A graph-based, semi-supervised, credit card fraud detection system

Bertrand Lebichot, Fabian Braun, Olivier Caelen and Marco Saerens

Abstract Global card fraud losses amounted to 16.31 Billion US dollars in 2014 [1]. To recover this huge amount, automated Fraud Detection Systems (FDS) are used to deny a transaction before it is granted. In this paper, we start from a graph-based FDS named APATE [2]: this algorithm uses a collective inference algorithm to spread fraudulent influence through a network by using a limited set of confirmed fraudulent transactions. We propose several improvements from the network data analysis literature [3] and semi-supervised learning [4] to this approach. Furthermore, we re-designed APATE to fit to e-commerce field reality. Those improvements have a high impact on performance, multiplying Precision@100 by three, both on fraudulent card and transaction prediction. This new method is assessed on a three-months real-life e-commerce credit card transactions data set obtained from a large credit card issuer.

1 Introduction

Nowadays, e-commerce becomes more and more important for global trade: sales of goods and services represented more or less 2,000 billion dollars in 2014 and it was estimated that on 7,223 millions peoples on earth, 20 % were e-shoppers [5]. Part of the reasons of this success is easy online credit card transactions and

Bertrand Lebichot, Marco Saerens
Universite catholique de Louvain, Place des Doyens 1, 1348 Louvain-la-Neuve, Belgium
e-mail: [bertrand.lebichot,marco.saerens]@uclouvain.be

Fabian Braun
Worldline GmbH, R&D, Pascalstrasse 19, 52076 Aachen, Germany
e-mail: fabian.braun@worldline.com

Olivier Caelen
Worldline SA/NV, R&D, Chaussee de Haecht 1442, 1130 Bruxelles, Belgium
e-mail: olivier.caelen@worldline.com

cross-border purchases. Furthermore, most organizations, companies and government agencies have adopted e-commerce to increase their productivity or efficiency in trading products or services [6].

Of course, e-commerce is used by both legitimate users and fraudsters. The Association of Certified Fraud Examiners (ACFE) defines fraud as: ”the use of one’s occupation for personal enrichment through the deliberate misuse or misapplication of the employing organization’s resources or assets ”[7].

Global card fraud losses amounted to 16.31 Billion US dollar in 2014 and is forecast to continue to increase [1]. This huge number of losses has increased the importance of fraud fighting: in a competitive environment, fraud have a serious business impact if not managed, and prevention (and repression) procedures must be undertaken.

For those reasons e-commerce and credit card issuers need automated systems that identify incoming fraudulent transactions or transactions that do not correspond to a normal behavior. Data mining and machine learning offer various techniques to find patterns in data; here, the goal is to discriminate between genuine and fraudulent transactions. Such Fraud Detection Systems (FDS) exist and are similar to detection approaches in Intrusion Detection System (IDS). FDS use misuse and anomaly based approaches to detect fraud [8].

However, there are issues and challenges that hinder the development of an ideal FDS for e-commerce system [9]; such as,

- Concept drift: fraudsters conceive new fraudulent ways/methods over time. Furthermore, normal behavior also varies with time (peak consumption at Christmas for instance).
- Six-seconds rule [2]: acceptance check must be processed quickly as the algorithm must decide within six seconds if a transaction can be pursued.
- Large amount of data: millions of transactions occur per day whereas have to be analyzed and acceptance must be granted in seconds.
- Unbalanced data: frauds represents hopefully only less than 1% of transactions but predicting a pattern is harder with unbalanced dataset.

The presence of those challenges leads to high false alert rate, low detection accuracy or slow detection (see [10] for more details).

This work focuses on automatically detecting e-commerce fraudulent transactions using network (or graph) related features. Our work is based on a recent paper [2] which introduced an automated and field-oriented approach to detect fraudulent patterns in credit card transactions by applying supervised data mining techniques. More precisely, this algorithm uses a collective inference algorithm to spread fraudulent influence through a network by using a limited set of confirmed fraudulent transactions and take a decision based on risk scores of suspiciousness of transactions, card holder and merchants.

In this paper, several improvements from graph literature and semi-supervised learning are proposed. The resulting fraud detection method is tested on a three-months real-life e-commerce credit card transaction data set obtained from a large credit card issuer in Belgium.

The following questions are addressed:

1. Can we enhance graph-based existing FDS in terms of performance?
2. How can we make FDS as suitable for real application as possible?
3. Is semi-supervised learning [4] or feedback [9] useful for this Graph-based FDS?

Our approach takes into account various field/ground realities such as the six-second rule, concept drift, dealing with large datasets and unbalanced data. It also has been conceived in accordance with field experts to guarantee its applicability.

The rest of this paper is divided as follows: Section 2 introduces background and notation. Section 3 reviews related work. Section 4 details the proposed contributions. Experimental comparisons are presented and analyzed in Section 5. Finally, Section 6 concludes this paper.

2 Background and Notation

This section will first introduce some basic facts about fraud detection, since behavior of fraudsters has to be taken into account in the development of algorithms designed to counter them. Then some useful graph notation is reviewed.

2.1 Frauds

There are many fraud detection domains but internet e-commerce presents a challenging data mining task (see Section 1) because it blurs the boundaries between fraud detection systems and network intrusion detection systems.

As in many domains, profit-motivated fraudsters interact with the affected business. [11, 12] describes comprehensively this interaction: the fraudster can be internal or external to the business, can either commit fraud as a customer (consumer) or as a supplier (provider), and has different basic profiles. From this description, it comes out that professional fraudsters' (as opposed to occasional ones) modus operandi changes over time. Therefore, fraud detection system algorithms should also adapt themselve to new behaviors. This is refered as ”Concept drift”: the constant change in fraudsters behavior.

2.2 Graphs

Consider a weighted directed graph or network, G, assumed strongly connected with a set of n nodes V (or vertices) and a set of edges E (or arcs, links) [13, 14]. The **adjacency matrix** of the graph, containing non-negative affinities between nodes, is denoted as $\mathbf{A}$, with elements $[\mathbf{A}]_{ij}$ (also written a_{ij}) ≥ 0. The **natural random walk** on G is defined in a standard way. In node i, the random walker chooses the next edge to follow according to reference transition probabilities

$$p_{ij} = \frac{a_{ij}}{\sum_{j'=1}^{n} a_{ij'}} \tag{1}$$

representing the probability of jumping from node i to node $j \in Succ(i)$, the set of successor nodes of i. The corresponding transition probability matrix will be denoted as $\mathbf{P}$. In other words, the random walker chooses to follow an edge with a probability proportional to the affinity (apart from the sum-to-one normalization), therefore favoring edges associated to a large affinity. The matrix $\mathbf{P}$, containing the p_{ij}, is stochastic and is called the **reference transition matrix**.

3 Related Work

Credit-card Fraud detection received a lot of attention, but the number of publications available is limited. Indeed, credit card issuers protect data sources and most algorithms are produced in-house concealing the model's details [2].

As for any machine learning modeling process, two main approaches can be used: a supervised and an unsupervised scheme. Supervised learning uses labels (the observed prediction of an instance, here the fraud tag) to build the classification model, where unsupervised simply extracts clusters of similar data that are then processed. Common unsupervised techniques are peer group analysis [15] and self-organizing maps [16] while common supervised techniques are artificial logistic regression, neural networks (ANN) and random forests, meta-learning, case-based reasoning, Bayesian belief networks, decision trees, logistic regression, hidden Markov models, association rules, support vector machines, Bayes minimum risk and genetic algorithms. The reader is advised to consult [17] for more detail about credit card fraud detection, and [11] for a wider review on fraud detection.

According to [2], APATE was the only one to include network knowledge in the prediction models for fraud detection: This model first builts a tripartite graph (see below) and then extracts relevant risk scores for each node.

In this work, we follows the methodology of APATE [2] (which is described in this section, to make this paper self-contained), and propose several improvements in the next section. Other types of graph were also investigated (bipartite,...) but they did not provide better results and are therefore not presented here.

In particular, APATE starts with a set of time stamped, labeled, transactions. The goal is, of course, to fit a model to infer future fraudulent/genuine transactions. Furthermore, for each transaction of this dataset, the card holder (or user) and merchant (or retailer) is known. APATE thus create a tripartite adjacency matrix $\mathbf{A}^{\text{tri}}$ (there are three type of node: transactions, card users and merchants) as follows:

$$\mathbf{A}^{\text{tri}} = \begin{pmatrix} \mathbf{0}_{t\times t} & \mathbf{A}_{t\times c} & \mathbf{A}_{t\times m} \\ \mathbf{A}_{c\times t} & \mathbf{0}_{c\times c} & \mathbf{0}_{c\times m} \\ \mathbf{A}_{m\times t} & \mathbf{0}_{m\times c} & \mathbf{0}_{m\times m} \end{pmatrix} \tag{2}$$

where $\mathbf{A}_{t\times c} = (\mathbf{A}_{c\times t})^{\mathrm{T}}$ is an adjacency matrix where transactions are linked with their corresponding card holders , $\mathbf{A}_{t\times m} = (\mathbf{A}_{m\times t})^{\mathrm{T}}$ is an adjacency matrix where transaction are linked with corresponding merchants and $\mathbf{0}_{\ldots\times\ldots}$ is a correctly sized matrix full of zeros. From $\mathbf{A}^{\mathrm{tri}}$, transition matrix $\mathbf{P}$ is derived (see Section 2.2).

A column vector $\mathbf{r}_0 = [\mathbf{r}_0^{\mathrm{Trx}}, \mathbf{r}_0^{\mathrm{CH}}, \mathbf{r}_0^{\mathrm{Mer}}]^{\mathrm{T}}$ of length equal to the total number of transactions (hence the superscript Trx), card holders (CH) and merchants (Mer) is also created. The vector is full of zeros, except for known fraudulent transactions where it is equal to one (and therefore is always zero for merchants and card holders). Finally, element k of a vector $\mathbf{r}_0$ is noted $[\mathbf{r}_0]_k$.

Then, in a convergence procedure similar to the *PageRank* algorithm [18], $\mathbf{r}_0$ is updated to spread the fraud label through the tripartite graph. This is known as a random walk with restart procedure (RWWR) [19]:

$$\mathbf{r}_k = \alpha \cdot \mathbf{P}^{\mathrm{T}} \mathbf{r}_{k-1} + (1-\alpha) \cdot \mathbf{r}_0 \tag{3}$$

where α is the probability to continue the walk and $(1-\alpha)$ is the probability to restart the walk from a fraudulent transaction. This parameter could be tuned, but was fixed to 0.85 in the experimental comparisons (see [18]). The procedure diffuses the information about the transactions through the network.

Eq. 3 is iterated until convergence. Then, from $\mathbf{r}_{kc}$ (where kc stands for k at convergence) $\mathbf{r}_{kc}^{\mathrm{Trx}}$, $\mathbf{r}_{kc}^{\mathrm{CH}}$ and $\mathbf{r}_{kc}^{\mathrm{Mer}}$ can be extracted and considered as a risk measure for each transaction, card holder and merchant respectively.

As fraud detection models should adapt dynamically to a changing environment, this procedure is repeated several times, introducing a time decay factor. Each non-zero entry of $\mathbf{A}^{\mathrm{tri}}$ and $\mathbf{r}_0$ is modified to characterize transactions based on current and normal customer's past behavior (see [2] for more details):

$$\begin{cases} [\mathbf{A}^{\mathrm{tri}}]_{ij} \leftarrow e^{-\gamma\cdot t([\mathbf{A}^{\mathrm{tri}}]_{ij})} & \text{or 0 if no relation} \\ [\mathbf{r}_0]_k \quad \leftarrow e^{-\gamma\cdot t([\mathbf{r}_0]_k)} & \text{or 0 if no fraud} \end{cases} \tag{4}$$

where $t(\cdot)$ is the (scalar) time where transaction between i and j in matrix $\mathbf{A}^{\mathrm{tri}}$ occurred (or k for vector $\mathbf{r}_0$), and γ is a scalar set in such a way that the half-life of the exponential is: one day, one week and one month (i.e. elements are equal to 0.5 at half-life). For instance, if a transaction occured two weeks ago, the corresponding element of $\mathbf{A}^{\mathrm{tri}}$ with week decay is equal to 0.25 and is $1/(2^{14})$ with day decay.

Therefore, for each transaction of our starting dataset, we have 12 new features: Transaction risk for transaction, card holder and merchant, each for four (no decay, day decay, week decay and month decay) time windows.

However, this procedure cannot be computed in less than a few minutes, which is not suitable with the six-seconds rule. Convergence on a graph with millions of nodes is expensive and is therefore daily re-estimated over night. Transactions made during the testing day are evaluated using the model trained on previous night. For card holders and merchants, the graph-based feature values are extracted (looked up) from the trained model, since they are likely to be part of the previous data.

Naturally, for the new transaction not part of the model, transaction-based features have to be estimated, which is done through the formula:

$$\text{score}(Trx_{i,k}) = \frac{1}{\sum_{j=1}^{n} p_{ji} + 1}\text{score}(Mer_i) + \frac{1}{\sum_{j=1}^{m} p_{jk} + 1}\text{score}(CH_k) \tag{5}$$

where score$(Trx_{i,k})$ stands for the new transaction score between merchant i and card holder k, score(Mer_i) stands for the score of merchant i and score(CH_k) stands for the score of card holder k. It represents the score of a new transaction l after one new iteration of Eq. 3 when this transaction is added to $\mathbf{P}$ (with $p_{li} = 1$ and $p_{lk} = 1$). If a new transaction involves a new merchant and/or card holder, score(Mer_i) and/or score(CH_k) are set to zero accordingly.

Finally, those 12 new features (plus geographical features, see Table 1) are fed to a random forest classification model, as this model proved to perform well for the problem at hand, predicting fraudulent transaction [20, 17].

Table 1 Features used by the random forest classifier. First group are demographical features and second group are graph-based features. Notice that each transaction is linked with a card holder and with a merchant at a certain date: those information are only used to build the tripartite graph.

Variable name	Description
inBEL/EURO/OTH	Issuing region: Belgium/Europa/World
TX AMOUNT	Amount of transaction
TX 3D SECURE	Transaction used 3D secure
AGE	Age of card holder
langNED/FRE/OTH	Card holder language: Dutch/French/Other
isMAL/FEM	Card holder is Male/Female
isFoM	Card holder gender unknown
BROKER	Code of card provider
cardMCD/VIS/OTH	Card is a Mastercard/Visa/Other
01 Mer score	Merchant risk score (boolean, no time damping)
ST/MT/LT Mer score	Day/week/month decay merchant risk score (3 features)
01 CH score	Card Holder risk score (boolean, no time damping)
ST/MT/LT CH score	Day/week/month decay Card Holder risk score (3 features)
01 Trx score	Transaction risk score (boolean, no time damping)
ST/MT/LT Trx score	Day/week/month decay Transaction risk score (3 features)
TX FRAUD	Target variable: Fraud/Guenuine

4 The Proposed Model

While showing good performance, APATE can be improved in various ways.

4.1 Dealing with hubs

From the literature, it is known that presence of hubs in a network can harm the classifier [21, 22, 23]: hubs are nodes having a high degree and are therefore neighbors of a large number of nodes. In our dataset, it corresponds to popular nodes such as popular online shops like Amazon (as an example, the dataset is anonymised). Those nodes tend to accumulate a high value of risk score since they are connected to a lot of transactions. A simple way to counterbalance this accumulation is to divide the risk score by the node degree after convergence. In general, it is possible to divide by any power of the node degree and/or by different powers for the three types of nodes of the tripartite graph (transactions, card holders and merchants). In practice however, we did not find any combination that significantly beats the simple divide-by-node-degree option (results are not reported here).

Furthermore, it allows us to make a link with the regularized commute time kernel which is $\mathbf{K} = (\mathbf{D} - \alpha\mathbf{A})^{-1}$ (where $\mathbf{D}$ is the degree matrix) : element i, j of this kernel can be interpreted as the discounted cumulated probability of visiting node j when starting from node i (see [24, 3, 25] for details). The (scalar) parameter $\alpha \in]0, 1]$ corresponds to an evaporating or killed random walk where the random walker has a $(1 - \alpha)$ probability of disappearing at each step (therefore it has the same interpretation than RWWR). This method provided the best results in a recent comparative study on semi-supervised classification [3] and the second best results in another one [26]. We will therefore use this name (RCTK) to refer to this improvement.

4.2 *Introducing a time gap*

On the other hand, unlike in [2], the model cannot be based on past few days. Indeed, fraudulent transaction tags (the variable we want to predict) cannot be known with certainty without human investigators feedback. Moreover, since the fraudsters' modus operandi is known to change over time (see 2.1), it is not acceptable to built our model on old, less reliable (but fully inspected) data. However, it takes several days to inspect all transactions, mainly because it is sometime card holders that report undetected frauds. Of course, this makes our fraud detection problem harder [27].

In arrangement with field experts, we designed a real-life scenario containing three sets of data:

1. Training set: data where the transaction fraud labels can be taken as reliable.
2. Gap set: data where the transaction fraud labels are unknown.
3. Test set: data of the day on which the algorithm is currently tested.

It corresponds therefore to a semi-supervised learning scheme (SSL), as training data are partially labeled. If the Gap set is totally left aside, this is an usual supervised learning (SL) problem again. Once again, in arrangement with field experts, 15 days of training data and seven days for the gap set were chosen [28, 9]. This scenario is depicted on Fig. 1. Notice that on this figure, τ controls the testing day and that models are systematically built (overnight) on the 22 previous days. By changing τ, we get different testing days.

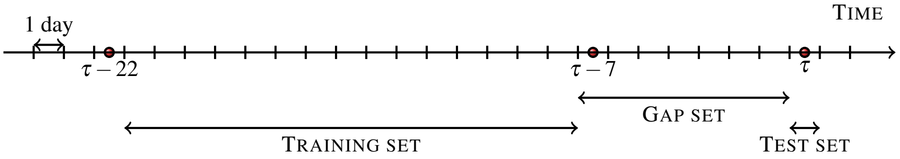

Fig. 1 Real-life FDS scenario with three sets of data. It takes several days to inspect all transactions, mainly because it is sometime the card holder who reports undetected frauds. Hence, in practice, the fraud tags of the Gap set are unknown. This scenario is repeated each day, as the parameter τ is incremented.

4.3 Including investigators feedback

Finally, even if in this last scenario it is not possible to know all fraud tags for the gap set, it is still conceivable that a fraction of previous alerts have been confirmed or overturned by human investigators (typically when a fraud alert occurs, the card is blocked and the card holder is contacted by phone). In our case, we put this number of feedbacks per day to 100, in arrangement with field experts. It is a realistic average number of cards than a human investigator can check per day. So in each of our gap set (except in starting condition) 700 cards have been checked by human investigators. We will take advantage of these investigated cases in order to try to predict more accurately the fraudulent transactions. On average, it means that roughly 1400 transaction feedbacks (two transactions per card) from previous testing day (previous τ's of our model) are available. This option will be refered as +FB and only make sense in a SSL scheme.

4.4 Removing merchant scores

Finally, we observed that removing merchant scores rises the performance. This is surprising at first glance but, after investigation, it turns out that new transactions involving new merchants cause issues (with our set-up, it corresponds to roughly 20% of merchants). In this case, the risk score is set to zero, causing the method to under-evaluate the risk. This should clearly be tackled but we choose to let this for further work. This last option will be refers as noM.

5 Experimental comparisons

In this section, the possible variation of considered algorithms will be compared on a real-life e-commerce credit card transaction data set obtained from a large credit card issuer in Belgium. Those graph-based algorithms compute additional features and were presented in Section 3 and 4. For practical purposes, considered algorithms are recalled in Table 2. The database is composed of 25,445,744 transactions divided in 139 days and fraud ratio is 0.31%. Features list can be found in Table 1.

Table 2 The nine compared models, see Sections 3 and 4 for acronyms. Considered variations of the APATE Algorithm according to four dimensions: merchant score status, hubs status, learning scheme and utilisation of feedback. Precision@100 (see Section 5) both for fraudulent card and transaction prediction is also reported (formatted mean ± std)

Classifier name	Mer Score	Damp hubs	Learning	Feedback	Card Pr@100	Trx Pr@100
RWWR SL = APATE	used	no	Supervised	no	18.64 ± 4.66	27.78 ± 11.61
RWWR SSL	used	no	Semi-supervised	no	16.95 ± 4.46	20.85 ± 10.14
RWWR SSL +FB	used	no	Semi-supervised	yes	14.19 ± 4.43	13.89 ± 8.49
RCTK SL	used	yes	Supervised	no	23.78 ± 9.52	40.50 ± 18.00
RCTK SSL	used	yes	Semi-supervised	no	44.55 ± 9.55	50.58 ± 13.99
RCTK SSL +FB	used	yes	Semi-supervised	yes	37.15 ± 10.14	49.06 ± 14.70
RCTK noM SL	discarded	yes	Supervised	no	45.35 ± 9.06	62.25 ± 11.97
RCTK noM SSL	discarded	yes	Semi-supervised	no	56.08 ± 8.06	81.61 ± 9.00
RCTK noM SSL +FB	discarded	yes	Semi-supervised	yes	**56.65 ± 8.69**	**84.13 ± 8.42**

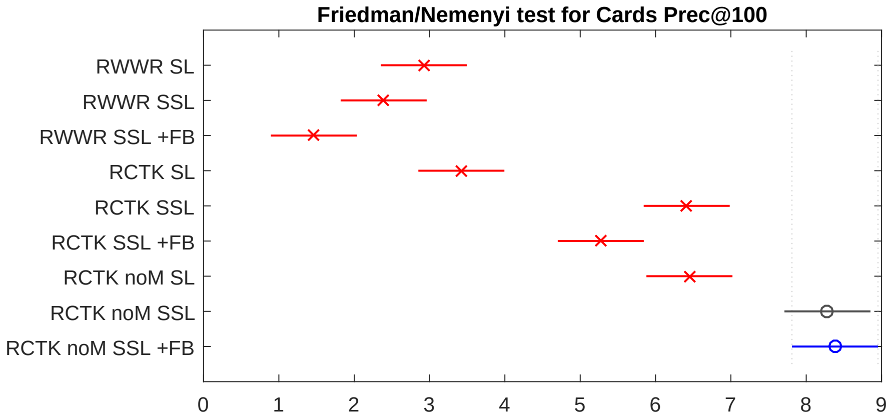

Fig. 2 Mean rank (circles and crosses) and critical difference (plain line) of the Friedman/Nemenyi test, obtained on a tree-months real-life e-commerce credit card transaction data set. The blue (bottom circle) method has the best mean rank and is significantly better than red (crosses) methods. The Critical difference is 1.14. Performance metric is Pr@100 (Precision@100) on fraudulent card prediction.

As a performance indicator, Precision@100 [29] was chosen, in accordance with field experts. It means that the 100 most probable (according to models) fraudulent transactions are checked by human investigators each day (and added as feedback in RWWR SSL +FB, RCTK w/ SSL +FB and RCTK noM w/ SSL +FB). Similarly all most probable fraudulent transactions are considered until 100 cards have been checked as usually human investigators verify all transactions of a card when they investigate. Precision@100 reports the number of true fraudulent transaction or card among 100 investigated cards. Notice that this last metric is more realistic as it is somehow the normal work charge for a human investigators team.

Figure 2 compares methods from Table 2 through a Friedman/Nemenyi test [30]. To do so, we adopt a sliding window approach: each day (different τ from Fig 1) is considered as a different (train-gap-test) dataset. This test compares the ranking provided by Table 2 methods. Friedman null hypothesis is rejected with $\alpha = 0.05$ and Nemenyi critical difference is equal to 1.14. A method is considered as significantly better than another if its mean rank is larger by more than this amount.

Firstly, RCTK always beats RWWR, RWWR noM was therefore discarded. This superiority indicates that tackling the hubs problem is actually important.

Secondly, SSL leads to a huge improvement, but only if hubs have been damped. SSL predicted frauds tend to contain more frauds with a fraudulent activity during gap days, compared to SL ones. As the fraud tag is hidden for the gap set, it means that this information is obtained by network analysis (train+gap).

Thirdly, even if +FB bring some kind of information, it only increases performance when hubs are tackled (RCTK) and merchant scores are removed (noM). By the way, results are not significantly better on our three-months dataset. Further analysis (not reported here) shows that with more data days and more checked cards, this improvement becomes significant (with $\alpha = 0.05$).

Lastly, removing merchant scores rises performance as explained in Section 4.

Overall, the best combination is RCTK noM SSL +FB, but it is not significantly better than RCTK noM SSL.

Finally, Figure 3 indicates the frequency of selected features by the random forest classifier. The method is RCTK SSL +FB and selects Mer scores most often. Sadly, new transactions involving new merchants cause issues. In this case, the risk score is set to zero, causing the method to under-evaluate the risk, resulting in a biased prediction. Discarding those four features (Mer scores) does increase the overall performance and selected variables of random forests stay similarly distributed.

6 Conclusion

In this paper, we start from an existing Fraud Detection Systems (FDS) APATE and bring several improvements: which have a huge impact on performances damping hub nodes (RCTK), introduce restrictions due to real application (SSL, Gap set, Pr@100 as a metric) and introduce feedback information from human investigators (+FB). Those improvements multiply the Pr@100 by three, both on fraudulent card or transaction prediction (for acronyms, see Section 4).

However, introducing feedback does not lead to a significant improvement: feedback impact can be increased if more cards are checked, but this is non-realistic for investigators. New transactions involving new merchants are still an issue (see noM in Section 4) which is let for further work: a possible way would be to mimic

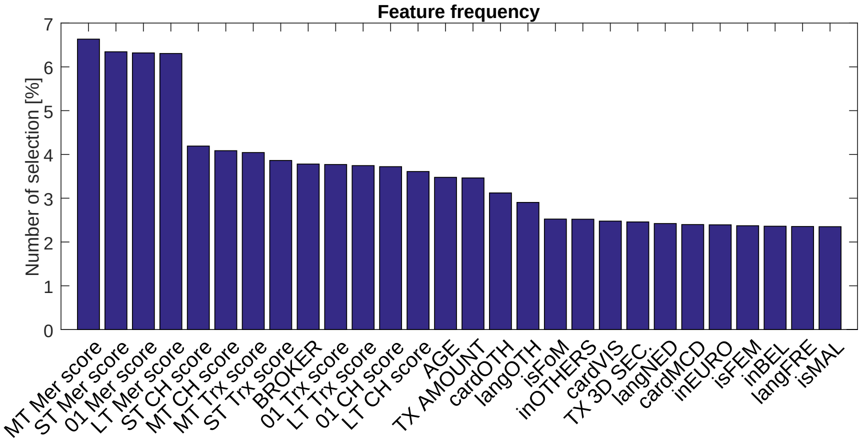

Fig. 3 Selected variables of random forests for the RCTK SSL +FB model for all days. Mer scores tend to bias the prediction. Discarding those four features does increase the overall performance (see Figure 2) and selected variables of random forests stay similarly distributed.

the learning procedure from [31]. Another envisaged further work is to introduce semi-supervised learning not only on graph analysis but also in main classifier.

Acknowledgements This work was partially supported by the Immediate and by the Brufence projects funded by Innoviris. We thank this institution for giving us the opportunity to conduct both fundamental and applied research.

References

1. HSN Consultants, Inc, The nilson report (2015).
URL https://www.nilsonreport.com/publication_newsletter_archive_issue.php?issue=1068
2. V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions., Decision Support Systems 75 (2015) 38–48.
3. F. Fouss, K. Francoisse, L. Yen, A. Pirotte, M. Saerens, An experimental investigation of kernels on a graph on collaborative recommendation and semisupervised classification, Neural Networks 31 (2012) 53–72.
4. O. Chapelle, B. Scholkopf, A. Zien (editors), Semi-supervised learning, MIT Press, 2006.
5. E-commerce Europe, Global b2c e-commerce light report 2015 (2014).
URL https://www.ecommerce-europe.eu/facts-figures/free-light-reports
6. R. Bolton, D. Hand, Statistical fraud detection: A review, Statistical science 17 (2002) 235–249.
7. Association of Certified Fraud Examiners, Report to the nation (2002).
URL http://www.acfe.com/uploadedFiles/ACFE_Website/Content/documents/2002RttN.pdf
8. T. Fawcett, Foster, F. Provost, Adaptive fraud detection, Data Mining and Knowledge Discovery 1 (1997) 291–316.

9. A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection and concept-drift adaptation with delayed supervised information., in: IJCNN, IEEE, 2015, pp. 1–8.
10. A. Abdallah, M. A. Maarof, A. Zainal, Fraud detection system, Journal of Network and Computer Applications 68 (2016) 90–113.
11. C. Phua, V. C. S. Lee, K. Smith-Miles, R. W. Gayler, A comprehensive survey of data mining-based fraud detection research, Computing Research Repository abs/1009.6119.
12. B. Baesens, V. V. Vlasselaer, W. Verbeke, Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection, Wiley Publishing, 2015.
13. U. Brandes, T. E. (editors), Network analysis: methodological foundations, Springer-Verlag, 2005.
14. M. Newman, Networks: an introduction, Oxford University Press, 2010.
15. D. J. Weston, D. J. Hand, N. M. Adams, C. Whitrow, P. Juszczak, Plastic card fraud detection using peer group analysis, Adv. Data Analysis and Classification 2 (1) (2008) 45–62.
16. V. Zaslavsky, A. Strizhak, Credit card fraud detection using self-organizing maps, Information and Security 18 (2006) 48.
17. A. D. Pozzolo, O. Caelen, Y.-A. L. Borgne, S. Waterschoot, G. Bontempi, Learned lessons in credit card fraud detection froma practitioner perspective, Expert Syst. Appl. 10 (41) (2014) 4915–4928.
18. L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web., Technical Report 1999-66, Stanford InfoLab, previous number = SIDL-WP-1999-0120 (November 1999).
19. J. G. Kemeny, J. L. Snell, Finite Markov Chains, Springer-Verlag, 1976.
20. S. Bhattacharyya, S. Jha, K. Tharakunnel, J. C. Westland, Data mining for credit card fraud: A comparative study, Decis. Support Syst. 50 (3) (2011) 602–613.
21. M. Radovanović, A. Nanopoulos, M. Ivanović, Hubs in space: Popular nearest neighbors in high-dimensional data, Journal of Machine Learning Research 11 (2010) 2487–2531.
22. M. Radovanović, A. Nanopoulos, M. Ivanović, On the existence of obstinate results in vector space models, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, ACM, 2010, pp. 186–193.
23. K. Hara, I. Suzuki, M. Shimbo, K. Kobayashi, K. Fukumizu, M. Radovanovic, Localized centering: Reducing hubness in large-sample data., in: Proceedings of the AAAI Conference, 2015, pp. 2645–2651.
24. D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Scholkopf, Learning with local and global consistency, in: Proceedings of the Neural Information Processing Systems Conference (NIPS 2003), 2003, pp. 237–244.
25. A. Mantrach, N. van Zeebroeck, P. Francq, M. Shimbo, H. Bersini, M. Saerens, Semi-supervised classification and betweenness computation on large, sparse, directed graphs, Pattern Recognition 44 (6) (2011) 1212 – 1224.
26. B. Lebichot, I. Kivimaki, K. Françoisse, M. Saerens, Semi-supervised classification through the bag-of-paths group betweenness, IEEE Transactions on Neural Networks and Learning Systems 25 (2014) 1173–1186.
27. A. D. Pozzolo, Adaptive machine learning for credit card fraud detection, Ph.D. thesis, Universite Libre de Bruxelles (2015).
28. R. J. Bolton, D. J. Hand, D. J. H, Unsupervised profiling methods for fraud detection, in: Proc. Credit Scoring and Credit Control VII, 2001, p. 235–255.
29. S. Theodoridis, K. Koutroumbas, Pattern recognition, 4th ed., Academic Press, 2009.
30. J. Demsar, Statistical comparaison of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
31. F. Braun, O. Caelen, E. Smirnov, S. Kelk, B. Lebichot, Improving card fraud detection through suspicious pattern discovery, Submitted for publication.