APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection using Network-Based Extensions*

Véronique Van Vlasselaer[1], Cristián Bravo[2], Olivier Caelen[3], Tina Eliassi-Rad[4], Leman Akoglu[5], Monique Snoeck[1], and Bart Baesens[1,6]

[1]Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium
[2]Departamento de Ingeniería Industrial, Universidad de Talca, Curicó, Chile
[3]Fraud Risk Management Analytics, Worldline, Brussels, Belgium
[4]Department of Computer Science, Rutgers University, Piscataway, NJ, USA
[5]Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
[6]School of Management, University of Southampton, Southampton, United Kingdom.

Abstract

In the last decade, the ease of online payment has opened up many new opportunities for e-commerce, lowering the geographical boundaries for retail. While e-commerce is still gaining popularity, it is also the playground of fraudsters who try to misuse the transparency of online purchases and the transfer of credit card records. This paper proposes ***APATE***, a novel approach to detect fraudulent credit card

[*] NOTICE: this is the author's version of a work that was accepted for publication in Decision Support Systems in May 8, 2015, published online as a self-archive copy after the 24 month embargo period. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. Please cite this paper as follows: Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. Decision Support Systems, 75, 38-48. Available Online: `http://www.sciencedirect.com/science/article/pii/S0167923615000846`

transactions conducted in online stores. Our approach combines (1) intrinsic features derived from the characteristics of incoming transactions and the customer spending history using the fundamentals of RFM (Recency - Frequency - Monetary); and (2) network-based features by exploiting the network of credit card holders and merchants and deriving a time-dependent suspiciousness score for each network object. Our results show that both intrinsic and network-based features are two strongly intertwined sides of the same picture. The combination of these two types of features leads to the best performing models which reach AUC-scores higher than 0.98.

Keywords: credit card transaction fraud, network analysis, bipartite graphs, supervised learning

1 Introduction

In recent years, e-commerce has gained a lot in popularity mainly due to the ease of cross-border purchases and online credit card transactions. Customers are no longer bound by the offers and conditions of local retailers, but can choose between a multitude of retailers all over the world and are able to compare their products, offered quality, price, services, etc. in just a few clicks. While e-commerce is already a mature business with many players, security for online payment lags behind. Recently, the European Central Bank (ECB) reported that the value of card fraud increased in 2012 by 14.8% compared to the year before (ECB, 2014). The main reason is the strong growth in online sales, resulting in many ”card-not-present” transactions (CNP), a means of payment that catches the attention of illicit people who try to mislead the system by pretending to be someone else. As a consequence, credit card issuers need an automated system that prevents the pursue of an incoming transaction if that transaction is highly sensitive to fraud, i.e. the transaction does not correspond to normal customer behavior.

This work focuses on *automatically* detecting online fraudulent transactions. Data mining offers a plethora of techniques to find patterns in data, distinguishing normal from suspicious transactions. A key challenge in fraud is to appropriately deal with the atypical character of fraud. That is, there are many legitimate transactions and only few evidence of fraudulent transactions to learn from, which complicates the detection process. Carefully

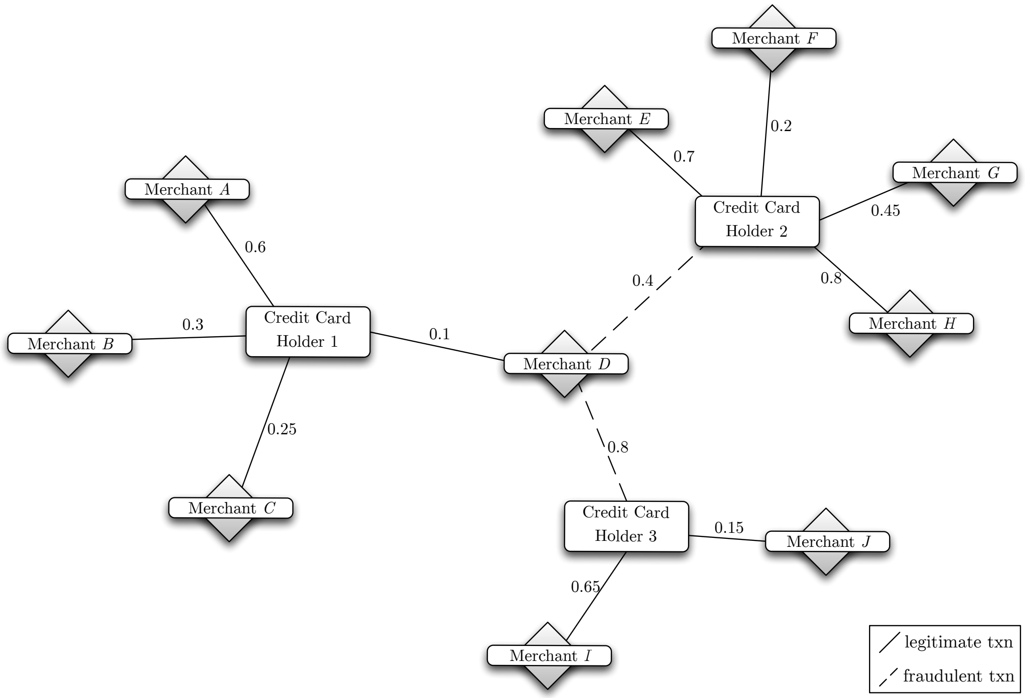

Figure 1: Toy example of a credit card fraud network. Weights depict the recency of the transaction between the merchant and credit card holder.

thinking about and creating significant characteristics that are able to capture irregular behavior, is an essential step in an efficient fraud detection process. In this paper, we combine both intrinsic and network-related features. Intrinsic features analyze the transaction as if it is an isolated entity, and compare whether the transaction fits in the normal customer profile. We create those features by deriving RFM attributes – Recency, Frequency and Monetary Value – of the credit card holder's past transactions. Network-based features, on the other hand, characterize each transaction by creating and analyzing a network consisting of credit card holders and merchants which are related by means of transactions. A sample network is given in Figure 1.

We use a collective inference algorithm to spread fraudulent influence through the network by using a limited set of confirmed fraudulent transactions and decide upon the suspiciousness of each network object by deriving an exposure score – i.e. the extent to which the transaction, the associated account holder and the merchant are exposed to past fraudulent influences.

In this work, we will answer the following questions: (1) Is a new incoming transaction in line with normal customer behavior, i.e. does it correspond to regular spending patterns of that customer in terms of (a) *frequency* or the average number of transactions over a certain time window (b) *recency* or the average time in between the current and previous transaction and (c) *monetary value* or the amount spent on that transaction? (2) Which merchants, credit cards and transactions are sensitive to fraud? Given past network-based information between merchants and credit card holders through the transactions made, how do we derive a suspiciousness score for (a) merchants indicating which merchants are often related to fraud, and as a consequence, form a risk of pursuing future fraudulent transactions; (b) credit card holders who act irregularly or whose credit card is stolen and (c) transactions by combining evidence of the associated credit card holder and merchant; (3) Does our detection approach which combines both intrinsic and network-based features, significantly boost the performance over traditional intrinsic-only models, and if so, which specific set of features contribute in detecting efficiently fraud?

We propose ***APATE*** (short for: Anomaly Prevention using Advanced Transaction Exploration), a novel, automated and real-time approach to tackle credit card transaction fraud by mapping past purchasing patterns and customer behavior into meaningful features and compare those features with the characteristics of a new, incoming transaction. We apply supervised data mining techniques to uncover fraudulent patterns from a real-life credit card transaction data set obtained from a large credit card issuer in Belgium. Our approach complies with the six-seconds rule, i.e. within six seconds the ***APATE*** algorithm needs to decide whether the transaction should or should not be pursued. We contribute by proposing a new propagation algorithm to propagate fraud from the network edges (i.e., the transactions) towards all the network components (i.e., the credit card holders and merchants) and derive for each transaction network-based features. Those features are combined with a set of intrinsic features to feed the learning algorithms. Our fraud detection model is able to *dynamically* adapt to a changing environment and continues to operate under the condition that fraudsters invent new ways to perpetrate their illegal activities.

The remainder of the paper is organized as follows. We introduce the credit card fraud domain in Section 2. Section 3 discusses the proposed methodology, and focuses on intrinsic and network-based feature extraction (Sections 3.1 and 3.2). In Section 4, we summarize the results. Section 5 concludes this paper.

2 Credit Card Transaction Fraud

2.1 Background

Credit card fraud detection is a widely studied research domain. Bhatla et al. (2003) and Delamaire et al. (2009) distinguishes between various types of fraud like application fraud (i.e., acquiring a credit card with false information), stolen or lost card, counterfeit card (i.e., card copying or using a card which does not belong to the owner) and card-not-present (CNP) fraud (i.e., using credit card details to make distance purchases). Our paper focuses on CNP fraud perpetrated through online credit card transactions.

As manually processing credit card transactions is a time-consuming and resource-demanding task, credit card issuers search for high-performing and efficient algorithms that *automatically* look for anomalies in the set of incoming transactions. Data mining is a well-known and often suitable solution to big data problems involving risk such as credit risk modelling (Baesens et al., 2003), churn prediction (Verbeke et al., 2011) and survival analysis (Backiel et al., 2014). Nevertheless, fraud detection in general is an atypical prediction task which requires a tailored approach to address and predict future fraud. We say that fraud is an uncommon, well-considered, imperceptibly concealed, time-evolving and often carefully organized crime which appears in many types and forms:

- ***Uncommon*** The number of legitimate transactions outnumbers the number of fraudulent transactions drastically. Many credit card fraud detection studies report a fraud ratio of less than 0.5% (Brause et al., 1999; Shen et al., 2007; Sánchez et al., 2009; Bhattacharyya et al., 2011; Duman and Elikucuk, 2013; Bahnsen et al., 2013, 2014; Dal Pozzolo et al., 2014).

- ***Well-considered*** Once fraudsters find a way to swindle, they exploit it until that type of fraud is discovered and prevention actions are taken. Extracting the right features and minimizing the opportunities of fraudsters to perpetrate fraud without being caught is an essential step in the fraud detection process.

- ***Imperceptibly concealed*** Fraudulent transactions often exhibit the same characteristics as legitimate transactions. Maes et al. (2002) formulated this as the presence of *overlapping data*. While many studies solely focus on customer profiling – *intra-account equivalence*, i.e. the extent to which the current behavior differs from previous customer behavior – models should take advantage of the knowledge sprouted from previous accounts used by fraudsters and compare this with currently legitimate customer – *inter-account equivalence*, i.e. the extent to which the customer profile differs from fraudulent profiles.

- ***Time-evolving*** An efficient fraud detection process is dynamic. There are two reasons. First, fraudsters change their way of working. Models should be fed with the most recent data to capture new types of fraud, and at the same time, should be able to prevent "existing" fraud. Second, customer changes in lifestyle might affect the spending patterns. Models that contrast a new transaction against the customer's transaction history mark changes in spending patterns as suspiciously.

- ***Carefully organized*** Once a credit card is stolen, it is used in many fraudulent transactions. Analogously, certain merchants are more sensitive to fraud – merchants that perpetrate fraud by themselves or that are easily accessible by fraudsters. Efficient detection models need to exploit the relational structure among credit card holders and merchants.

Each of the aforementioned requirements need to be addressed before a detection model can efficiently work in practice. In the remainder of the paper, we formulate a solution that systematically incorporates all of these requirements.

2.2 Credit Card Fraud Detection Process

The credit card detection process is summarized in Figure 2. The ultimate goal of such detection processes is to prevent the pursue of all transactions that do not comply with the imposed regularities. When a new transaction

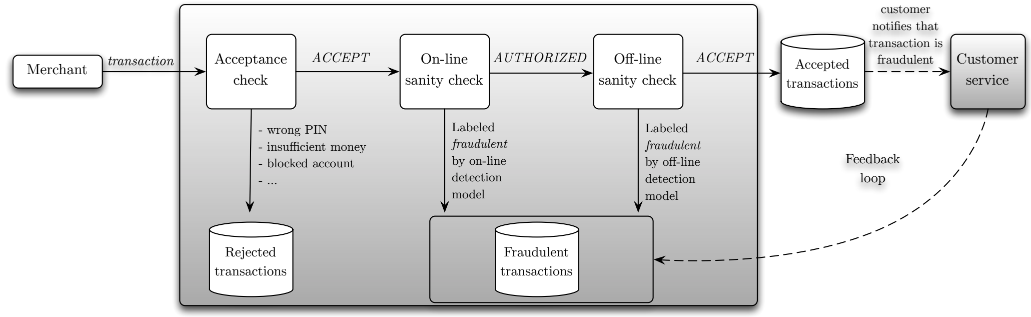

Figure 2: Credit Card Detection Process

arrives in the system, a series of acceptance checks is performed. The transaction processing system checks for example whether the user entered the right PIN or whether the spending amount is yet sufficient. If the transaction clears the acceptance checks, it is passed on to the sanity check of the detection system. Here, the system computes the probability that the transaction is fraudulent, e.g. by applying a detection model learned from past transactions. If the probability exceeds a certain threshold, the transaction does not proceed and is aborted. The sanity check has both an on-line (i.e., in real time) and off-line module. The data set under consideration consists of all processed transactions by Worldline Belgium. A transaction is fraudulent if the transaction does not pass the (a) on-line or (b) the off-line sanity check, or (c) by customer notification. While (a) is known in real-time, (b) and (c) can take up to one week.

The on-line detection process is liable to the "six-seconds rule" of decision. This means that both the acceptance check and the on-line sanity check need to be processed within six seconds. Our approach discusses the sanity check, and can be implemented both in an on-line as an off-line environment.

2.3 Related Work

Although fraud detection in the credit card industry is a much-discussed topic which receives a lot of attention, the number of publicly available works is rather limited. One of the reasons is that credit card issuers protect the sharing of data sources and most algorithms are produced in-house concealing the model's details. In particular, credit card fraud detection techniques can be divided into two broad categories: supervised and unsupervised methods. Unsupervised methods solely use the customer (or transaction) characteristics to group them into small, similar clusters while maximizing the difference between the extracted clusters. If a new transaction of a certain customer is not allocated to the normal customer group, then an alarm is raised for that transaction (Bolton et al., 2001). Unsupervised techniques include peer group analysis (Bolton et al., 2001; Weston et al., 2008) and self-organizing maps (Zaslavsky and Strizhak, 2006; Quah and Sriganesh, 2008). More studies focus on supervised techniques using evidence of past fraudulent transactions to infer the suspiciousness of future transactions. The most prevalent technique for supervised credit card fraud detection is artificial neural networks (ANN's) (Ghosh and Reilly, 1994; Aleskerov et al., 1997; Dorronsoro et al., 1997; Brause et al., 1999; Maes et al., 2002; Syeda et al., 2002; Shen et al., 2007). While ANN's generally achieve a high performance, they are black box models which lack interpretability. Recently, the use of ensemble methods like random forests is found to perform well in credit card fraud (Whitrow et al., 2009; Bhattacharyya et al., 2011; Dal Pozzolo et al., 2014). Random forests work especially well when there are many input features to learn from, which is often the case in network-related classification problems (Henderson et al., 2011). Other techniques for supervised learning in fraud are meta-learning (Chan et al., 1999), case-based reasoning (Wheeler and Aitken, 2000), Bayesian belief networks (Maes et al., 2002), decision trees (Shen et al., 2007), logistic regression (Shen et al., 2007; Bhattacharyya et al., 2011), hidden Markov models (Srivastava et al., 2008), association rules (Sánchez et al., 2009), support vector machines (Bhattacharyya et al., 2011), Bayes minimum risk (Bahnsen et al., 2013, 2014) and genetic algorithms (Duman and Elikucuk, 2013).

The aforementioned models do not exploit the relational structures. To the best of the authors' knowledge, ***APATE*** is the first to include network knowledge in the detection models.

3 Proposed Methodology

In this section, we discuss how the ***APATE*** detection process is implemented. Note that the detection process comprises the sanity check as illustrated in Figure 2. Particularly, we start from a list of time stamped, labeled transactions and learn a model to infer future fraudulent transactions. As fraud detection models should adapt dynamically to a changing environment, we introduce a sliding time window which characterizes a transaction based

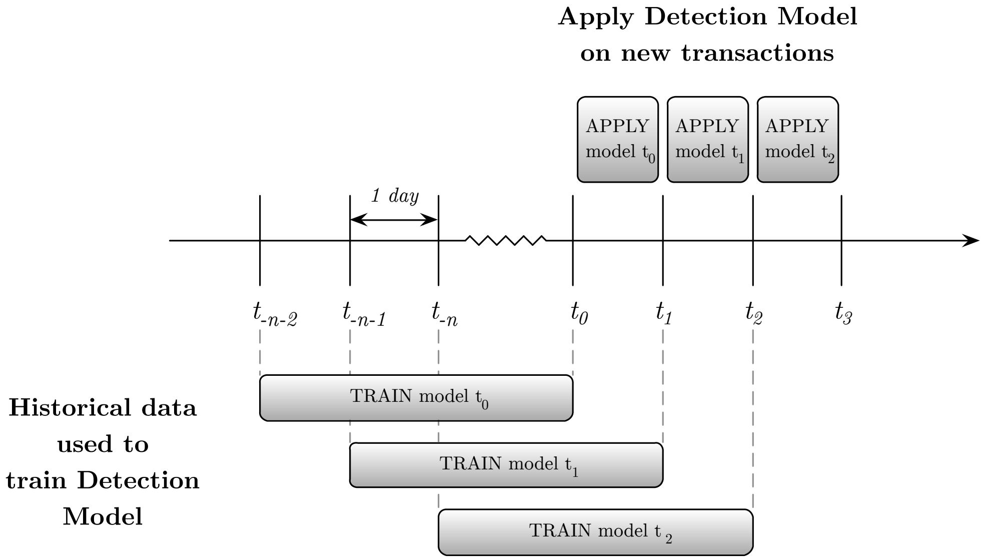

Figure 3: ***APATE***'s re-estimation process of the detection models using a sliding window.

on current (i.e., short term), and normal (i.e., medium and long term) customer's past behavior. Both intrinsic and network-based features are derived using those three time windows. Since model estimation often cannot be executed within six seconds, we choose to daily re-estimate the detection models at midnight the day before. Transactions made during the next day are evaluated using the model trained on data of the day before. The transaction features are extracted at real-time and fed into the model. This is depicted in Figure 3.

The ***APATE*** fraud detection process consists of two featurization steps:

1. ***Intrinsic feature extraction*** How does the incoming transaction differ from the previous transactions performed by that credit card holder?

2. ***Network-based feature extraction APATE*** exploits the relationships between credit card holders and merchants by means of transactions. The set of network-related features measures the exposure of each network object to fraud.

[1] Independent of time window

Table 1: Summary of input features on short (ST), medium (MT) and long (LT) term.

Variable	Description	Summary statistics					
		ST		MT		LT	
		μ	σ	μ	σ	μ	σ
Transaction features[1]							
Location (dummy)	Issuing region						
Belgium		0.16	0.37				
EU		0.76	0.43				
MC Category (dummy)	Category sensitivity to fraud						
Low		0.67	0.47				
Medium		0.31	0.47				
Amount	Amount of transaction	78.7	202.9				
Recency	Time passed since last transaction						
MC	at the merchant	8.95	13.01	235.69	417.62	2483.8	2887.5
MC Category	at the merchant category	8.97	13.04	232.9	415.2	2652.03	2988.6
Global	across all transactions	10.31	14.39	318.3	455.8	2996.4	3011.0
Country	in the same country	9.29	13.63	242.6	420.1	2582.0	2950.7
Currency	with the same currency	9.99	14.13	292.9	446.4	2957.6	3034.2
Frequency	Total number of transactions						
MC	at the merchant	0.12	0.70	0.25	1.54	0.85	5.85
MC Category	at the merchant category	0.13	0.74	0.28	1.63	0.92	6.11
Global	across all transactions	0.23	1.53	0.52	2.62	1.82	9.99
Country	in the same country	0.17	0.98	0.37	2.09	1.27	9.48
Currency	with the same currency	0.19	1.04	0.42	2.18	1.52	9.79
Monetary Value	Average amount of transactions						
MC	at the merchant	5.24	120.84	9.64	158.21	30.08	558.6
MC Category	at the merchant category	6.54	139.09	13.39	198.45	47.05	783.66
Global	across all transactions	18.49	259.54	60.1	1083.7	288.63	7041.3
Country	in the same country	11.56	199.12	43.26	1002.4	227.67	6619.9
Currency	with the same currency	13.57	220.06	50.45	1068.8	261.9	7033.4
Event occurrence	First purchase?						
MC	at the merchant	0.93	0.24	0.89	0.30	0.80	0.40
MC Category	at the merchant category	0.93	0.26	0.89	0.32	0.78	0.41
Global	across all transactions	0.89	0.31	0.80	0.40	0.52	0.50
Country	in the same country	0.91	0.29	0.86	0.35	0.72	0.45
Currency	with the same currency	0.90	0.30	0.83	0.38	0.60	0.49
Average Transactions	Average per time frame and level						
Global	across all transactions					78.5	181.09
Merchant	at the same merchant					78.3	199.26
Exposure Score	Extent to which transaction (TXN),						
Transaction (TXN)	merchant (MC) and credit card	0.11e-2	0.018	0.46e-3	0.45e-2	0.40e-4	0.29e-3
Merchant (MC)	holder (CCH) are influenced by fraud	0.063	0.500	0.092	0.390	0.141	0.259
Credit card holder (CCH)	given the network.	0.26e-4	0.85e-2	0.34e-4	0.36e-2	0.27e-4	0.76e-3

All features are summarized in Table 1. In the remainder of this section, we will discuss in more details how we extracted each of the features to include in our ***APATE*** models.

3.1 Intrinsic Feature Extraction

Traditionally, attempting to predict fraud using supervised data mining has been supported by the characterization of the purchase patterns that the customers present previous to the fraud event (Bhattacharyya et al., 2011). Most models are constructed using an aggregation of the transactions and their value. (Krivko, 2010) uses both the number of transactions and the monetary value of them to estimate rolling windows that are then used to train the model, Whitrow et al. (2009) use several aggregation techniques on the data and study the effects of the aggregation on the results, and Jha et al. (2012) construct a detailed data set that contains several transaction aggregations, plus information on the country in which the transaction occurred, to name a few. The literature seems to agree that there are three conditions that assist in predicting fraud: the transaction details, the time framework, and the location in which they occur.

Following this, the first set of variables that we propose for studying this problem are a mixture of literature variables, plus some other indicators that arose during our research, and it refers to the characteristics of the transactions themselves. We start by constructing variables inspired by the transaction analysis, including all variables that we were able to replicate from the studies in the literature. Our variables include the number of transactions that occur in a given time framework (frequency), the amount of money spent in those transactions (monetary value), and the time between two subsequent transactions in a particular time period if any (recency).

These variables fit within the Recency - Frequency - Monetary Value (RFM) framework, which is widely used in marketing (Blattberg et al., 2008). There is no agreement in the literature regarding which one is an appropriate time framework to estimate these variables, ranging from hourly to averages over three months, so we propose to study both the short, the medium and the long term: the last hour of transactions (attempting to capture cards that are heavily used and then dropped), the last day of transactions (attempting to capture specific, consumption-prone days), and the last week of transactions (attempting to capture the normal behavior of the customer). As will be shown in the experimental part (Section 4), we have one month of transactions available, so analysis of longer time periods were not possible. Jha et al. (2012) suggests that useful information can be extracted regarding the merchant at which the purchases occur. Data that is available, and that will be used to aggregate the merchants, concerns the merchant itself, a gross category in which the spending occurs (i.e. supermarkets, clothing stores, etc.), and an aggregated global variable with all merchants. The literature (e.g. Bhattacharyya et al. (2011)) seems to suggest that performing the RFM analysis segmented by the currency and the country in which the transactions occurred would also bring information relevant to the study.

An additional set of binary variables was created to mark for when no purchase has occurred. These variables (FirstPurchase) mark if the transaction is the first one in that measured time frame, for each of the dimensions that are measured (see Table 1). This information is relevant mostly to a generalized linear model such as logistic regression, as discussed in Allison (2001). We construct 15 variables accounting for each level of aggregation and time period.

In summary, using three time periods, three types of RFM variables, and five types of transaction aggregations (single merchant, category, country, currency and global), we develop a set of 60 ($3 \times 3 \times 5 + 3 \times 5$) variables aggregating the past transactions. All variables have the following naming scheme: Level of Aggregation, RFM Type, Time Period. So for example, GlobalRecencyHour refers to the Recency (time between consecutive purchases) within one hour, when considering all available merchants.

The second step is to characterize the transaction itself using the location in which it occurred and the merchant info. Given the characteristics of the European credit card users, there is a strong pattern of credit card use European Union-wide, rather than in the country where the card is emitted. Transactions that occur outside the EU (mostly in the US) are rarer. We include dummy variables for these three zones (EU, Belgium, and Rest-of-World, ROW) to capture this information. Table 2 shows relevant information supporting this segmentation for the data set available for this work.

We completed this part of the data set with the variables from the literature that do not fit the RFM framework. There were some variables from the literature which could not be implemented in our study, given the availability of data: some works in the literature use three months of data (Bhattacharyya et al., 2011), but we only have one month available so it was impossible. We also have available only online transactions of one issuer, so bank-related and POS related variables are not applicable, such as in Sánchez

Table 2: Transactions per Region and Fraud Percentage

Region	% of Transactions	% Fraudulent
Belgium	16,13%	0,05%
European Union	75,39%	0,45%
ROW	8,48%	5,36%
Total	100%	0,78%

et al. (2009) and Whitrow et al. (2009). We included dummy variables representing the currency in which the transaction occurred, categorizing them in euros, US dollars, and other currencies. We also included variables regarding the average amount of the transactions during the last week, as suggested by Bhattacharyya et al. (2011) and Whitrow et al. (2009), which were estimated both at global transaction level and at merchant level.

The last set of constructed variables deal with the categories of merchants. The data provider manifested that there were suspicions that fraudulent transactions tended to accumulate in certain categories. Using this information, the available categories (19) were segmented into three large categories using the individual categories' fraud percentage. This leads to three dummy variables (CategoryLow, CategoryMid, and CategoryHigh) capturing this assumption.

After constructing the data set along the intrinsic (transaction related) variables, we complement the information by exploring a novel approach of network analysis, as described in the next subsection.

3.2 Network Feature Extraction

3.2.1 Network definition

Complex network analysis (CNA) studies the structure, characteristics and dynamics of networks that are irregular, complex and dynamically evolving in time (Boccaletti et al., 2006). Those networks often consist of millions of closely interconnected units. Most real-life networks are complex. CNA uses graph theory to extract useful statistics from the network. Boccaletti et al. (2006) define graph theory as the natural framework for the exact mathematical treatment of complex networks, and consequently, they formalize a complex network as a graph. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices or nodes $v \in \mathcal{V}$ which are connected to each other by a set of edges or links

$e \in \mathcal{E}$. Graphs can be directed or undirected, depending on whether the edges impose a direction or an order in the network. When edges express the intensity of relationships between nodes, we say that the graph $\mathcal{G}^{\mathbf{w}} = (\mathcal{V}, \mathcal{E})$ is weighted.

Mathematically, a graph can be represented by an adjacency matrix $\mathbf{A}_{n\times n} = (a_{i,j})$ with n the number of vertices in the network. The adjacency matrix is an $n \times n$ matrix, and $a_{i,j} = 1$ if a link between node i and j exists, $a_{i,j} = 0$ otherwise. A weighted matrix $\mathbf{W}_{n\times n} = (w_{i,j})$ is the matrix representation of a weighted graph and $w_{i,j} \in]0, 1]$ expresses the intensity between node i and j if a link exists, or $w_{i,j} = 0$ otherwise.

A graph that represents heterogeneous node types, is a multipartite graph. In particular, the credit card fraud network in this work is represented as a *bipartite* graph $\mathcal{G} = (\mathcal{V}_{\mathbf{1}}, \mathcal{V}_{\mathbf{2}}, \mathcal{E})$, containing two node types – i.e. credit card holders and merchants – and satisfies the following property:

$$e(v_1, v_2) \text{ subset of } \mathcal{V}_{\mathbf{1}} \times \mathcal{V}_{\mathbf{2}} \tag{1}$$

with $v_1 \in \mathcal{V}_{\mathbf{1}}$ the set of credit card holder nodes, and $v_2 \in \mathcal{V}_{\mathbf{2}}$ the set of merchant nodes. Property 1 enforces that a transaction can only exist between different node types, i.e. credit card holders and merchants. A toy example of the credit card network is shown in Figure 1.

The corresponding adjacency matrix $\mathbf{A}_{\mathbf{c}\times\mathbf{m}} = (a_{i,j})$ of a bipartite graph is a matrix of size $c \times m$ with c and m the total number of credit card holder and merchant nodes respectively. The weight matrix $\mathbf{W}_{\mathbf{c}\times\mathbf{m}} = (w_{i,j})$ represents the weighted graph as a matrix.

In order to address the dynamic character of fraud, we integrate time into the network such that the edges express the *recency* of the transaction. Inspired by the half-life decay of atoms, we exponentially decay the intensity of a relationship in time, where:

$$\begin{cases} w_{i,j} = e^{-\gamma h} & \text{if a relationship exists between node } i \text{ and } j \\ w_{i,j} = 0 & \text{otherwise} \end{cases}$$

with γ the decay constant and h the time passed since the transaction pursued, measured according to the interval being studied (minutes for short term, hours for medium term and days for long term). We set the decay constant γ such that the edge weight is close to zero after one month (long-term: $\gamma = 0.0001$), one week (medium-term: $\gamma = 0.004$) and one day (short-term: $\gamma = 0.03$) respectively. A high weight represents a recent transaction.

3.2.2 Network fraud propagation

Given a credit card network, how can we use the fraud label of the edges – i.e. the transactions – to infer a score for each network object? That is, we want to infer a score for each credit card holder, merchant and transaction. The derived score expresses the extent to which the network object is exposed to fraud, and is therefore called the *exposure* or *suspiciousness score*.

Influence propagation in networks is a widely researched topic, with many good results in fraud detection (Akoglu et al., 2014, 2013; Pandit et al., 2007). In earlier work, we proposed *GOTCHA!*'s fraud propagation algorithm for social security fraud to analyze bipartite graphs. *GOTCHA!*'s propagation algorithm is an iterative fraud scoring algorithm that is designed such that it scores two node types (cfr. bipartite graphs) based on the label of one node type. Assume that a graph consists of c type-one nodes and m type-two nodes. After k iterations, the vector containing the exposure scores of each node equals:

$$\vec{\xi}_k = \alpha \cdot \boldsymbol{Q_{norm}} \cdot \vec{\xi}_{k-1} + (1-\alpha) \cdot \vec{z}_{norm} \tag{2}$$

with $\vec{\xi}_k$ the $(c+m)$-vector containing the exposure scores of each node after k iterations, $\vec{\xi}_0$ a random vector with values between $[0,1]$, $1-\alpha$ the restart probability (according to Page et al. (1999), we choose $\alpha = 0.85$), $\boldsymbol{Q_{norm}}$ the column-normalized weight matrix of size $((c+m)\times(c+m))$, and $\vec{z}_{norm}$ the normalized degree-adapted starting vector of size $(c+m)$. Both $\boldsymbol{Q_{norm}}$ and $\vec{z}_{norm}$ are weighted in time to address the dynamic characteristic of fraud.

Equation 2 starts from a limited set of labeled *nodes* to infer a score for the remaining *nodes*. However, in the credit card fraud case, we require to start from a limited set of labeled *edges* to derive a score for both the *edges* and *nodes*. Therefore, ***APATE***'s network propagation algorithm adapts Equation 2 by making two changes: (1) $\boldsymbol{Q_{norm}}$ is transformed into a tripartite graph including transactions as a node in the network; (2) $\vec{z}_{norm}$ is a time-dependent normalized vector indicating the fraudulent transactions. These adaptations are discussed next.

(1) Edge-to-node transformation

In order to be able to propagate influence from edges, we include the edges as a separate entity in the network. That is, we transform the edges into nodes and create a tripartite graph $\mathcal{G} = (\boldsymbol{\mathcal{V}_1}, \boldsymbol{\mathcal{V}_2}, \boldsymbol{\mathcal{V}_3}, \mathcal{E})$ with $\mathcal{E} \subseteq (\boldsymbol{\mathcal{V}_1} \times \boldsymbol{\mathcal{V}_3}) \cup (\boldsymbol{\mathcal{V}_2} \times \boldsymbol{\mathcal{V}_3})$, such that the following property holds:

$$\begin{aligned} \forall\, v_3 \in \boldsymbol{\mathcal{V}_3} : \exists!\, v_1 \in \boldsymbol{\mathcal{V}_1} \,|\, e(v_1, v_3) \in \mathcal{E} \\ \&\, \exists!\, v_2 \in \boldsymbol{\mathcal{V}_2} \,|\, e(v_2, v_3) \in \mathcal{E}. \end{aligned} \tag{3}$$

with $v_1 \in \boldsymbol{\mathcal{V}_1}$ the set of credit card holder nodes, $v_2 \in \boldsymbol{\mathcal{V}_2}$ the set of merchant nodes and $v_3 \in \boldsymbol{\mathcal{V}_3}$ the set of transaction nodes. Property 3 enforces that credit card holder nodes and merchant nodes can only be connected to transaction nodes. We note that the edge weight in the tripartite graph between the transaction and both the credit card holder and merchant is equal to the edge weight between the credit card holder and the merchant in the original bipartite graph $\mathbf{W}_{c \times m}$. Let's say that c, m and t are the total number of credit card holder nodes, merchant nodes and transaction nodes respectively, then the weighted matrix $\mathbf{M}_{(c+m) \times t}$ is the mathematical representation of the tripartite graph which is exponentially decayed over time (see Section 3.2.1).

As Equation 2 requires a symmetric matrix, we transform the tripartite graph into a symmetric unipartite graph. Mathematically,

$$\boldsymbol{Q}^{\text{tri}} = \begin{pmatrix} 0_{(c+m) \times (c+m)} & \boldsymbol{M} \\ \boldsymbol{M'} & 0_{t \times t} \end{pmatrix} \tag{4}$$

Matrix $\mathbf{Q}^{\text{tri}}$ is a matrix with $c + m + t$ rows and columns. After normalizing the columns such that each column sums up to 1, the resulting matrix is $\boldsymbol{Q}^{\text{tri}}_{\boldsymbol{norm}}$.

(2) Starting vector

The starting vector is originally created to personalize the ranking of web pages by guiding the algorithm with the user's interests (Page et al., 1999). Rather than initializing the starting vector as a uniformly distributed vector, the starting vector can be used to emphasize the influence of certain nodes on the final ranking. The same reasoning holds for fraud. As we are not interested in any influence to propagate through the network, but only in fraudulent influence, we guide the algorithm by specifying the confirmed fraudulent transactions using the starting vector. That is, the starting vector $\vec{z}^{\text{ tri}}$ of size $(c + m + t)$ equals:

$$\begin{cases} z_i^{\text{tri}} = e^{-\gamma h} & \text{if node } i \text{ is a fraudulent transaction} \\ z_i^{\text{tri}} = 0 & \text{otherwise} \end{cases}$$

with γ the decay constant, and h the time passed since the transaction is labeled as fraudulent. Dependent on the time window of analysis, we exponentially decay the fraudulent influence on long ($\gamma = 0.0001$), medium ($\gamma = 0.004$) or short ($\gamma = 0.03$) term. All credit card holder and merchant nodes have a zero weight for the starting vector. Remark that we assign a higher weight to fraudulent transactions that occurred more recently.

The starting vector is normalized to $\vec{z}^{\,\text{tri}}_{norm}$, summing up to 1.

Using the previous modification to the bipartite propagation algorithm as stated in Equation 2, we derive ***APATE***'s propagation algorithm for edge and node labeling, where:

$$\vec{\xi}_k = \alpha \cdot \boldsymbol{Q}^{\text{tri}}_{\boldsymbol{norm}} \cdot \vec{\xi}_{k-1} + (1-\alpha) \cdot \vec{z}^{\,\text{tri}}_{norm} \tag{5}$$

The resulting score $\vec{\xi}_k$ is computed using the power-iteration method, iterating until convergence. Convergence is reached after a maximum number of iteration steps or when the change in the scores is marginal.

3.2.3 Feature extraction

As we use a long-, medium- and short-term time window in the analysis, matrix $\boldsymbol{Q}^{\text{tri}}_{\boldsymbol{norm}}$ and $\vec{z}^{\,\text{tri}}_{norm}$ in Equation 5 are computed with different α values ($\alpha = 0.0001, 0.004, 0.03$) to infer an exposure score of each node and edge using information up until one month, week and day respectively. For example, the long-term exposure score indicates the extent to which the transaction (or merchant, or credit card holder) is sensitive to fraud during the last month. In general, the higher the exposure score of a network object, the more the node or edge is surrounded by fraud in its neighborhood.

For each new incoming transaction, the following features are computed: (a) credit card holder exposure score (CCHScore), (b) merchants exposure score (MCScore) and (c) transaction (TXScore) exposure score on long (LT), medium (MT) and short (ST) term. We re-estimate the exposure scores for every network object each day at midnight in order to extract the evidential features for transactions that occur the next day.

The credit card holder and merchant exposure score are derived from Equation 5. If the credit card holder or merchant did not yet appear in the network – i.e., he/she did not perform any transaction during the time period of analysis – we assign a score of zero, as they are not yet exposed to fraudulent influences.

The transaction exposure score combines the influence of the associated credit card holder and merchant. If a transaction already occurred between the credit card holder and the merchant, we use the exposure score as calculated in Equation 5. If multiple transactions occurred between the same credit card holder and merchant, we use the score assigned to the most recent transaction. When a transaction did not yet happen between a certain credit card holder and merchant, we compute the exposure score of that transaction by using the exposure scores of its direct neighborhood. Therefore, we say that we **locally update** the exposure scores in the network, where:

$$\mathrm{TXN}_{i,k,\mathbf{score}} = \frac{1}{\sum_{j=1}^{n} w_{i,j} + 1}\mathrm{CCH}_{i,\mathbf{score}} + \frac{1}{\sum_{j=1}^{m} w_{k,j} + 1}\mathrm{MC}_{k,\mathbf{score}} \qquad (6)$$

with $\mathrm{TXN}_{i,k,\mathbf{score}}$ the exposure score of a transaction between credit card holder i and merchant k, $\mathrm{CCH}_{i,\mathbf{score}}$ the exposure score of credit card holder i, $\mathrm{MC}_{k,\mathbf{score}}$ the exposure score of merchant k, $w_{x,y}$ the link weight between node x and y, and n and m the total number of links from credit card holder i and merchant k respectively. The local updating algorithm redivides the fraudulent influences. Instead of propagating the exposure score of the credit card holder/merchant only among the past transactions, the exposure score is now partly absorbed by the newly added transaction. We note that the edge weight of the new transaction is set to 1, as it represents a current relationship.

The network feature extraction step results in 9 features for each transaction: long-, medium- and short-term exposure scores for the transaction, associated credit card holder and merchant.

In the following section, we will estimate analytic models using three sets of variables: intrinsic (18), network-based (9) and demographics (5); and measure the capabilities they have for predicting fraud. In all cases, we seek to estimate the probability of fraud given the variables available, that is:

$$P(Y = fraud | X_{Intrinsic}, X_{Network}, X_{Demographics}) \qquad (7)$$

4 Results

To test the proposed approach, a unique data set of approximately 3.3M transactions from a large Belgian credit card issuer has been used. The data consists of a supervised data set with all the information related to transactions occurring during five consecutive weeks, plus a fraud or no fraud mark added for each transaction by the company after suspicious transactions were investigated (after two weeks at most). The data set is highly imbalanced, with only 48 000 frauds among the transactions (< %1).

We test the approach seeking to answer three questions: What is the best model for the approach? How can the model be applied in a real life-situation? And finally, what is the added value of using network variables for this problem? For all questions we will create an out-of-time test set consisting of all transactions that occur in the last week (approximately 500k), while the first two weeks will be used as the data pool for creating the RFM and network variables for the following two weeks of data (the training set).

During data cleansing and pre-processing, all transactions that were rejected due to normal banking reasons (wrong PIN, input errors, and other non-purchase related reasons) were eliminated from the data set. These transactions account for 15% of all transactions. Additionally, all transactions over 5000 EUR were also dropped from the data set, to avoid distortions in the set. These transactions are clear outliers: they consist of less than 1% of all transactions (none of them fraudulent) and they were almost 25 standard deviations from regular transactions, as shown in Table 1, so eliminating them leads to more stable models. The final training set consists of 2.2M transactions, and the final test set consists of 500k transactions. For each case, the variables described in Section 3 are calculated, accounting for 78 different variables, 9 which are network-based, 60 RFM variables, and the remaining variables being the non-RFM literature variables, or demographic and location-related.

4.1 Prediction Results

According to the findings of related research (see Section 2.3), we will benchmark three of them: logistic regression, the standard general linear model for classification used in many banking related activities, which is the less powerful of the group in terms of predictive capabilities, but is very simple to understand; a feed-forward, one hidden layer, neural network, one of the most powerful non-linear models, but that is considered a black box; and a random forest, a very powerful ensemble of decision trees which has brought very good results in many publications dealing with multiple applications.

To tackle the imbalance problem, we will apply standard case weighting for neural networks and logistic regression. For random forests we will use the sub-sampling capabilities of random forests, with each tree constructed using all fraudulent transactions and a randomly selected subset of the non-fraudulent ones such that they account for two times the number of fraudulent ones, as explained in Chen et al. (2004). We used 500 trees for the random forest model, which gives non-fraudulent cases an *a priori* chance of being selected similar to the one of simple random sampling. For parameter tuning, in the case of neural networks, 20% of the training data set was reserved for tuning the parameters, selecting the best combination of epochs and number of neurons over the grid given by $(Neurons, Epochs) \in [16, 156] \times [100, 1000]$, with the epochs increased in increments of 50, and the neurons in increments of one.

Table 3: Comparison of models

Model	AUC	Accuracy
Logistic Regression	0.972	95.92%
Neural Networks	0.974	93.84%
Random Forests	0.986	98.77%

The results, in Table 3, show a very high accuracy and Area-Under-the-ROC-Curve (AUC) values. The models are almost perfect, correctly predicting 98.7% of cases in the case of the random forest (the highest value), and with an AUC of 0.987. The relatively lower accuracy in the other two models is caused by a higher fraud detection rate when contrasted with false positives: the models are good at detecting frauds, but that comes at a cost of some extra non-fraudulent transactions being detected as fraudulent, which does not occur with random forests. This hits accuracy given the high imbalance of the data set. The very high AUC obtained can be seen in Figure 4.

To make a fairer comparison, possibly closer to a real application of the model, we will study the case when at most a 1% false positives are acceptable. The rationale behind this is that there is a reputational cost whenever a false positive occurs, given that users of the credit card will get a rejection on a non-fraudulent transaction, with all the consequences and annoyances

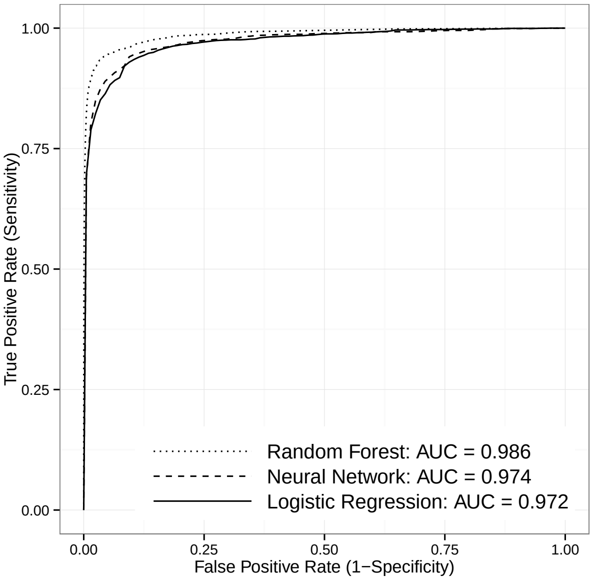

Figure 4: ROC Curve for Different Models

that such an action brings. Table 4 shows the obtained results.

Table 4: Accuracy and AUC (test set) at 1% Maximum False Positive Rate

Model	Cut-Off	Balanced Acc.	Specificity
Logistic Regression	0.85	87.4%	75.7%
Neural Networks	0.99	87.9%	76.8%
Random Forests	0.53	93.2%	87.4%

The results continue to be very good, but now the effects of the highly imbalanced problem are apparent. Random forest is the best model overall, with an 87.4% accuracy in the positive (fraudulent) cases, and a balanced accuracy of 93.2%. It is followed by neural networks, with a 76.8% specificity. The results hint at a highly non-linear problem, since there is a clear advantage when using non-linear models, which can be as large as the 12% increase in specificity when comparing random forests with logistic regression. The difference between neural networks and the random forest also suggests that the problem is not only highly non-linear, but that it is necessary to apply an ensemble model that searches for patterns in the sub-spaces that arise when applying a random forest. In any case, the results are very good. A user could use the model and detect close to 90% of all fraudulent transactions, flagging incorrectly only 1% of non-fraudulent ones.

4.2 Variable Importance and Network Variable Impact

The final question we would like to answer is which variables are more important, and try to measure their effect in the model overall. There are three main sets of variables in the problem: The RFM and demographic variables, the variables that are suggested in the literature that extend the RFM methodology, and the network variables. In order to contrast these sets we will estimate three additional random forests – since they give the best results –, one for each subset of variables. The results of these models can be seen in Table 5.

Table 5: AUC for Different Subsets of Variables.

Type	AUC	Accuracy
Only RFM	0.953	97.83%
Literature	0.955	97.87%
All Variables - First transaction	0.971	99.46%
Only Social Networks	0.920	94.37%
All variables	0.986	98.77%

It can be seen that only using the 9 network variables available the model reaches an AUC of 0.920. A model with only the RFM and demographic variables reaches an AUC 0.953, slightly higher. The inclusion of currency and country variables, together with the transaction averages (from the literature) make the AUC increase only slightly to 0.955. From these results we can conclude that the RFM variables are a very good set of variables to predict fraud, permitting to reach a very high AUC measure. The inclusion of the extended literature variables increase only slightly the AUC from a pure RFM approach, which might be caused due to regional behavior described in the data set we have available: variables representing currency and country do not present a strictly different behavior in Europe – with a unified currency, small travel distances and an integrated market, which might be even stronger when considering online sales – than what it might occur on different regions, such as North America or Australia. We can conclude that the inclusion of transaction averages, currency, and country variables has a minor, albeit positive, impact on the description of fraud for our data set.

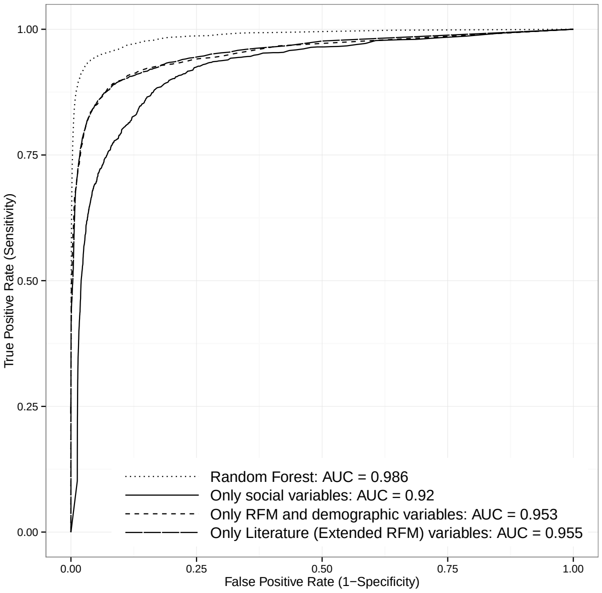

Figure 5: ROC Curve for Different Subsets of Variables

The inclusion of social network variables in combination with all the RFM variables has a very strong impact on the prediction results, reaching an AUC of 0.987. The main conclusion that we can derive from this result is that, considering that the social network variables have a very small correlation with respect to the other sets (the largest is 0.1), the information that these variables bring allows increasing the capabilities of the data set, interacting multidimensionally with the other two sets of variables, which translates into an increase of 5% in the AUC of the model. The ROC curves of the three different models (Figure 5) show that the models perform similarly in terms of the separation of false positives and false negatives, but the full model has less false positives in the early stages of the model, and that gain comes from the combination of the data sets.

When dealing with fraud, it is common to see several transactions that occur in a very short period of time, with a very high accumulated monetary value. As such, detecting the first transaction that is fraudulent is an interesting problem. In Table 5 we estimated the AUC of first transactions only (the ones with GlobalFrequencyHour equal to zero). It can be seen that the AUC, although lower, is still very high, which suggests that the purchasing patterns that precede fraud in the long term are the most relevant for pre-

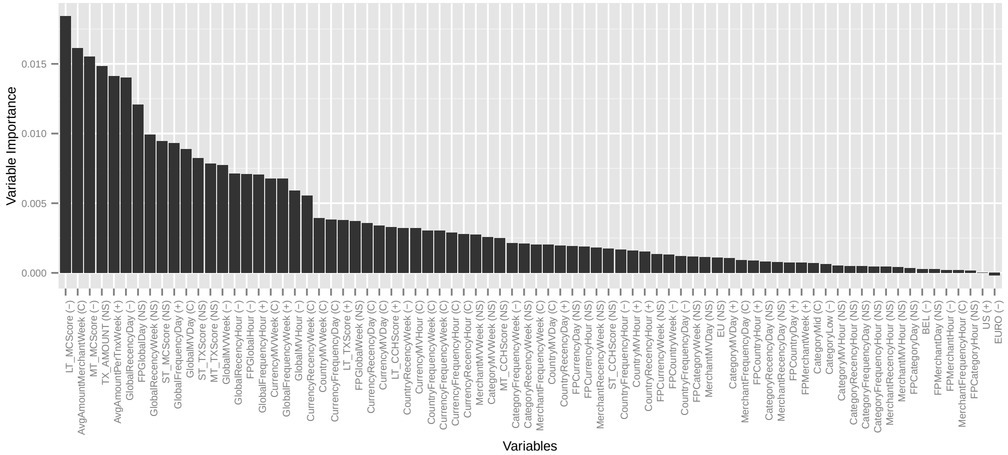

Figure 6: Importance of each variable for the random forest built using all available variables. In parentheses after each variable is the relevant information from the logistic regression output: A + or - sign of significant coefficients, NS when the variable was not significant, and C when it was highly correlated (greater than 0.9) with another variable in the data set.

dicting it, or, conversely, that it is the contrast between the current and the past behaviors that allow to correctly estimate fraud, and this is correctly captured by the variables in the model.

The exact relevance of the variable can also be extracted from the random forest model, and sheds light on the multidimensional increase in predictive capabilities of the model. Figure 6 shows the relative importance of each variable according to the random forest. It is interesting to note that the top two (with very similar importance) are one for each set, and are both related to the merchant at which the purchase occurs: AvgAmountMerchantWeek corresponds to the average monetary value per week before the current transaction at the merchant, so it shows the normal behavior on any given week, whereas LT_TXScore shows the long-term behavior of the network associated with the transaction itself, representing the normal, long-term, relation between merchants and the user of the credit card, weighting in the expected patterns of both fraud and non-fraud given the structure of the network. The next set of variables are again some literature and RFM variables mixed with network variables, but now referring to the medium-term (day) purchases, followed by the short-term network scores for both the merchant and the transaction. The variables representing transactions during the last hour seem to be of lower importance, and the currency variables close the list, which suggest, as shown before, that the purchase pattern in Europe is marked by the euro, so effects on currency that were present in previous works in the literature are annulled. It follows that it is from these relationships between the social network variables and the purchase patterns that the learning process is able to extract a significant amount of information that allows for a very high accuracy and AUC. As was expected, the first purchase variables are of limited importance in the random forest, but they can be significant in the logistic regression, considering that we knew that the information of those variables was included in a mixture of information from other variables which cannot be recovered easily in generalized linear models.

Regarding the signs and significance, most short term variables are non-significant in the logistic model, which suggests that the hourly behavior requires a deeper multivariate analysis that random forests delivers. All currency-related, and many country-related variables are highly correlated with other variables in the data set which suggest that the purchasing patterns of the data set are highly localized. The signs of the significant variables show that the hourly behavior tends to have a positive sign, which increases the odds of fraud, showing that when there are short-term increases in purchasing there is a higher risk of fraud. Something similar happens with the global variables: a higher global frequency is related to higher fraud, but a higher monetary value is related to lower odds of fraud. All long term social network variables are relevant, with varying signs: the long-term merchant score has a negative sign, showing that there are less risky merchants when dealing with fraud, but the transaction and customer long term score has a positive sign, which suggests that there are riskier customers, more prone to be subject to fraudulent activities.

5 Conclusions

In summary, this paper tackles credit card transaction fraud by proposing a novel, automated and real-time approach ***APATE*** (short for: Anomaly Prevention using Advanced Transaction Exploration). For each new incoming transaction, ***APATE*** decides whether the transaction might hint towards fraud and whether or not it should be pursued. A major component of ***APATE*** is the feature extraction part, where we opt to combine both intrinsic and network-based attributes. Our approach uses the RFM framework (Recency - Frequency - Monetary Value) complemented with demographic information of the transaction to define intrinsic features. As opposed to many previous studies, we enrich the detection models with network variables. The credit card fraud network consists of a network where credit card holders are connected to the merchants through the transactions they make. In particular, this paper discusses a new technique for fraud propagation through the network starting from a limited set of labeled edges (i.e., fraudulent transactions) and inferring a score for all the network components (i.e., credit card holders, merchants and transactions).

We tested the proposed approach on a company data set with more than three million transactions, and estimated a logistic regression, a neural network and a random forest model. Results show that our proposed approach leads to a very high AUC score and accuracy, especially for the random forest model. Even after adjusting the model to only allow 1% false positives, we obtain a high specificity, meaning that our models efficiently identify fraudulent transactions. Although each set of features separately results in a good model performance, the best results are reached when we combine both intrinsic and network variables, which suggest that there is a multidimensional component that is inherent of the combinations of the RFM and network approaches, potentially capturing both a short term change in behavior – contrasting the short term purchase pattern with the normal ones, either daily or weekly one – and a long term structure of the transactions, which arises from analyzing the different networks that can be inferred from the data. Finally, we show that ***APATE*** is not only able to find almost all fraudulent transactions, but also accurately pick out the first transaction in a series of fraudulent transactions, which is an important requirement in curtailing credit card transaction fraud.

While this work focuses on finding individual fraud, future work should investigate group behavior, i.e. the existence of fraudulent setups in the network of credit card holders and merchants.

Acknowledgments

All authors acknowledge Worldline Belgium that provided the data for this research. This work has been partially funded by CONICYT Becas Chile PD-74140041, and the Institute of Complex Engineering Systems (ICM: P-

05-004-F, CONICYT: FBO16). This material is based upon work supported by the ARO Young Investigator Program grant with Contract No. W911NF-14-1-0029, NSF with Grant No. IIS-1017181, an ONR SBIR grant under Contract No. N00014-14-P-1155, and the Stony Brook University Office of Vice President for Research. Any findings and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

Akoglu, L., Chandy, R., Faloutsos, C., 2013. Opinion fraud detection in online reviews by network effects. In: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media. Cambridge, Massachussets, USA, pp. 2 – 11.

Akoglu, L., Tong, H., Koutra, D., 2014. Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery Available Online, doi:10.1007/s10618–014–0365–y.

Aleskerov, E., Freisleben, B., Rao, B., 1997. Cardwatch: A neural network based database mining system for credit card fraud detection. In: Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997. IEEE, pp. 220–226.

Allison, P. D., 2001. Missing Data. Quantitative Applications in the Social Sciences. SAGE University Paper.

Backiel, A., Baesens, B., Claeskens, G., 2014. Mining telecommunication networks to enhance customer lifetime predictions. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L. A., Zurada, J. M. (Eds.), Artificial Intelligence and Soft Computing. Vol. 8468 of Lecture Notes in Computer Science. Springer International Publishing, pp. 15–26.

Baesens, B., Setiono, R., Mues, C., Vanthienen, J., 2003. Using neural network rule extraction and decision tables for credit-risk evaluation. Management Science 49 (3), 312–329.

Bahnsen, A. C., Stojanovic, A., Aouada, D., Ottersten, B., 2013. Cost sensitive credit card fraud detection using bayes minimum risk. In: Machine

Learning and Applications (ICMLA), 2013 12th International Conference on. Vol. 1. IEEE, pp. 333–338.

Bahnsen, A. C., Stojanovic, A., Aouada, D., Ottersten, B., 2014. Improving credit card fraud detection with calibrated probabilities. In: Proceedings of the 2014 SIAM International Conference on Data Mining. Pennsylvania, USA, pp. 677–685.

Bhatla, T. P., Prabhu, V., Dua, A., 2003. Understanding credit card frauds. Cards business review 1 (6), 1 – 17.

Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J. C., 2011. Data mining for credit card fraud: A comparative study. Decision Support Systems 50 (3), 602–613.

Blattberg, R. C., Kim, B.-D., Neslin, S. A., 2008. Database Marketing. Vol. 18 of International Series in Quantitative Marketing. Springer, New York, USA.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U., 2006. Complex networks: Structure and dynamics. Physics reports 424 (4), 175–308.

Bolton, R. J., Hand, D. J., et al., 2001. Unsupervised profiling methods for fraud detection. In: Proceedings of the VII Conference on Credit Scoring and Credit Control. Edinburgh, United Kingdom, pp. 235 – 255.

Brause, R., Langsdorf, T., Hepp, M., 1999. Neural data mining for credit card fraud detection. In: Proceedings. 11th IEEE International Conference on Tools with Artificial Intelligence. IEEE, pp. 103–106.

Chan, P. K., Fan, W., Prodromidis, A. L., Stolfo, S. J., 1999. Distributed data mining in credit card fraud detection. Intelligent Systems and their Applications, IEEE 14 (6), 67–74.

Chen, C., Liaw, A., Breiman, L., 2004. Using random forest to learn imbalanced data. Statistics Tech. Report 666, University of California, Berkeley.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., Bontempi, G., 2014. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications 41 (10), 4915–4928.

Delamaire, L., Abdou, H., Pointon, J., 2009. Credit card fraud and detection techniques: a review. Banks and Bank systems 4 (2), 57–68.

Dorronsoro, J. R., Ginel, F., Sgnchez, C., Cruz, C., 1997. Neural fraud detection in credit card operations. Neural Networks, IEEE Transactions on 8 (4), 827–834.

Duman, E., Elikucuk, I., 2013. Solving credit card fraud detection problem by the new metaheuristics migrating birds optimization. In: Rojas, I., Joya, G., Cabestany, J. (Eds.), Advances in Computational Intelligence. Vol. 7903 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 62–71.

ECB, February 2014. Third report on Fraud. European Central Bank.

Ghosh, S., Reilly, D. L., 1994. Credit card fraud detection with a neural-network. In: Proceedings of the Twenty-Seventh International Conference on System Sciences. Vol. 3. IEEE, pp. 621–630.

Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., Faloutsos, C., 2011. It's who you know: graph mining using recursive structural features. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 663–671.

Jha, S., Guillen, M., Westland, J. C., 2012. Employing transaction aggregation strategy to detect credit card fraud. Expert Systems with Applications 39 (16), 12650 – 12657.

Krivko, M., 2010. A hybrid model for plastic card fraud detection systems. Expert Systems with Applications 37 (8), 6070 – 6076.

Maes, S., Tuyls, K., Vanschoenwinkel, B., Manderick, B., 2002. Credit card fraud detection using bayesian and neural networks. In: Proceedings of the 1st international naiso congress on neuro fuzzy technologies.

Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford InfoLab.

Pandit, S., Chau, D. H., Wang, S., Faloutsos, C., 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In:

Proceedings of the 16th international conference on World Wide Web. ACM, pp. 201–210.

Quah, J. T., Sriganesh, M., 2008. Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications 35 (4), 1721–1732.

Sánchez, D., Vila, M., Cerda, L., Serrano, J.-M., 2009. Association rules applied to credit card fraud detection. Expert Systems with Applications 36 (2), 3630–3640.

Shen, A., Tong, R., Deng, Y., 2007. Application of classification models on credit card fraud detection. In: Service Systems and Service Management, 2007 International Conference on. IEEE, pp. 1–4.

Srivastava, A., Kundu, A., Sural, S., Majumdar, A. K., 2008. Credit card fraud detection using hidden markov model. Dependable and Secure Computing, IEEE Transactions on 5 (1), 37–48.

Syeda, M., Zhang, Y.-Q., Pan, Y., 2002. Parallel granular neural networks for fast credit card fraud detection. In: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems. Vol. 1. IEEE, pp. 572–577.

Verbeke, W., Martens, D., Mues, C., Baesens, B., 2011. Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Systems with Applications 38 (3), 2354–2364.

Weston, D. J., Hand, D. J., Adams, N. M., Whitrow, C., Juszczak, P., 2008. Plastic card fraud detection using peer group analysis. Advances in Data Analysis and Classification 2 (1), 45–62.

Wheeler, R., Aitken, S., 2000. Multiple algorithms for fraud detection. Knowledge-Based Systems 13 (2), 93–99.

Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., Adams, N. M., 2009. Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery 18 (1), 30–55.

Zaslavsky, V., Strizhak, A., 2006. Credit card fraud detection using self-organizing maps. Information and Security 18, 48.