will cost what is called "The Administrative Cost" [8], which is related to analyzing the transaction by the human investigators and contacting the card holders, meanwhile, classifying a transaction as not fraud but in reality it is (false negative) will cost the amount of the transaction that we have lost, which will vary from one transaction to another.

The literature in cost-sensitive classification can differentiate between class-dependent problems where the cost of misclassification is associated with the class (Fraud or Not-Fraud), and example-dependent problems where the cost of misclassification is associated with each example [9], as in our case. The class-dependent is highly restrictive, as assuming that the different costs are constant within the same class which is not realistic in fraud detection.

Another way of categorizing the algorithms in the literature is the time the cost-sensitive is applied in the training process [9], which is: prior training, during training and after training shown in Fig. 1.

In this paper we will apply the Bayes minimum risk as a wrapping step to the classifiers, which some of them are new as XGBoost and CatBoost and have achieved state-of-the-art results in multiple fields, with and without SMOTE as a re-balancing step, and we will compare the results using F1-score, AUC and Savings.

2 Related work

The high importance of this problem urges the researchers to pay attention to find solutions, that was by using existing methods or developing new ones. In [10] the researchers compare 11 classifiers tested on 71 datasets and they conclude that Gradient Boosting Decision Trees (GBDT) has sometimes better and most of the time faster results comparing to SVM or RF, this encourage us to use the boosting family of classifiers.

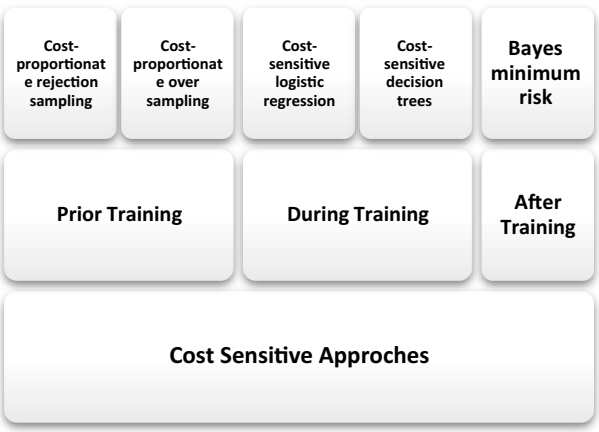

Fig. 1 Categorizing cost-sensitive algorithms accourding to the training process stage [9]

Meanwhile in [11] the researchers worked with fraud detection using three classifiers: RF, balanced bagging ensemble (BBE) and Gaussian Naïve Bayes and they concluded that BBE has the best prediction but RF is the most acclimatized with large data size, in [12] the researchers proposed a classifier combined of number of classifiers and compare it with RF and XGboost using the traditional measures. All the previous studies had compared classifiers without rebalancing or cost-sensitive approach using the traditional measures, but others had looked to the fraud detection problem as unsupervised problem and proposed solutions accordingly, like in [13] they used anomaly detection technics (one-class SVM and T2 control charts) using real-word data which gave them high accuracy and low FP rate, and in [14] they used autoencoders with Restricted Boltzmann Machine. In our work we focused on the supervised approach while considering the unsupervised in the future.

However, there was plenty researches discussing the problem as cost-sensitive, like in [15] the researchers improve the known decision tree (DT) to a cost-sensitive version and the results were better than the algorithms (DT, ANN, SVM) using SLR (saved loss rate) and TPR (true positive rate) measures in fraud detection problem. Meanwhile, the researchers in [16] have been working on improving the SVM to be able to handle the unbalanced data problems with cost-sensitive, which show that the cost-sensitive SVM has better results in most of the used datasets which were 21 datasets, using AUC to compare. In [17], one of their methods to improve credit scoring results was cost-sensitive logistic regression and they used AUC as a measure. However, they train it on credit scoring data set and the result was that proper variable discretization and cost-sensitive logistic regression with the best class weights can reduce the model bias and/or variance. [8] also worked in improving DT, it proposes an example-dependent cost-sensitive decision tree algorithm using three real-world applications: credit card fraud detection, credit scoring and direct marketing, which showed better results for the three datasets using accuracy, F1 and Savings as measures.

Although, [3] used BMR wrapping and they used LR, RF and DT as classifiers they have not used SMOTE to rebalance the data, instead they used under-sampling.

In [18] they altered the cost function of SVM to produce a cost-sensitive version and they trained it on 21 datasets from KDD98 not including fraud detection dataset, they even compare the results with balancing using SMOTE, they compared the results using AUC (and risk for only datasets with costs included) and their proposed algorithm had the best results in most of the datasets, meanwhile in [19] the researchers propose a cost-sensitive random forest based ensemble learning technique and their algorithm outcome two existing cost-sensitive implementation of random forest.

Although [20] have altered three boosting classifiers to be cost-sensitive it didn't use XGBoost nor CatBoost, however the researchers managed to get better results using F-score and Cost as measures.

In this paper, we will compare the example-dependent cost-sensitive BMR wrapping of four algorithms (LR, RF, Xgboost and CatBoost) with and without using SMOTE as a rebalancing pre-process step, while using the F1-score, AUC and Savings (the latter is implemented manually) measures.

Table 3 shows a comparison between the related and our work.

3 Datasets

One of our problem's difficulties was not having standard datasets for research purposes until 2015, in this paper we will use the Credit Card Dataset which consists of 284,807 transaction with 492 fraud cases (0.172% fraud) [1].

The known features are:

- *Time* the seconds between each transaction and the first transaction in the dataset.
- *Amount* is the transaction amount, (which will be used for example-dependent cost-sensitive learning).
- *Class* is the response variable and it takes value "1" in case of fraud and "0" otherwise.

The meaning of the other 28 features is not revealed for confidentiality reason. Also, they have been transformed by means of principal components.

The cardholder identifier is not given in the dataset, so each transaction can be considered independent from the others. Which was the reason why we could not use this dataset to study the concept driven in our problem.

4 Measurement/metrics

There are many metrics to measure performances, but they vary depending on the data we are using. First, we will mention the most common metrics, then we will explain which is more suitable in our case and why.

The coincidence or confusion matrix is defined as follows [21].

The numbers in the upper-left and the lower-right represent the correct decisions made, meanwhile the others represent the errors. The true positive rate (recall or Sensitivity) of a classifier is estimated by dividing the correctly classified positives (the true positive count) by the total positive count (Sensitivity $= \frac{TP}{P}$). The overall accuracy of a classifier is estimated by dividing the total correctly classified positives and negatives by the total number of samples (Accuracy $= \frac{TP+TN}{P+N}$) [21].

These metrics are commonly used in classification problems and even in FD as in [22], but they are not preferred in our problem because the data is unbalanced, therefore any classifier, even random classifier, will give a high accuracy if it classifies all the transactions as not fraud. It is more accurate to use other measures as ROC (receiving operating characteristic), AUC (Area Under the ROC Curve) in this type of problems, where ROC curve draws the relation between True Positive percent and False Positive percent and the AUC measure the area under this curve, where the higher AUC the better [21].

Although these measures derived from the confusion matrix are very common, they assume the misclassification errors carry the same cost, so they may not be the most suitable for evaluation in our problem. Thence, we can use the proposed cost matrix in [23] which is example-dependent.

And from this matrix we can calculate the cost and savings respectively as follows:

$$Cost = \sum_{i=1}^{N} y_i\left(1 - c_i\right)Amt_i + c_i C_a \ldots \tag{1}$$

$$Savings = 1 - \frac{Cost}{Cost_l}, \ldots \tag{2}$$

$$where \quad Cost_l = \sum_{i=1}^{N} y_i Amt_i \ldots \tag{3}$$

N: the number of examples. $Cost_l$: the cost of not using any algorithm and predict all the examples as not fraud.

Fig. 2 OSEMN process [24]

5 Methodology

In our work we used the OSEMN process shown in Fig. 2 described in [24] which consists of five steps: Obtain, Scrub, Explore, Model and iNterpret. For the training phase the Obtain was from an CSV file (the dataset) meanwhile, the prediction was from a stream (NiFi simulation). The Scrub has only the scaling (there was no null values in the data and we did not delete outliers for their importance). In the Explore we tested the correlation between the features, and between them and the class. The Model, it has the model parameters setting (with/without SMOTE and with/without BMR) and the cross validation, and in the end the iNterpret, compares one or more model with the F1-score, AUC and Savings measures.

Figure 3 shows the details of all the previous steps.

In this paper, we will discuss four main algorithms (LR, RF, Xgboost, CatBoost), which some of them is sensitive to unbalanced data (ex: RF) and the others are not, then we will combine them with data-level sampling algorithm (SMOTE) and finally, we will wrap them with example-dependent cost-sensitive method (BMR) and compare them using Savings measure among others.

5.1 Smote

"Synthetic Minority Over-sampling Technique, is an algorithm which over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining any/all the 'k' minority class nearest neighbours" [2]. The idea is to form new minority examples by interpolating between samples of the same class. This has the effect of creating clusters around each minority observation. By creating synthetic observations, the classifier builds larger decision regions that contain nearby instances from the minority class.

5.2 Binary classifiers

As we already mentioned, our problem is supervised binary classification, where the dataset includes examples, each is consisted of input and output to train the model,

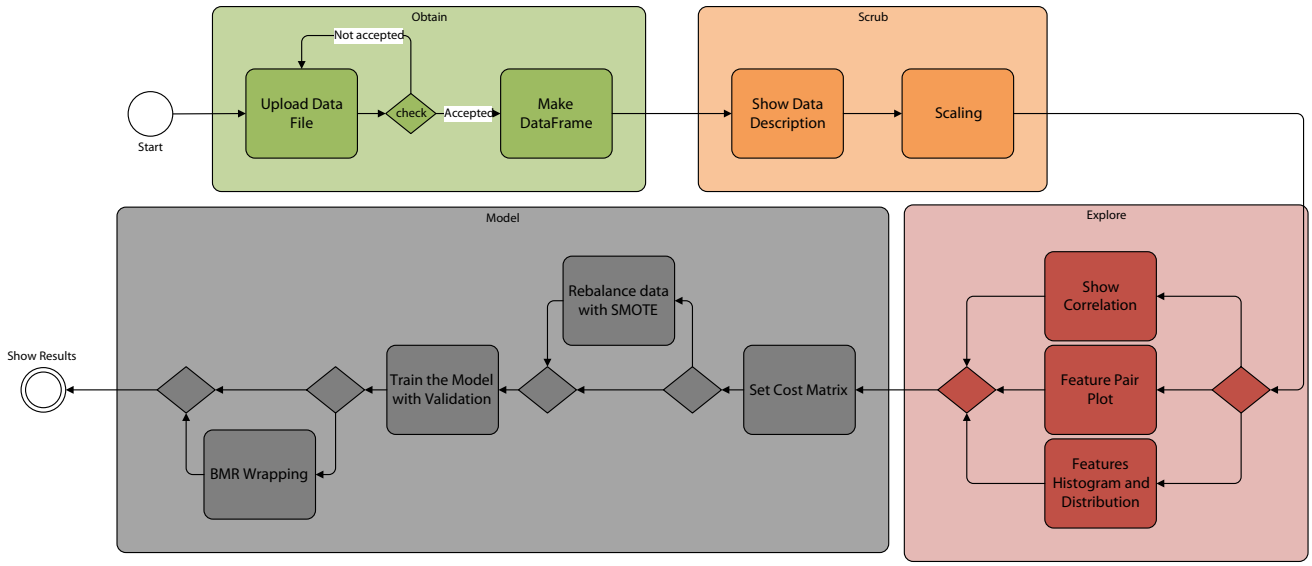

Fig. 3 OSEMN in our work

and predict the output of a new example by having the input features.

In our work we will use $y \in \{0, 1\}$ to refer to the output, where "1" means fraud and "0" means not a fraud.

5.3 Random forest

"RF is the algorithm which has solved the overfitting problem in Decision Trees. It consists of a collection of decision trees" [25]. The prediction of the random forest is obtained by a majority vote over the predictions of the individual trees. To specify a particular random forest, we need to define the algorithm each tree applies and the distribution over its independent and identically distributed random variables.

The reason why we used LR and RF is to have some base to compare with, duo that these classifiers are some of the traditional yet still used tremendously in the literature.

5.4 Boosting

"Boosting is a powerful technique for combining multiple 'base' classifiers to produce a form of committee whose performance can be significantly better than that of any of the base classifiers" [26]. Although, base classifiers or known as weak learners, can perform only slightly better than random, boosting gives good results in most cases. The main reasons why we are using boosting are:

- Our data is heterogeneous data.
- Results can be interpretable.
- More useful for a medium dataset.
- Training time is relatively very fast.
- Require short time to tune parameters.

Although boosting is not new, but many new interpretations and versions appear every year. So, we are using the two latest boosting algorithms in our work, Xgboost and CatBoost.

5.5 XgBoost

Extreme Gradient Boosting is a scalable tree boosting algorithm which is widely used by data scientists, it provides state-of-the-art results on many problems. It handles sparse data and it proposes a theoretically justified weighted quantile sketch for approximate learning [4].

5.6 CatBoost

A new gradient boosting toolkit introduced by [5], this algorithm competes other available boosting implementations in terms of quality. CatBoost introduces two advances, according to researches:

- The implementation of ordered boosting, alternative to the classic algorithm.
- Algorithm for processing categorical features.

These two advances will solve the target leakage in currently existing implementations of gradient boosting algorithms.

Table 1 Binary confusion matrix [21]

		True class	
		p	n
Hypothesized class	Y	True positives	False negatives
	N	False negatives	True negatives
	Column totals	P	N

Table 2 Example-dependent cost-sensitive matrix

		True class	
		$y_i = 1$	$y_i = 0$
Hypothesized	$c_i = 1$	C_a	C_a
	$c_i = 0$	Amt_i	0

Where c_i is the predicted class, y_i is the true class, C_a the administrative cost, Amt_i is the amount of the transaction which we have lost because it is a false negative

5.7 BMR

In this paper we will use a cost-sensitive wrapping to the classifiers after training using the estimated probability and the cost matrix. The Bayes Minimum Risk is the way to wrap our classifiers, it takes the estimated probability resulted after the classifier's training and calculate the risk of predicting each of the classes, and choose the one with the minimum risk (Table 1).

As [3] the risk of the fraud and non-fraud classes respectively is defined as:

$$R(p_f|x) = L(p_f|y_f)P(p_f|x) + L(p_f|y_l)P(p_l|x) \ldots \quad (4)$$

$$R(p_l|x) = L(p_l|y_l)P(p_l|x) + L(p_l|y_f)P(p_f|x) \quad (5)$$

where p_f, p_l: are the prediction of fraud and non-fraud; y_f, y_l: are the true label of fraud and non-fraud; $L(a|b)$: is

the loss function when a transaction is predicted as a and the real label is b. $P(p_f|x), P(p_l|x)$ are the estimated probabilities for fraud and non-fraud classes.

The transaction will be predicted as fraud if:

$$R(p_f|x) \leq R(p_l|x), \ldots \quad (6)$$

In our problem this will mean:

$$C_a P(p_f|x) + C_a P(p_l|x) \leq Amt_i P(p_f|x) \ldots \quad (7)$$

According to Table 2.

6 Experimental

In our work, we used R (Rstudio) then python (Jupyter Notebook), and we experiment the following:

- As we explored the dataset, we noticed that the "Time" and "Amount" features have different scale from the others, so we started by pre-processing these two features with Standard Scaler (Table 3).
- When using SMOTE, it has increased the number of samples from 284,807 to 568,630 with 284,315 fraud sample instead of 492. Then we pre-process the data with Standard Scaler.
- We also validate our models using 5 splits k-fold cross validation.

We first start by testing the classifiers (LR, RF, XGBoost, Catboost) with traditional measures and the results are shown in Table 4, and then we focused on the Savings, F1-score and AUC measures which are more useful in our problem, in Table 5 with SMOTE rebalancing and Table 6 without it.

In Table 7 we show the results of the wrapped classifiers using BMR also with and without SMOTE using the Savings, F1-score and AUC measures.

And we will compare with the cost-sensitive implementation of RF proposer in [9] in the CostCla library (CSRF).

The parameters tuning was implemented by cross-validation using R, and the following parameters were used depending on the best AUC:

- Logistic Regression: threshold = 0.5
- Random Forest: mtry = 16.
- XGBoost: nrounds = 100, gamma = 0.3, max_depth = 10, objective = binary:logistic.
- CatBoost: default parameters.

7 Results and discussion

As mentioned before the main challenges in our problem is the unbalanced data and the concept drift, in this paper we were concerned with the first challenge, meanwhile the dataset cannot be used to study the second one, due to the independence of the transactions. As we already mentioned the dataset was altered for privacy reasons by deleting the id of the credit card so we cannot connect two or more transaction belongs to the same credit card.

In addition, we scaled the Time and Amount features so they will have the same effect as the others, but we used the Amount to weight the risk in BMR wrapping and again in the Savings measure.

As we can see in Table 4 the results show the traditional measures after tuning the classifiers (the parameters will be mentioned in Appendix) the confusion matrix values show the difference between the classifiers, where XG has the worst FN after LR, but it has the best FP, which mean although it couldn't discover that much of frauds as the others (RF, CB) but at least it gives the least false alarms. This difference will be highly considered from the one working in this field, what is more important, the amount of the discovered cases no matter the overhead of the wrongly predicted cases, or discovering as much as possible with minimal overhead.

Meanwhile in Table 5 even without SMOTE or BMR the CB is giving the best Savings with the best F1-score, but XG has the best Savings at the expense of F1-score when using BMR only as shown in Table 7.

Tables 6 and 7 shows that SMOTE has enhanced results for all the model especially for the RF which has improved a lot and outperform the rest, CB has the best results as the RF when using SMOTE.

Noticeably, our proposed wrapping using BMR outperforms the CostCla cost-sensitive implementation of RF in Savings and F1-score.

From Figs. 4 and 5 we can see that using SMOTE increase the savings with increasing the F1-score which indicates that increasing the detected fraud cases, meanwhile BMR increase the savings on the expense of the F1-score, which indicates less number of detected fraud cases but detecting higher amounts.

Figure 6 compares the AUC of the classifiers where we can see that RF was the worst.

To choose the wrapping while using SMOTE is for sure the best choice regarding the savings, but it will cause an overhead regarding the performance, in the end it will depend on the problem, the number of the training examples and the training process repeating rate.

Table 3 Related work comparison

References	Year	Algorithm(s)	Results	Disadvantages	Our work
[3]	2013	BMR wrapping and with LR, RF and DT	BMR Wrapping has better results	Under-sampling to rebalance the data	SMOTE for rebalancing and boosting algorithms
[10]	2017	Compare 11 classifiers tested on 71 datasets	Gradient Boosting Decision Trees has better and faster results	Use the Accuracy and AUC as measures	Encourage us to use the boosting family of classifiers
[11]	2018	RF, BBE and Gaussian Naïve Bayes	BBE has the best prediction but RF is the better with large data size	Use the Accuracy as measures, not taking cost into considerations, no balancing	Cost sensitive wrapping, Savings as measure, SMOTE for rebalancing
[12]	2018	Classifier combined, RF and xgboost	Similarity in the results	Using traditional measures, not taking cost into considerations, no balancing	
[13]	2018	One-class SVM and T2 control charts	High accuracy and low FP rate	Accuracy as measure, real-word data (cannot be compared), not taking cost into considerations, no balancing	Standard dataset, cost sensitive wrapping, Savings as measure, SMOTE for rebalancing
[14]	2018	Autoencoders with Restricted Boltzmann Machine	Better AUC	AUC as measure, not taking cost into considerations, no balancing	Cost sensitive wrapping, Savings as measure, SMOTE for rebalancing
[15]	2013	Cost-sensitive DT	Results were better than the algorithms DT, ANN and SVM	No balancing	Savings as measure, SMOTE for rebalancing
[16]	2019	Cost-sensitive SVM, 21 datasets	Better results in most of the used dataset	AUC as measure, no balancing	Savings as measure, SMOTE for rebalancing
[17]	2018	Cost-Sensitive LR	Better AUC	AUC as measure, no balancing, credit scoring problem not fraud detection	Savings as measure, SMOTE for rebalancing, boosting algorithms
[18]	2019	Cost-sensitive SVM, 21 datasets	Better in risk and AUC	Not using Fraud Detection dataset, AUC as measure	Savings as measure, boosting algorithms
[19]	2019	Cost-sensitive weighted random forest	G-mean, F-measure and AUC values	AUC as measure, no balancing	Savings as measure, SMOTE for rebalancing, boosting algorithms
[20]	2020	ICSAdaBoost, ICSRealBoost and ICSGentleBoost	Better F-Score and Cost	Not using Fraud Detection dataset	Savings as measure, SMOTE for rebalancing

Table 4 The classifiers results with traditional measures

	LR	RF (Down)	RF (SMOTE)	XGBoost	CatBoost
TP	58	84	84	72	80
FP	8	928	540	**5**	617
FN	34	8	8	20	11
TN	56,861	55,941	56,329	56,864	56,253
Acc	0.9993	0.9836	0.9904	0.9996	0.989
Recall	0.6304	0.9130	0.9130	0.7826	0.8791
NPV	0.9994	0.9999	0.9999	0.9996	0.9998
PPV	0.8788	0.083	0.1346	0.9351	0.1148

Bold value indicate the maximum result in each column/measure

Table 5 The classifiers with and without SMOTE

	F1-score	AUC	Savings
LR	0.7318	0.9723	0.5084
RF	0.8448	0.9475	0.6408
XG	0.8470	0.9699	0.7024
CB	0.8624	0.9688	**0.7158**

Bold value indicate the maximum result in each column/measure

Table 6 The classifiers with SMOTE

	F1-score	AUC	Savings
LR + SMOTE	0.9802	0.9973	0.9282
RF + SMOTE	1.00	0.9999	0.9704
XG + SMOTE	0.9876	0.9994	0.9424
CB + SMOTE	1.00	0.9999	**0.9710**

Bold value indicate the maximum result in each column/measure

Table 7 The wrapped classifiers with and without SMOTE

	F1-score	AUC	Savings
CSRF	0.5510	0.9398	0.6262
LR + BMR	0.3420	0.9723	0.7426
RF + BMR	0.3574	0.9475	0.730
XG + BMR	0.2890	0.9699	**0.7570**
CB + BMR	0.5694	0.9688	0.7460
LR + SMOTE + BMR	0.7666	0.9973	0.9720
RF + SMOTE + BMR	0.8192	0.9999	0.9760
XG + SMOTE + BMR	0.7656	0.9994	0.9726
CB + SMOTE + BMR	0.8250	0.9999	**0.9762**

Bold values indicate the maximum result in each column/measure

8 Conclusion

In this paper we studied the fraud detection problem in credit cards, presenting the methods to reduce the unbalancing of the data using resampling SMOTE as a preprocess. We compare some common classifiers with and without cost-sensitive wrapping by F1-score, AUC and Savings measures. Finally, we found that XG has given good Savings when wrapped with BMR, but CB and RF has outperformed when using SMOTE.

As future work, we can consider testing XG and CB for example-dependent cost-sensitivity by modifying their loss function as future work, which consider during training cost-sensitive implementation. In addition, we consider using another dataset to study the concept driven, which must have a connection between the transactions made by the same card to study the user profile and this is not provided by the used dataset in this paper, and need a longer period of collected data.

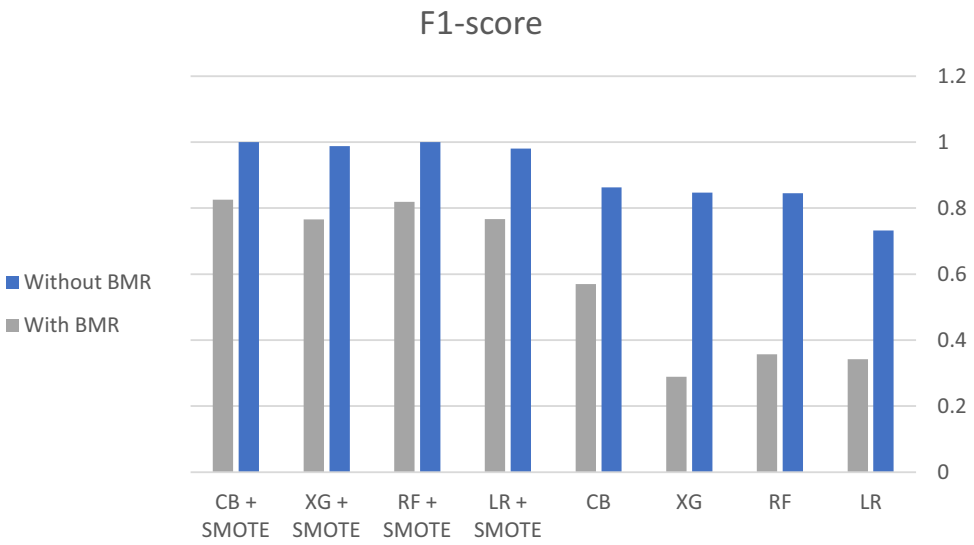

Fig. 4 Comparison of F1-score with and without BMR

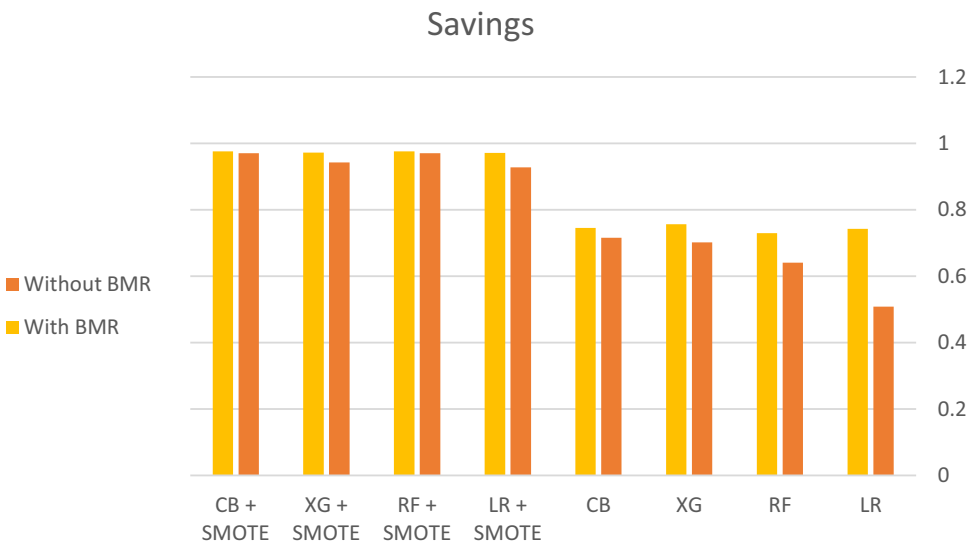

Fig. 5 Comparison of Savings with and without BMR

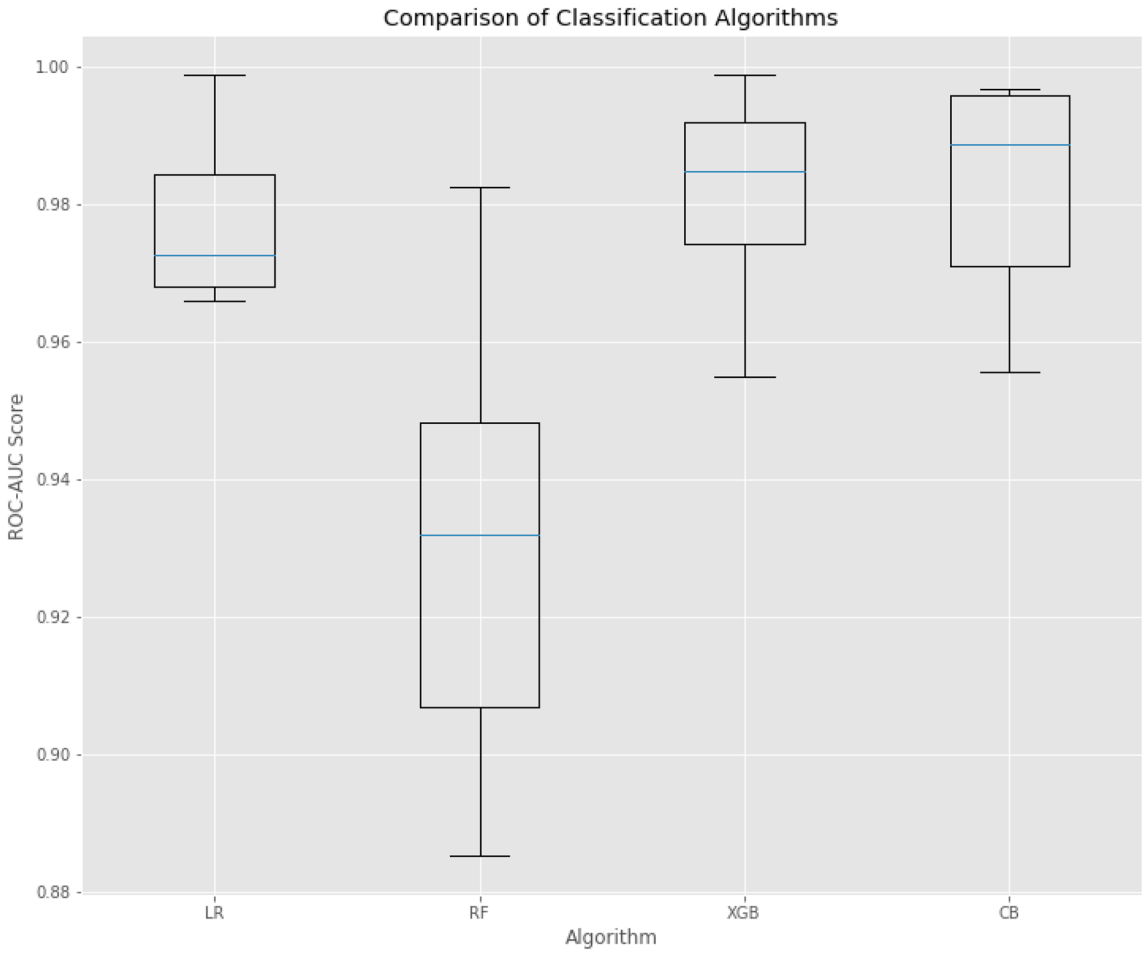

Fig. 6 AUC comparison of the classifiers

Author contributions DA; Designed and performed experiments, analyzed data and wrote the paper, AJ; Designed experiments and supervised the research, MA; Provided final approval of the version to publish.

Availability of data and materials The dataset supporting the conclusions of this article is available in the kaggle repository [https://www.kaggle.com/mlg-ulb/creditcardfraud].

Compliance with ethical standards

Conflict interest The authors declare that they have no conflict interest.

References

1. Dal Pozzolo A, Bontempi G (2015) Adaptive machine learning for credit card fraud detection. Unpublished doctoral dissertation, Université libre de Bruxelles, Faculté des Sciences—Informatique, Bruxelles
2. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
3. Bahnsen AC, Stojanovic A, Aouada D, Ottersten B (2013) Cost sensitive credit card fraud detection using Bayes minimum risk. In: 2013 12th international conference on machine learning and applications, vol 1. IEEE, pp 333–338
4. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp 785–794
5. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2018) CatBoost: unbiased boosting with categorical features. In: Advances in neural information processing systems. pp 6638–6648
6. Dal Pozzolo A, Boracchi G, Caelen O, Alippi C, Bontempi G (2017) Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Trans Neural Netw Learn Syst 29(8):3784–3797
7. Daumé III H (2012) A course in machine learning. Ciml Inf pp 5–73
8. Bahnsen AC, Aouada D, Ottersten B (2015) Example-dependent cost-sensitive decision trees. Expert Syst Appl 42(19):6609–6619
9. Correa Bahnsen A (2015) Example-dependent cost-sensitive classification with applications in financial risk modeling and marketing analytics. Doctoral dissertation, University of Luxembourg, Luxembourg

10. Zhang C, Liu C, Zhang X, Almpanidis G (2017) An up-to-date comparison of state-of-the-art classification algorithms. Expert Syst Appl 82:128–150
11. Mohammed RA, Wong KW, Shiratuddin MF, Wang X (2018) Scalable machine learning techniques for highly imbalanced credit card fraud detection: a comparative study. In: Pacific rim international conference on artificial intelligence. Springer, Cham, pp 237–246
12. Dhankhad S, Mohammed E, Far B (2018) Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In: 2018 IEEE international conference on information reuse and integration (IRI), IEE, pp 122–125
13. Tran PH, Tran KP, Huong TT, Heuchenne C, HienTran P, Le TMH (2018) Real time data-driven approaches for credit card fraud detection. In: Proceedings of the 2018 international conference on e-business and applications. pp 6–9
14. Pumsirirat A, Yan L (2018) Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine. Int J Adv Comput Sci Appl 9(1):18–25
15. Sahin Y, Bulkan S, Duman E (2013) A cost-sensitive decision tree approach for fraud detection. Expert Syst Appl 40(15):5916–5923
16. Park Y, Luo L, Parhi KK, Netoff T (2011) Seizure prediction with spectral power of EEG using cost-sensitive support vector machines. Epilepsia 52(10):1761–1770
17. Zhang L, Ray H, Priestley J, Tan S (2020) A descriptive study of variable discretization and cost-sensitive logistic regression on imbalanced credit data. J Appl Stat 47(3):568–581
18. Iranmehr A, Masnadi-Shirazi H, Vasconcelos N (2019) Cost-sensitive support vector machines. Neurocomputing 343:50–64
19. Devi D, Biswas SK, Purkayastha B (2019) A Cost-sensitive weighted random forest technique for credit card fraud detection. In: 2019 10th international conference on computing, communication and networking technologies (ICCCNT). IEEE, pp 1–6
20. Sharifnia E, Boostani R (2020) Instance-based cost-sensitive boosting. Int J Pattern Recognit Artif Intell 34(03):2050002
21. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874
22. Dhankhad S, Mohammed E, Far B (2018) Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In: 2018 IEEE international conference on information reuse and integration (IRI). IEEE, pp 122–125
23. Bahnsen AC, Aouada D, Stojanovic A, Ottersten B (2016) Feature engineering strategies for credit card fraud detection. Expert Syst Appl 51:134–142
24. Mason H, Wiggins CH (2010) A taxonomy of data science. Retrieved November 2017, from http://www.dataists.com/2010/09/a-taxonomy-of-data-science
25. Shalev-Shwartz S, Ben-David S (2014) From theory to algorithms. Understanding machine learning. Cambridge University Press, Cambridge
26. Nasrabadi NM (2007) Pattern recognition and machine learning. J Electron Imaging 16(4):049901