Multiple Behavioral Models: A Divide and Conquer Strategy to Fraud Detection in Financial Data Streams

Roberto Saia, Ludovico Boratto and Salvatore Carta
Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy

Keywords: Fraud Detection, Pattern Recognition, User Model.

Abstract: The exponential and rapid growth of the E-commerce based both on the new opportunities offered by the Internet, and on the spread of the use of debit or credit cards in the online purchases, has strongly increased the number of frauds, causing large economic losses to the involved businesses. The design of effective strategies able to face this problem is however particularly challenging, due to several factors, such as the heterogeneity and the non-stationary distribution of the data stream, as well as the presence of an imbalanced class distribution. To complicate the problem, there is the scarcity of public datasets for confidentiality issues, which does not allow researchers to verify the new strategies in many data contexts. Differently from the canonical state-of-the-art strategies, instead of defining a unique model based on the past transactions of the users, we follow a Divide and Conquer strategy, by defining multiple models (user behavioral patterns), which we exploit to evaluate a new transaction, in order to detect potential attempts of fraud. We can act on some parameters of this process, in order to adapt the models sensitivity to the operating environment. Considering that our models do not need to be trained with both the past legitimate and fraudulent transactions of a user, since they use only the legitimate ones, we can operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past. Such a way to proceed also overcomes the data imbalance problem that afflicts the machine learning approaches. The evaluation of the proposed approach is performed by comparing it with one of the most performant approaches at the state of the art as Random Forests, using a real-world credit card dataset.

1 INTRODUCTION

Any business that carries out activities on the Internet and accepts payments through debit or credit cards, also implicitly accepts all the risks related to them, like for some transaction to be fraudulent. Although these risks can lead to significant economic losses, nearly all the companies continue to use these powerful instruments of payment, as the benefits derived from them will outweigh the potential risks involved. Fraud is one of the major issues related with the use of debit and credit cards, considering that these instruments of payment are becoming the most popular way to conclude every financial transaction, both online and in a traditional way. According to a study of some years ago conduct by the *American Association of Fraud Examiners*[1], fraud related with the financial operations are the 10-15% of the whole fraud cases. However, this type of fraud is related to the 75-80% of all involved finances with an estimated average loss per fraud case of 2 million of dollars, in the USA alone. 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.

Open Problems. Considering that the number of fraudulent transactions is typically much smaller than that of the legitimate ones, the distribution of data is highly unbalanced (Batista et al., 2004), reducing the effectiveness of many learning strategies used in this field (Japkowicz and Stephen, 2002). The problem of the unbalanced data distribution is further complicated by the scarcity of information in a typical record of a financial transaction, which generates an overlapping of the classes of expense of a user (Holte et al., 1989). A fraud detection system can basically operate following two different learning strategies: static and dynamic (Pozzolo et al., 2014). Through the static strategies, the model used to detect the frauds is completely generated after a certain time period, while in the dynamic strategies it is generated one time, then

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

Saia, R., Boratto, L. and Carta, S..
Multiple Behavioral Models: A Divide and Conquer Strategy to Fraud Detection in Financial Data Streams.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 496-503
ISBN: 978-989-758-158-8

updated after a new transaction. Most of the state-of-the-art approaches used in this context are based on the detection of the suspicious changes in the user behavior, a quite trivial approach that in several cases leads toward false alarms. This because numerous of these approaches do not include some non-numeric data in the evaluation process, due to their incapacity to manage them (e.g., the machine learning approaches, such as the Random Forests, are not able to manage many categories, typically not more than 32).
Our Contribution. The vision behind this paper is to extend the canonical criteria, integrating to them the ability to operate with heterogeneous information, and by adopting multiple behavioral patterns of the users. This approach reduces the problems previously underlined, related with the scarcity, heterogeneity, non-stationary distribution, and presence of an imbalanced class distribution, of the transactions data. This is possible because we take into account all parts of a transaction, considering more information about it, contrasting the scarcity of information that leads toward an overlapping of the classes of expense. By means of the generation of multiple behavioral models of a user, made by dividing the sequence of transactions in several event-blocks, we face instead the problem of the non-stationarity of data, modeling anyway the user behavior effectively.

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 its behavioral pattern to each of the behavioral patterns of the user. 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 (i.e., numeric and non-numeric), converting them in absolute numeric variations between each pair of contiguous events;
- definition of the *Transaction Determinant Field* (TDF) set, a series of distinct values extracted from a field of the transaction, and used to give more importance to certain elements of a transaction, during the fraud detection process;
- introduction of the *Event-block Shift Vector* (EBSV) operations, made by sliding a vector of size *eb* (event-block) over the sequence of absolute variations previously calculated, in order to store, in the behavioral patterns of a user, the average values of the variations measured in each event-block;
- definition of a discretization process used to adjust the sensitivity of the system in the fraud detection process, by converting the continuous values in the behavioral patterns in output to the EBSV process, in a number of *d* levels (*discretization*);
- formalization of the process of evaluation of a new transaction, performed by comparing, through the *cosine similarity*, its behavioral pattern with the user behavioral patterns in *P*, in order to assign it a certain level of reliability.

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

2 RELATED WORK

The 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, an so on), exploiting a user model built on the basis of the past transactions of the user. In (Assis et al., 2013), the authors show how in the field of automatic fraud detection there is lack of real datasets (publicly available) indispensable to conduct experiments, as well as a lack of publications about the related methods and techniques.
Supervised and Unsupervised Approaches. In (Phua et al., 2010) it is underlined how the *unsupervised* fraud detection strategies are still a very big challenge in the field of E-commerce. Bolton and Hand (Bolton and Hand, 2002) show how it is possible to face the problem with strategies based both on statistics and on *Artificial Intelligence* (*AI*), two effective approaches in this field able to exploit powerful instruments (such as the *Artificial Neural Networks*) in order to get their results. In spite the fact that every *supervised* strategy in fraud detection needs a reliable training set, the work proposed in (Bolton and Hand, 2002) takes in consideration the possibility to adopt an *unsupervised* approach during the fraud detection process, when no dataset of reference containing an adequate number of transactions (legitimate and non-legitimate) is available. Another approach based on two *data mining* strategies (*Random*

Forests and *Support Vector Machines*) is introduced in (Bhattacharyya et al., 2011), where the effectiveness of these methods in this field is discussed.

Data Unbalance. As previously underlined, the unbalance of the transaction data represents one of the most relevant issues in this context, since almost all of the learning approaches are not able to operate with this kind of data structure (Batista et al., 2000), i.e., when an excessive difference between the instances of each class of data exists. Several techniques of pre-processing have been developed to face this problem (Japkowicz and Stephen, 2002; Drummond et al., 2003).

Detection Models. The *static approach* (Pozzolo et al., 2014) represents a canonical way to operate to detect fraudulent events in a stream of transactions. It is based on the initial building of a user model, which is used for a long period of time, before its rebuilding. In the so-called *updating approach* (Wang et al., 2003), instead, when a new block appears, the user model is trained by using a certain number of latest and contiguous blocks of the sequence, then the model can be used to infer the future blocks, or aggregated into a big model composed by several models. In another strategy, based on the so-called *forgeting approach* (Gao et al., 2007), a user model is defined at each new block, by using a small number of non-fraudulent transactions, extracted from the last two blocks, but keeping all previous fraudulent ones. Also in this case, the model can be used to infer the future blocks, or aggregated into a big model composed by several models. In any case, regardless of the adopted approach, the problem of the non-stationary distribution of the data, as well as that of the unbalanced classes distribution, remain still unaltered.

Differences with Our Approach. The proposed approach introduces a novel strategy that, firstly, takes into account all elements of a transaction (i.e., numeric and non-numeric), reducing the problem related with the lack of information, which leads toward an overlapping of the classes of expense. The introduction of the *Transaction Determinant Field* (TDF) set, also allows to 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 an unique model, but instead on multiple user models that involve the entire set of data. This allows us to evaluate a new transaction by comparing it with a series of behavioral models related with many parts 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. This because the operative domain is represented by the limited event blocks, and not by the entire dataset. The discretization of the models, according to a certain value of d, permit us to adjust their sensitivity to the peculiarities of the operating environment.

3 PROBLEM DEFINITION

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

Definition 3.1 (Input Set)**.** *Given a set of users* $U = \{u_1, u_2, \dots, u_M\}$*, a set of transactions* $T = \{t_1, t_2, \dots, t_N\}$*, and a set of fields* $F = \{f_1, f_2, \dots, f_X\}$ *that compose each transaction t (we denoted as* $V = \{v_1, v_2, \dots, v_W\}$*, 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}*).*

Definition 3.2 (Fraud Detection)**.** *The main objective of a fraud detection system is the isolation and ranking of the potentially fraudulent transactions (Fan and Zhu, 2011) (i.e., by assigning a high rank to the potential fraudulent transactions), since in the real-world applications this allows a service provider to focus the investigative efforts toward a small set of suspect transactions, maximizing the effectiveness of the action, and minimizing the cost. For this reason, we evaluate the ability of our fraud detection strategy in terms of its capacity to assign a high rank to frauds, using as measure the average precision (denoted as* α*), since it is considered the correct metric in this context (Fan and Zhu, 2011). Others metrics commonly used to evaluate the fraud detection strategies, such as the AUC (a measure for unbalanced datasets), and PrecisionRank (a measure of precision within a certain number of observations with the highest rank) (Pozzolo et al., 2014), then are not taken in consideration in this work.*

The formalization of the average precision is shown in Equation 1, where N is the number of transactions in the set of data, and $\Delta R(t_r) = R(t_r) - R(t_r - 1)$*. Denoting as* π *the number of fraudulent transactions in the set of data, 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* $precision(t) = h(t)/t$ *values, then the value of* α*.*

$$\alpha = \sum_{r=1}^{N} P(t_r)\Delta R(t_r) \tag{1}$$

Lemma 1. *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.*

Corollary 1. *When the set processed by the Equation 1 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 Definition 3.2 we have $\pi = 1$ and $t = 1$. Consequently, from the previous Lemma 1, 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.*

Definition 3.3 (Performed Tasks). *In order to operate with only numeric elements, able to characterize the sequence of transaction events, we transform the set T in the set $\hat{T} = \{\hat{t}_1 = |t_2 - t_1|, \hat{t}_2 = |t_3 - t_2|, \ldots, \hat{t}_N = |t_N - t_{N-1}|\}$, where $|\hat{T}| = (|T| - 1)$, and each subtraction operation is performed on all fields $f \in F$ of the considered transactions, by using a different criterion for each type of data. We also denote as $I = \{i_1, i_2, \ldots, i_Z\}$ the set of behavioral patterns generated at the end of the shift process, performed on the set $\hat{T}$, where the shift operation aims to extract the average value of a certain number (defined by the event-block parameter) of contiguous variations of the set $\hat{T}$. The purpose of this process is the definition of a set of behavioral patterns, which takes into account a series of contiguous events (i.e., the average variation), instead of only one (or all). To uniform all the variations in I in a certain range of values, we define a new set $P = \{p_1, p_2, \ldots, p_Y\}$, with contains the same elements of I, but where the value of each field $f \in F$ is discretized, according to certain number of levels (defined by the discretization parameter d, with $d \geq 2$)). It should be noted that $|I| = |P|$.*

Problem 1. *For the reasons explained in Definition 3.2, our objective is to maximize the α value, by ordering the new transactions on the basis of their similarity with the behavioral patterns in P, in order to rank the fraudulent transactions ahead the legal ones:*

$$\max_{0 \leq \alpha \leq 1} \alpha = \sum_{r=1}^{N} P(t_r) \Delta R(t_r) \tag{2}$$

4 OUR APPROACH

The steps needed to implement our strategy can be grouped into the following five steps:

- **Absolute Variation Calculation:** conversion of the transactions set T of a user into a set of absolute numeric variations between two contiguous transactions $t \in T$, adopting a specific criterion for each type of data in the set F;
- **TDF Definition:** creation of a *Transaction Determinant Field* (TDF) set, a series of distinct terms, extracted from the field *place*, used to define a binary element in each pattern of the set P, allowing to give more relevance to this field during the fraud detection process;
- **EBSV Operation:** application of a *Event-block Shift Vector* (EBSV) over the set of absolute numeric variations $\hat{T}$, aimed to calculate the average value of the elements in the event-block *eb*, storing the results as patterns in the set I;
- **Discretization Process:** discretization of the average values in the set I, in accord with a defined number of levels d (discretization). It allows to adjust the sensitivity of the system during the fraud detection process. The result of this operation, along with the result of the TDF query, defines the set of behavioral patterns P;
- **Transaction Evaluation:** assignation of a level of reliability to a new transaction, by comparing all patterns in the set P with the pattern obtained by inserting the transaction to evaluate as last element of the set T, repeating the process previously described only for the last *eb* transactions.

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 *Transaction Determinant Field*). Differently from a canonical preprocessing approach, which in such contexts usually has the task to convert the non-numeric values into numeric ones, the output of this step are the absolute numeric variations calculated between contiguous transaction events.

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 $f_x^{(t_{n-1})}$), as shown in Equation (3). The result is the absolute difference between the values taken into account.

$$NAV = |f_x^{(t_n)} - f_x^{(t_{n-1})}| \tag{3}$$

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 $f_x^{(t_{n-1})}$), as shown in Equation 4). The result is the absolute difference in days, between the two dates taken in account.

$$TAV = |days(f_x^{(t_n)} - f_x^{(t_{n-1})})| \tag{4}$$

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 5.4.2, as shown in Equation 5). The result is a value in the range from 0 (complete dissimilarity) to 1 (complete similarity).

$$DAV = \mathrm{lev}_{f_x^{(t_n)}, f_x^{(t_{n-1})}} \tag{5}$$

4.2 TDF Definition

In order to define the *Transaction Determinant Field* (TDF) from a field that we decide to consider as crucial in the fraud detection process (in our case, the field *place*), we extract from the set of transactions all distinct values $v_1, v_2, \ldots, v_W$ of this field, storing them in a new set $\hat{V} = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_W\}_{\neq}$, according with the formalization introduced in Section 3. The set $\hat{V}$ 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, defined as described in Section 4) is set to 1, otherwise to 0. It should be noted that this value is always set to 1 in the behavioral patterns related with the past transactions of the user. In other words, the TDF process operates like a *drift detector* (Kuncheva, 2008), and allows us to give more importance to certain parts of the transaction, during the building of the behavioral models.

4.3 EBSV Operation

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 the shift operation by sliding the *Event-block Shift Vector* over the sequence of absolute variation values stored in $\hat{T}$, one step at a time, extracting the average value of the variations present in the defined event-block *eb*. Given a event-block $eb = 3$, a set of variations $\hat{T} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, we can execute a maximum of $|C|$ shift operations, with $|C| = |I| = (|\hat{T}| - |eb| - 1)$, as shown in the Equation 6.

$$\begin{aligned}
&\hat{T} = [v_1, v_2, v_3, v_4, v_5, v_6]\\
&\qquad\Downarrow\\
&c_1 = \tfrac{v_1+v_2+v_3}{|eb|}, c_2 = \tfrac{v_2+v_3+v_4}{|eb|}\\
&c_3 = \tfrac{v_3+v_4+v_5}{|eb|}, c_4 = \tfrac{v_4+v_5+v_6}{|eb|}\\
&\qquad\Downarrow\\
&I = [c_1, c_2, c_3, c_4]
\end{aligned} \tag{6}$$

The sequence of values calculated in each event-block *eb*, for each considered 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 shift process, adding to them the binary value determined by querying the *Transaction Determinant Field* set (as described in Section 4.2), before using them in the evaluation process of a new transaction. It is a process quite similar to that performed in the context of the *time series* (Hamilton, 1994), but in this case the data in input are the numeric absolute variations measured between the numeric and non-numeric fields of all transactions, and the output is a set of user behavioral models.

4.4 Discretization Process

The continuous values $f \in F$ present in the pattern set I, obtained through the shift operation described in Section 4.3), must be transformed in discrete values, in accord with a certain level of *discretization d*. It allow us to determine the level of sensitivity of the system during the fraud detection process. The result is a set $P = \{p_1, p_2, \ldots, p_Y\}$ of patterns that represent the behavior of a user in different parts of her/his transaction history. Given a discretization value d, and a set of patterns I, each continuous value v_c of a field f (i.e., we process only the fields *description*, *date*, and *amount*, because the field *place* assumes a binary value determined by the TDF process) is transformed in a discrete value v_d, following the process shown in the Equation 7.

$$v_d = \left\lceil \frac{v_c}{\left(\frac{max(f) - min(f)}{d}\right)} \right\rceil \tag{7}$$

4.5 Transaction Evaluation

To evaluate a new transaction, we need to compare each behavioral pattern $p \in P$ with the single behavioral pattern $\hat{p}$ obtained by inserting the transaction to evaluate as last element of the set T, repeating the entire process previously described (variation calculation, shift, and discretization) only for the transactions present in the last event-block (i.e., the event-block composed by the last $|time\text{-}frame|$ transactions of the set T, were the last one element is the transaction to evaluate). The comparison is performed by using the *cosine similarity* metric (described in Section 5.4.1), and the result is a series of values in the range from 0 (transaction completely unreliable) to 1 (transaction completely reliable). It should be noted that the value of the field *place* depends on the result of the query operated on the TDF set, as described in the Section 4.2. The value of similarity is the average of the sum of the minimum and maximum values of cosine similarity $cos(\theta)$, measured between the pattern $\hat{p}$ and all patterns of the set P, i.e., $sim(\hat{p},P) = (min(\cos(\theta)) + max(\cos(\theta)))/2$. The result is used to rank the new transactions, on the basis of 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 private dataset related to one-year (i.e., 2014) of credit card transactions, 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 proposed EBSV approach was developed in Java, while the implementation of the state-of-the-art approach, used to evaluate its performance, was made in R^2, using the *randomForest* package.

5.2 Dataset

The dataset used for the training, in order to generate the set of behavioral patterns P, 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 *place* reported is *Internet*). Considering that all transactions in the dataset are legal, we have $T_+ = 204$ and $T_- = 0$. The fields of the transaction taken in consideration are five: *Type of transaction*, *City of transaction*, *Date of transaction*, and *Amount in Euro*. It should be noted that we do not consider any metadata (e.g., mean value of expenditure per week or month).

5.3 Strategy

Considering that it has been proved (Pozzolo et al., 2014) that the *Random Forests* (RF) approach outperforms the other approaches at the state of the art, in this work we chose to compare our EBSV approach only to this one. For the reason described in Section 3, we perform this operation by comparing their performance in terms of Average Precision (AP). 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, as shown in Table 1 (they have been marked as *unreliable*, as well as the other ones have been marked as *reliable*). We perform the experiments following the *k-fold cross-validation* criterion. Regarding the EBSV 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 user models. We repeat the same previous steps for the RF approach, 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.

Since the RF approach is not able to operate a textual analysis on the transaction description, and that is well-known that the RF approaches are biased by the categorical variables that generate many levels (such as the *Description* field), we do not use this field in the RF implementation. In addition, in order to work with the same type of data, in the RF implementation we converted the information of the field Date, in time intervals between transactions, expressed in days. For reasons of reproducibility of the RF experiments, we fix the seed value of the random number generator by the method *set.seed(123)* (the value is not relevant). The RF parameters (e.g., the number of trees to grow) have been defined in experimental way,

[2] https://www.r-project.org/

Table 1: Fraudulent Transactions Set.

TransactionID		Fields Values (1=anomalous 0=regular)				
From	*To*	*Description*	*Place*	*Date*	*Amount*	*Status*
1	2	1	0	0	0	unreliable
3	4	0	1	0	0	unreliable
5	6	0	0	1	0	unreliable
7	8	0	0	0	1	unreliable
9	10	1	1	1	1	unreliable

by researching those that minimized the *error rate* given as output during the RF process. The experiments are articulated in two steps: in the first step, we define the values to assign to the parameters that determine the performance of the EBSV approach (i.e., *event-block* and *discretization*), as described in Section 5.5; in the second step, we evaluate the EBSV performance, comparing to the RF approach, by testing the ability to detect a number of $2,4,\ldots,10$ fraudulent transactions (respectively, a frauds percentage of $2.8\%,5.5\%,\ldots,12.8\%$).

5.4 Metrics

This section reports the metrics used during the experiments, as well as those involved in our approach.

5.4.1 Cosine Similarity

In order to evaluate the similarity between the behavioral pattern of a transaction under analysis, and each of the behavioral patterns of the user, generated at the end of the process exposed in Section 4, we use the cosine similarity metric. The output of this measure is bounded in $[0,1]$, with 0 that means complete diversity, and 1 complete similarity. Given two vectors of attributes x and y (i.e., the behavioral patterns), the cosine similarity, cos(θ), is represented using a dot product and magnitude as shown in Equation 8.

$$\text{similarity} = \cos(\theta) = \frac{x \cdot y}{\|x\|\|y\|} = \frac{\sum\limits_{i=1}^{n} x_i \times y_i}{\sqrt{\sum\limits_{i=1}^{n} (x_i)^2 \times} \sqrt{\sum\limits_{i=1}^{n} (y_i)^2}} \quad (8)$$

5.4.2 Levenshtein Distance

The *Levenshtein Distance* is a metric able to measure the difference between two sequences of terms. Given two strings a and b, it indicates the minimal number of insertions, deletions, and replacements, needed to transforming the string a into the string b. Denoting as $|a|$ and $|b|$ the length of the strings a and b, the *Levenshtein Distance* is given by $\text{lev}_{a,b}(|a|,|b|)$, as shown in Equation 9.

$$\text{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if} \min(i,j) = 0 \\ \min \begin{cases} \text{lev}_{a,b}(i-1,j)+1 \\ \text{lev}_{a,b}(i,j-1)+1 \\ \text{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \quad (9)$$

Where $1_{(a_i \neq b_j)}$ is the *indicator function* equal to 0 when $a_i = b_j$ 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.

5.4.3 Average Precision

The average precision (*AP*) is considered as the correct measure to use in the fraud detection context, as described in Definition 3.2. Given N the number of transactions in the dataset, $\Delta Recall(t_r) = Recall(t_r) - Recall(t_r - 1)$, π the number of fraudulent transactions in the dataset (out of the percent t of top-ranked candidates), $h(t) \leq t$ the truly relevant transactions, $Recall(t) = h(t)/\pi$, and $Precision(t) = h(t)/t$, we can obtain the *AP* value as shown in Equation 10.

$$AP = \sum_{r=1}^{N} Precision(t_r)\Delta Recall(t_r) \quad (10)$$

5.5 Parameter Tuning

Considering that the performance of our approach depends on the parameters *eb* (*event-block*) and *d* (*discretization*), before evaluating its performance, we need to detect their optimal values. To perform this operation we test all pairs of possible values of *eb* and *d*, in a range from 2 to 99 (to be meaningful, both values must be greater than 1). The criterion applied to choose the best values is the average precision AP, as described in Section 3. The experiments detected $eb = 41$ as best value of event-block, and $d = 11$ as best value of discretization (i.e., the best performance measured in all subsets involved in the *k-fold cross-validation* process).

5.6 Experimental Results

The final result is given by the mean value of the results of all experiments performed, in accord with the *k-fold cross-validation* criterion. As we can observe in Figure 1, the performance of the EBSV approach reach those of the RF one, and this without train its models with the past fraudulent transactions (as occurs in RF). This result shows an important aspect, i.e., that EBSV is able to operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past.

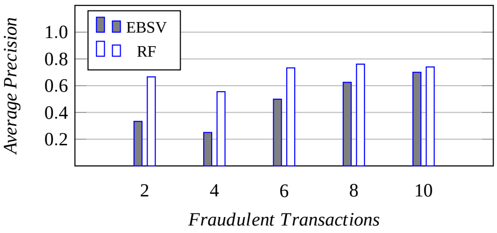

Figure 1: Experiment Results.

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 almost all 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, during the evaluation of a new transaction, the user behavioral in different temporal frames of her/his history. The possibility to adjust the levels of discretization and the size of the temporal frames, give us the opportunity to adapt the detection process to the operating environment characteristics. Considering that our approach does not need fraudulent transactions occurred in the past to build the behavioral models, it allows us to operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past, allowing also to overcome the problem of data imbalance, which afflicts the canonical machine learning approaches. The experimental results show that the performance of the proposed EBSV approach reach those of the state-of-the-art approach to which we compared (i.e., 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.

ACKNOWLEDGEMENTS

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

Assis, C., Pereira, A., Pereira, M., and Carrano, E. (2013). Using genetic programming to detect fraud in electronic transactions. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web*, pages 337–340. ACM.

Batista, G. E., Carvalho, A. C., and Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In *MICAI 2000: Advances in Artificial Intelligence*, pages 315–325. Springer.

Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–29.

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

Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, pages 235–249.

Drummond, C., Holte, R. C., et al. (2003). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11. Citeseer.

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

Gao, J., Fan, W., Han, J., and Philip, S. Y. (2007). A general framework for mining concept-drifting data streams with skewed distributions. In *SDM*, pages 3–14. SIAM.

Hamilton, J. D. (1994). *Time series analysis*, volume 2. Princeton university press Princeton.

Holte, R. C., Acker, L., Porter, B. W., et al. (1989). Concept learning and the problem of small disjuncts. In *IJCAI*, volume 89, pages 813–818. Citeseer.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449.

Kuncheva, L. I. (2008). Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. In *2nd Workshop SUEMA*, pages 5–10.

Phua, C., Lee, V. C. S., Smith-Miles, K., and Gayler, R. W. (2010). A comprehensive survey of data mining-based fraud detection research. *CoRR*, abs/1009.6119.

Pozzolo, A. D., Caelen, O., Borgne, Y. L., Waterschoot, S., and Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Syst. Appl.*, 41(10):4915–4928.

Wang, H., Fan, W., Yu, P. S., and Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. ACM.