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

Roberto Saia, Ludovico Boratto, Salvatore Carta
Dip.to di Matematica e Informatica, Università di Cagliari
Via Ospedale 72 - 09124 Cagliari, Italy
Email: {roberto.saia, ludovico.boratto, salvatore}@unica.it

***Abstract*—** 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. 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 almost all strategies at the state of the art, instead of producing a unique model based on the past transactions of the users, in this paper we present an approach that generates a set of models (behavioral patterns) that allow us to evaluate a new transaction, by considering the behavior of the user in different temporal frames of her/his history. The size of the temporal frames and the number of levels (granularity) used to discretize the values in the behavioral patterns, can be adjusted in order to adapt the system 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 at the state of the art. 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.

***Index Terms*—**Fraud detection, Pattern Mining, Rule learning.

I. Introduction

The exponential and rapid growth of the electronic commerce (E-commerce) based both on the new opportunities offer 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. 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 Issues. Considering that the number of fraudulent transactions is typically much smaller than legitimate ones, the distribution of data is highly unbalanced, reducing the effectiveness of many learning strategies used in this field [1]. 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 [2]. A fraud detection system can basically operate following two different learning strategies: static and dynamic [3]. 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 updated after a new transaction. There are several kind of approaches that are used in this context, such as those based on Data Mining [4], Artificial Intelligence [5], Fuzzy Logic [6], Machine Learning [7], or Genetic Programming [8]. The strategy used in many of the cited approaches is based on the detection of the suspicious changes in the user behavior, a quite 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 state-of-the-art approaches exclude some non numeric data from the evaluation process, due to their incapacity to manage it. This happens because employing machine learning approaches, such as the Random Forests, all the types of data that involve a lot of categories (typically 32) cannot be handled. Thinking about real-world transactional data, they usually involve much more than 32 categories (e.g., the places in the transactions).

Our Contribution. The vision behind this paper is to extend the canonical criteria, integrating them the ability to operate with heterogeneous information (i.e., numeric and non numeric data), 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 dis-

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

tribution, 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 time-frame, we face instead the problem of the non stationarity of data, modeling anyway the user behavior effectively.

The block diagram in Fig. 1 introduces the proposed approach step by step. As we can observe, the past transactions of a user are processed in order to define a series of behavioral models that characterize different parts of the transaction history of the user. Such process takes into account the importance of certain transaction elements in the fraud detection process, such as, for instance, the place where the transaction happens. The first block in Fig. 1, labeled *Transactions Set*, contains the initial set of transactions (past transactions of a user) to process in order to define a set of behavioral patterns. Its output depends on the presence of a new transaction te to evaluate in the input channel: in absence of it, we have as output all transactions; otherwise we have only the $tf-1$ transactions (where tf denotes the size of time-frame), followed by the te transaction to evaluate. This happens because in this case we need as output only a single behavioral pattern of tf size. As input of the second block (*Calculate Variations*), we have a set of transactions T, composed by the output of the previous block, after the removal of field (in our case, the field *place*) designed as *Transaction Field Keywords* (TFK), the part of a transaction to which we have decided to give more relevance during the fraud detection process, in accord with the operations of the block *TFK Process*, described in Section IV-B. The set of transactions T is processed by the block *Calculate Variations*, in order to convert it into absolute numeric variations measured between each pair of contiguous transactions, as described in Section IV-A. The absolute variations in the set $\hat{T}$ are processed in the *Convolution Operations* block, in accord with the value in the input channel tf, that defines the size of the *Time-frame Convolution Vector* (TFCV), as described in Section IV-C. The result is a set I of behavioral patterns. The next *Discretization Process* block converts the continuous values present in the set I, in output to the previous block, in discrete values, according with the value of granularity defined in the input channel g, as described in Section IV-D. The final set of *Behavioral Patterns* P, in the output of the entire process, is built by integrating the output of the block *Discretization Process* block, with the TFK information of the block *TFK Process*. The level of reliability of a new transaction is evaluated by comparing, through the *cosine similarity* the behavioral pattern P obtained by performing the entire process with the transaction to evaluate applied to input channel te, with the set of behavioral patterns generated following the same process without any transaction applied in this channel, as described in Section IV-E.

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

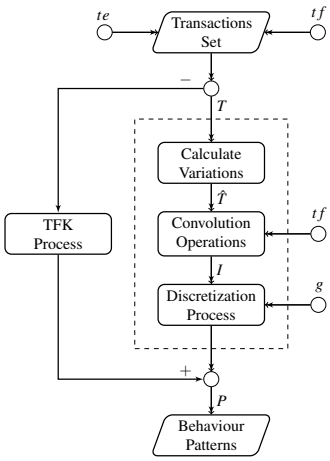

Fig. 1: System Architecture

definition need 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 (i.e., numeric and non numeric), converting them in absolute numeric variations between each pair of contiguous events;
- definition of the *Transaction Field Keywords* (TFK) 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 *Time-frame Convolution Vector* (TFCV) operations, made by sliding a vector of size tf (time-frame) 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 time-frame;
- 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 TFCV process, in a number of g levels (*granularity*);
- 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 II provides a background on the concepts handled by our proposal; Section III provides a formal notation and definition of the problem faced in this work; Section IV introduces the proposed model, presents the block diagram of a fraud detection system based on our strategy, and provides all the details about its implementation; Section V describes the experimental environment,

the adopted metrics, and the experimental results; the last Section VI reports some concluding remarks and future work.

II. Related Work

As highlighted in many studies, frauds represent the biggest problem in the E-commerce environment. The credit card fraud detection represents one of the most important context, 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 [8], 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.

The most common causes of this problem are the policies (for instance, competitive and legal) that usually stand behind every E-commerce activity, which makes it very difficult to obtain real data from business. Furthermore, such datasets composed by real information about user transactions could also reveal the potential vulnerabilities in the related E-commerce infrastructure, with a subsequent *loss of trust*.

Supervised and Unsupervised Approaches. In [9] it is underlined how the *unsupervised* fraud detection strategies are still a very big challenge in the field of E-commerce. Bolton and Hand [10] 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 [10] 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 [11], where the effectiveness of these methods in the field of the fraud detection 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 [12], i.e., when an excessive difference between the instances of each class of data exists. To face this problem, several techniques of pre-processing have been developed, aimed to balance the set of data [1]. The *undersampling* technique randomly removes the transactions, until the balancing has been reached, while the specular *oversampling* technique, obtains the balancing by additioning new transactions, created through an interpolation of the elements that belong to a same class [13].

Detection Models. The *static approach* [3] 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. An approach characterized by a simple learning phase, but not able to follow the changes of user behavior during the time.

In a static approach, the data stream is divided into blocks of the same size, and the user model is trained by using a certain number of initial and contiguous blocks of the sequence, which use to infer the future blocks.

In the so-called *updating approach* [14], 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* [15], 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.

The main disadvantages related of these approaches of user modeling are: the incapacity to follow the changes in the users behavior, in the case of the *static approach*; the ineffectiveness to operate in the context of small classes, in the case of the *updating approach*; the computational complexity in the case of the *forgetting approach*. However, regardless of the used 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 in 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 Field Keywords* (TFK) 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 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. This because the operative domain is represented by the limited temporal frames, and not by the entire dataset. The discretization of the models, according to a certain value of granularity, permit us to adjust their sensitivity to the peculiarities of the operating environment. In more details, regarding the analysis of the textual information related to the transactions, the literature presents several ways to operate, and most of them work in accord with the *bag-of-words* model, an approach where the words (for instance, type and description of the transaction) are processed without taking into account of the correlation between terms [16], [17]. This trivial way to manage the information does usually not lead toward good results, and just for this reason the basic approaches are usually flanked by complementary techniques aimed to improve their effectiveness [18], [19], or they are replaced by more sophisticated alternative based on the semantic analysis of the text [20], which proved to be effective in many

contexts, such as the recommendation one [21]. Considering the nature of the textual data related to a financial transaction, the adoption of semantic techniques could lead toward false alarms, as well as a trivial technique based on simple matching between words. This happen because, a conceptual extension of a the textual field of a transaction could evaluate as similar two transactions instead very different, while a simple matching technique could lead to consider as different some string of text, due to the existence of some slight differences (i.e., plural forms instead of singular, words different but with a common root, and so on). For this reason, in this work we adopt the *Levenshtein Distance*, a metric that measure the similarity between two textual fields in terms of minimal number of insertions, deletions, and replacements, needed to transforming the content of the first field into the content of the second one.

III. 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 sets): Given a set of users $U = \{u_1, u_2, \ldots, u_M\}$, a set of transactions $T = \{t_1, t_2, \ldots, t_N\}$, a set of absolute variations $\hat{T} = \{v_1 = |t_2 - t_1|, v_2 = |t_3 - t_2|, \ldots, v_N = |t_N - t_{N-1}|\}$, where $|\hat{T}| = (|T| - 1)$, and a set of fields $F = \{f_1, f_2, \ldots, f_X\}$ that compose each transaction t (we denoted as $k_1, k_2, \ldots, k_W$, 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}).

Definition 3.2 (Output sets): We denote as $I = \{i_1, i_2, \ldots, i_Z\}$ 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 F), and as $P = \{p_1, p_2, \ldots, p_Y\}$ 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 $|I| = |P|$.

Definition 3.3 (Fraud detection): The main objective of a fraud detection system is the isolation and ranking of the potentially fraudulent transactions [22] (i.e., by assigning an 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. In [22], the average precision (here denoted as α) is considered as the correct measure to use in this kind of process. Its formalization 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 3.3 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*.

Problem 1: An ideal fraud detection approach should have a value of α as close as possible 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, improving the effectiveness in the fraud attempts detection, as shown in Equation 2.

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

IV. Our Approach

The steps needed to implement our strategy, schematically shown in the block diagram in the Introduction (Fig. 1), 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;
- **TFK Definition**: creation of a *Transaction Field Keywords* (TFK) set, a series of distinct k 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;
- **TFCV Operation**: application of a *Time-frame Convolution Vector* (TFCV) over the set of absolute numeric variations $\hat{T}$, aimed to calculate the average value of the elements within the time-frame tf, which stores the results as patterns of the set I;
- **Discretization Process**: discretization of the average values in the set I, in accord with a defined number of levels g (granularity). 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 TFK 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 entire process previously described only for the last tf transactions.

A. 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 III, 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*). In our case, in accord with the adopted dataset (described in Section V-B), we need to define three type of operations: numeric absolute variation, temporal absolute variation, and textual absolute variation.

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 in Euros (since it is the currency used in the dataset), between the two amounts taken in 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 V-D2, as shown in Equation 5). The result is a value in the range from 0 (complete dissimilarity) to 1 (complete similarity).

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

B. TFK Definition

In order to define the *Transaction Field Keywords* (TFK) from a field that we decide to consider 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 with the formalization introduced in Section III. 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, defined as described in Section **??**) 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.

C. 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 IV-A, we operate a convolution operation 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 present 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 6.

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

The sequence of values calculated in each time-frame tf, 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 convolution process, adding to them the binary value determined by querying the *Transaction Field Keywords* in K (as described in Section IV-B), before using them in the evaluation process of a new transaction.

D. Discretization process

The continuous values v_c present in the patterns set I, obtained through the convolution operation described in Section IV-C), must be transformed in discrete values v_d, in accord with a certain level of *granularity* g. 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 behaviour of a user in different parts of her/his transaction history. Given a granularity $g = 10$, 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 TFK process) is transformed in a discrete value v_d, following the process shown in the Equation 7.

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

E. 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, convolution, and discretization) only for the transactions present in the last time-frame (i.e., the time-frame composed by the last $|time - frame|$ 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 V-D1), 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 TFK set, as described in the Section IV-B. 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.

V. 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.

A. Experimental Setup

In order to evaluate the proposed strategy, we perform a series of experiments using a real-world dataset related to one-year (i.e., 2014) of credit card transactions[2]. The proposed TFVC approach was developed in Java, while the implementation of the state-of-the-art approach, used to evaluate its performance, was made in the R[3] environment, using the *randomForest* package.

B. 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 $T_- = 0$. As shown in Table I, the fields that compose a transaction are 5, but in this work we do not take in account the *Transaction ID* field (TID), nor any metadata (e.g., mean value of expenditure per week or month).

C. Strategy

Considering that it has been proved [3] 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, excluding alternative approaches, such as *Support Vector Machine* (SVM), or *Neural Network* (NNET). For the reason described in Section III, 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 II (they have been marked as *unreliable*, as well as the other ones have been marked as *reliable*).

NR	*Field*	*Explanation*	*Type*
1	TID	Transaction ID	Numeric
2	Description	Type of transaction	Textual
3	Place	City of transaction	Textual
4	Date	Date of transaction	Date
5	Amount	Amount in Euro	Currency

TABLE I: Transaction Fields

During the experiments aimed to compare the performance of our TFCV approach, with those of the RF one, we adopt the *k-fold cross-validation* criterion. 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 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, by researching those that minimized the *error rate* given as output during the RF process. The experiments are articulated in the following two steps:

- definition of the values to assign to the parameters that determine the performance of the FTCV approach (i.e., *time-frame* and *granularity*), as described in Section V-E;
- evaluation of the TFCV performance, comparing to the RF approach, by testing the ability to detect a number of $2,4,\ldots,10$ fraudulent transactions (respectively, a fraudulent transactions percentage of $2.8\%, 5.5\%, \ldots, 12.8\%$).

D. Metrics

In this section, we present the metric used during the experiments.

[2] A private dataset provided by a researcher
[3] https://www.r-project.org/

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

TABLE II: Fraudulent Transactions Set

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 **??**, we use the cosine similarity metric. It allows to measure the similarity between two vectors (i.e., the behavioral patterns) of an inner product space that measures the cosine of the angle between them. Considering that the cosine of 0° is 1, and it is less than 1 for any other angle, in two vectors with the same orientation we measure a cosine similarity of 1. The output of this measure is then bounded in $[0,1]$, with 0 that means complete diversity, and 1 complete similarity. Given two vectors of attributes x and y, the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as shown in Equation 8.

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

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.

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

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.

E. Parameter Tuning

Considering that the performance of our approach depends on the parameters tf (*time-frame*) and g (*granularity*), before evaluating its performance, we need to detect their optimal values. To perform this operation we test all pairs of possible values of tf and g, 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 III. The experiments detected $tf = 46$ as best value of time-frame, and $g = 11$ as best value of granularity (i.e., the best performance measured in all subsets involved in the *k-fold cross-validation* process).

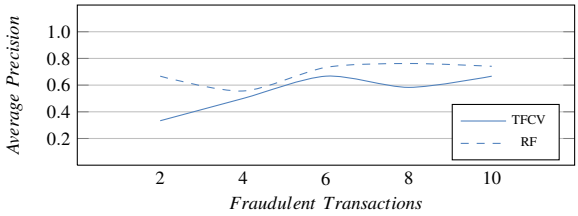

Fig. 2: Experiment Results

F. Experimental Results

As introduced in the Sections V-A and V-C, we test our TFCV strategy by using a real-world dataset T related to one-year of credit card transactions, where we have added 10 fraudulent transactions, the nature of which is defined in Table II. We adopt the *k-fold cross-validation* criterion, with $k = 3$, during all experiments, as specified in Sections V-C. The TFCV process generates a set of user behavioral patterns P, which we compare (i.e., using the cosine similarity metric) 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 IV-E. 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 Fig. 2, 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). This result shows an important aspect, as it means that TFCV approach is able to operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past.

VI. 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 granularity and the size of the temporal frames, give us the opportunity to adapt the detection process to the operating environment characteristics. The most important aspect to consider is however tied to the fact that, in our approach, the building of the behavioral models does not need examples of past fraudulent transactions, but is performed exclusively by exploiting the legitimate cases. This allow 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 *Time Frame Convolution Vector* approach are very close to those of the *Random Forests* (i.e., the state-of-the-art approach, to which we compared), and this without training our models with the past fraudulent transactions (as occurs

in *Random Forests*). 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] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.

[2] R. C. Holte, L. Acker, and B. W. Porter, "Concept learning and the problem of small disjuncts," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, N. S. Sridharan, Ed. Morgan Kaufmann, 1989, pp. 813–818.

[3] A. D. Pozzolo, O. Caelen, Y. L. Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014.

[4] M. Lek, B. Anandarajah, N. Cerpa, and R. Jamieson, "Data mining prototype for detecting e-commerce fraud," in *Proceedings of the 9th European Conference on Information Systems, Global Co-operation in the New Millennium, ECIS 2001, Bled, Slovenia, June 27-29, 2001*, S. Smithson, J. Gricar, M. Podlogar, and S. Avgerinou, Eds., 2001, pp. 160–165.

[5] A. J. Hoffman and R. E. Tessendorf, "Artificial intelligence based fraud agent to identify supply chain irregularities," in *IASTED International Conference on Artificial Intelligence and Applications, part of the 23rd Multi-Conference on Applied Informatics, Innsbruck, Austria, February 14-16, 2005*, M. H. Hamza, Ed. IASTED/ACTA Press, 2005, pp. 743–750.

[6] M. J. Lenard and P. Alam, "Application of fuzzy logic fraud detection," in *Encyclopedia of Information Science and Technology (5 Volumes)*, M. Khosrow-Pour, Ed. Idea Group, 2005, pp. 135–139.

[7] D. G. Whiting, J. V. Hansen, J. B. McDonald, C. C. Albrecht, and W. S. Albrecht, "Machine learning methods for detecting patterns of management fraud," *Computational Intelligence*, vol. 28, no. 4, pp. 505–527, 2012.

[8] C. Assis, A. M. Pereira, M. de Arruda Pereira, and E. G. Carrano, "Using genetic programming to detect fraud in electronic transactions," in *A Comprehensive Survey of Data Mining-based Fraud Detection Research*, C. V. S. Prazeres, P. N. M. Sampaio, A. Santanchè, C. A. S. Santos, and R. Goularte, Eds., vol. abs/1009.6119, 2010, pp. 337–340.

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

[10] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," *Statistical Science*, pp. 235–249, 2002.

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

[12] G. E. A. P. A. Batista, A. C. P. L. F. de Carvalho, and M. C. Monard, "Applying one-sided selection to unbalanced datasets," in *MICAI 2000: Advances in Artificial Intelligence, Mexican International Conference on Artificial Intelligence, Acapulco, Mexico, April 11-14, 2000, Proceedings*, ser. Lecture Notes in Computer Science, O. Cairó, L. E. Sucar, and F. J. Cantu, Eds., vol. 1793. Springer, 2000, pp. 315–325.

[13] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003.

[14] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, L. Getoor, T. E. Senator, P. M. Domingos, and C. Faloutsos, Eds. ACM, 2003, pp. 226–235.

[15] J. Gao, W. Fan, J. Han, and P. S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions," in *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*. SIAM, 2007, pp. 3–14.

[16] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. J. Palakal, "Detection of shifts in user interests for personalized information filtering," in *SIGIR*, 1996, pp. 317–325.

[17] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "Learning user interest dynamics with a three-descriptor representation," *JASIST*, vol. 52, no. 3, pp. 212–225, 2001.

[18] G. Armano, A. Giuliani, and E. Vargiu, "Studying the impact of text summarization on contextual advertising," in *2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011*, F. Morvan, A. M. Tjoa, and R. Wagner, Eds. IEEE Computer Society, 2011, pp. 172–176.

[19] A. Addis, G. Armano, and E. Vargiu, "Assessing progressive filtering to perform hierarchical text categorization in presence of input imbalance," in *KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Valencia, Spain, October 25-28, 2010*, A. L. N. Fred and J. Filipe, Eds. SciTePress, 2010, pp. 14–23.

[20] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity: Measuring the relatedness of concepts," in *Demonstration Papers at HLT-NAACL 2004*, ser. HLT-NAACL–Demonstrations '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 38–41.

[21] R. Saia, L. Boratto, and S. Carta, "Semantic coherence-based user profile modeling in the recommender systems context," in *Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2014, Rome,Italy, October 21-24, 2014*. SciTePress, 2014.

[22] G. Fan and M. Zhu, "Detection of rare items with target," *Statistics and Its Interface*, vol. 4, pp. 11–17, 2011.