Discovering Internal Fraud Models in a Stream of Banking Transactions

Fabien Vilar[1], Marc Le Goc[1], Philippe Bouche[2] and Pierre-Yves Rolland[1]
[1]*Laboratory for Sciences of Information and Systems (LSIS), UMR CNRS 7296, Marseille, France*
[2]*TOM4, Pelissanne, France*

Keywords: Data Mining, Knowledge Engineering, Online and Real Time Fraud Detection, Fraud Modelisation.

Abstract: Internal frauds in the banking industry represent a huge cost and this problem is particularly difficult to solve because, by construction, swindlers being very imaginative persons, the fraud schemata evolves continuously. Fraud detection systems must then learn from the continuously new fraud schematas, making them difficult to design. This paper proposes a new theoretical and practical approach to detect internal frauds and to model fraud schematas. This approach is based on a particular method of abstraction that reduces the complexity of the problem from $O(n^2)$ to $O(n)$ making its implementation in a an Java program that detects and models the frauds in real time and online with a simple professional personal computer. The results of this program are presented with its application on a real-world fraud provided by a world wide French bank.

1 INTRODUCTION

Since the last three decades, the theoretical researches in the econometrics and the financial analysis domain have developed a stochastic approach of the analysis and the modeling of financial datas that are currently largely used in the bank and the financial industries (Fliess et al., 2011). During this period, the exponential development of the information system of the bank and the financial companies allowed the usage of Data Mining or Machine Learning algorithms to define new services, notably to bring elements to solve the problem of the internal fraud detection in the bank industry. But this problem is particularly difficult to solve because, by construction, swindlers being very imaginative persons, the models of fraud evolve continuously so that the fraud detection systems must learn from the new fraud schematas. This paper proposes a new theoretical and practical approach to detect internal frauds and to model fraud schematas thanks to a reduction of the complexity of the problem from $O(n^2)$ to $O(n)$. The proposed solution allows to detect and model internal frauds online, in real time, with a simple professional personal computer, this latter being able of handling more than 4 billions of transactions a day. The next Section describes the works related with the proposed approach. Section 3 introduces the mains concepts of the Timed Observations Theory (TOT, (Le Goc, 2006)) that are required for the online fraud detection and modeling from bank transactions. This framework has been implemented in the TOM4FFS program (Timed Observations Mining for Fraud Fighting System) and its results on the detection and the modeling of a real world, and particularly complex, example of internal fraud schemata concerning a world wide French bank are provided in Section 4. The Section 5 concludes this paper.

2 RELATED WORKS

To solve the general fraud detection problem, many statistic and machine learning techniques have been developed such as neural networks (Fanning and Cogger, 1998; Green and Choi, 1997), genetic algorithms (Hoogs and al., 2007), bayesian network and decision tree (Kirkos and al., 2007; Phua et al., 2004), K-nearest neighbour (Kotsiantis and al., 2006), logistic regression (Altman et al., 1994) and even rule-based fuzzy reasoning system (Deshmukh and Talluru, 1998). They have been used with more or less success in operation (cf. for example (Roddick and Spiliopoulou, 2002) for a complete state of the art about these techniques). Technically speaking, the main difficulties with these approaches are concerned with (i) the data set representativity and completeness problem and (ii) the high power of computation that most of the learning algorithms require to learn and to detect the fraud schemata from the huge amount

Vilar, F., Goc, M., Bouche, P. and Rolland, P..
Discovering Internal Fraud Models in a Stream of Banking Transactions.
In *Proceedings of the 7th International Joint Conference on Computational Intelligence (IJCCI 2015) - Volume 1: ECTA*, pages 346-351
ISBN: 978-989-758-157-1

of transactions that a bank handles each day. As a consequence, most of the solutions can't be used online in real-time at low costs. On the other hand, the paper (Fliess et al., 2011) puts on the light a conceptual problem that is inherent to all statistics-based approaches: the effect of the *sampling* on the datas. Usually, the quantity of raw data is so huge that one uses stochastic models to aggregate the raw datas in a minimal set of samples, with a reduced set of dimensions that maximizes the compactness of the learning space so that learning algorithms can work efficiently. This usual method introduces, in a devious way (often without the analysts realize it), the problem of the *loss of causality in data*. Having very small effects on most of the cases, this problem can be (and is) generally neglected. Nevertheless, it is significant and important when the timestamps of the raw data are pertinents for the calculations: in that case, biases are then introduced into both structure and value of the parameters of the learning models. And it becomes crucial when dealing with online data flows that must be analyzed in real time. Since most of the proposed Data Mining algorithms have been designed for non timed data, some extensions have been proposed to this aim as for example, AprioriAll (Agrawal and Srikant, 1995) or Winepi/Winepi (Mannila et al., 1995; Mannila et al., 1997). But, according to (Mannila, 2002) or (Han and Kamber, 2006), these algorithms present two main drawbacks: the number of discovered models increases in a nonlinear way with (i) the *a priori* setting of the values of the algorithm parameters and (ii) the threshold values of the decision criteria, even when only a very small fraction of these models are interesting. There are few papers dealing with the problem of fraud detection in online banking (Phua and al., 2010; Jans et al., 2009; Wei and al., 2012). Most of them concern external fraud detection in application domains far from banking industry. To solve these problems, the TOT defines a new Knowledge Discovery in Databases (KDD) approach, specifically designed to learn directly from timed data, without any parameter. The theoretical basis of this KDD approach, implemented in the TOM4FFS program, are described in the next sections.

3 INTRODUCTION TO THE TOT

The TOT defines a *dynamic process* as an arbitrarily constituted set $X(t) = \{x_1(t), ..., x_n(t)\}$ of timed functions $x_i(t)$ of continuous time t. The set $X(t)$ of functions implicitly defines a set $X = \{x_1, x_2, ..., x_n\}$ of n variable names denoted x_i for simplicity. A dynamic process $X(t)$ is said to be *observed* by a program Θ when this latter aims at writting *timed messages* describing the modifications over time of the functions $x_i(t)$ of $X(t)$. The aim of the TOT is precisely to model *observed processes* defined as a couple $(X(t), \Theta(X, \Delta))$. To this aim, the TOT defines a *timed observation* to provide a *meaning* to a timed message:

Definition 1. *Timed Observation*
Let $X(t) = \{x_i(t)\}_{i=1...n}$ be a set of time functions describing the evolution of a process that is observed by a program Θ; let $\Gamma = \{t_k\}_{t_k \in \Re}$ be a set of arbitrary time instants in which Θ observes the functions; let $\theta(x_\theta, \delta_\theta, t_\theta)$ be a predicate implicitly determined by Θ; and let $\Delta = \{\delta_j\}$ be a set of constant values.

A timed observation $(\delta_j, t_k) \in \Delta \times \Gamma$ made on the time function $x_i(t)$ is the assignation of values x_i, δ_j and t_k to the predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ such that $\theta(x_i, \delta_j, t_k)$.

The TOT notion of *observation class* makes the link between a variable x_i and a constant δ_j.

Definition 2. *Observation Class*
Let $X(t) = \{x_i(t)\}_{i=1...n}$ be a set of time functions that are observed by an abstract program $\theta(X, \Delta)$ where $\Delta = \{\delta_j\}_{j=1...m}$ is the set of all the constants the abstract program can use and $X = \{x_i\}_{i=1...n}$ is the set of variable names corresponding to $X(t)$.

$\forall i \in [1, n]$, $\forall j \in [1, m]$ and $\forall k \in \mathbb{N}$, an observation class $O_k = \{..., (x_i, \delta_j), ...\}$ is a subset of $X \times \Delta$.

A timed observation (δ_j, t_k) always can be considered as an occurrence $O_j(t_k)$ of an observation class $O_j = \{(x_i, \delta_j)\}$. In practice, temporal functions $x_i(t)$ describing the evolution of the process are piecewise functions (see section 4).

Definition 3. *Concrete Unary Observer*
A concrete unary observer $\Theta'(\{x_i\}, \mathbb{Z})$ is a program observing a piecewise function and implementing equation 1 to produce timed observations of the form $O_i(t_{k_i}) = (x_i(t_{k_i}) - x(t_{k_i - 1}), t_{k_i})$:

$$\forall t_{k_i}, t_{k_i - 1}, x_i(t_{k_i - 1}) \neq x_i(t_{k_i}) \Rightarrow write(O_i(t_{k_i})) \quad (1)$$

This generates a sequence $\omega_i = \{..., O_i(t_{k_i}), ...\}$ of instances of the observation class occurrences $O_i(t_{k_i})$.

A temporal binary relation is the representation of a *sequential* relation between two observations classes O_i and O_j:

Definition 4. *Temporal Binary Relation*
A temporal binary relation $r_{ij}(O_i, O_j, [\tau_{ij}^-, \tau_{ij}^+])$, $\tau_{ij}^- \in \Re$, $\tau_{ij}^+ \in \Re$, is an oriented relation between two observation classes O_i and O_j that is timed constrained with the $[\tau_{ij}^-, \tau_{ij}^+]$ interval.

Temporal constraint $[\tau_{ij}^-, \tau_{ij}^+]$ is the time interval to observe the timed observation $O_j(t_j) = (\delta_j, t_j)$ of the output observation class O_j after the timed observation $O_i(t_i) = (\delta_i, t_i)$ of the input observation class O_i, $t_j - t_i \in [\tau_{ij}^-, \tau_{ij}^+]$. In that case, the temporal binary relation $r_{ij}(O_i, O_j, [\tau_{ij}^-, \tau_{ij}^+])$ is said to be observed and is denoted $r_{ij}(O_i(t_i), O_j(t_j), [\tau_{ij}^-, \tau_{ij}^+])$.

A program $\Theta_i(O_i, O_j)$ designed to recognize a temporal binary relation of the form $r_{ij}(O_i, O_j, [\tau_{ij}^-, \tau_{ij}^+])$ is called an *abstract binary observers* (ABO).

Definition 5. *Abstract Binary Observer*
Any program $\Theta_i(O_i, O_j)$ implementing the equation 2 is an Abstract Binary Observer.

$$\forall t_{k_i} \in \Gamma_i, \forall t_{k_j} \in \Gamma_j, t_{k_j} \geq t_{k_i}$$
$$\exists O_i(t_{k_i}) \in \omega_i \wedge \exists O_j(t_{k_j}) \in \omega_j \wedge (t_{k_j} - t_{k_i}) \in [\tau_{ij}^-, \tau_{ij}^+]$$
$$\Rightarrow write(O_{ij}(t_{k_j})) \qquad (2)$$

$O_{ij}(t_{k_j})$ is a timed observation and an occurrence of an abstract observation class $O_{ij} = \{(x_{ij}, \delta_{ij})\}$ linking an *abstract variable x_{ij}* with the *abstract binary constant* $\delta_{ij} \equiv (\delta_i, \delta_j)$.

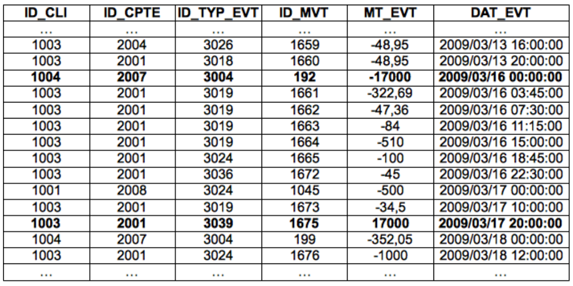
4 APPLICATION

The aim of this section is to present the application of the TOM4FFS program to detect and to model schemata of *potential* internal fraud. An internal fraud is a particular sequence of non-compliant transactions the aim of which is to move money from clients accounts to some accounts of a tactless manager of a bank. The role of TOM4FFS is to detect and to model, online and in real time, a schemata of *potential* internal fraud from a continuous flow of transactions. The detected transactions remain *potentially* fraudulent until their *non-compliance* have been confirmed. The fraud schemata of this example is based on a set of pairs $(a_{k_i}(t_{k_i}), a_{k_j}(t_{k_j}))$ of transactions where the first transaction $a_{k_i}(t_{k_i})$ sells an amount a_{k_i} of money from an account of a customer *before* the second transaction $a_{k_j}(t_{k_j})$ credits the *same amount* to one of the accounts of the administrator of the bank. The problem is then to find the minimal set of transaction pairs $(a_{k_i}(t_{k_i}), a_{k_j}(t_{k_j}))$ satisfying the following constraints: (i) from a customer account which is not an account of the manager, (ii) to an account of a manager, (iii) where $a_{k_i} = -a_{k_j}$ and (iv) $t_{k_i} \leq t_{k_j}$. It is clear that given a database of n transactions, the complexity of this problem is $O(n^2)$: when n is evaluated in millions (i.e. 10^6), the number of pairs to evaluate must be evaluated in millions of millions (i.e. 10^{12}), making the problem difficult for humans as for computers. As an illustration, the internal fraud schemata studied in this section has been detected by a client and required an analysis of 6 months for the bank's expert in internal fraud. It is to precise that the studied example is considered as particularly complicated by this expert. According to the TOT, a bank transaction $a_{k_i}(t_{k_i})$ is a timed observation $((x_i(t_{k_i}) - x_i(t_{k_i-1}), t_{k_i})$ associated with a bank account $x_i(t)$ of a particular customer. In this application, since it will be seen that the cents can be neglected in a first step, $x_i(t)$ is considered as a piecewise constant time function defined over $\mathbb{Z}$ (cf. figure 2 for an illustration). The constant $(x_i(t_{k_i}) - x_i(t_{k_i-1}))$ is the *amount* of money that has been moved from the account $x_i(t)$ at time t_{k_i}: it is a natural number of cents, which is positive when the account $x_i(t)$ is credited and negative either. For example, when considering the transaction (`1|8|24|1169|-189.64`, `2009/09/29 15:45:25`), the symbol "`|`" structures the sequence of characters `1|8|24|1169|-189.64` in different items: the customer number `1`, the account number `8`, the type of the transaction `24`, the index of the transaction `1169` and the amount of the transaction `-189.64`. The sequence of characters `2009/09/29 15:45:25` being the timestamps, the transaction is then represented with the timed observation $(-189.64, 2009/09/29\ 15:45:25)$. The other items, the customer number, the account number, the type of the transaction and the index of the transaction are then considered as attributes of the timed observations. These attributes are defined with an ontology describing the customers, the accounts and the types of transactions with frames similarly to the Manchester OWL syntax. As a consequence, a timed observation $(-189.64, 2009/09/2915:45:25)$ is both an instance of a frame customer, a frame account or a frame type of transactions. In the suite of this section, a fraud schemata is represented with a set of binary relations between customer's accounts. The problem with this representation of the transaction is that, the number of constant δ is *a priori* infinite (i.e. equal to $\aleph_0$, the cardinal of $\mathbb{Z}$). Recalling that an observation class $O_i = \{(x_i, \delta_i)\}$ is a singleton associating a constant δ_i with one and only one variable name x_i (cf. definition 2), the number of observation classes is also infinite. As a consequence, the set of timed binary relations that are required to recognize all the possible pairs of transactions $(a_{k_i}(t_{k_i}), a_{k_j}(t_{k_j}))$ from (i) a customer account to (ii) an account of a manager where (iii) $a_{k_i} = -a_{k_j}$ and (iv) $t_{k_i} \leq t_{k_j}$ is also infinite. When forgetting the cents, the constants $\delta_i \equiv ((x_i(t_{k_i}) - x_i(t_{k_i-1}))$ of each of these timed observations can then be any natural number of $\mathbb{Z}$. So, an infinite set of timed binary relations is required to constitute the pairs of transactions satisfying the required logical constraints. To solve this problem, the idea is to define a compact representation of the amounts, inspired from the Benford's Law (Benford, 1938), also called the *First-Digit Law*. An amount $m = (x_i(t_{k_i}) - x_i(t_{k_i-1}))$ of a transaction can be represented with a signed sum of powers of 10: $\forall z \in \mathbb{Z}, z = s(z) \cdot \sum_{k=0}^{n} a_k \cdot 10^k$. In this representation, (i) $s(z)$ is the sign function of z i.e. $s : \mathbb{Z} \to \{-1, 1\}, z \mapsto -1$ if $z < 0$, 1 otherwise, (ii) n is the highest power of 10 of z ($n \geq 0$), (iii) $a_k \in D = \{1, 2, ..., 8, 9\}$ is the digit defining the value of the coefficient of the k^{th} power of z. With the First-Digit Law in mind, the advantage of this representation, when using only the digits of D, it is possible to the following classification function to represent the set of transaction's amount with a much more smaller set $O = \{O_i\}$ of observation classes O_i:

Definition 6. *Classification function*

The classification function μ maps any $z \in \mathbb{Z}$ to a particular $\mu(z)$ of the set $M = \{$..., -21, ..., -11, -9, -8, ..., -2, -1, 0, 1, 2, ..., 8, 9, 11, ..., 21, ... $\}$, $\mu : \mathbb{Z} \to M, z = s(z) \cdot \sum_{k=0}^{n} a_k \cdot 10^k \mapsto \mu(z) = s(z).(10.n + a_n)$.

The classification function $\mu(z)$ only uses the *first digit* of z and n, its *highest power of 10*. In practice, the TOM4FFS program is set with a maximum value n_{max} of power of 10 to create a finite set O of $(20 \cdot n_{max} + 1)$ observation classes to analyze transactions the amounts of which are contained in the range $[-(10.n_{max} + 1); 10.n_{max} + 1]$. For example, with $n_{max} = 9$, the highest digit of D, TOM4FFS creates 181 observation classes $O = \{O_{-91}, O_{-90}, ..., O_{-1}, O_0, O_{+1}, ..., O_{+90}, O_{+91}\}$ to take into account amounts contained in the range $]-10^9; 10^9[$ (i.e. a billion). This interval is largely sufficient to the aim of the internal fraud detection. With these 181 observation classes, TOM4FFS creates (i) a unique concrete unary observer $\Theta_i(\{\phi_i\}, \mathbb{Z})$ implementing the classification function μ of definition 6 where ϕ_i is the name of an abstract time function $\phi_i(t)$ that transforms the transaction's amounts $s(z) \cdot \sum_{k=0}^{n} a_k \cdot 10^k$ in occurrences $O_k(t_k)$, and (ii) a network of 181 ABO's, each of them being specified with a timed binary relation of the form $r_{ij}(O_i, O_j, [\tau_{ij}^-, \tau_{ij}^+])$ where O_i and O_j are opposite value of the μ classification function. So, since the μ classification function is only concerned with $10.n + a_n$, the default structure of the ABO (cf. equation 2) is *independent* of the customer, the account and the type of transaction: it is only concerned with the checking of the classes and the time constraint $[\tau_{ij}^-, \tau_{ij}^+]$. The ABO's are then set with (i) a predicate `equal` to evaluate the constraint of equality of the *true values* of the amount transactions (i.e. the cents are taken into account) and (ii) two simple propositions to check if the transactions are concerned with a costumer (and not the manager) for constraint 1 and the manager uniquely for constraint 2. The studied database Ω is composed of 1492 transactions (cf. figure 1) between the bank administrator (ID_CLI = 1003) and his 3 clients (ID_CLI $\in \{1001, 1002, 1004\}$). This concerns 30 banking accounts (column ID_CPTE) and 40 transactions types (column ID_TYP_EVT) numbered from 3001 to 3040. Client 1001 owns 8 accounts: 2006, 2008, 2009, 2012, 2013, 2014, 2015 and 2022. Client 1002 owns 5 accounts: 2005, 2011, 2019, 2023 and 2027. Bank administrator owns 10 accounts: 2001, 2002, 2003, 2004, 2024, 2026, 2028, 2029, 2030 and 2031. Client 1004 owns 7 accounts 2007, 2010, 2016, 2017, 2018, 2020, 2021. Those transactions cover a period of more than one year, from 2009/01/02 to 2010/03/12 and involve amount of money from -445 200€ to 460 614.36€. The aim is to find among these transactions, those who are potentially fraudulent. The set of constant Δ is the set M of the definition 6 so that O is the set of observation classes. Let $\Gamma = \{t_k, t_k \in \Omega\}$ be the set of time instants contained in the database Ω. Fraud detection and modelisation are presented here by observing three different processes: accounts, clients and transactions types processes. Each process $X(t) = \{x_i(t), i = 1...n_X\}$ is made with n_X piecewise timed functions. So each unary observer $\Theta_i'(\{x_i\}, \mathbb{Z})$ implementing equation 1 generates a sequence of timed observations $\omega_i = \{O_i(t_{k_i}) = (\delta_i, t_{k_i}), \delta_i \in \Delta, t_{k_i} \in \Gamma\}$. This creates a network of 181 ABOs of the form $r_{ij}(O_i, O_j, [\tau_{ij}^-, \tau_{ij}^+])$. The time constraints of each ABO are arbitrary set to $\tau_{ij}^- = 0$ day and $\tau_{ij}^+ = 30$ days so that only pairs of transactions $(O_i(t_{k_i}), O_j(t_{k_j}))$ where $t_{k_j} - t_{k_i} \in [0, 30]$ are considered by the ABO. To observe accounts process, let's consider here the process $X(t) = \{x_{2001}(t), ..., x_{2031}(t)\}$ made with 30 piecewise timed functions $x_i(t)$ defined on $\mathbb{Z}$ corresponding to the 30 banking accounts $ID\_CPTE \in \{2001, ..., 2031\}$ contained in the database Ω. Let's take the case of the account number 2007 whose piecewise timed function $x_{2007}(t)$ is represented on figure 2. The unary observer $\Theta_{2007}'(\{x_{2007}\}, \mathbb{Z})$ observes this function. At



ID_CLI	ID_CPTE	ID_TYP_EVT	ID_MVT	MT_EVT	DAT_EVT
...	...	...	...	...	...
1003	2004	3026	1659	-48,95	2009/03/13 16:00:00
1003	2001	3018	1660	-48,95	2009/03/13 20:00:00
1004	**2007**	**3004**	**192**	**-17000**	**2009/03/16 00:00:00**
1003	2001	3019	1661	-322,69	2009/03/16 03:45:00
1003	2001	3019	1662	-47,36	2009/03/16 07:30:00
1003	2001	3019	1663	-84	2009/03/16 11:15:00
1003	2001	3019	1664	-510	2009/03/16 15:00:00
1003	2001	3024	1665	-100	2009/03/16 18:45:00
1003	2001	3036	1672	-45	2009/03/16 22:30:00
1001	2008	3024	1045	-500	2009/03/17 00:00:00
1003	2001	3019	1673	-34,5	2009/03/17 10:00:00
1003	**2001**	**3039**	**1675**	**17000**	**2009/03/17 20:00:00**
1004	2007	3004	199	-352,05	2009/03/18 00:00:00
1003	2001	3024	1676	-1000	2009/03/18 12:00:00
...	...	...	...	...	...

Figure 1: Extract of the Banking Transactions Data Base.

time $t_0 = 2009/03/16\ 00:00:00$, the value of $x_{2007}(t)$ changes and is -17000 which is mapped by the classification function (see definition 6) to -41. So unary observer creates a timed observation $O_{-41}(2009/03/16\, 00:00:00) = (-41, 2009/03/16\ 00:00:00)$ which is a representation of the transaction $a_1 =$ (`1004|2007|3004|192|-17000.00, 2009/03/16 00:00:00`) (see first line in bold of figure 1). Finally it adds it to the sequence of timed observations $\omega_{2007} = \{..., O_{-41}(2009/03/16\, 00:00:00), ...\}$. In parallel, unary observer $\Theta'_{2001}(\{x_{2001}\}, \mathbb{Z})$ does the same job, observing piecewise function $x_{2001}(t)$ corresponding to the account 2001 of the manager. It thus generates a sequence of timed observations ω_{2001}. Each timed observation of ω_{2001} is a reprensentation of a transaction in database Ω. In particular, $O_{+41}(2009/03/17\ 20:00:00)$ represents transaction $a_2 =$ (`1003|2001|3039|1675|17000.00, 2009/03/1720:00:00`) (see second bold line of figure 1). As a consequence, the ABO implementing the relation $r_{-4141}(O_{-41}, O_{+41}, [0,30])$ will be activated when receiving the timed observation $O_{+41}(2009/03/17\ 20:00:00)$ after the observation $O_{-41}(2009/03/16\ 00:00:00)$, the time constraint $[0,30]$ being satisfied. As a consequence, the ABO will write the binary timed observation $O_{-4141}(2009/03/17\ 20:00:00)$ denoting that the corresponding transactions a_1 and a_2 are potentially fraudulent. Doing so with the 1492 successive transactions of the studied database, the network of 181 ABO's of the TOM4FFS program produces the 8 binary timed observations of figure 3. The schemata of these potential fraudulent transactions is represented in figure 4. In this figure, the triangles represent the observation classes of the manager, the two other forms representing those of two other costumers (1001 and 1004). The labels represents total of the moved money under the form of an interval $[m_i, m_j]$: for example, the label [-17 000€, 17 000€] of the relation between the accounts 2007 and 2001 means that -17 000€ have been moved from the account 2007 and 17 000€ have been moved to the account 2001. The fraud schemata of figure 4 has been validated by the internal fraud expert of the French bank: the manager stole a total of 104 000€ from two costumers, 1004 (49 000€) and 1001 (55 000€). Now let's observe clients process and consider here the process $X(t) = \{x_{1001}(t), ..., x_{1004}(t)\}$ made with 4 piecewise timed functions $x_i(t)$ defined on $\mathbb{Z}$ corresponding to the 3 client ids (ID_CLI $\in \{1001, 1002, 1004\}$) and the bank administrator id (ID_CLI = 1003) contained in the database Ω. The same methodology as previously presented is applied and leads to the fraud scheme shown in figure 5. This confirms the fact that the manager has stolen a total of 104 000€ from two clients: 49 000€ from client 1004 and 55 000€ from client 1001. Finally, to observe transactions types process, let's consider here the process $X(t) = \{x_{3001}(t), ..., x_{3040}(t)\}$ made with 40 piecewise timed functions $x_i(t)$ defined on $\mathbb{Z}$ corresponding to the 40 transactions types ids contained in the database Ω. Figure 6 shows which types of transactions are involved in the fraud scheme. The manager uses five types of transactions among the 40 to steel his clients.

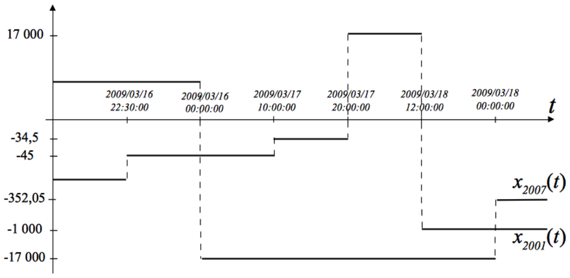
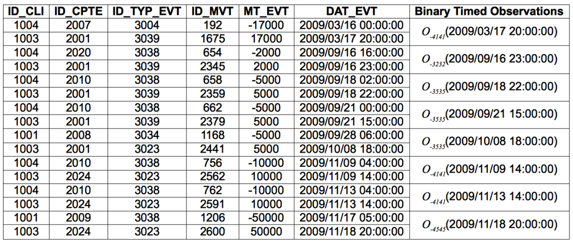


Figure 2: Piecewise functions for accounts 2001 and 2007.



ID_CLI	ID_CPTE	ID_TYP_EVT	ID_MVT	MT_EVT	DAT_EVT	Binary Timed Observations
1004	2007	3004	192	-17000	2009/03/16 00:00:00	O_{-4141}(2009/03/17 20:00:00)
1003	2001	3039	1675	17000	2009/03/17 20:00:00	
1004	2020	3038	654	-2000	2009/09/16 16:00:00	O_{-3232}(2009/09/16 23:00:00)
1003	2001	3039	2345	2000	2009/09/16 23:00:00	
1004	2010	3038	658	-5000	2009/09/18 02:00:00	O_{-3535}(2009/09/18 22:00:00)
1003	2001	3039	2359	5000	2009/09/18 22:00:00	
1004	2010	3038	662	-5000	2009/09/21 00:00:00	O_{-3535}(2009/09/21 15:00:00)
1003	2001	3039	2379	5000	2009/09/21 15:00:00	
1001	2008	3034	1168	-5000	2009/09/28 06:00:00	O_{-3535}(2009/10/08 18:00:00)
1003	2001	3023	2441	5000	2009/10/08 18:00:00	
1004	2010	3038	756	-10000	2009/11/09 04:00:00	O_{-4141}(2009/11/09 14:00:00)
1003	2024	3023	2562	10000	2009/11/09 14:00:00	
1004	2010	3038	762	-10000	2009/11/13 04:00:00	O_{-4141}(2009/11/13 14:00:00)
1003	2024	3023	2591	10000	2009/11/13 14:00:00	
1001	2009	3038	1206	-50000	2009/11/17 05:00:00	O_{-4545}(2009/11/18 20:00:00)
1003	2024	3023	2600	50000	2009/11/18 20:00:00	

Figure 3: Binary Timed Observations.

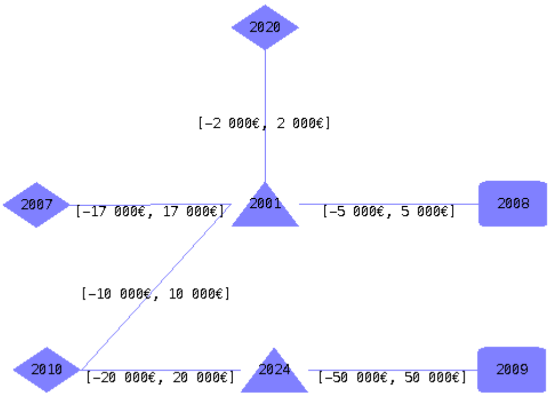


Figure 4: Fraud scheme for accounts.

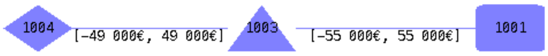


Figure 5: Fraud scheme for clients.

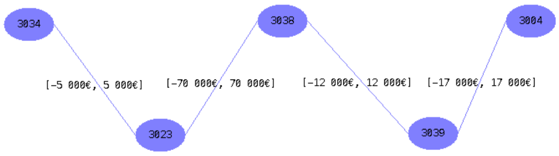


Figure 6: Fraud scheme for transactions types.

5 CONCLUSION

Since the last three decades, Data Mining or Machine Learning algorithms are used to pursue the delicate problem of the fraud detection in the bank industry. These algorithms pose three main problems: (i) the strong reluctance with which the bankers agree to supply in a third party a set of real-world transactions for confidentiality reasons, (ii) the problem of the data set representativity, and to a lesser extent, its completeness, and (iii) the huge amount of transactions that must be analyzed to detect the potential frauds. This paper presents an operational program, called TOM4FFS, to solve these three problems. The main advantages of this method are (i) to be purely syntactic what guarantees a strict confidentiality and (ii) to reduce the complexity of the problem of the fraud detection from $O(n^2)$ to $O(n)$. The TOM4FFS program is then able to handle more than 4 billions of transactions a day, online and in real time, with a standard personal computer. This paper describes the TOM4FFS program and its application to a real-world fraud example of a world wide French bank. Our current works are concerned with the extension of the approach to more complex fraud schemata and its application to the general problem of the conformity in the banking and other industries.

REFERENCES

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. *Proceedings of the 11th International Conference on Data Engineering (ICDE95)*, pages 3–14.

Altman, E., Marco, G., and Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience). *Journal of banking & finance*, 18(3):505–529.

Benford, F. (1938). The law of anomalous numbers. *Proceedings of the American Philosophical Society*.

Deshmukh, A. and Talluru, L. (1998). A rule-based fuzzy reasoning system for assessing the risk of management fraud. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 74:223–241.

Fanning, K. and Cogger, K. (1998). Neural network detection of management fraud using published financial data. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 7:21–41.

Fliess, M., Join, C., and Hatt, F. (2011). Is a probabilistic modeling really useful in financial engineering? In *Conférence Méditerranéenne sur L'Ingénierie Sûre des Systèmes Complexes*.

Green, B. and Choi, J. (1997). Assessing the risk of management fraud through neural network technology. *Auditing*, 161:14–28.

Han, J. and Kamber, M. (2006). *Data Mining. Concepts and Techniques*. Morgan Kaufmann.

Hoogs, B. and al. (2007). A genetic algorithm approach to detecting temporal patterns indicative of financial statement fraud. *Intelligent Systems in Accounting, Finance and Management*, 15:41–56.

Jans, M., Lybaert, N., and Vanhoof, K. (2009). A framework for internal fraud risk reduction at it integrating business processes: the ifr framework. *The International Journal of Digital Accounting Research*, 9:1–29.

Kirkos, E. and al. (2007). Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*.

Kotsiantis, S. and al. (2006). Forecasting fraudulent financial statements using data mining. *International Journal of Computation Intelligence*, 3:104–100.

Le Goc, M. (2006). *Notion d'observation pour le diagnostic des processus dynamiques: Application à Sachem et à la découverte de connaissances temporelles*. Hdr, Aix-Marseille University, Faculté des Sciences et Techniques de Saint Jérôme.

Mannila, H. (2002). Local and global methods in data mining: Basic techniques and open problems. *29th International Colloquium on Automata, Languages and Programming*.

Mannila, H., Toivonen, H., and Verkamo, A. I. (1995). Discovering frequent episodes in sequences. In Fayyad, U. M. and Uthurusamy, R., editors, *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, Montreal, Canada. AAAI Press.

Mannila, H., Toivonen, H., and Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289.

Phua, C. and al. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

Phua, C., Alahakoon, D., and Lee, V. (2004). Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59.

Roddick, F. J. and Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, (14):750–767.

Wei, W. and al. (2012). Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web: Internet and Web Information Systems*, 16:449–475.