A Proactive Time-frame Convolution Vector (TFCV) Technique to Detect Frauds Attempts in e-Commerce Transactions

Roberto Saia*, Ludovico Boratto, Salvatore Carta
Dipartimento di Matematica e Informatica, Università di Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy.

*Corresponding author. Email: roberto.saia@unica.it

Manuscript submitted June 25, 2015; accepted October 25, 2015.
doi: 10.17706/ijeeee.2015.5.4.229-236

Abstract: Any business that operates on the Internet and accepts payments through debit or credit cards, also implicitly accepts that some transaction may be fraudulent. The design of effective strategies to face this problem is challenging, due to factors such as the heterogeneity and the non stationary distribution of the data, as well as the presence of an imbalanced class distribution, and the scarcity of public datasets. Differently from the state-of-the-art strategies, instead of producing a unique model based on the past transactions of the users, our approach generates a set of models (behavioral patterns) to evaluate a new transaction, by considering the behavior of the user in different temporal frames of her/his history. By using only the legitimate past transactions of a user, we can operate in a proactive manner, by detecting the fraudulent ones that have never occurred. This also overcomes the data imbalance that afflicts the state-of-the-art approaches. We evaluate our proposal by comparing it with one of the most performing approaches at the state of the art (i.e., Random Forests), using a real-world credit card dataset.

Key words: Fraud detection, pattern mining, rule learning.

1. Introduction

Fraud is one of the major issues related with the use of debit and credit cards, considering that they are becoming the most popular way to conclude every financial transaction. The research of efficient ways to face this problem has become an increasingly crucial imperative in order to eliminate, or at least minimize, the related economic losses. Considering that the number of fraudulent transactions is typically much smaller than that of the legitimate ones, the data distribution is highly unbalanced, reducing the effectiveness of many learning strategies. This problem is also complicated by the scarcity of information in a typical financial transaction record, which generates an overlapping of the classes of expense of a user.

A fraud detection system operates with either *static* or *dynamic* learning strategies. Through the static ones, the model used to detect the frauds is generated after a certain time period, while in the dynamic strategies it is generated once, then updated after a new transaction. The strategy used in many of the state-of-the-art approaches is based on the detection of the suspicious changes in the user behavior, a trivial approach that in several cases leads toward false alarms. Most of these false alarms are related to the absence of extended criteria during the evaluation of the suspect activities, since numerous approaches exclude some non numeric data from the evaluation process, due to their incapacity to manage it.

In this paper, we extend the canonical criteria, integrating the ability to operate with heterogeneous data (i.e., numeric and non numeric), and by adopting multiple behavioral patterns of the users. This reduces the previously underlined problems, related with the data scarcity, heterogeneity, non stationary distribution, and imbalanced class distribution. This is possible because we take consider all the parts of a transaction, thus extracting more information, contrasting the overlapping of the classes of expense. By generating multiple behavioral models of a user, made by dividing the sequence of transactions in several time-frames, we also face the problem of the non stationarity of data, effectively modeling anyway the user behavior.

Differently from the canonical machine learning approaches at the state of the art (e.g., the Random Forests approach to which we compared in this work), our models do not need to be trained with the fraudulent transactions, because their definition needs only the legitimate ones. This overcomes the problem of data imbalance that afflicts the machine learning approaches. The level of reliability of a new transaction is evaluated by comparing (through the *cosine similarity* measure) its behavioral pattern to each of the behavioral patterns of the user, generated at the end of the previously described process.

This work provides the following main contributions to the current state of the art:

- introduction of a strategy able to manage heterogeneous parts of a financial transaction;
- definition of the *Transaction Field Keywords* (TFK) set, able to give more weight to certain information;
- introduction of the *Time-frame Convolution Vector* (TFCV) operations, which store, in the behavioral patterns of a user, the average values of the variations measured in each time-frame;
- definition of a discretization process, able to adjust the sensitivity of the fraud detection system;
- formalization of the process of evaluation of a new transaction.

The paper is organized as follows: Section 2 provides a background on the concepts of our proposal; Section 3 provides a formal notation and the problem definition; Section 4 introduces the proposed model and provides all the implementation details; Section 5 describes the experimental environment, the adopted metrics, and the results; the last Section 6 reports some concluding remarks and future work.

2. Related Work

Credit card fraud detection represents one of the most important contexts, where the challenge is the detection of a potential fraud in a transaction, through the analysis of its features (i.e., description, date, amount, etc.), exploiting a user model built on the basis of the past transactions of a user. In [1], the authors show how in the field of automatic fraud detection there is lack of publicly available real datasets indispensable to conduct experiments, as well as a lack of publications about the related techniques.

In [2], it is underlined how the *unsupervised* fraud detection strategies are still a very big challenge in the field of E-commerce. Bolton and Hand [3] show how it is possible to face the problem with strategies based both on statistics and on *Artificial Intelligence*, two effective approaches in this field able to exploit powerful instruments (such as the *Artificial Neural Networks*) in order to get their results. Another approach based on two *data mining* strategies (*Random Forests* and *Support Vector Machines*) is introduced in [4], where the effectiveness of these methods in the field of the fraud detection is discussed.

Data imbalance represents one of the most relevant issues, since almost all of the learning approaches do not work when excessive difference between the instances of each class of data exists [5].

The *static approach* [6] represents a canonical way to detect frauds. This approach uses a simple learning phase, but it is not able to follow the changes of the user behavior. Regardless of the approach, the problem of the non stationary distribution of the data, and that of the unbalanced classes distribution, still remain.

Our approach introduces a novel strategy that, firstly, takes into account all elements of a transaction (i.e., numeric and non numeric), reducing the lack of information problem, which leads toward an overlapping of the classes of expense. By introducing a *Transaction Field Keywords* (TFK) set, we also give more importance to certain elements of the transaction, during the model building. Secondly, differently from the canonical approaches at the state of the art, our approach is not based on a unique model, but on multiple user models that involve the entire dataset. This allows us to evaluate a new transaction by comparing it with a series of behaviors captured in many temporal frames of the user transaction history.

The main advantage of this strategy is the reduction, or removal, of the issues related with the stationary distribution of the data, and the unbalancing of the classes. Indeed, the domain is represented by limited temporal frames, and not by the entire dataset. The discretization of the models, according to a certain value of granularity, permits us to adjust their sensitivity to the peculiarities of the operating environment.

3. Notation and Problem Definition

This section defines the problem faced by our approach, preceded by a set of definitions aimed to introduce its notation.

Input Sets: given a set of users $U = \{u_1, u_{2,} \dots, u_M\}$, a set of transactions $T = \{t_1, t_{2,} \dots, t_N\}$, a set of absolute variations $\hat{T} = \{v_1 = |t_2 - t_1|, v_1 = |t_3 - t_2|, \dots, v_n = |t_N - t_{N-1}|\}$, where $|\hat{T}| = (|T| - 1)$, and a set of fields $F = \{f_1, f_{2,} \dots, f_X\}$ that compose each transaction t (we denoted as $k_1, k_{2,} \dots, k_W$, the values that each field f can assume), we denote as $T_+ \subseteq T$ the subset of legal transactions, and as $T_- \subseteq T$ the subset of fraudulent transactions. We assume that the transactions in the set T are chronologically ordered (i.e., t_n occurs before t_{n+1}).

Output sets: we denote as $I = \{i_1, i_{2,} \dots, i_Z\}$ the set of behavioral patterns generated at the end of the convolution process performed on the set $\hat{T}$ (before the discretization process of the values in the set P, and as $P = \{p_1, p_2, \dots, p_Y\}$ the same set after the discretization process in g levels (with $g \geq 2$) of the continuous values in the set F. It should be noted that $|I| = |P|$.

Fraud Detection: the objective of a fraud detection system is the isolation and ranking of the potentially fraudulent transactions, and the average precision (denoted as α) is considered as the correct measure to use in this kind of process [7]. Its formalization is $\alpha = \sum_{r=1}^{N} P(t_r)\Delta R(t_r)$, where N is the number of transactions in the dataset, and $\Delta R(t_r) = R(t_r) - R(t_{r-1})$. Denoting as π the number of fraudulent transactions in the data set, out of the percent t of top-ranked candidates, denoting as $h(t) \leq t$ the *hits* (i.e., the truly relevant transactions), we can calculate the $recall(t) = h(t)/\pi$, and $precision(t) = h(t)/t$, then the value of α.

The values $R(t_r)$ and $P(t_r)$ represent, respectively, the *recall* and *precision* of the r^{th} transaction, then we have $\Delta R(t_r) = (1/\pi)$ when the r^{th} transaction is fraudulent, and $\Delta R(t_r) = 0$ otherwise. When the set processed is a set composed by a certain number of legitimate transactions, but with only one potential fraudulent transaction to evaluate $\hat{t}$ (i.e., $T_+ \cup \hat{t}$), according to the previous *Fraud Detection* definition, we have $\pi = 1$ and $t=1$. Consequently, from the previous Lemma we can define a binary classification of the transaction $\hat{t}$, since $\Delta R(t_r) = 1$ when the r^{th} transaction is fraudulent, and $\Delta R(t_r) = 0$ otherwise, which allow us to mark a new transaction as *reliable* or *unreliable*.

Problem definition: an ideal fraud detection approach should have a value of α close to *1*, since it means that all fraudulent transactions π have been ranked ahead the legal ones. Our objective is then to maximize the α value, in order to reduce the false alarms and improve the effectiveness in the fraud attempts detection, i.e., $\max_{0 \leq \alpha \leq 1} \alpha = \sum_{r=1}^{N} P(t_r)\Delta R(t_r)$.

4. Our Approach

In this section we describe the five steps needed to implement our strategy.

4.1. Absolute Variations Calculation

In order to convert the set of transactions T in the set of absolute variations $\hat{T}$, according with the criterion exposed in Section 3, we need to define a different kind of operation for each different type of data in the set F (excluding the field *place*, used in the *Transactions Field Keywords*).

Numeric Absolute Variation: given a numeric field $f_x \in F$ of a transaction $t_n \in T$ (i.e., in our case the field *amount*), we calculate the Numeric Absolute Variation (NAV) between each pair of fields that belong to two contiguous transactions (denoted as $f_x^{(t_n)}$ and $f_x^{(t_{n-1})}$), i.e., $NAV = \left| f_x^{(t_n)} - f_x^{(t_n-1)} \right|$.

Temporal Absolute Variation: given a temporal field $f_x \in F$ of a transaction $t_n \in T$ (i.e., in our case the field *date*), we calculate the *Temporal Absolute Variation* (TAV) between each pair of fields that belong to two contiguous transactions (denoted as $f_x^{(t_n)}$ and $f_x^{(t_{n-1})}$), i.e., $TAV = \left| days(f_x^{(t_n)} - f_x^{(t_n-1)}) \right|$.

Descriptive Absolute Variation: given a textual field $f_x \in F$ of a transaction $t_n \in T$ (i.e., in our case the *description* field), we calculate the Descriptive Absolute Variation (DAV) between each pair of fields, that belong to two contiguous transactions (denoted as $f_x^{(t_n)}$ and $f_x^{(t_{n-1})}$), by using the *Levenshtein Distance* metric described in Section 5.4, i.e., $DAV = lev_{f_x^{(t_n)}, f_x^{(t_n-1)}}$.

4.2. TFK Definition

In order to define the *Transaction Field Keywords* (TFK) for a field considered as crucial in the fraud detection process (in our case, the field *place*), we select from the set of transactions all distinct values of this field, then we store them in a vector $K = \{k_1, k_2, \ldots, k_W\}_{\neq}$, according to the formalization introduced in Section 3. The vector K will be queried in order to check if the place of the transaction under analysis is a place already used by the user, or not. When it is true, the binary value of the corresponding element of the behavioral pattern (i.e., the field *place* of the behavioral pattern of the transaction to evaluate) is set to 1, otherwise to 0. It should be noted that the TFK process allows us also to manage some particular fields (e.g., those related to the card type, which contain terms such as VISA, MASTERCARD, AMEX, and so on), otherwise hard to manage through a typical text analysis process.

4.3. TFCV Operation

The convolution is a mathematical operation between two functions *f* and *g*, which produces a third function that represents a modified version of one of the original functions. In our context, after we have converted the set of transaction T into a set of absolute variations $\hat{T}$, adopting the criteria exposed in Section 4.1, we operate a convolution by sliding the *Time-frame Convolution Vector* over the sequence of absolute variation values stored in $\hat{T}$, one step at a time, extracting the average value of the variations in the defined time-frame *tf*. Given a time-frame *tf*=3, a set of variations $\hat{T} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, we can execute a maximum of $|C|$ convolution operations, with $|C| = |I| = (|\hat{T}| - |tf| - 1)$, as shown in the Equation 1.

$$\begin{gathered} \hat{T} = [v_1, v_2, v_3, v_4, v_5, v_6] \\ \Downarrow \\ c_1 = \frac{v_1 + v_2 + v_3}{|tf|}, c_2 = \frac{v_2 + v_3 + v_4}{|tf|}, c_3 = \frac{v_3 + v_4 + v_5}{|tf|}, c_4 = \frac{v_4 + v_5 + v_6}{|tf|} \Rightarrow I = [c_1, c_2, c_3, c_4] \end{gathered} \tag{1}$$

The sequence of values calculated in each time-frame *tf*, for each field (i.e., *description*, *amount*, and *date*), represents the set I of behavioral patterns of the user. It should be observed that we have to discretize the patterns obtained through the convolution process, by adding the binary value determined by querying the *Transaction Field Keywords* in K (as described in Section 4.2), before using them in the evaluation process.

4.4. Discretization process

The continuous values $\boldsymbol{v_c}$ in the patterns set I, obtained through the convolution (Section 4.3), are transformed in discrete values $\boldsymbol{v_d}$, in accord with a certain *level of granularity* g. (i.e., the sensitivity of the system when detecting the frauds). The result is a set $P = \{p_1, p_2, \ldots, p_Y\}$ of patterns that represent the user behavior in different parts of her/his transaction history. Given a granularity g, and a set of patterns I, each value $\boldsymbol{v_c}$ of a field f is transformed in a discrete value $\boldsymbol{v_d}$, following the process in the Equation 2.

$$v_d = \left\lceil \frac{v_c}{\left(\frac{\max(f)-\min(f)}{g}\right)} \right\rceil \tag{2}$$

4.5. Transaction Evaluation

To evaluate a new transaction, we need to compare each behavioral pattern $\boldsymbol{p} \in \boldsymbol{P}$ with the single behavioral pattern $\hat{\boldsymbol{p}}$ obtained by inserting the transaction to evaluate as last element of the set T, repeating the entire process previously described only for the transactions present in the last time-frame.

The comparison is done by using the *cosine similarity*, and the results are a range from 0 (transaction completely unreliable) to 1 (transaction completely reliable). The similarity value is the average of the sum of the minimum and maximum values of similarity $\cos(\theta)$, measured between a pattern $\hat{\boldsymbol{p}}$ and all patterns of the set P. The result is used to rank the new transactions, based on their potential reliability.

5. Experiments

This section describes the experimental environment, the adopted dataset and strategy, as well as the involved metrics, the parameters tuning process, and the results of the performed experiments.

5.1. Experimental Setup

In order to evaluate the proposed strategy, we perform a series of experiments using a real-world dataset related to one-year of credit card transactions (a private dataset provided by a researcher). Due to the scarcity of datasets publicly available, that are relevant to our context and that are not synthetic (or too old), in order to test our strategy we chosen to adopt this real and updated dataset, even considering that the detection of potential frauds, using for the training a small set of data, is more hard than using a big set of data. The TFVC algorithm was developed in Java, while the implementation of the state-of-the-art approach, used to evaluate its performance, was made in the R (https://www.r-project.org) environment, using the *randomForest* package.

5.2. Dataset

The dataset used for the training, contains one year of data related to the credit card transaction of a user. It is composed by 204 transactions, operated from January 2014 to December 2014, with amounts in the range from 1.00 to 591.38 Euro, 55 different descriptions of expense, and 7 places of operation (when the transaction is operated online, the reported *place* is Internet). Considering that all transactions are legal, we have T_+=204 and T_-=0. The fields that compose a transaction are four: *Description*, *Place*, *Date*, and *Amount*.

5.3. Strategy

Considering that it has been proved [6] that the *Random Forests* (RF) approach outperforms the other approaches at the state of the art, in this work we chose to compare our TFVC approach only to this one. Since we do not have any real-world fraudulent transactions to use, we first define a synthetic set of data T_-, composed by 10 transactions aimed to simulate several kind of anomalies.

The experiments compare our TFCV approach, with the RF one, by employing a *k-fold cross-validation*. Regarding the TFCV approach, we first partitioned the entire dataset T_+ into k equal sized subsets (according with the dataset size, we set k=3), which denote as $T_+^{(k)}$. Thus, each single subset $T_+^{(k)}$ is retained as the validation data for testing the model, after adding to it the set of fraudulent transactions T_- (i.e., $T_-^{(k)} \cup T_-$). The remaining *k-1* subsets are merged and used as training data to define the models. We repeat the previous steps for RF, with the difference that, in this case, we add the set T- also to training data. In both cases, we consider as final result the average precision (AP) related to all k experiments.

We perform two experiments: in the first, we define the values to assign to the parameters that determine the performance of the TFCV approach (i.e., *time-frame* and *granularity*), as described in Section 5.5; in the second, we compare TFCV and RF, by testing the ability to detect a number of 2, 4,…, 10 fraudulent transactions (respectively, a fraudulent transactions percentage of 2.8%, 5.5%,…, 12.8%).

5.4. Metrics

The metrics employed in the experiments are the *cosine similarity* and the *Levenshtein Distance*. In order to evaluate the similarity between the behavioral pattern of a transaction under analysis, and each of the behavioral patterns of the user, we use the cosine similarity. It measures the similarity between two vectors of an inner product space that measures the cosine of the angle between them, as shown in Equation 3

$$similarity = \cos(\theta) = \frac{x \cdot y}{\|x\|\|y\|} = \frac{\sum_{i=1}^{N} x_i \times y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{3}$$

The *Levenshtein Distance* is a metric that measures the difference between two term sequences. Given two strings a and b, it indicates the minimal number of insertions, deletions, and replacements, needed to transform a into b. Denoting as $|a|$ and $|b|$ the length of the strings, the formula is shown in Equation 4.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & if \min(i,j)=0 \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases} \tag{4}$$

Where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i \neq b_j$ and equal to 1 otherwise. It should be noted that the first element in the minimum corresponds to deletion (from a to b), the second to insertion and the third to match or mismatch, depending on whether the respective symbols are the same.

5.5. Parameter Tuning

Our approach considers the parameters *tf* (*time-frame*) and *g* (*granularity*), so we need to detect their values. We tested all the pairs of possible values of *tf* and *g*, in a range from 2 to 99 (to be meaningful, both must be greater than 1). By measuring the average precision AP, the chosen values are *tf*=46 and *g*=11.

5.6. Experimental Results

The TFCV process generates a set of user behavioral patterns *P*, which we compare (i.e., using the cosine similarity) to the behavioral pattern related to each transaction in the subset of test, in order to retrieve a level of reliability for each of them, following the process described in Sections 4.5. The final result is given by the mean value of the results of all the experiments, in accord with the *k-fold cross-validation* criterion.

As we can observe in Fig. 1, our TFCV approach obtained values very close to the RF one, and this without train its models with the past fraudulent transactions (as occurs in RF).

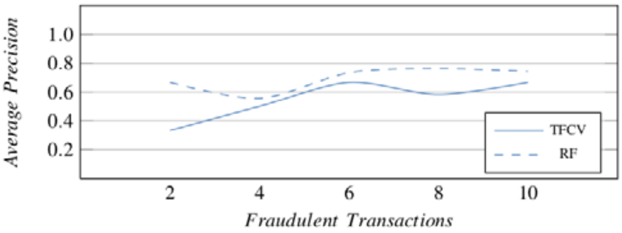

Fig. 1. Experiment results.

6. Conclusions and Future Work

In this paper we proposed a novel approach able to reduce or eliminate the threats connected with the frauds operated in the electronic financial transactions. Differently from the strategies at the state of the art, instead of exploiting a unique model defined on the basis of the past transactions of the users, we adopt multiple models (behavioral patterns), in order to consider the user behavioral in different temporal frames. Our approach, by building the behavioral models do not consider past fraudulent transactions, allows us to operate in a proactively, by detecting fraudulent transactions that have never occurred.

The experiments show that the performance of the proposed *Time Frame Convolution Vector* approach are very close to those of the *Random Forests*, and this without training our models with the past fraudulent transactions. A possible follow up of this work could be its development and evaluation in scenarios with different kind of financial transaction data, e.g., those generated in an E-commerce environment.

Acknowledgments

This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione “Industria Artigianato e Servizi” (annualità 2010), and by MIUR PRIN 2010-11 under project “Security Horizons”.

References

[1] Assis, C., Pereira, A. M., de Arruda Pereira, M., & Carrano, E. G. (2010). Using genetic programming to detect fraud in electronic transactions. *In a Comprehensive Survey of Data Mining-based Fraud Detection Research*, *1009(6119)*, 337–340.

[2] Phua, C., Lee, V. C. S., Smith-Miles, K., & Gayler, R. W. (2010). A comprehensive survey of data mining-based fraud detection research. *CoRR, 1009(6119)*.

[3] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 235–249.

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

[5] Batista, G. E. A. P. A., de Carvalho, L. F., & Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In O. Cairò, L. E. Sucar, & F. J. Cantu (Eds.), *Proceedings of MICAI 2000:* Vol. *1793. Advances in Artificial Intelligence, Mexican International Conference on Artificial Intelligence* (pp. 315–325). Springer.

[6] Pozzolo, A. D., Caelen, O., Borgne, Y. L., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection. *Expert Syst. Appl.*, *41(10)*, 4915–4928.

[7] Fan, G., & Zhu, M. (2011). Detection of rare items with target. *Statistics and Its Interface*, *4*, 11–17.

Roberto Saia is a Ph.D.c at the Department of Mathematics and Computer Science of the University of Cagliari. He got a master degree in computer science at the same University. His current research activity is focused on the development of techniques and algorithms able to improve the effectiveness of the user profiling and item recommendation.

Ludovico Boratto is a research assistant at the University of Cagliari - Italy. He graduated with full marks and honor and received his PhD in 2012 at the same University. His research focuses mainly on recommender systems and data mining in social networks.

Salvatore Carta received a PhD in electronics and computer science from the University of Cagliari in 2003. He is an assistant professor in computer science at the University of Cagliari since 2005. Recently, he has focused on topics related to the social web, ubiquitous computing and computational societies.