Journal of Information Security and Applications 46 (2019) 13–22

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model

Salvatore Carta, Gianni Fenu, Diego Reforgiato Recupero*, Roberto Saia

Department of Mathematics and Computer Science, University of Cagliari, Palazzo delle Scienze, Via Ospedale 72, 09124 Cagliari

ARTICLE INFO

Article history:
Available online 22 February 2019

Keywords:
Information security
Credit card
Fraud detection
Machine learning

ABSTRACT

More and more financial transactions through different E-commerce platforms have appeared now-days within the big data era bringing plenty of opportunities but also challenges and risks of stealing information for potential frauds that need to be faced. This is due to the massive use of tools such as credit cards for electronic payments which are targeted by attackers to steal sensitive information and perform fraudulent operations. Although intelligent fraud detection systems have been developed to face the problem, they still suffer from some well-known problems due to the imbalance of the used data. Therefore this paper proposes a novel data intelligence technique based on a Prudential Multiple Consensus model which combines the effectiveness of several state-of-the-art classification algorithms by adopting a twofold criterion, probabilistic and majority based. The goal is to maximize the effectiveness of the model in detecting fraudulent transactions regardless the presence of any data imbalance. Our model has been validated with a set of experiments on a large real-world dataset characterized by a high degree of data imbalance and results show how the proposed model outperforms several state-of-the-art solutions, both in terms of ensemble models and classification approaches.

1. Introduction

Nowadays, the employment of credit cards for financial transactions represent the backbone of the E-commerce dynamics and business, since they allows purchasing in real-time of goods and services all over the world and using any device (smart-phone, tablet, pc) connected to the Internet. As it can be noticed, there are risks associated to this operation that might cause the theft of sensitive information associated to the customers' credit cards. A recent report from the European Payments Council[1] shows that a certain percentage of the Internet electronic payments is related to frauds.

One more analysis has been performed by the *Euromonitor International*[2] in the Europe, Middle East and Africa (*EMEA*) area, which shows that the number of frauds, and the associated budget in euros, within the *EMEA* area kept growing from 2006 to 2016, the year of the publication of the study. The values showing that are reported in Fig. 1. Although these data refer only to the *EMEA* area, they clearly underline the seriousness of the problem. In US, the *American Association of Fraud Examiners*[3] found out that *15%* of all the frauds are somehow connected to credit cards transactions, and this represents the *80%* of the whole financial value.

According to the *FBI*'s Internal Crime Complaint Center (*IC3*)[4], the term *credit card fraud* is defined as:

"a wide-ranging term for fraud committed using a credit card or any similar payment mechanism as a fraudulent source of funds in a transaction."

It can be carried out in two different ways, *off-line* or *on-line* [1]. If the fraud is off-line, that means the credit card has been previously stolen and then used to perform fraudulent payments, assuming the identity of the legitimate owner. For this case the thief has a limit amount of time which lasts from the time of the theft of the credit card to the time when the owner reports to his/her bank and the bank consequently disables the card. When the fraud is on-line, and this is the most common, the information been stolen is digital and it has been obtained in several ways (e.g., *skimming*, *shimming*, *cloning*, or *phishing*). Once the fraudster obtains this information, he/she can purchase through the Internet, until either the legitimate owner does not notice the problem and blocks the card or the budget in the card ends. The latter (fraud

* Corresponding author.
E-mail address: diego.reforgiato@unica.it (D. Reforgiato Recupero).

[1] https://bit.ly/2yQC7G1.

[2] http://www.euromonitor.com/.

[3] http://www.acfe.com.

[4] https://www.ic3.gov/.

https://doi.org/10.1016/j.jisa.2019.02.007

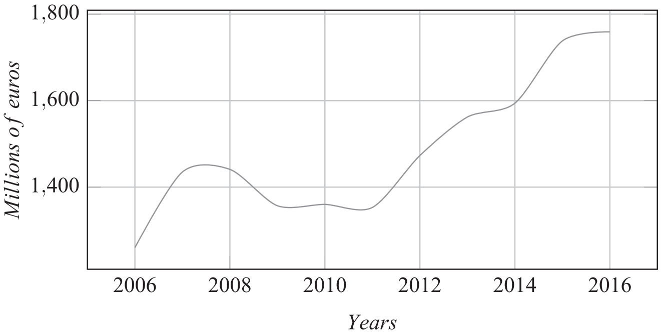

Fig. 1. Fraud amount in EMEA area.

on-line) is the case we take into account within the proposed paper.

Several research institutions and industries have made huge investments with the aim of designing effective methods capable of tackling the problem by employing machine learning, deep learning, big data, and computational intelligence technologies.

The efforts in this context have led to a large number of solutions that are able to automatically distinguish legitimate credit card transactions from the fraudulent ones.

However, regardless of the used approach, there are some common problems that reduce its performance. The most common is represented by the unbalanced distribution nature of the training data characterizing the past transactions which generates different problems of over-fitting and leads to low performances of the adopted classifiers. In other words, such a problem arises because the number of available *fraudulent* samples is usually much lower than the *legitimate* ones and this high grade of unbalance does not allow the definition of a reliable model of evaluation [2–4].

This happens because the fraudulent transactions collected in the past by the fraud detection systems are much less frequent than the legitimate ones. Moreover, (i) the heterogeneity nature of the data and the (ii) presence of overlaps among the data [5] are two elements that worsen the problem.

The two elements mentioned above highly affect the effectiveness of any fraud detection systems generating a large number of miss-classifications.

The approach we propose in this paper aims at improving the classification performances of several state-of-the-art classification algorithms, by adopting an ensemble approach where the final classification is given by the combination of the different elements of the ensemble through a novel model regulated by a twofold policy. The policy is further defined with probabilistic and prudential criteria in order to maximize the effectiveness of single approaches.

More in detail, the main scientific contributions of our proposed approach are the following:

(i) We introduce a formalization of the *Prudential Multiple Consensus* (*PMC*) model aimed at combining the classification made by each single approach by adopting both a probabilistic and a prudential criterion;
(ii) We defined the algorithm used to classify the new transactions as *legitimate* or *fraudulent* depending on the *PMC* model previously formalized, according to the performed ensemble criteria analysis.

The remainder of this paper is organized as it follows. Section 2 introduces background and related work of the scenario taken into account. Section 3 includes the adopted formal notation defining formally the problem we face. In Section 4 we describe the implementation of the proposed approach whereas in Section 5 we perform a preliminary study aimed at selecting and ensembling a set of classification algorithms. Section 6 describes the characteristics of the experimental environment as well as the description of the adopted datasets, strategy, and metrics whereas Section 7 shows the obtained results along with a related discussion. Remarks and future work where we are headed are given in Section 8 which also ends the paper.

2. Background and related work

After an introduction on the most common approaches and methods used to tackle the fraud detection problem, this section underlines the current open problems, introducing the ensemble classification methods and the most suitable performance metrics that are used for evaluation.

2.1. Fraud detection approaches

Different approaches in literature which tackle the fraud detection problem exploits the following techniques:

- *Data Mining*: an example is provided by the work of researchers in [6] where the generation of ad-hoc patterns are presented to recognize frauds. In one more example [7], authors investigate several combinations of manual and automatic approaches of classification.
- *Artificial Intelligence*: as in [8], which uses a technique to obtain a reduction of the number of false alarms, during the evaluation process.
- *Machine Learning*: the work presented in [9] makes use of several types of classification (single and ensemble). Another work, [10], takes into account the combination of unsupervised and supervised strategies.
- *Genetic Programming*: an example is represented by [11]), where an *evolutionary computation* technique has been implemented to improve the fraud detection process, taking into account the dynamics of the credit card transactions.
- *Reinforcement Learning*: as in [12], which formalizes the interactions between fraudsters and card-issuers as a *Markov Decision Process*.
- *Transformed-domain-based*: as in [13–16], where the evaluation process has been performed in a non-canonical domain (e.g., time, frequency, or frequency-time).
- *Combined Criteria*: as in [17,18], where a multidimensional technique is exploited to improve the classification performance. One more example is given in [17], where the authors introduce several fraud indicators in the classification process.

2.2. Operating modalities

Under a different point of view, all the state-of-the-art fraud detection approaches can be divided in two categories which depend on the involvement of artificial intelligence technology [19]: there are *supervised approaches* and *unsupervised approaches*. In the following we will show a brief description of each of the two classes.

- *Supervised approaches* define their evaluation model by exploiting the past fraudulent and non-fraudulent transactions collected by the fraud detection system. These methods do not work well without a considerable number of examples (training data) which have annotations in both classes (i.e., *legitimate* and *fraudulent*). As these methods must learn from the training data how to predict future data, and therefore they discover only known patterns, it is important to provide them with a fully consistent and complete training data.
- *Unsupervised approaches* operate by searching anomalies in the features that compose the transaction under evaluation. The problem in this case is that a *fraudulent* transaction might not

have any anomalies in its values and, consequently, the design of effective unsupervised fraud detection approaches continues being a hard research challenge [20].

In addition, the evaluation model definition can be performed using three different modes: *static, updating*, or *forgetting*:

- By following the *static mode*, the data under analysis are divided into several blocks of equal size and the training of the model is made by exploiting a defined number of contiguous blocks [21]. A drawback of this mode is the absence of a dynamic model of evaluation able to follow users behaviour changes.
- The *updating mode* does not work by using a unique evaluation model, since it updates the model when a new block arrives, involving in this process a defined number of the most recent and contiguous blocks [22]. The problem in this case is the impossibility to operate with small classes of data.
- The *forgetting mode* also updates the evaluation model at each new block, but it performs this operation by involving all the past *fraudulent* transactions and the *legitimate* ones present in the last two blocks [23]. However, this mode presents high computational cost.

2.3. Current open problems

In addition to the intrinsic issues mentioned in Section 2.2, in the following we report the most important open problems that affect the fraud detection domain, regardless of the used approach.

- **Data Scarcity**: it happens because for different reasons (commercial operators policies, privacy, legal constraints, etc.) there is not much availability of real-world datasets to use to develop and verify novel fraud detection approaches [11]. This scenario is quite understandable, given the intrinsically private nature of the involved data and it represents a big problem for the researchers, which in many cases are forced to use synthetic data [24].
- **Data Heterogeneity**: this problem is related to the difficulty to model the relationships among transaction features that are represented differently in various sets of data [25]. In other words, it is presented because each card issuer processes every day a high number of transactions, and each transaction is composed by a number of features whose values periodically might change within a single user's account or between all the accounts.
- **Model Staticity**: the classification approaches define their evaluation model based on the available data (i.e., past transactions). Considering the high level of heterogeneity of the involved information, this is a problem where the pattern that characterizes a new transaction is not present among those used to define the evaluation model [26].
- **Cold Start**: in the fraud detection scenario and in others needing the training step of an evaluation model, this problem happens when the available data are not enough [27]. In the context taken into consideration in this paper, we have a cold-start situation when a fraud detection system does not collect a sufficient number of *fraudulent* and *legitimate* transactions to perform the model training.
- **Data Imbalance**: this issue is given by the small number of *fraudulent* cases usually collected by a fraud detection system, respect to the *legitimate* cases. Considering the past transactions are used to train the evaluation models, such an occurrence leads towards a reduction of the fraud detection approaches effectiveness [2–4]. The literature presents many approaches able to face this problem, such as those proposed in [28,29].

2.4. Ensemble methods

Ensemble methods are largely adopted in order to perform data classification [30], since they can improve the performance achieved by a single classification method. Such methods have been much investigated in the past, as for example the work in [31], which evaluates the advantages related to the computational, statistical, and representational aspects.

It should be observed how the combination of more classification algorithms does not always lead to better results, because this operation can also reduce the overall classification performance. This means that the effectiveness of the resulting approach is strictly related to the strategy adopted to aggregate the single results, thus it highly depends on the definition of the global assessment model [30].

In scenarios such as the one considered in this paper, the ensemble methods are mainly aimed at improving the correct evaluation of a minority class label (e.g., that related to the *fraudulent* cases), since it represents an important result when the available data are strongly unbalanced [32].

The literature indicates the ensemble methods as one of the most effective approaches able to face the class imbalance problems. Moreover, such a scenario has been well outlined in a survey [33], where out of *527* papers taken into consideration, *218* referred to ensemble models.

The strategy used to combine several classification algorithms usually operates in the following two steps:

(i) In the first step a series of different algorithms is selected based on the complementarity of their results (i.e., they get misclassifications in different places of the test set);
(ii) In the second step their results are combined by adopting a consensus criterion, such as complete agreement, majority, absolute, correction, multi stage, weighted, confidence and ranked voting [34].

Approaches similar to ours are listed in the following: [35], where the authors analyzed the performance of three state-of-the-art data mining techniques in the context of a bagging ensemble classifier based on *decision tree* algorithms; [36], where the authors propose a strategy that drops a certain number of classifiers periodically and uses only a part of them for the evaluation; [37], where the authors combine the bagging and boosting techniques. However, regardless the adopted ensemble strategy, the state-of-the-art solutions do not implement any prudential criterion, such us that in our *PMC* approach, which is based on the observation that in the context taken into account (credit card fraud detection) a wrong classification of a transaction as *fraudulent* is preferable rather than a wrong classification of a transaction as *legitimate*.

The reader notices that some classification algorithms are ensemble in nature, such as *AdaBoost* [38] and *Gradient Boosting* [39]. Such algorithms operate by exploiting a prediction model based on an ensemble of weak prediction models [33].

2.5. Performance assessment

Within the fraud detection scenario, especially where credit card transactions are involved, the performance assessment must follow certain criteria, due to the particular configuration of the involved data. This is necessary because some canonical metrics (e.g. the accuracy) usually used to evaluate the classification algorithms performance might lead to unreliable results.

This happens especially when the experiments involve unbalanced data [3,33,40]. For example, let us assume a dataset in which the *fraudulent* cases represent the 0.01% of the entire dataset (*legitimate* and *fraudulent*), an algorithm that classifies all the samples as *legitimate* achieves the 99% of accuracy.

Table 1
Confusion matrix.

Real class		Algorithm classification		
		fra	leg	total
	fra′	TruePositive	FalseNegative	|*fra*′|
	leg′	FalsePositive	TrueNegative	|*leg*′|
	total	|*fra*|	|*leg*|	

It should be also underlined that such an event is not rare, and it is in accordance with the real-world data.

For the aforementioned considerations, the suitable assessment metrics should be oriented to evaluate the algorithms performance by taking into account the unbalanced configuration of data.

For this reason it is preferable to use the metrics based on the *confusion matrix* shown in Table 1 (avoiding their use in aggregate form), where *fra* stands for *fraudulent* and *leg* stands for *legitimate*).

Simple metrics as the *Sensitivity* (true positive rate) and the *Fallout* (false positive rate) give us information about the classification algorithms effectiveness in terms of *fraudulent* cases correctly classified, while metrics as the *Specificity* (true negative rate) and the *Miss Rate* (false negative rate), provide specular information on the performance related to the detection of the *legitimate* cases.

In addition to the aforementioned metrics, it is also preferable to add another one such as the *AUC* (Area Under the *ROC* Curve), since it is able to investigate the ability to discriminate between the possible destination classes (i.e., *legitimate* and *fraudulent* in our case) of the adopted evaluation model [41,42].

3. The proposed approach, notation and problem formulation

This section introduces the proposed approach, the formal notation which includes the formulation of the problem we address in this paper.

3.1. Proposed approach

This paper proposes a combined approach aimed at maximizing the effectiveness of several single approaches. We employ an ensemble strategy regulated by a *Prudential Multiple Consensus* model, which is based on a twofold criterion, probabilistic and majority based.

Such an idea relies on the observation that the results given by different classification algorithms are not the same in terms of correct classifications and misclassifications. It means that in many cases their results do not agree on the identification of certain *legitimate* or *fraudulent* transactions.

In more detail, in a preliminary study we observed that the classifications made by different algorithms are frequently in conflict, also when the different algorithms achieve good individual performances in terms of *Specificity* (i.e., legitimate cases correctly classified) and *Sensitivity* (*fraudulent* cases correctly classified). This can be exploited to increase the classification reliability, by adopting strategies that take into account the classifications made by multiple algorithms.

On the basis of these considerations, the proposed approach wants to exploit an aggregation strategy able to conveniently combine the correct evaluations made by the single algorithms, maximizing their effectiveness in the *fraudulent* transactions detection.

Instead of adopting a canonical aggregation criteria (e.g., *complete agreement, majority voting*, or *weighted voting*, etc.) to determine the class of destination of a new transaction and the basis of the results of the single algorithms, our approach adopts a novel *prudential criterion* which works as it follows:

(i) Each single algorithm classifies a new transaction as *legitimate* only if its classification is *legitimate* and the classification probability is above the average value of the probabilities of the classifications made by all the algorithms for that transaction. When this does not happen (i.e., the algorithm classification is *fraudulent* or the classification probability is below that average value of probability) the transaction is classified as *fraudulent*;

(ii) A canonical consensus criterion based on the *majority voting* is then taken into account and the final classification of the transaction under analysis will depend on the results of all the algorithms.

3.2. Formal notation

Given a set of transactions $T = \{t_1, t_2, \dots, t_N\}$ collected in the past and already classified and the subsets $T_+ = \{t_1, t_2, \dots, t_K\}$ and $T_- = \{t_1, t_2, \dots, t_J\}$, respectively related to the *legitimate* and *fraudulent* transactions in T (i.e., $T_+ \subseteq T$ and $T_- \subseteq T$), we denote as $F = \{f_1, f_2, \dots, f_M\}$ the set of features that compose each transaction $t \in T$.

In addition, we denote as $\hat{T} = \{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_U\}$ a set of new transactions to classify and as $C = \{legitimate, fraudulent\}$ the possible classes of destination of each transaction, meaning that a transaction can belong to only one class $c \in C$.

Finally, we denote as $A = \{a_1, a_2, \dots, a_Z\}$ a set of classification algorithms.

Let Ψ be the classification process made by using our *PMC* model. Then we evaluate the correctness of each classification performed by *PMC* through the function ***Evaluation***$(\hat{e}, \Psi)$ that returns a Boolean value β: *1* in case of correct classification, *0* otherwise. In this way we can formalize our classification problem in terms of maximization of the sum of the values returned by this function, as indicated in Eq. (1).

$$\max_{0\leq\beta\leq|\hat{E}|} \beta = \sum_{u=1}^{|\hat{E}|} \boldsymbol{Evaluation}(\hat{e}_u, \Psi) \tag{1}$$

4. Implementation

The architecture of our approach is shown in Fig. 2.

In such an architecture, the activity made in the *Data Preprocessing* block depends on the input data and it can involve, for instance, a *class binarization* (a mapping of a multi-class learning problem to several two-class learning problems) or a *minority class oversampling* (in order to adjust the class distribution of the dataset).

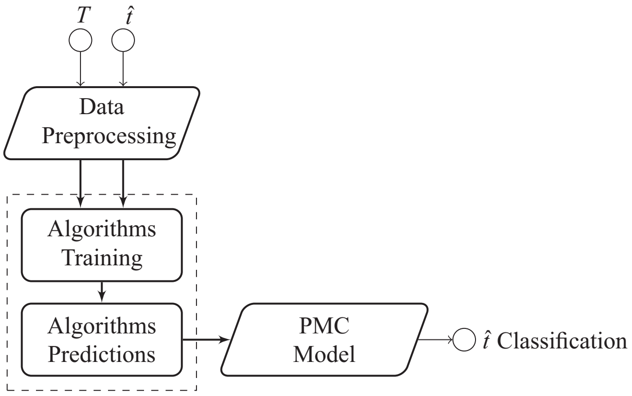

Fig. 2. PMS high-level architecture.

4.1. *Step 1:* model definition

The proposed *Prudential Multiple Consensus* (*PMC*) model operates by combining the results of five classification algorithms (described in Section 6.3) on the basis of two criteria, one based on the *classification probability* and one based on the *majority voting*.

To get the *classification probability* we use *Logistic Function*, since it is able to measure the probability of a binary response based on more independent predictors. More formally, the probability that a new transaction $\hat{t} \in \hat{T}$ belongs to a class $c \in C$ is calculated by mapping the algorithm predictions in terms of probabilities through the sigmoid σ function[5] Such a method is formalized in the Eq. (2), where $\sigma(a_z(p))$ is the probability estimate for the prediction p through the algorithm a_z, whose result is given in the range [0,1] and e denotes the base of natural log.

$$\sigma\left(a_z(p)\right) = \frac{1}{1+e^{-p}} \tag{2}$$

Subsequently, in each classification performed by a single algorithm, a transaction is considered as *legitimate* only when its probability is above a certain value, otherwise, prudentially, the transaction is classified as *fraudulent*. The final classification is given according to the results of all the algorithms (by using the *majority voting* criterion), as shown in Eq. (3), where $|A|$ is the number of classification algorithms and c is the transaction classification.

$$c = \begin{cases} legitimate, & if\ w_1 > w_2 \\ fraudulent, & otherwise \end{cases}$$

with

$$\begin{aligned} \mu &= \tfrac{1}{|A|} \cdot \sum_{z=1}^{|A|} \sigma\left(a_z(p)\right) \\ w_1 &= \sum_{z=1}^{|A|} 1 \ \ if \ \ \sigma\left(a_z(p)\right) > \mu \wedge a_z(p) = legitimate \\ w_2 &= \sum_{z=1}^{|A|} 1 \ \ if \ \ \sigma\left(a_z(p)\right) \leq \mu \vee a_z(p) = fraudulent \end{aligned} \tag{3}$$

It should be observed that *Logistic Function* represents only one of the possible approaches able to estimate the probability of a binary response given by a predictor. It means that also other approaches able to perform the same operation can be used in our model.

4.2. *Step 2:* data classification

According to the model previously formalized in Section 4.1, each new transaction $\hat{t} \in \hat{T}$ is classified by using the Algorithm 1.

The input of the Algorithm 1 is represented by the classification algorithms in the set A, the previous transactions E already classified, and a new transaction $\hat{t} \in \hat{T}$ to evaluate. The output will be the classification of the event $\hat{t}$ as *legitimate* or *fraudulent*. At *step 4* the evaluation models related to the set A of the classification algorithms are defined, while their classifications for the transaction $\hat{e}$ are calculated at *step 5*. The average probability value of all the performed classifications is calculated at *step 6* and saved in μ. A control aimed at checking whether the classification probability of each algorithm is above the average value in μ is performed (*step 7* to *step 13*). In particular, we increase by one the value of w_1 when *p=legitimate* and the prediction probability is above the μ value, otherwise we increment the value of w_2. The transaction $\hat{e}$ is classified as *legitimate* when all predictions have been processed and $w_1 > w_2$, otherwise the transaction is classified as *fraudulent*. Such a classification is returned, and the algorithm ends, at *step 19*.

Algorithm 1 Transaction classification.

```
Input: $A$=Set of algorithms, $T$=Past classified transactions, $\hat{t}$=Unevaluated transaction
Output: $result$=Transaction $\hat{t}$ classification
 1: procedure CLASSIFICATION($A$, $T$, $\hat{t}$)
 2:     $w_1 \leftarrow 0$
 3:     $w_2 \leftarrow 0$
 4:     $models = trainingModels(A, T)$
 5:     $predictions = getPredictions(A, models)$
 6:     $\mu \leftarrow getProbabilityAverage(predictions)$
 7:     for each $p$ in $predictions$ do
 8:         if $getProbability(p) > \mu \wedge p == legitimate$ then
 9:             $w_1 \leftarrow w_1 + 1$
10:         else
11:             $w_2 \leftarrow w_2 + 1$
12:         end if
13:     end for
14:     if $w_1 > w_2$ then
15:         $result \leftarrow legitimate$
16:     else
17:         $result \leftarrow fraudulent$
18:     end if
19:     return $result$
20: end procedure
```

It should be noted that the functions *getProbabilityAverage()* and *getProbability()* are both based on the *Logistic Function* model formalized in Eq. (2).

5. Classification algorithms

This section first explains the used criteria for the selection of the algorithm to use in our approach, then it describes the adopted ensemble criteria.

5.1. Selection criteria

In order to implement the proposed evaluation model, we need that the classification algorithms, those in the set A of Section 3.2, not only predict the class label, but also provide the probability related to each class label. It should be observed that not all the classification algorithms provide this type of information, which represents a kind of confidence level about the prediction. For this reason, during the composition of the set of algorithm A, we kept out the algorithms not providing this information and the algorithms performing a poor estimation of the class probabilities (i.e., those that, instead of a continuous probability value in [0,1], returned only the *0* or *1* values).

The five algorithms that have been thus chosen for the experiments are: *Multilayer Perceptron* (*MLP*), *Gaussian Naive Bayes* (*GNB*), *Adaptive Boosting* (*ADA*), *Gradient Boosting* (*GBC*), and *Random Forests* (*RFC*), and their settings are shown in Table 2.

The reason why we limited the number of algorithms to five derived by several analysis and work in literature, such as [43], which fixes to five the maximum number of algorithms to be used within an ensemble approach to obtain the best classification performances.

5.2. Ensemble criteria

We performed a set of experiments in order to try different combinations for the proposed *Prudential Multiple Consensus* model.

As a first step, we trivially used our model with single algorithms applying the prudential voting defined in Section 4.1, as

[5] A mathematical function characterized by a *sigmoid curve*, which maps any real value into the interval [0,1].

Table 2
Algorithms configuration.

Algorithm	Parameters
MLP	activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, hidden_layer_sizes=(100,), learning_rate='constant', learning_rate_init=0.001, max_iter=200, momentum=0.9, nesterovs_momentum=True, power_t=0.5, random_state=None, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False
GNB	priors=None
ADA	algorithm='SAMME.R', base_estimator=None, learning_rate=1.0, n_estimators=50, random_state=None
GBC	criterion='friedman_mse', init=None, learning_rate=0.1, loss='deviance', max_depth=3, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, presort='auto', random_state=None, subsample=1.0, verbose=0, warm_start=False
RFC	bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False

Table 3
Single algorithms performance.

Evaluation algorithm	*Native Sensitivity*	*Native Fallout*	*Native AUC*	*Model Sensitivity*	*Model Fallout*	*Model AUC*
MultilayerPerceptron(MLP)	0.146	0.000	0.573	0.781	0.629	0.576
GaussianNaiveBayes(GNB)	0.709	0.012	0.848	0.739	0.016	0.862
AdaptiveBoosting(ADA)	0.614	0.000	0.807	0.970	0.667	0.651
GradientBoosting(GBC)	0.506	0.000	0.753	0.699	0.082	0.808
RandomForests(RFC)	0.653	0.000	0.827	0.807	0.016	0.896

Table 4
Ensemble algorithms performance by pairs and triples.

Algorithms by pairs	*Sensitivity*	*Fallout*	*AUC*	*Algorithms by triples*	*Sensitivity*	*Fallout*	*AUC*
MLP, GNB	0.753	0.016	0.869	MLP, GNB, ADA	0.730	0.012	0.859
MLP, ADA	1.000	0.992	0.504	GNB, ADA, GBC	0.747	0.012	0.867
MLP, GBC	0.869	0.727	0.571	ADA, GBC, RFC	0.782	0.001	0.890
MLP, RFC	0.807	0.016	0.896	MLP, ADA, RFC	0.782	0.001	0.890
GNB, ADA	1.000	0.992	0.504	MLP, GNB, RFC	0.744	0.002	0.871
ADA, GBC	1.000	0.992	0.504	MLP, GBC, RFC	0.699	0.002	0.848
GBC, RFC	0.828	0.041	0.894	GNB, ADA, RFC	0.794	0.013	0.890
GNB, GBC	0.824	0.074	0.875	MLP, ADA, GBC	0.635	0.000	0.817
GNB, RFC	0.817	0.022	0.898	MLP, GNB, GBC	0.631	0.007	0.812

Table 5
Ensemble algorithms performance by quadruples.

Algorithms by quadruples	*Sensitivity*	*Fallout*	*AUC*	*Algorithms by quadruples*	*Sensitivity*	*Fallout*	*AUC*
GNB, ADA, GBC, RFC	0.807	0.016	0.895	MLP, GNB, ADA, RFC	0.807	0.016	0.895
MLP, ADA, GBC, RFC	0.797	0.005	0.896	MLP, GNB, ADA, GBC	0.768	0.013	0.878
MLP, GNB, GBC, RFC	0.800	0.005	0.897				

Table 6
Ensemble all algorithms performance.

Algorithms	*Sensitivity*	*Fallout*	*AUC*
MLP, GNB, ADA, GBC, RFC	0.769	0.002	0.884

shown in Table 3. In order to underline the differences with respect to the native performance gained by each single algorithm, the table reports this information (*Native* columns) beside the performance gained by using the proposed approach (*Model* columns).

Afterwards, we tested our model by combining the algorithms in pairs, triples, quadruples, and finally by using all of the algorithms. The experimental results are reported in Tables 4–6.

They show that our model applied on a single algorithm reaches the best result by using *Random Forests*, which compared to the native performance indicates an improvement in terms of *fraudulent* transactions correctly detected (0.807% instead of 0.653%), slightly increasing the value of *Fallout* (0.016% instead of 0.000%) but improving that of *AUC* (0.896% instead of 0.827%).

Also by combining the algorithms in pairs, triples, and quadruples, we obtain the best results when *Random Forests* is involved. This is in line with other studies in literature which indicate this algorithm [44] as one of the best approaches in these kind of tasks within the proposed domain [4,45,46].

Moreover, the results in Table 5 indicate the configuration based on four algorithms as the most promising, since by using the combination of the *MLP, GNB, GBC*, and *RFC* we get the best performances in terms of all the considered metrics.

The reader notices that the above results (even in the case of the single algorithms), have been calculated using our *PMC* as decision strategy.

6. Experimental environment

This section provides details on the experimental environment, on the dataset and the performed strategy, and on the metrics used to evaluate the classification performance.

6.1. Technological environment

The development environment used to implement the approach presented in this paper is based on the *Python* language: the

Table 7
Dataset details.

Transactions $\vert T\vert$	Legitimate $\vert T_+\vert$	Fraudulent $\vert T_-\vert$	Features $\vert F\vert$	Classes $\vert C\vert$
284,807	284,315	492	30	2

scikit-learn[6] libraries have been used to implement the state-of-the-art algorithms. In order to ensure the reproducibility of the experiments we have carried out, the seed of the pseudo-random number generator used by the *scikit-learn* classification algorithms has been set to *1*.

6.2. Dataset

The real-world dataset[7] used for the experiment contains a series of transactions related to European cardholders and executed in two days of *2013*. As shown in Table 7, such a dataset presents a high degree of data imbalance [47], since only *492* out of *284,807* transactions are classified as *fraudulent* (i.e., the *0.0017%*). All the information in the dataset have been anonymized, except those related to the *time* and the *amount*, which contain, respectively, the seconds elapsed since the first transaction in the dataset and the amount of the underlying transaction.

6.3. Strategy

In order to respect the transaction chronology, instead of a canonical *k-fold cross-validation* criterion we used the *TimeSeriesSplit scikit-learn* function to perform a *time series cross-validation* criterion. Such a function allows us splitting our dataset in a series of training and test sets, respecting the transactions chronology. For the experiments we used the *TimeSeriesSplit* function with *n_splits=10*.

The data imbalance problem, previously described in Section 2.3, has not been faced during the experiments. As suggested in [48], we have preferred to evaluate the effectiveness of our approach without any kind of data preprocessing (e.g., *under-sampling* or *over-sampling* balancing process) because in some cases the undersampling can potentially remove important samples whereas the oversampling can lead to overfitting and increase the computational load when the dataset is already fairly large.

The existence of a statistical significance between the obtained results has been verified by using the independent-samples *two-tailed Student's t-tests* ($p < 0.05$).

6.4. Metrics

According to the considerations made in Section 2.5, the performance of the involved algorithms has been evaluated by using three metrics: the *Sensitivity*, the *Fallout*, and the *AUC* (i.e., Area Under the *ROC* Curve). As we mentioned before, such metrics have been chosen because they provide information about the performance in terms of *fraudulent* transactions correctly classified (*Sensitivity* and *Fallout*), a crucial indicator in the context taken into account, and in terms of effectiveness of the adopted evaluation model (*AUC*).

In order to evaluate the algorithm performance also in terms of correct and incorrect classification of the *legitimate* transactions, we took into account two more metrics, which provide specular information with respect to the *Sensitivity* and *Fallout*: the *Specificity* and the *Miss Rate*.

The formulation of all the aforementioned metrics is presented below:

6.4.1. Sensitivity

The *Sensitivity* is calculated as reported in Eq. (4), where $\hat{T}$ is the set of new transactions to classify, *TP* is the number of transactions correctly classified as *fraudulent*, and *FN* is the number of *legitimate* transactions erroneously classified as *fraudulent*.

$$Sensitivity(\hat{T}) = \frac{TP}{(TP + FN)} \tag{4}$$

6.4.2. Fallout

The *Fallout* is calculated as reported in Eq. (5), where $\hat{T}$ is the set of new transactions to classify, *FP* is the number of *fraudulent* transactions erroneously classified as *legitimate*), and *TN* is the number of transactions correctly classified as *legitimate*.

$$Fallout(\hat{T}) = \frac{FP}{(FP + TN)} \tag{5}$$

6.4.3. AUC

The *AUC* is calculated as reported in Eq. (6), where given the subsets of the past *legitimate* T_+ and the past *fraudulent* transactions I_-, Ψ indicates all the possible comparisons between these subsets (i.e., T_+ and T_-). Its result will be given by averaging all the comparisons and will lie within the interval [0,1], where *1* denotes the best performance.

$$\Psi(i_+, i_-) = \begin{cases} 1, & if\ i_+ > i_- \\ 0.5, & if\ i_+ = i_- \\ 0, & if\ i_+ < i_- \end{cases} \qquad AUC = \frac{1}{|I_+| \cdot |I_-|} \sum_{1}^{|I_+|} \sum_{1}^{|I_-|} \Psi(i_+, i_-) \tag{6}$$

6.4.4. Specificity

The *Specificity* is calculated as reported in Eq. (7), where $\hat{T}$ is the set of new transactions to classify, *TN* is the number of transactions correctly classified as *legitimate*, and *FP* is the number of *fraudulent* transactions erroneously classified as *legitimate*.

$$Specificity(\hat{T}) = \frac{TN}{(TN + FP)} \tag{7}$$

6.4.5. Miss rate

The *Miss Rate* is calculated as reported in Eq. (8), where $\hat{T}$ is the set of new transactions to classify, *FN* is the number of *legitimate* transactions erroneously classified as *fraudulent*, and *TP* is the number of transactions correctly classified as *fraudulent*.

$$Miss\ Rate(\hat{T}) = \frac{FN}{(FN + TP)} \tag{8}$$

7. Results

This section reports the results of the performed experiments by comparing our solution to single and multiple algorithms approaches. Discussions on the results are also highlighted.

7.1. Single algorithm

Fig. 3 summarizes the comparison between our solution based on the *PMC* model and each single algorithm. Results indicate *Sensitivity, Fallout, Specificity, Miss Rate*, and *AUC* values. Please note that the *Sensitivity, Fallout*, and *AUC* values for the *PMC* model are the same shown in Table 5 using the best combination found (*MLP, GNB, GBC, RFC*). The reader notices that, we have not included the *Adaptive Boosting* (*ADA*) algorithm as indicated by a preliminary study discussed in Section 5.

[6] http://scikit-learn.org.

[7] https://www.kaggle.com/mlg-ulb/creditcardfraud.

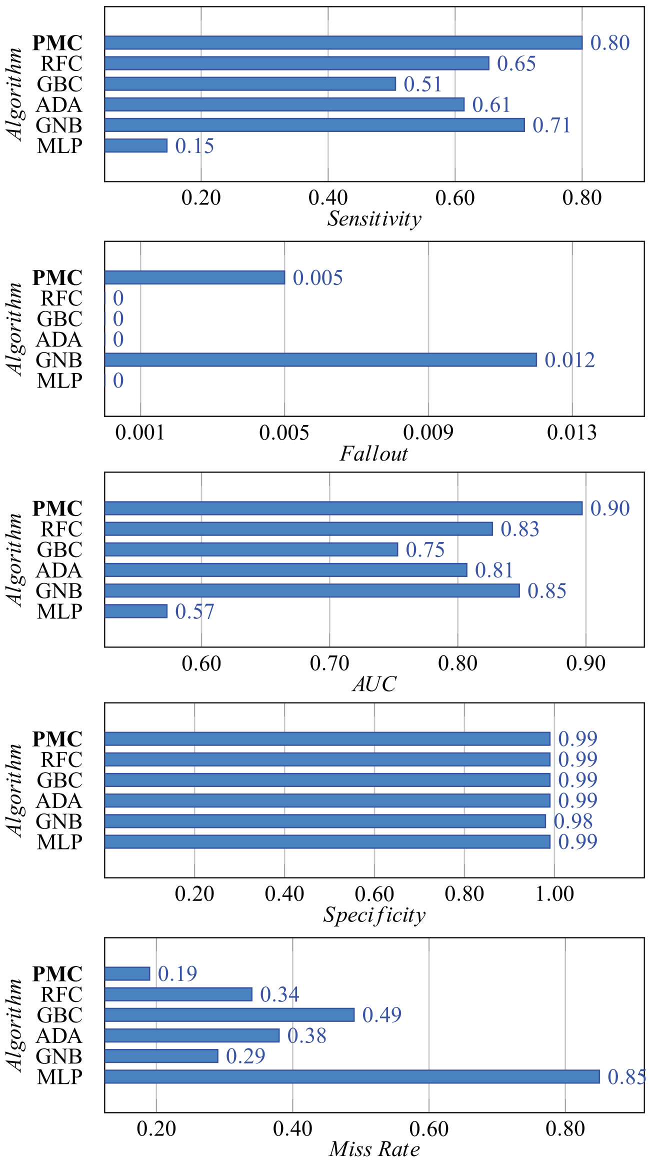

Fig. 3. *Specificity, Miss Rate, AUC, Sensitivity, and Fallout.*

As reported in Section 6.3, all the experiments have been performed according to a *time series cross-validation* criterion and after a thorough analysis of their results shown in Fig. 3, we can made the following observations:

- As *fraudulent* transactions in the dataset were *492*, our approach correctly detected *394* of them (*Sensitivity* = 0.800), with only *2* misclassifications (*Fallout* = 0.005);
- Compared to the best single approach (i.e., *GNB*), which correctly detected *349 fraudulent* transactions, with *6* misclassifications, this means that our approach had a gain of *9.1%*;
- Our solution was able to outperform the other algorithms in terms of *AUC* metric;
- The improvement of our approach is further confirmed in terms of *Specificity* and *Miss Rate*, as they prove that the obtained gain in terms of *Sensitivity* does not depend on a mere increase of *fraudulent* classifications made by our evaluation model.

7.2. Multiple algorithms

Here, we report the results of the last sets of experiments, which were aimed to compare the performance of the proposed approach, based on the *PMC* model, with that of the canonical state-of-the-art models used to manage the ensemble strategy of classification, and to a state-of-the-art solution that operates by adopting a *Bagging* method based on the *Decision Tree* algorithm [35].

Table 8
Ensemble strategies comparison (four algorithms).

Strategy	*Sensitivity*	*Fallout*	*AUC*	*Specificity*	*Miss rate*
Completeagreement	0.08	0.000	0.54	0.99	0.91
Majorityvoting	0.68	0.000	0.84	0.99	0.31
Weightedvoting	0.55	0.000	0.77	0.99	0.44
PMC	0.80	0.005	0.89	0.99	0.19

Table 9
Ensemble strategies comparison (all algorithms).

Strategy	*Sensitivity*	*Fallout*	*AUC*	*Specificity*	*Miss rate*
Completeagreement	0.06	0.000	0.53	1.00	0.93
Majorityvoting	0.63	0.000	0.81	0.99	0.36
Weightedvoting	0.63	0.000	0.81	0.99	0.36
PMC	0.80	0.005	0.89	0.99	0.19

Table 10
Dataset composition.

Dataset part	Legitimate	Fraudulent	Total	Fraud rate
*P*1	60,415	163	60,578	0.0026%
*P*2	77,338	101	77,439	0.0013%
*P*3	110,942	189	111,131	0.0017%
*P*4	35,620	39	35,659	0.0010%
Total	284,315	492	284,807	

7.2.1. Model comparison

This set of experiments is aimed at comparing our strategy based on the *PMC* model with the strategies such as the *complete agreement*, the *majority voting* (i.e., classification based on the majority of the classifications made by the algorithms), and the *weighted voting* (i.e., classification based on the weight of the classifications made by the algorithms, in terms of class probability) between the algorithms. The experiments have been conducted by using the four chosen algorithms presented in Section 5 (i.e., the same algorithms used in our approach) and all the algorithms. The results reported in Tables 8 and 9 indicate that our *PMC* approach outperforms other ensemble approaches which use different decision strategies. Please note that the *Sensitivity, Fallout, AUC, Specificity*, and *Miss Rate* values for the *PMC* model have been found using the best combination found of Table 5 (MLP, GNB, GBC, RFC).

7.2.2. Algorithm comparison

The last set of experiments has been performed in order to evaluate our approach with a state-of-the-art approach employing a *Bagging* method based on the *Decision Tree* algorithm [35] (denoted as *BDT*). According to the experimental criteria formalized in [35], our dataset has been divided into four parts by following the same percentage criterion (i.e., $P1 = 21.27\%$, $P2 = 27.19\%$, $P3 = 39.02\%$, $P4 = 12.52\%$), as shown in Table 10. The results of the experiments are reported in Table 11, which contains both the performances in terms of *Sensitivity, Fallout*, and *AUC* calculated on each part of the dataset (Table 10), and their average value calculated on all the dataset parts.

7.3. Discussion

The results highlighted in Section 7 indicate that the proposed approach based on our *PMC* model is able to improve the

Table 11
Ensemble algorithm comparison (bagging and decision tree).

Approach	*Dataset*	*Sensitivity*	*Fallout*	*AUC*
BDT	P1	0.75	0.010	0.81
BDT	P2	0.70	0.010	0.80
BDT	P3	0.65	0.019	0.79
BDT	P4	0.71	0.019	0.78
BDT	Average	0.70	0.014	0.79
PMC	P1	0.85	0.005	0.88
PMC	P2	0.80	0.005	0.89
PMC	P3	0.83	0.004	0.87
PMC	P4	0.77	0.006	0.80
PMC	Average	0.81	0.005	0.86

performance of a fraud detection system in terms of number of *fraudulent* transactions correctly classified.

Such an achievement is related to the value of *Sensitivity* (i.e., *0.800*) and *Fallout* (i.e., *0.005*) that, respectively, indicate its capability to correctly classify 9.1% of *fraudulent* transactions more than the best competitor algorithm (GNB, which has a *Sensitivity* value of *0.71*).

The results in terms of *AUC* metric underline the effectiveness of our evaluation model, proving its ability to classify new transactions as *legitimate* or *fraudulent*).

The evaluation in terms of *Specificity* and *Miss Rate* confirms the above results, showing that the increase of correctly classified *fraudulent* transactions implies a more robust and precise model.

The experiments aimed at comparing our method with other combined approaches show the effectiveness of our evaluation model (compared to the state-or-art based on the *complete agreement, majority voting*, and *weighted voting* criteria) in two different configurations, i.e. by using the four algorithms selected in Section 5 and by using all the algorithms.

The last set of experiments, where the performance of our approach has been compared to that of a performing state-of-the-art approach that uses a *Bagging* method based on the *Decision Tree* algorithm, also show that our *PMC* model outperforms its competitor, since it obtains best average performances in terms of *Sensitivity, Fallout*, and *AUC*.

Summarizing, we have proved that in real-world scenarios, characterized by a high degree of data imbalance, the proposed *PMC* model can significantly improve a fraud detection system, reducing the losses related to the misclassification of the fraudulent events.

The reader notices that the rationale of the *PMC* method, and the reason why it works well in the proposed domain, is because legitimate transactions are much higher in number and usually share a similar pattern easy to recognize. During the classification, several algorithms are thus able to assess with higher precision whether a transaction is legitimate. On the other hand, when a sample is fraudulent, most of the algorithms return a lower probability (confidence value) on their classification (either legitimate or fraudulent) and that is likely to be fraudulent. We have modeled this behaviour in our proposed *PMC* algorithm and this is why we obtain such high performances.

8. Conclusions and future work

In our era of big data, data intelligence and data security are very important research topics, and present constant challenges for academia and industry. The rapid evolution of the E-commerce platforms is an example of the increasing number of financial transactions made by electronic instruments of payment such as credit cards. Malicious people try to steal sensitive information from these transactions creating huge risks for the entire ecosystem. This is why, *Fraud Detection Systems*, especially those oriented to discover credit card frauds, are becoming more and more important.

The approach proposed in this paper, based on a novel *Prudential Multiple Consensus* model, addresses this problem and its risks associated with the aim of identify fraudulent transactions with higher precision than several state-of-the-art classification approaches. Our ensemble approach is able to reduce some well known problems that affect this kind of classification tasks, first of all the issue related to the data imbalance, improving the classification performance in terms of number of frauds correctly detected.

All the performed experiments have been conducted by involving a real-world dataset characterized by a high degree of data imbalance, and the performances of our approach have been compared to those of several state-of-the-art solutions, both single and combined strategies, proving its effectiveness in terms of *Sensitivity* and *AUC*.

A future work would be to evaluate the proposed approach in other scenarios also characterized by a high degree of data imbalance, as well as the experimentation of new aggregation strategies based on the Artificial Neural Network.

Acknowledgments

This research is partially funded by: *Regione Sardegna* under project *Next generation Open Mobile Apps Development* (NOMAD), *Pacchetti Integrati di Agevolazione* (PIA) - *Industria Artigianato e Servizi* - Annualitá 2013; Italian Ministry of Education, University and Research - Program Smart Cities and Communities and Social Innovation project ILEARNTV (D.D. n.1937 del 05.06.2014, CUP F74G14000200008 F19G14000910008); Sardinia Regional Government (Convenzione triennale tra la Fondazione di Sardegna e gli Atenei Sardi Regione Sardegna – L.R. 7/2007 annualità 2016 – DGR 28/21 del 17.05.2016, CUP: F72F16003030002).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jisa.2019.02.007.

References

[1] Abdallah A, Maarof MA, Zainal A. Fraud detection system: a survey. J Netw Comput Appl 2016;68:90–113.

[2] Japkowicz N, Stephen S. The class imbalance problem: a systematic study. Intell Data Anal 2002;6(5):429–49.

[3] He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng 2009;21(9):1263–84. doi:10.1109/TKDE.2008.239.

[4] Brown I, Mues C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst Appl 2012;39(3):3446–53. doi:10.1016/j.eswa.2011.09.033.

[5] Holte RC, Acker L, Porter BW. Concept learning and the problem of small disjuncts. In: Sridharan NS, editor. Proceedings of the 11th international joint conference on artificial intelligence. Detroit, MI, USA, August 1989. Morgan Kaufmann; 1989. p. 813–18.

[6] Lek M, Anandarajah B, Cerpa N, Jamieson R. Data mining prototype for detecting e-commerce fraud. In: Smithson S, Gricar J, Podlogar M, Avgerinou S, editors. Proceedings of the 9th european conference on information systems, global co-operation in the New Millennium, ECIS 2001, Bled, Slovenia, June 27–29; 2001. p. 160–5.

[7] Carneiro N, Figueira G, Costa M. A data mining based system for credit-card fraud detection in e-tail. Decis Support Syst 2017;95:91–101.

[8] Hoffman AJ, Tessendorf RE. Artificial intelligence based fraud agent to identify supply chain irregularities. In: Hamza MH, editor. IASTED international conference on artificial intelligence and applications, part of the 23rd multi--conference on applied informatics, Innsbruck, Austria, February 14–16, 2005. IASTED/ACTA Press; 2005. p. 743–50. ISBN 0-88986-459-4.

[9] Whiting DG, Hansen JV, McDonald JB, Albrecht CC, Albrecht WS. Machine learning methods for detecting patterns of management fraud. Comput Intell 2012;28(4):505–27.

[10] Nolan I. Transaction fraud detection using random forest classifier and logistic regression. Neural Netw Mach Learn 2017;1(1). 2–2

[11] Assis C, Pereira AM, de Arruda Pereira M, Carrano EG. Using genetic programming to detect fraud in electronic transactions. In: Prazeres CVS, Sampaio PNM, Santanchè A, Santos CAS, Goularte R, editors. A comprehensive survey of data mining-based fraud detection research; 2010. p. 337–40. arXiv:1009.6119. ISBN 978-1-4503-2559-2.

[12] Mead A, Lewris T, Prasanth S, Adams S, Alonzi P, Beling P. Detecting fraud in adversarial environments: a reinforcement learning approach. In: Systems and information engineering design symposium (SIEDS), 2018. IEEE; 2018. p. 118–22.

[13] Wang W, Lu D, Zhou X, Zhang B, Mu J. Statistical wavelet-based anomaly detection in big data with compressive sensing. EURASIP J Wirel Comm Netw 2013;2013:269.

[14] Saia R. A discrete wavelet transform approach to fraud detection. NSS. Lecture notes in computer science, 10394. Springer; 2017. p. 464–74.

[15] Saia R, Carta S. Evaluating credit card transactions in the frequency domain for a proactive fraud detection approach. In: SECRYPT. SciTePress; 2017. p. 335–42.

[16] Saia R, Carta S. A frequency-domain-based pattern mining for credit card fraud detection. In: IoTBDS. SciTePress; 2017. p. 386–91.

[17] Huang SY, Lin C, Chiu A, Yen DC. Fraud detection using fraud triangle risk factors. Inf Syst Front 2017;19(6):1343–56.

[18] Saia R. Unbalanced data classification in fraud detection by introducing a multidimensional space analysis. In: IoTBDS. SciTePress; 2018. p. 29–40.

[19] Bolton RJ, Hand DJ. Statistical fraud detection: a review. Stat Sci 2002:235–49.

[20] Phua C, Lee VCS, Smith-Miles K, Gayler RW. A comprehensive survey of data mining-based fraud detection research; 2010. arXiv:1009.6119.

[21] Pozzolo AD, Caelen O, Borgne YL, Waterschoot S, Bontempi G. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl 2014;41(10):4915–28. doi:10.1016/j.eswa.2014.02.026.

[22] Wang H, Fan W, Yu PS, Han J. Mining concept-drifting data streams using ensemble classifiers. In: Getoor L, Senator TE, Domingos PM, Faloutsos C, editors. Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, Washington, DC, USA, August 24, - 27, 2003. ACM; 2003. p. 226–35. doi:10.1145/956750.956778.

[23] Gao J, Fan W, Han J, Yu PS. A general framework for mining concept-drifting data streams with skewed distributions. In: Proceedings of the seventh SIAM international conference on data mining, April 26–28, 2007, Minneapolis, Minnesota, USA. SIAM; 2007. p. 3–14. doi:10.1137/1.9781611972771.1.

[24] Barse EL, Kvarnström H, Jonsson E. Synthesizing test data for fraud detection systems. In: ACSAC. IEEE Computer Society; 2003. p. 384–94.

[25] Chatterjee A, Segev A. Data manipulation in heterogeneous databases. ACM SIGMOD Record 1991;20(4):64–8.

[26] Sorournejad S, Zojaji Z, Atani RE, Monadjemi AH. A survey of credit card fraud detection techniques: data and technique oriented perspective; 2016. arXiv:1611.06439.

[27] Attenberg J, Provost FJ. Inactive learning?: difficulties employing active learning in practice. SIGKDD Explor 2010;12(2):36–41. doi:10.1145/1964897.1964906.

[28] Zareapoor M, Yang J. A novel strategy for mining highly imbalanced data in credit card transactions. Intell Autom Soft Comput 2017:1–7.

[29] Vinciotti V, Hand DJ. Scorecard construction with unbalanced class sizes. J Iran Stat Soc 2003;2(2):189–205.

[30] Gomes HM, Barddal JP, Enembreck F, Bifet A. A survey on ensemble learning for data stream classification. ACM Comput Surv 2017;50(2) 23:1-23:36.

[31] Dietterich TG. Ensemble methods in machine learning. Multiple classifier systems. Lecture notes in computer science, 1857. Springer; 2000. p. 1–15.

[32] Akila S, Reddy US. Risk based bagged ensemble (RBE) for credit card fraud detection. In: Inventive computing and informatics (ICICI), international conference on. IEEE; 2017. p. 670–4.

[33] Guo H, Li Y, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: review of methods and applications. Expert Syst Appl 2017;73:220–39.

[34] Faghani M, Nordin MJ, Shojaeipour S. Optimization of the performance face recognition using AdaBoost-based. In: Chen R, editor. Intelligent computing and information science. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 359–65. ISBN 978-3-642-18134-4.

[35] Zareapoor M, Shamsolmoali P. Application of credit card fraud detection: based on bagging ensemble classifier. Procedia Comput Sci 2015;48:679–85.

[36] Wu D, Liu Y, Gao G, Mao Z, Ma W, He T. An adaptive ensemble classifier for concept drifting stream. In: Computational intelligence and data mining, 2009. CIDM'09. IEEE symposium on. IEEE; 2009. p. 69–75.

[37] Bian Y, Cheng M, Yang C, Yuan Y, Li Q, Zhao JL, et al. Financial fraud detection: a new ensemble learning approach for imbalanced data. In: PACIS; 2016. p. 315.

[38] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55(1):119–39.

[39] Natekin A, Knoll A. Gradient boosting machines, a tutorial. Front Neurorobot 2013.

[40] Adewumi AO, Akinyelu AA. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. Int J Syst Assur Eng Manage 2017;8(2):937–53.

[41] Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation; 2011.

[42] Faraggi D, Reiser B. Estimation of the area under the ROC curve. Stat Med 2002;21(20):3093–106.

[43] Sergio AT, de Lima TPF, Ludermir TB. Dynamic selection of forecast combiners. Neurocomputing 2016;218:37–50.

[44] Breiman L. Random forests. Mach Learn 2001;45(1):5–32. doi:10.1023/A:1010933404324.

[45] Lessmann S, Baesens B, Seow H, Thomas LC. Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur J Oper Res 2015;247(1):124–36. doi:10.1016/j.ejor.2015.05.030.

[46] Bhattacharyya S, Jha S, Tharakunnel KK, Westland JC. Data mining for credit card fraud: a comparative study. Decis Support Syst 2011;50(3):602–13. doi:10.1016/j.dss.2010.08.008.

[47] Pozzolo AD, Caelen O, Johnson RA, Bontempi G. Calibrating probability with undersampling for unbalanced classification. In: IEEE symposium series on computational intelligence, SSCI 2015, Cape Town, South Africa, December 7–10, 2015. IEEE; 2015. p. 159–66. doi:10.1109/SSCI.2015.33.

[48] Chawla NV, Japkowicz N, Kotcz A. Special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl 2004;6(1):1–6.