An Assessment of Streaming Active Learning Strategies for Real-Life Credit Card Fraud Detection

Fabrizio Carcillo*, Yann-Aël Le Borgne*, Olivier Caelen† and Gianluca Bontempi*
*Machine Learning Group
Computer Science Department
Faculty of Sciences ULB
Université Libre de Bruxelles
Brussels, Belgium
Email: fcarcill, yleborgn@ulb.ac.be and gbonte@ulb.ac.be
†R&D, Worldline
Brussels, Belgium
Email: olivier.caelen@worldline.com

***Abstract*—Credit card fraud detection raises unique challenges due to the streaming, imbalanced, and non-stationary nature of transaction data. It additionally includes an active learning step, since the labeling (fraud or genuine) of a subset of transactions is obtained in near-real time by human investigators contacting the cardholders. These challenges and characteristics have traditionally been studied separately in the literature. In this paper, we investigate how previously proposed techniques can be combined to improve fraud detection accuracy. In particular, we highlight the existence of an exploitation/exploration tradeoff for active learning in the context of fraud detection, which has so far been overlooked in the literature. Relying on a real-world dataset of millions of transactions provided by our industrial partner Worldline, we performed an extensive experimental analysis in order to assess how traditional active learning strategies can be improved by using complementary machine learning techniques. We find that the baseline active learning strategy, denoted High Risk Querying, is a robust strategy, which can be further improved by combining it with Semi-Supervised learning.**

1. Introduction

The use of machine learning for credit card fraud detection requires to address three major problems: the class imbalance of the training set (many more genuine transactions than fraudulent ones), the nonstationarity of the phenomenon (due to changes in the behavior of customers as well as in fraudsters) and the labeling bottleneck (due to the cost of assessing transactions by human investigators) [1]. In this context, an automatic Fraud Detection System (FDS) should support the activity of the investigators by letting them focus on the transactions with the highest fraud probability. From the perspective of the transactional service company, this is crucial in order to reduce the costs of the investigation activity and keep high the confidence of the customers. From a machine learning perspective, however, it is important to keep an adequate balance between exploitation and exploration, i.e. between the short-term needs of providing good alerts to investigators, and the long-term goal of maintaining a high accuracy of the system (e.g. in the presence of concept drift).

The issue of labeling the most informative data by minimizing the cost has been extensively addressed by active learning which can be considered as a specific instance of semi-supervised learning [2], [3], the domain studying how unlabeled and labeled data can both contribute to a better learning process. Active learning (AL) literature proposed a number of techniques to select, within a large set of unlabeled instances, the most informative to label. Active learning can be typically described by the iteration of three steps [3]:

1) *Selection*: given a query budget of size k, a model is used to choose which k data points, once labeled, could better describe the data generating process;
2) *Querying*: selected points are submitted to an *oracle* (in our case an investigator) for labeling;
3) *Training*: the labeled data points are used to train/update the model;

The *Selection* and *Querying* steps differentiate active learning from conventional (passive) learning, which is limited to the *Training* dataset. Note also that, in order to bootstrap the procedure, a random initialization or an unsupervised model is commonly used [3].

We can distinguish between two main AL approaches according to the nature of the dataset: *pool-based* and *streaming* AL. In pool-based AL, the algorithm performs queries in the same set of unlabeled points, while in stream-based AL, the set of unlabeled data points is periodically updated. The accuracy of a pool-based AL classifier is

expected to grow in time, since more and more labeled data points are used for the *Training*. This is not always true in the case of the streaming approach, since data received in different periods may differ significantly (e.g. concept drift).

Though the number of AL works is large [4], [5], very few address the other aspects of credit card fraud detection (Section 2), notably the class imbalance [6], [7] (in our case study approximately only 0.2% of transactions are fraudulent), the restricted labeling budget [8], [9] and the specific nature of the assessment [10]. In fact, it is worth to remark that in credit card fraud detection, though the observations are at the level of transactions, the ultimate goal is to detect fraudulent cards, i.e. there exists a one-to-many relation between cards and transactions. Also what is peculiar is that the labeling and the assessment phase are coincident (and consequently strongly dependent), in the sense that what matters more for the transactional company is that investigations are as successful as possible (i.e. low false positive rate). This means that the accuracy of a FDS is measured in terms of precision over the top k alerted credit cards [11], [12]. This is not always the case in other AL tasks where the labels of the k queried points are not directly related to the accuracy of the training process. In a FDS it is not only important to minimize the labeling cost but also that this labeling allows to discover as many frauds as possible. This means that the nature and the intensity of the exploration step has a strong impact on the final accuracy of detection and that, accordingly, the set of state-of-the-art AL strategies which are effective in practice is much more limited than expected. In other terms no real-life FDS could afford a totally random labeling process since this would necessarily imply an unacceptable short-term random performance. This exploitation/exploration tradeoff inherent to fraud detection has, to the best of our knowledge, not yet been addressed in the research literature.

The main contributions of this paper are: i) an extensive comparison of streaming AL techniques (and a number of their variants) for the detection of both fraudulent transactions and cards and ii) an experimental assessment of their performances on the basis of a massive set of 12 million transactions and in terms of real-life criteria (defined by our industrial partner, Worldline, a leader company in transactional services). The outcome is an original analysis of the exploitation/exploration trade-off in the context of a real-world FDS.

The work is organized as follows. Section 2 presents the related state-of-the-art. Section 3 provides an overview of our Fraud Detection System. Section 4 discusses a number of AL strategies (as well as possible variants) for dealing with streaming credit card transactions. Finally, Section 5 presents an extensive experimental session based on a real stream of transactions.

2. Related Work

The application of active learning to the specific characteristics of fraud detection has only been partially addressed in the research literature. Fan et al. carried out an empirical analysis on a fraud detection dataset [13] to assess AL approach in presence of concept drift. They focused on the adaptation ability of the AL strategy, but did not address the detection accuracy. Pichara et al. [14] tested a large scale anomaly detection approach in a synthetic dataset emulating the fraud process. Their AL schema was able to detect the whole subset of frauds using a number of queries smaller than a bayesian network detection approach. Multiple tests were repeated using different data-noise levels, and their AL consistently outperformed the other technique. The fact that they have used a synthetic dataset, is a limitation to their results. It is very difficult to create a fair synthetic dataset, since frauds are very diverse and they evolve in an unpredictable way. Van Vlasselaer et al. [15] applied active inference, a network-based algorithm, to fraud discovery in social security real data. They found that committee-based strategies, based on uncertainty, resulted in a slightly better classification performance than expert-based strategies. Nevertheless expert-based strategies are often preferred in order to obtain unbiased training sets from queries.

The relationship between active learning and streaming data, notably the sampling bias issue, is discussed in [16]. Authors showed that in stream-based active learning, the estimated input-output dependency changes over time and depends on previously queried instances. Since those instances are typically selected next to the class decision boundary of the classifier, this may lead to a biased representation of the underlying data distribution. AL and concept drift is also addressed in [17], who stressed how concept drift may be missed in regions far from where AL queries normally take place (e.g. boundary regions between classes). The authors showed that techniques based on classical uncertainty sampling favor close concept drift adaptation while techniques based on random sampling are more effective in dealing with remote concept drift. Nevertheless, the best performing techniques can strongly depend on the characteristics of the data and the size of the query budget.

The integration of AL and semi-supervised learning technique is discussed in Xie and Xiong [18]. They introduced a Stochastic Semi-supervised Learning (SSSL) process to infer labels in case of large imbalanced datasets with small proportion of labeled points. The approach relies on the consideration that since the number of unlabeled points is huge and the minority class is rare, the probability of making a wrong majority assignment is very low. Consequently they proposed the assignment of the majority class to random selection of points and adopted it with success in the context of a data competition.

Finally, an original approach that may be used to deal with the one-to-many relationships between cards and transactions, is discussed in [10]. They present an AL approach for multiple-instance problems where instances are organized into *bags*. Typical examples of multiple-instance problems are found in text classification and content-based image retrieval. In those type of problems a bag is said to be positive if it includes at least one instance which is positive, while the bag is negative if no positive instances are observed in it.

3. The FDS classifier

Let us consider a FDS whose goal is to detect automatically frauds in a stream of transactions. Let x be the vector coding the transaction (e.g. including features like the transaction amount, the terminal) and $y \in \{+, -\}$ the associated label, where $+$ denotes a fraud and $-$ a genuine transaction. A detection strategy needs a measure of risk (score) associated to any transaction. In a machine learning approach this score is typically provided by the estimation of the a posteriori probability $\mathcal{P}_{\mathcal{C}}(+|x)$ returned by a classifier $\mathcal{C}$. We consider a streaming setting where unlabeled transactions arrive one at a time or in small batches.

The FDS goal is to raise every day a fixed and small number of k alerts. In our industrial case study k is set to 100 on the basis of cost and work organization considerations.

The issuing of those alerts has two consequences: the trigger of an investigation and the consequent labeling of the associated transactions. The outcome of the investigation determines both the success rate of the FDS and the new set of labeled transactions.

Section 5 will present two levels of experimental validations: the first will concern the detection of fraudulent transactions, while the second will focus on fraudulent cards. In the first experiment, the classifier $\mathcal{C}$ is implemented by a conventional Random Forest, while in the second, we use a more complex approach (ensemble of classifiers) dictated by the more challenging nature of the detection tasks. This approach has been presented in [11], [12] and consists of the weighted average of two classifiers

$$\mathcal{P}_{\mathcal{C}}(+|x) = w^A \mathcal{P}_{\mathcal{D}_t}(+|x) + (1 - w^A)\mathcal{P}_{\mathcal{F}_t}(+|x) \quad (1)$$

where $\mathcal{D}_t$ and $\mathcal{F}_t$ stand for *Delayed classifier* and *Feedback classifier*, respectively, and $w^A \in [0, 1]$ is a weight controlling the contribution of the two classifiers. $\mathcal{D}_t$ is implemented as an ensemble of Balanced Random Trees [19], [20] trained on old transactions for which we can reasonably consider the class as known. $\mathcal{F}_t$ is trained on recently alerted transactions, for which a *Feedback* was returned by investigators. It is therefore alimented by the active learning component of the fraud detection system. This *Feedback* component is very important to address concept drift.

This architecture is the result of an extensive model selection and assessment procedure which have been discussed in our previous work [11], [12]. Since the aim of this paper is to discuss the impact of different AL strategies, we will not take into consideration alternative classifier architectures.

4. Active learning strategies

The rationale of AL is to select (on the basis of current information) unlabeled training samples which, once labeled, can improve the accuracy. However, there are two main unknowns concerning the effectiveness of AL in credit card fraud detection. The first concerns the strong imbalancedness of the class distribution: since the selection of adequate queries is the most important step of an AL procedure, this step should take into account that in such a large imbalanced problem, selecting majority class points will inevitably have a negligible impact on accuracy. The second concerns the definition of accuracy: measures of detection accuracy are strictly related to the capacity of discovering frauds, i.e. querying minority class samples. This means that an AL strategy for fraud detection requires some specific tuning for being successful.

Algorithm 1 Active Learning process

Require: k ▷ total number of alerts
Require: q ▷ exploration budget
Require: m ▷ SSSL budget
Require: D ▷ initial training set
for any new day **do**
  $\mathcal{C} \leftarrow$ learning(D)
  $inTrx \leftarrow$ unlabeled set
  $scores \leftarrow \{\mathcal{P}_{\mathcal{C}}(x), x \in inTrx\}$
  $sel \leftarrow$ points with highest risk $scores$ ▷ HRQ
  if $(q > 0)$ **then** ▷ EAL
    $Esel \leftarrow q$ explorative points
    $sel \leftarrow \{sel, Esel\}$
  end if
  $queries \leftarrow$ investigator labeling of sel
  if $(m > 0)$ **then** ▷ SSSL
    $SSSLset \leftarrow m$ points based on a SSSL criterion
    $SSSLset \leftarrow$ set label $y(SSSLset) = 0$
    $queries \leftarrow \{queries, SSSLset\}$
  end if
  $D \leftarrow \{D, queries\}$
end for

To illustrate the impact of AL on FDS, we will start by considering a baseline strategy which simply queries the highest risk transactions on the basis of the current classification model. This strategy will be denoted as the Highest Risk Querying (HRQ). Thereafter, we will introduce and assess a number of modifications of HRQ according to several principles. In order to make the comparison easier we will define each AL strategy as an instance of a generic AL strategy detailed in Algorithm 1. The Algorithm requires the specification of three parameters: the budget k of queries (i.e. maximum number of transaction that can be investigated per day), the number of q queries defined for exploration purposes and the number m of unlabeled transactions that can be set as genuine without investigation (see 4.3). The entire list of discussed AL strategies is presented in Table 1.

4.1. Highest Risk Querying

The idea of Highest Risk Querying is simple: given a classifier $\mathcal{C}$ and a budget of queries, HRQ returns the unlabeled transactions with the highest estimated *a posteriori* probability $\mathcal{P}_{\mathcal{C}}(+|x_i)$. Highest Risk Querying (HRQ) is the most intuitive active learning strategy for our problem if we consider that the final FDS accuracy depends on the

TABLE 1. SUMMARY OF ACTIVE LEARNING AND SEMI-SUPERVISED STRATEGIES DESCRIBED IN THE PAPER

Id	Strategy	Type
HRQ	Highest Risk Querying	Baseline / BL (section 4.1)
R	Random Querying	Exploratory Active Learning / EAL (section 4.2)
U	Uncertainty Querying	
M	Mix of Random and Uncertainty Querying	
SR	Stochastic Semi-supervised Learning (SSSL) on Random points	Stochastic Semi-Supervised Learning / SSSL (section 4.3)
SU	SSSL on Uncertain points	
SM	SSSL on Random/Uncertain points	
SE	SSSL on points most likely to be genuine	
SR-U	SSSL on Uncertain points + Random Sampling	SSSL + EAL (section 4.3)
SR-R	SSSL on Random points + Random Sampling	
SR-M	SSSL on Random/Uncertain points + Random Sampling	
ROS	Random Oversample	Oversample (section 4.4)
SMOTE	SMOTE	
QFU	Querying by Frequent Uncertainty	Multiple Instance Learning (sections 4.5)
MF-...	Max combining function	
SM-...	Softmax combining function	
LF-...	Logarithmic combining function	

amount of minority class querying. Note that in terms of the pseudocode in Algorithm 1, HRQ is obtained by setting $q = 0$ and $m = 0$.

HRQ is expected to have a positive impact on accuracy by discovering new instances from the minority class and improving consequently the balance of the training set. HRQ has also some drawbacks: since its querying strategy relies on the classifier accuracy, this selection step could be inaccurate especially at the very beginning.

4.2. Exploratory Active Learning

Exploratory Active Learning (EAL) strategies modify HRQ by trading exploitation for exploration. The idea is to convert a subset of the labeling budget in explorative queries. The size of the exploration budget is represented by $0 < q \leq k$ in Algorithm 1.

We may consider a number of exploration techniques for selecting the q exploratory transactions. The simplest one is random querying (denoted by EAL-R) which consists in choosing randomly the q query points. This solution can be sub-optimal since it may query points for which the classifier is already highly confident about the class.

An alternative is represented by uncertainty querying (EAL-U) which returns unlabeled data points for which the current classifier has low confidence [21]. Given a binary classifier $\mathcal{C}$, the uncertainty querying strategy gives priority to the transactions x_i for which $\mathcal{P}_{\mathcal{C}}(+|x_i) \approx 0.5$.

Žliobaitė et al. [17] proposed the mix of the two techniques, uncertainty querying and randomization, to tackle remote concept drift (Section 2). The technique (denoted by EAL-M) consists in querying by uncertainty most of the points and in querying random points from time to time.

4.3. Stochastic Semi-supervised Learning

The SSSL strategy has been introduced by Xie and Xiong [18] to infer labels in case of highly imbalanced datasets with a large number of unlabeled points. The strategy relies on a simple consideration: since the ratio between the number of frauds and the total number of transactions is very small, the probability of randomly selecting a fraud is very low.

The resulting AL learning schema is made of four steps:

1) *Selection*: the current model is used to annotate all unlabeled transactions with an estimated risk;
2) *Querying*: the highest risk transactions are labeled by the investigators;
3) *Majority assumption*: a number of transactions are labeled as genuine by majority assumption; in this paper we explore a number of criteria to attribute the majority class: pure random attribution (SR), uncertainty (SU), mix of randomness and uncertainty (SM) and low predicted risk (SE).
4) *Training*: the labeled data points, obtained by the previous steps, are used to train/update a supervised model.

It appears that this strategy differs in terms of the usage of the current model $\mathcal{C}$: the predicted risk is not only used to alert and trigger the investigation but also to label (without investigation) a number of low risk transactions.

In order to illustrate the reliability of the majority assumption, we report in Figure 4.5.1 the distribution of the scores $\mathcal{P}_{\mathcal{C}}(+|x_i)$ over 15 days. In particular the histograms (a), (b) and (c) refer to the score distribution for all transactions, genuine transactions and fraudulent transactions, respectively. The plot (d) represents the proportion of fraudulent and genuine transactions for a given score in the range $[0, 1]$. Note that, though the a priori proportion of fraudulent cards in the dataset is 0.13%, it becomes 23.33% for scores

higher than 0.95 and 61.90% for scores beyond 0.99. Also, in the area of maximal uncertainty for $\mathcal{C}$ (e.g between 0.49 and 0.51), we find only 0.35% of frauds.

On the basis of those considerations, it is possible to define a number of Stochastic Semi-Supervised strategies:

- SR: no exploration budget ($q = 0$) and attribution of the majority class to $m > 0$ random transactions;
- SU: no exploration budget ($q = 0$) and attribution of the majority class to the $m > 0$ most uncertain points;
- SM: no exploration budget ($q = 0$) and attribution of the majority class to the $0.7 \times m$ most uncertain points and to $0.3 \times m$ random points;
- SE: no exploration budget ($q = 0$) and attribution of the majority class to the $m > 0$ lowest risk points;

Additional variants can be created by simply allowing an exploration budget ($q > 0$). The SR-U, SR-R and SR-M strategies are hybrid strategies by combining an exploration strategy (e.g. U in SR-U) and a SSSL strategy (e.g. SR in SR-U).

4.4. Oversampling

It is worth noting that a side-effect of the adoption of SSSL (Section 4.3) is to add a number of majority class samples to the training set. This goal is typically achieved by oversampling techniques, with the main difference that here the target class is the majority class and not the minority one. In order to assess how SSSL situates with respect to conventional oversampling, we also consider a comparison with the two main oversampling techniques: Random Oversample (ROS) [22] and SMOTE [23]. ROS consists in duplicating some random instances from the class to be oversampled until a given sample size is reached. SMOTE creates artificial instances from the target class in the following manner: once the k nearest neighbors from the same class have been identified, new artificial transactions are generated moving along the line segment joining the original instance and its k neighbors.

4.5. Multiple Instance Learning

This section deals with another specificity of the credit card fraud detection problem: the observations take place at the level of transactions but what is relevant for the company is the detection at the card level, since the investigation is performed at the card level and not at the transaction level.

From an AL perspective, since multiple transactions map to the same card, we could select query points by taking advantage of such one-to-many relationship.

4.5.1. Querying by Frequent Uncertainty. The rationale of Querying by Frequent Uncertainty (QFU) boils down to query those cards which are mapped to the largest number of uncertain transactions. We associate to each card c a counter representing how many of its associated transactions $x_i \in c$ are uncertain, i.e. have a score $\mathcal{P}_{\mathcal{C}_t}(+|x_i) \in [0.5-v, 0.5+v]$ where v is set by the user. The counters are updated in real-time and the AL selection returns the k cards with the highest counters.

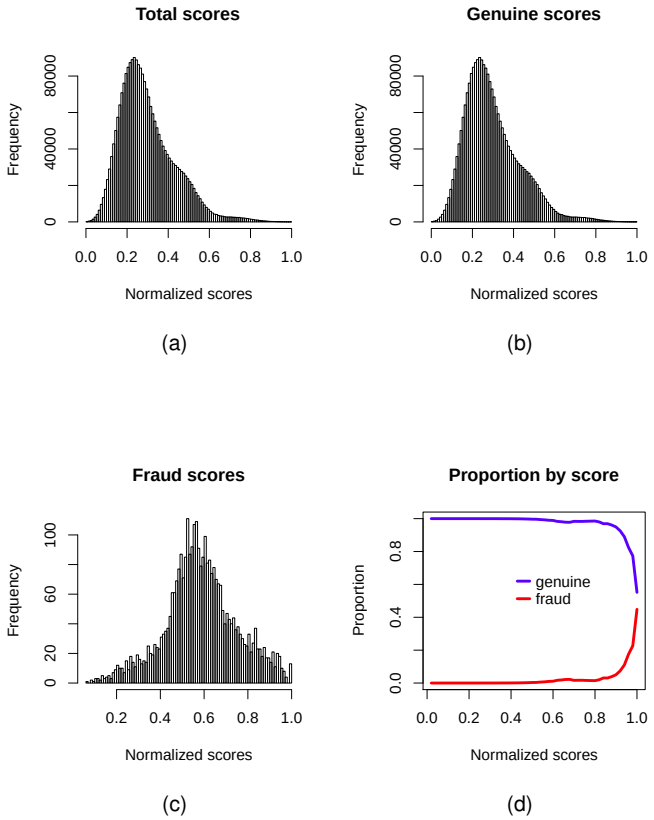

Figure 1. Distribution of the scores obtained by $\mathcal{P}_{\mathcal{A}_t}(+|x_i)$ in a range of 15 days involving 2.4 millions of cards for: (a) all the transactions (b) only genuine transactions and (c) only fraudulent transactions. In (d) the proportion of genuine and fraudulent transactions is plotted while changing the score obtained by $\mathcal{P}_{\mathcal{A}_t}(+|x_i)$.

4.5.2. Combining Function. A more advanced strategy to deal with card detection is inspired by [10]. A *combining function* can be used to aggregate all the posterior probabilities $p_i^c = \mathcal{P}_{\mathcal{C}}(+|x_i)$ of the transactions $x_i \in c$ and derive the posterior probability $\mathcal{P}_{\mathcal{C}}(+|c)$.

The simplest combining function is the *max* function (denoted MF), which returns

$$\mathcal{P}_{\mathcal{C}}^{MF}(+|c) = \max_{x_i \in c} p_i^c \tag{2}$$

Alternatively, authors in [10] propose the *softmax combining function*:

$$\mathcal{P}_{\mathcal{C}}^{SM}(+|c) = \frac{\sum_{x_i} p_i^c e^{\alpha p_i^c}}{\sum_{x_i} e^{\alpha p_i^c}} \tag{3}$$

where α is a constant that determines the extent to which *softmax* approximates a *max* function.

In order to i) increase the sensitivity of the card risk to high risk transactions and ii) reduce its sensitivity to low risk

TABLE 2. SCORING OF TRANSACTIONS

Rank	Card Id	Trx Id	p_i^c
1	A	A7	0.90
2	B	B3	0.88
3	B	B5	0.87
4	A	A2	0.83
...	...	...	...

TABLE 3. SCORING OF CARDS ON THE BASIS OF TRANSACTIONS FROM TABLE 2 WITH THREE COMBINING FUNCTIONS

Card Id	**max**(p_i^c)	**softmax**(p_i^c)	**logarithmic** (p_i^c)
A	**0.90**	0.87	34.21
B	0.88	**0.88**	**34.55**
...	...	...	...

TABLE 4. ADDITIONAL TRANSACTION

Rank	Card Id	Trx Id	p_i^c
20000	B	B6	0.20

TABLE 5. SCORING OF CARDS ON THE BASIS OF TRANSACTION FROM TABLES 2 AND 4 WITH THREE COMBINING FUNCTIONS

Card Id	**max**(p_i^c)	**softmax**(p_i^c)	**logarithmic** (p_i^c)
A	**0.90**	**0.87**	34.21
B	0.88	0.74	**34.55**
...	...	...	...

transactions, we propose a *logarithmic combining function* returning the score

$$\sum_{x_i \in c} -\frac{1}{\log s_i^c} \tag{4}$$

where $s_i^c = \begin{cases} p_i^c - \epsilon & \text{if } p_i^c > 0.5 \\ \epsilon & \text{otherwise.} \end{cases}$ and ϵ is a very small number.

Table 3 illustrates the scores associated to the transactions of Table 2 for the three combining functions presented above. It appears that, unlike the *max* function, the other two functions are able to take into account the impact of multiple risky transactions on the overall risk of a card. In other terms two high risk transactions weight more than a simple one with a marginal higher risk. However the softmax and the logarithmic functions differ in the importance they give to low risk transactions. Suppose we add a low risk transaction (Table 4) for card "B" to the set of transactions of Table 2. Table 5 shows that the sensitivity of the card risk to such additional transaction is much larger in the *softmax* than in the logarithmic case. The counter-intuitive consequence is that according to the *softmax* function the card "B" becomes now less risky than the card "A".

5. Experiments

This section relies on a large dataset of 12 million credit card transactions provided by our industrial partner Worldline. In this realistic case-study, only a very small number ($k = 100$) of cards per day can be queried, amounting to roughly 0.2% of labeled points. The dataset covers 60 days, each day including roughly $200K$ transactions.

Two sets of experiments are performed: the first measures the detection accuracy at the level of the transactions, while the second measures the detection accuracy at the card level. In the first study, for the sake of simplicity, the classification model $\mathcal{C}$ is a conventional random forest model $\mathcal{RF}$ while a more realistic model $\mathcal{A}$ (discussed in [11] and in Section 3) is used for the cards. Since the randomization process in $\mathcal{RF}$ and $\mathcal{A}$ may induce variability in the accuracy assessment, we present the results of twenty repetitions of the streaming.

All the AL strategies are compared in identical situations and initialized with the same random and balanced set (initial training set D presented in algorithm 1). The results are presented as box-plots summarizing the fraud detection performance over the 60 days. In particular we considered the following accuracy measures: Top100 Precision, Area Under the Precision-Recall Curve and Area Under the Receiver Operator Characteristic Curve. In all the plots, the dark boxes are used to denote the most accurate AL strategy as well as the ones which do not differ significantly from it (paired Wilcoxon signed rank test with 5% significance level).

Note that the expected precision over the Top100 alerts is expected to be larger for $\mathcal{RF}$ than $\mathcal{A}$ since multiple positive alerts for the same card will be accounted as several true positives in the transaction case but as a single success in the card case. We made all the code available on Github[1].

5.1. Transaction-based Fraud Detection

In Figure 2, we report the detection accuracy of the AL techniques discussed in Section 4. A horizontal line is added in order to make the comparison with the baseline strategy HRQ easier. The experiments are performed with $k = 100$, $q = 5$ and $m = 1000$. These hyperparameters have been set by trial-and-error and are compatible with the kind of exploration effort that our industrial partner could ask to its investigators.

It appears that exploratory AL alone is not able to outperform the standard HRQ strategy. Instead, the best results are obtained by combining SSSL with either randomization (SR) or uncertainty sampling (SR-U). In particular, the SR strategy leads to an improvement in precision of 5.84%, while SR-U leads to an improvement of 5.15%. Similar improvement are observed for the AU-PRC (Figure 2b), while a wider range of techniques perform better in terms of AU-ROC curve (Figure 2c).

1. https://github.com/fabriziocarcillo/StreamingActiveLearningStrategies

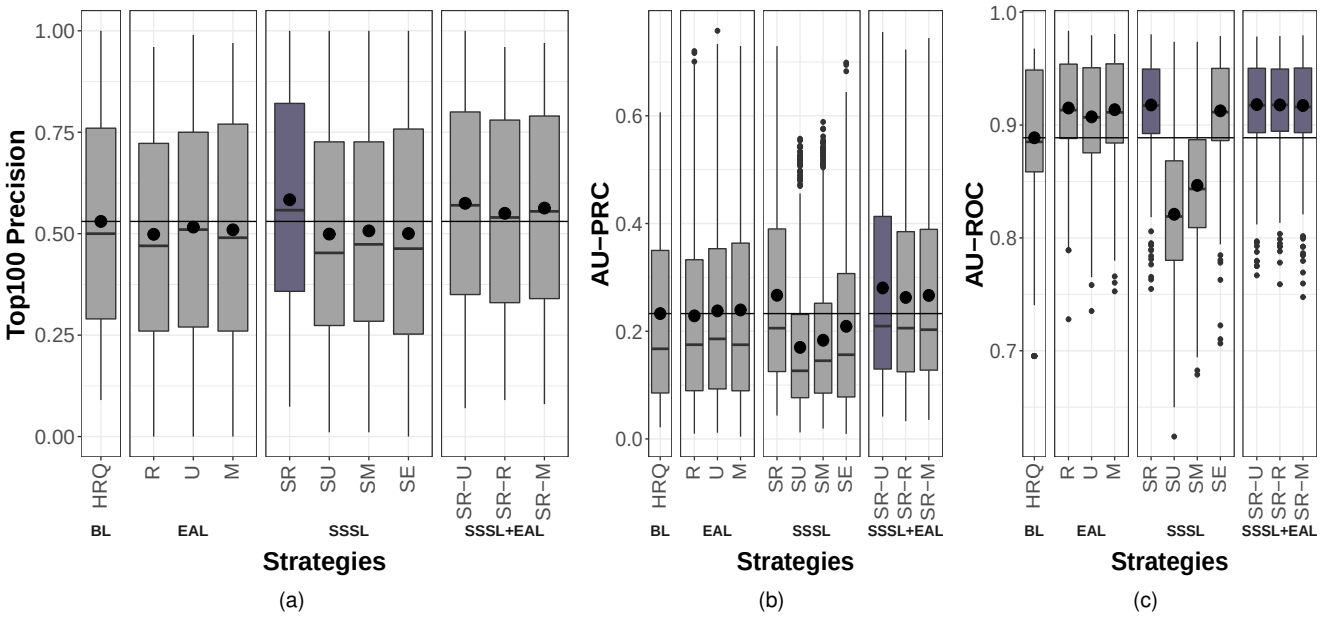

Figure 2. Transaction-based case study. Box-plots summarize the accuracy measures obtained over 60 days and 20 trials. Black points indicate the mean value for each box and the horizontal line indicates the mean for the baseline HRQ. Dark boxes indicate the best strategy and those which are not statistically different (paired Wilcoxon test). The extended names for the strategies listed on the horizontal axes can be found in Table 1. Figure (a): Top100 precision. Figure (b): Area Under the Precision-Recall Curve. Figure (c): Area Under the Receiver Operator Characteristic Curve.

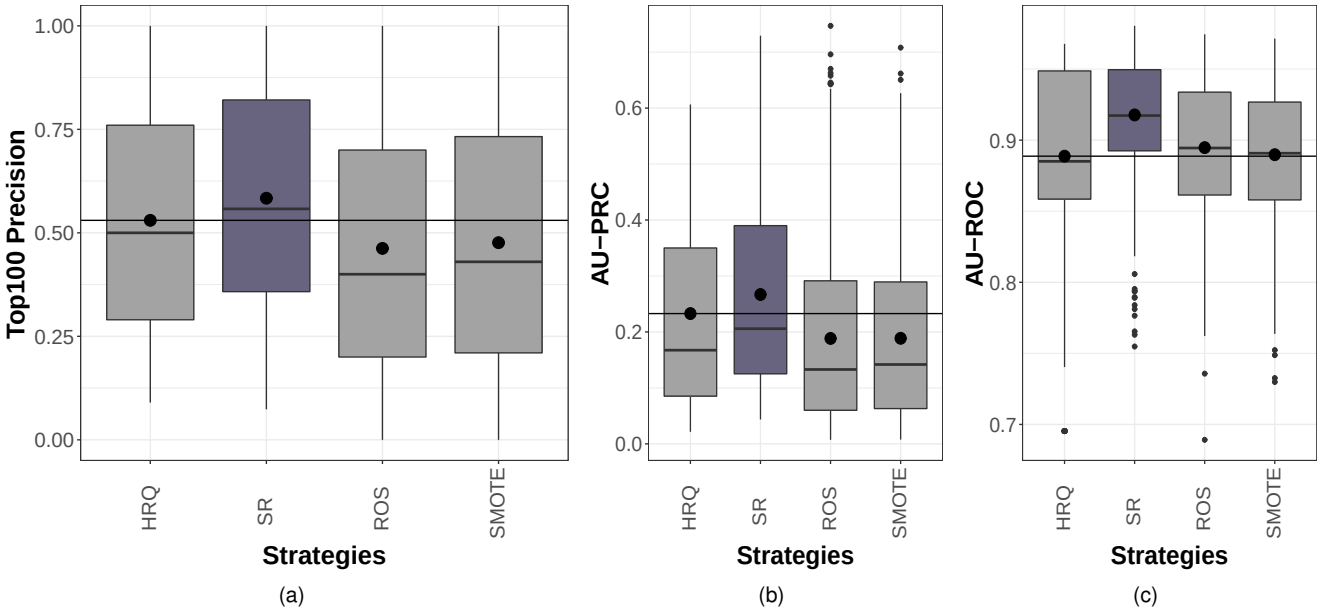

Figure 3. Transaction-based case study. Figure (a): Top100 precision. Figure (b) Area Under the Precision-Recall Curve. Figure (c) the Area Under the Receiver Operator Characteristic Curve.

The most efficient combination in our setting is therefore obtained by a combination of stochastic semi supervised approach with the standard HRQ strategy for active learning. We also have compared the semi-supervised technique SR with the standard Random Oversample and SMOTE over-sampling techniques. As shown in Figure 3, SR appears to be better in terms of Precision, AU-PRC and AU-ROC. ROS and SMOTE outperform HRQ only in terms of AU-ROC.

5.2. Card-based Fraud Detection

In this second case study, we retain the most promising techniques form the transaction assessment (namely, SR and

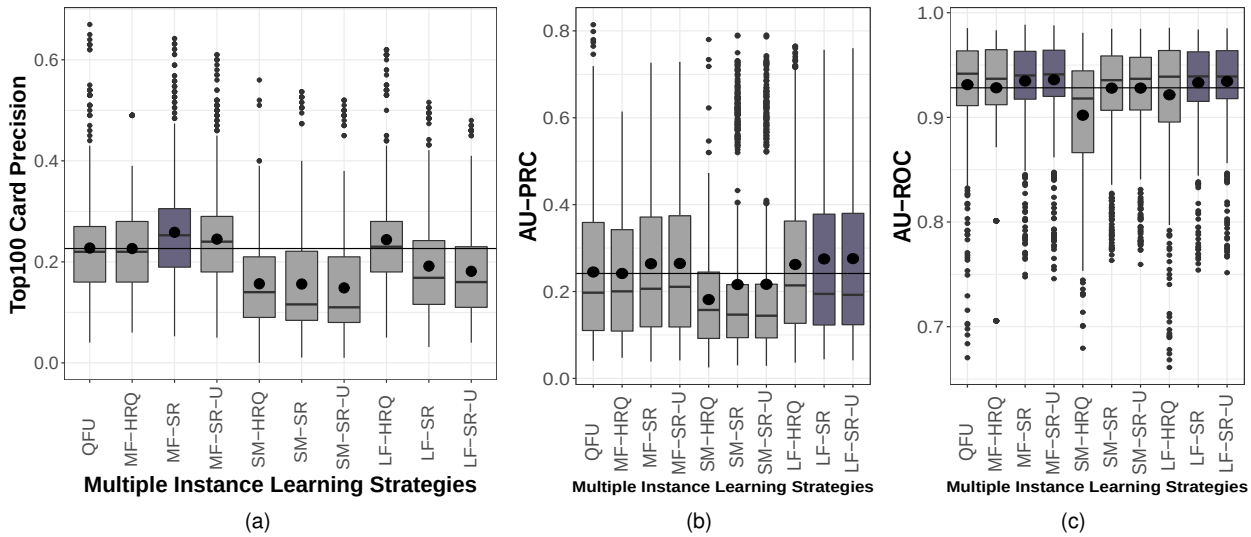

Figure 4. Card-based case study. Figure (a): Top100 Precision. Figure (b) Area Under the Precision-Recall Curve. Figure (c) Area Under the Receiver Operator Characteristic Curve.

SR-U), and we compare them with the Multiple Instance Learning strategies described in Section 4.5.1 ($v = 0.05$) and 4.5.2 ($\epsilon = 1e-3$). Figure 4 summarizes the results for the card detection study, using Top100 Card Precision, AU-PRC and AU-ROC as accuracy metrics.

Using the Top100 precision as the performance metric, the best results are obtained using stochastic semi-supervised with random labeling (SR) when using the *max* combining function (MF). The strategy also performs well in terms of AU-ROC, but is outperformed (by a small margin) by the logarithmic combining function when using AU-PR as the performance metric.

Similarly to the results obtained at the transaction level, the best strategies are those combining the baseline High Risk Querying with stochastic semi-supervised with random labeling (SR) or uncertainty labeling (SU). The addition of an exploratory part (QFU, or SR-U strategies) did not allow to improve the detection accuracy. The performances are even slightly decreased in terms of Top100 precision for SR-U strategies.

Finally, the results show that the combining function plays an important role. While the *max* and *logarithmic* performed best overall, the *softmax* clearly hampered the fraud detection accuracy.

To conclude, stochastic semi-supervised labeling (SR and SU) combined with HRQ remained the best strategies, confirming the results obtained at the transaction level. Regarding combining functions, while the *logarithmic* function provided the best performances in terms of AU-ROC, they were however significantly outperformed by *max* in terms of Top100 precision. Overall, the *max* combining function was observed to provide the most stable improvements throughout the range of explored performance metrics.

6. Conclusion

This paper investigated the combination of semi-supervised and active learning techniques in the context of streaming fraud detection. Using a real-world dataset of several millions of transactions over sixty days, we provided an extensive analysis and comparison of different strategies, involving standard active learning, exploratory active learning, semi-supervised learning and combining functions, and we made the code available on Github.

Our results show that the baseline active learning for fraud detection, the Highest Risk Querying, can be noticeably improved by combining it with Stochastic Semi-supervised Learning, thereby allowing to increase the fraud detection accuracy by up to five percent. Exploratory active learning techniques were not observed to improve the fraud detection task, which we attribute to the highly imbalanced nature of the data and the small exploration budget that can be reasonably allocated in a fraud detection system.

Last, our results on combining functions for bags of transactions showed that the baseline strategy, implemented with the *max* strategy, was the most stable across different accuracy metrics, but that alternative functions could be worth considering.

Future work will aim at further investigating Stochastic Semi-supervised Learning strategies and combining functions. In particular, two promising research axes are to better characterize the ratio of unlabeled transactions that can be labeled in a semi-supervised way, and the use combining functions as part of the semi-supervised sampling strategies.

Acknowledgments

The authors FC, YLB and GB acknowledge the funding of the Brufence project (Scalable machine learning for au-

tomating defense system) supported by INNOVIRIS (Brussels Institute for the encouragement of scientific research and innovation).

References

[1] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[2] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[3] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[4] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[5] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.

[6] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007, pp. 127–136.

[7] X. Zhu, P. Zhang, X. Lin, and Y. Shi, "Active learning from data streams," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 757–762.

[8] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*. Citeseer, 2000, pp. 839–846.

[9] S. Vijayanarasimhan, P. Jain, and K. Grauman, "Far-sighted active learning on a budget for image and video recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3035–3042.

[10] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Advances in neural information processing systems*, 2008, pp. 1289–1296.

[11] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection with alert-feedback interaction," *Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy*, 2017.

[12] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "Scarff: a scalable framework for streaming credit card fraud detection with spark," *(under submission)*, 2017. [Online]. Available: https://hub.docker.com/r/fabriziocarcillo/scarff/

[13] W. Fan, Y.-a. Huang, H. Wang, and P. S. Yu, "Active mining of data streams," in *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, 2004, pp. 457–461.

[14] K. Pichara, A. Soto, and A. Araneda, "Detection of anomalies in large datasets using an active learning scheme based on dirichlet distributions," in *Ibero-American Conference on Artificial Intelligence*. Springer, 2008, pp. 163–172.

[15] V. Van Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "Afraid: fraud detection via active inference in time-evolving social networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*. IEEE, 2015, pp. 659–666.

[16] S. Dasgupta, "Two faces of active learning," *Theoretical computer science*, vol. 412, no. 19, pp. 1767–1781, 2011.

[17] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with evolving streaming data," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 597–612.

[18] J. Xie and T. Xiong, "Stochastic semi-supervised learning on partially labeled imbalanced data," *Active Learning Challenge Challenges in Machine Learning*, 2011.

[19] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, 2004.

[20] L. Rokach, "Decision forest: Twenty years of research," *Information Fusion*, vol. 27, pp. 111–125, 2016.

[21] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *In Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 1994, pp. 148–156.

[22] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.