Miguel, Amblard, Barceló & Madella (eds.) Advances in Computational Social Science and Social Simulation
Barcelona: Autònoma University of Barcelona, 2014, DDD repository <http://ddd.uab.cat/record/125597>

Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research

Edgar Alonso Lopez-Rojas
Department of Computer Science and Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
Email: edgar.lopez@bth.se

Stefan Axelsson
Department of Computer Science and Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
Email: stefan.axelsson@bth.se

***Abstract*—We present a social simulation model that covers three main financial services: Banks, Retail Stores, and Payments systems. Our aim is to address the problem of a lack of public data sets for fraud detection research in each of these domains, and provide a variety of fraud scenarios such as money laundering, sales fraud (based on refunds and discounts), and credit card fraud. Currently, there is a general lack of public research concerning fraud detection in the financial domains in general and these three in particular. One reason for this is the secrecy and sensitivity of the customers data that is needed to perform research. We present *PaySim, RetSim, and BankSim* as three case studies of social simulations for financial transactions using agent-based modelling. These simulators enable us to generate synthetic transaction data of normal behaviour of customers, and also known fraudulent behaviour. This synthetic data can be used to further advance fraud detection research, without leaking sensitive information about the underlying data. Using statistics and social network analysis (SNA) on real data we can calibrate the relations between staff and customers, and generate realistic synthetic data sets. The generated data represents real world scenarios that are found in the original data with the added benefit that this data can be shared with other researchers for testing similar detection methods without concerns for privacy and other restrictions present when using the original data.**

I. Introduction

Modelling the social financial behaviour of individuals is not a simple task. The social behaviour of individuals include many complex transactions. These interactions are driven by many factors and are constrained by the context surrounding them. In this paper we cover an important topic concerning the human financial interactions in the financial transactions domain. Unfortunately, whenever money is involved, there is been a risk of fraud.

Fraud is an important problem in a number of different situations. The economic impact can be substantial. The detection of fraud is therefore a worthwhile endeavour. However, in order to investigate, develop, test and improve fraud detection techniques there is a need for detailed information about the domain and its peculiarities.

All these needs can be satisfied if we had access to publicly available data of financial transactions so that different approaches could be compared and contrasted. Unfortunately for several reasons, including confidentiality, protection of privacy, the law, internal policies and regulations, it is hard if not impossible for an outside researcher to get access to such a data. Hence, research has historically been hampered by a lack of publicly available relevant data sets. Our aim with this work is to address that situation.

This paper is an effort to deal with the lack of public available financial data, with the idea that if we can not get access to public financial records due the restrictions mentioned before, then one good alternative is to use a simulator to generate financial data. However simulating a financial environment and generating data brings new challenges, specifically those related to characteristics of the generated data such as quality, privacy, realistic and usefulness.

We present three different case studies in the area of the social simulation of financial transactions for fraud detection research. The first consists of a new payment system that uses mobile phones to ease the payments, called *PaySim* [1]. Our access to base level data was poor at the time of that research being performed. Thus, we experienced difficulties to build an accurate model. This lead our research to our second case study called *RetSim* [2]. *RetSim* is a simulation tool that generates realistic scenarios of a retail store based on transactional data from one of the biggest shoe retailers in Scandinavia. The last case study is called *BankSim*, which is our first approach towards the simulation of bank transactions; payments and transfers between different people and merchants. *BankSim* is based on the public available aggregated transactions shared by a bank in Spain, with the main objective of promoting applications for Big Data uses of their services.

The main goal of developing these simulators is that it enables us to produce and share realistic fraud data with the research community, without exposing potentially sensitive and private information about the actual source.

Simulation also have other benefits: it can produce more data much faster and with less cost than for instance, collecting data, and one can try different scenarios of fraud, detection algorithms, and personnel and security policy approaches, in an actual store, for example, the introduction of new supervisors, security cameras, auditing routines, etc. The latter also risks incurring e.g. unhappiness among the staff, due to trying e.g. an ill advised policy, which leads to even greater expense and problems.

The main contribution of our approach is a method to generate anonymous synthetic data of a "typical" financial chain, that can then be used as part of the necessary input data for the research, development and testing of fraud detection techniques, both research prototypes and commercially available systems. Also, the data set generated could be the basis for research in other fields, such as marketing, demand prediction, logistics and demand/supply research.

The rest of this paper is organised as follows: Section II presents related work on simulation and fraud detection for the financial domain. Section III describes the methodology used for our research. Section IV presents *PaySim*. Section V presents *RetSim*. Section VI presents *BankSim*. We present a description of the model, evaluation and results for the simulators and finish with a discussion in section VII and conclusions, including future work in section VIII.

II. Background and Related Work

Simulations in financial domains have traditionally been built to predict markets changes, stocks prices and more specifically in the domain of retail stores for finding answers to logistics problems such as inventory management, supply management, staff scheduling and for customer queue reductions [3]. Our work uses similar techniques of financial modelling but has a different focus, which is the generation of synthetic data sets for fraud detection research.

Some of the benefits of using a synthetic data set for testing machine learning algorithms have been previously addressed by us [4]. We argue that: data that represent realistic scenarios can be made readily available; the privacy of the customer is not impacted; disclosure of results is not affected by policies or legal issues; the generated data set can be made available for other researchers to reproduce experiments; and different scenarios can be modelled by changing the parameters controlled by the researcher.

There has been work in the area of privacy preserving methods for data mining [5]. However, since the main problem in our experience usually is to get access to the data in the first place, our approach is to try and generate data that can then be shared without problems from a privacy perspective. The actual analysis method then does not need to be privacy preserving.

Social Network Analysis is a topic that is currently being combined with Social Simulation [6]. Both topics support each other in the representation of interactions and behaviour of agents in the specific context of social networks. However, there is no work in the field of customer/salesman interaction that we are aware of.

Money laundering threatens the economic and social development of countries. Due to the high amount of transactions and the variety of money laundering tricks and techniques, it is difficult for the authorities to detect money laundering and prosecute the wrongdoers. Thus, it is not only the ever increasing amount of transactions, but the ever changing characteristics of the methods used to launder money that are constantly being modified by the fraudsters which makes this problem interesting to study.

In Sweden and other countries, most companies in the financial sector are required by law to implement money laundering detection. The cost of implementing such controls for AML is quite high, mainly because of the amount of manual labour required. In Sweden alone the cost is estimated to be around 400 million SEK annually [7]. The most recent notorious case of money laundering is the HSBC Bank case [8], where the lack of AML controls lead to large amounts of money being laundered and injected into the U.S. financial system from countries under strict control, such as Mexico and Iran.

The most common method today used for preventing illegal financial transactions consists on flagging different clients according to perceived risk and restricting their transactions using thresholds [9]. Transactions that exceed these thresholds require extra scrutiny whereby the client needs to declare the precedence of the funds. These thresholds are usually set by law without distinction made between different economic sectors or actors. This of course leads to fraudsters adapting their behaviour in order to avoid this kind of controls, by e.g. making many smaller transactions that fall just below the threshold. Hence, these and other similar methods have proven insufficient [7].

New promising research in the field of data mining based methods have also been used to detect fraud [10]. This leads to the observation that machine learning algorithms can identify novel methods of fraud by detecting those transactions that are different (anomalous) in comparison to the benign transactions. Supervised learning algorithms have been used on synthetic data to prove the performance of outliers detection in different domains [11].

Several machine learning techniques have been used for the detection of fraud, and more specifically money laundering [12]. The application of machine learning to the problem is advantageous in many situations [13], [14]. However, to our knowledge, there are not a sufficient number of studies on this topic with public financial data to determine whether one detection method is better than another. Our simulators aim to close this gap and allow these researchers and organisations interested in fraud detection research to test, compare and develop new methods.

III. Methodology

We developed three different case studies on financial transactions. The first consists of a payment system that uses mobile phones to easy the payments *PaySim*, introduced in [1]. The second is *RetSim*, a simulation tool that generates realistic scenarios based on transactional data from one of the biggest shoe retail stores in Scandinavia. [2], [15]. The last, is *BankSim*, a simulator built on a sample of aggregated transactional data that one Spanish bank made available for a contest to encourage the development of applications in the *big data* field and specifically based on their data set. All simulators use the same Multi-Agent Based Simulation toolkit, called MASON, which is implemented in Java [16].

PaySim was based only on the schema of the database and the described behaviour of the customers for the simulated system. At the time of development, the system was in a testing phase, which made it impossible for us to obtain realistic data to calibrate the behaviour of the agents. However, we used the generated data to illustrate the possibilities and usefulness of the model by first generating a synthetic data set and second by performing an example of fraud detection using labelled data and machine learning techniques to classify the injected malicious behaviour.

PaySim is still waiting for real data from our partner in order to move forward with the calibration of the simulation and experimentation on diverse fraud scenarios. This situation made us focus our attention on our second case study RetSim.

RetSim started with the contribution of real data from a new partner, one of the largest Nordic shoe retailers. This data contains several hundred million records of diverse transactional data from all their stores from a few years ago, and also covering several years. This data is recent enough to reflect current conditions, but old enough to not pose a serious risk from a competitor analysis standpoint.

To better understand the problem domain, specifically the normal operation of a store (which is the domain from where we have access to data), we began by performing a data analysis of the historical data provided by the retailer. We were interested in finding necessary and sufficient attributes to enable us to simulate a realistic scenario in which we could reason about and detect interesting cases of fraud. This information was useful to build a social network interaction between customers and salesmen.

Fraud analysis has traditionally been strongly associated with network analysis. This is because of the possibility of several actors participating in a specific fraud in order to confuse the investigators and dilute the evidence, hence describing a network of actors, companies, ownership etc. By doing this we aim to model the micro behaviour of the different agents that captures the observed macro behaviour and gives rise to a total picture of the store. We generated a social network from the relation between customers and salesmen. We measured and use its properties to simulate a similar network with the aim of preserving interesting properties from the original social network such as topology, average in-degree and out-degree distribution of the salesmen and customers that are relevant to fraud detection.

We have no known instances of fraud in the real data (as certified by the data owner). So we had to inject malicious behaviour, by programming agents that behave according to some known or hypothesised retail fraud case presented before: *Refunds* and *Discounts*.

IV. PaySim, a Mobile Money Payment Simulator

Mobile Money Payment Simulation case study is based on a real company that has developed a mobile money implementation that provides mobile phone users with the ability to transfer money between themselves using the phone as a sort of electronic wallet. The task at hand is to develop an approach that detects suspicious activities that are indicative of money laundering.

Unfortunately, this service has only been running in a demo phase. This situation prevent us to collect any data that can be used for analysis of possible detection methods of illegal money transfers.

We modelled and implemented a Multi-Agent Based Simulator that uses the schema of the real mobile money service, but can generate synthetic data based on unknown scenarios that we based on our guess of what could be possible when the real system starts operating.

The simulation contains one agent that represent the clients of the service. The agents are represented by the class *Client* which extends to two child classes (*ClientSimA and Fraudster*). The inherit model allows an agent to rewrite specific behaviour of a client but implement its own specific behaviour. We created different types of agents and instantiate them together in the class *Clients* to represent the normal behaviour of clients and fraudsters.

Each clients has four possible actions in each step of the simulation. They can either make a *deposit*, a *withdrawal*, a *transfer* or simply “decide” not to do anything. The autonomy of the agent is implemented by a probabilistic transition function that computes the type of operation and the action that an agent will perform in each step. This transition function depends on the attributes of the client such as *Age* and the amount is calculated according to the balance and the limits previously defined for each client profile. To reflect what a realistic scenario could look like, we used the thresholds imposed by the original money laundering system.

For each simulation we can modify the parameters and the probabilities of occurrence for the transitions in order to improve the quality of the simulation. It is difficult to find the right probabilities that model a realistic scenario. Our implementation is based on pseudo random transitions. The given probabilities are based on 3 different configurations for the percentage of account balance in comparison to the maximum limit allowed by the client profile (Lower than 15%, higher than 80% and *medium balance* which is between *low* and *high*). The agent has a higher probability of making a deposit when the balance is low. When the balance is high the agent has a higher probability of making a withdrawal or a transfer, rather than a deposit.

A. Description of Scenarios

Our chosen scenario is an hypothetical situation where 200 clients from 4 different cities perform several transactions with partners inside or outside their city. We decided to have around 10% of the clients behaving as malicious agents (fraudsters). In a real scenario it is more common to find a lower percentage of fraudsters. The idea behind a higher proportion of fraudsters is to prevent the class imbalance problem during the training of the detector. All of the fraudsters were connected in a network where the 3 roles of the money laundering chain were represented (injection, layering and integration).

The social network between the clients was built restricting the network to a maximum of five contacts per client inside the city, and two outside the city. The fraudsters can also interact with normal clients of the system.

All the transactions are stored in a log file. The simulation was run five times for 1000 steps. Each step represents a time unit that we assume is the transaction rate of the clients (1/3 per day). The files generated were merged and ultimately used as input for the machine learning algorithms presented in sect. IV-B.

B. Results

In total we simulated 486977 transactions over 5 simulations, each one with 200 agents performing 1000 steps. A total of 6006 transactions were generated by 107 malicious agents and labelled as *suspicious*. Each of the malicious agents was designed with a specific goal in mind, chosen from the money laundering cycle that involves the three stages: placement (40), layering (33), and integration (34). The data generated by the simulation represent a realistic situation of the class imbalance problem, where one of the classes is very large in comparison to the other one. In this case only 1.23% of the total data is suspicious. For the experiment we ran different supervised algorithms that were selected for the purpose of classifying the class labelled as suspicious transactions.

The results can be seen in Table I and II. We can see that *JRip* produces the best accuracy in TP (True Positive) rate and FP (False Positives) rate in comparison with the other algorithms. The MC (Misclassified) number of instances is a bit higher than for the other algorithms e.g J48graft or Random-Forest.

TABLE I
RESULTS FOR THE CLASS *money laundering* (SUSPICIOUS)

Algorithm	**TP**	**FP**	**MC**
Naive-Bayes	0.988	0.479	8543
Decision-Table	0.999	0.029	200
Jrip	0.999	0.012	115
Random-Forest	0.999	0.009	66
Random-Tree	0.999	0.015	173
J48graft	0.999	0.014	118

TABLE II
CONFUSION MATRIX

Algorithm	JRip		Random-Forest		J48graft	
class*	**a**	**b**	**a**	**b**	**a**	**b**
a	5934	72	5954	52	5922	84
b	43	480928	14	480957	34	480937
	* a=Normal b=Suspicious					

C. Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [17]. The verification ensures that the simulation correspond to the described model presented in the chosen scenarios. We can easily check the constraints in the generated data such as positive balance numbers, account age, consistency between the transfers, deposits and withdrawals with the changes in account balances. Validation of the model is a bit more complex, since we need to ascertain whether the model is an accurate representation of a real world situation. Since we do not have real world data at this time, we need to rely on a description of the desired scenario and the opinion of experts in the field to validate that the basic statistics and the overall process of the simulation design correspond to a real world scenario. The complexity of the agents also matter here, the simpler the agents the easier is to validate the model.

Calibrating the model to a realistic scenario was rather hard in this simulation. From this difficulty we learnt a lot about the importance of accessing and sharing real data for fraud detection. Hopefully soon we will be able to get our hands on real data from this system that will help to improve the accuracy of the simulator.

V. RetSim, a retail store simulator

Since we have access to several years worth of transaction data from one of the largest Scandinavian retail shoe chains, we developed *RetSim*, a **Ret**ail shoe store **Sim**ulation, built on the concept of Multi Agent Based Simulation (MABS). RetSim is intended to be used for developing and investigating fraud scenarios at a shoe retail store, while keeping business sensitive and private personal information about customers consumption secret from competitors and others.

The defence against fraud is an important topic that has seen some study. In the retail store the cost of fraud are of course ultimately transferred to the consumer, and finally impacts the overall economy. Our aim with the research leading to RetSim is to learn the relevant parameters that governs the behaviour in and of a retail store to simulate *normal* behaviour, which is our focus in this paper. However we also touch upon the simulation of malicious behaviour and detection. As fraud in the retail setting is usually perpetrated by the staff we have focused on that. Examples of such fraud is e.g: *Sales cancellations* The salesman cancels the purchase of some items on the receipt and doesn't tell the customer, pocketing the difference. *Refunds* The salesman creates fraudulent refund slips and keeps the cash refund. *Coupon reductions/discounts* The salesman registers a discount on the sale and doesn't tell the customer, pocketing the difference. In many of these cases the fraud is simplified if the customer is an accomplice.

A. Model

The design of *RetSim* was based on the ODD model introduced by Grimm et.al. [18]. ODD contains 3 main parts: *Overview*, *Design Concepts* and *Details*.

We aim to produce a simulation that resembles a real retail store. Our main purpose is to generate a synthetic data set of business transactions that can be used for the development and testing of different fraud detection techniques. It is important due to the difficulty of finding a sufficient amount of diverse cases of fraud in a real data set. However this is not the case in a simulated environment, where fraud can be injected following known patterns of fraud.

There are three agents in this simulation: *Manager*, *Sales clerk* and *Customer*.

Manager This agent decides the price, check inventory and order new items.

Sales clerk Is in charge of promoting the items and issues the receipt after each sale. A sales clerk can be in state busy when the clerk is serving its maximum amount of customers.

Customer The behaviour is determined by the goal of purchasing one or several items. A customer is in an active *need-help* state, when no sales clerk is assisting the customer with its shopping.

1) Process overview and scheduling: During a normal step of the simulation, a customer enters the simulation, and a sales clerk sense nearby customers in the *need-help* state and offers help. There are two different outcomes: Either a transaction takes place, with probability p, or no transaction takes place with, trivially, probability $1 - p$.

The time granularity of the simulation is each step representing a day of sales. So a normal week has seven steps and a month will consist of around 30 steps. We do not make any explicit distinction between specific days of the week. Instead we handle differences between days by using a different distribution of the customers per day.

2) Design Concepts: The *basic principle* of this model is the concept of a commercial transactions. We can observe an *emergent* social network from the relation between the customers and the sales clerks. Each of the customers have the *objective* of purchasing articles from the store. The sales clerks *objective* is to aid the customers and produce the receipt necessary for the generation of the data set. Managers play a special role in the simulation. They serve as the schedulers for the next step of the simulation. Given the specific step of the simulation the manager generate a supply of customers for the next day and activate or deactivate specific sales clerks in the store. In our virtual environment the *interaction* between agents is always between sales clerk and customer. Purchase articles from another customer or selling articles to a sales clerk is not permitted.

Customers and sales clerks can scout the store in any radial direction from their current position and search or offer help, respectively.

The agents do not perform any specific learning activities. Their behaviour is given by probabilistic Markov models where the probabilities are extracted from the real data set.

3) Details: The simulation starts with a number of sales clerks that serve the customers, an initial number of customers and one manager that does the scheduling.

The in-degree distribution is used as an indication of how good a sales clerk can be. Each sales clerk is assigned an in-degree value in each step of the simulation when the sales clerk searches for customers in need of assistance. The bigger their in-degree the more customers they can help.

RetSim has different inputs needed in order to run a simulation. The input data concerns the distributions of probabilities for scheduling the sales clerks, the items that can be purchased and different statistic measures for the customers. A CSV file which contains an identifier, description, price, quantity sold and total sales specify these inputs. For setting the parameters, including the name of the CSV-file, we use a parameter file that is loaded as the simulation starts or the can also be set manually in the GUI.

Figure 1 shows the different use cases of the agents. This model represent the different actions that an agent can take inside the system.

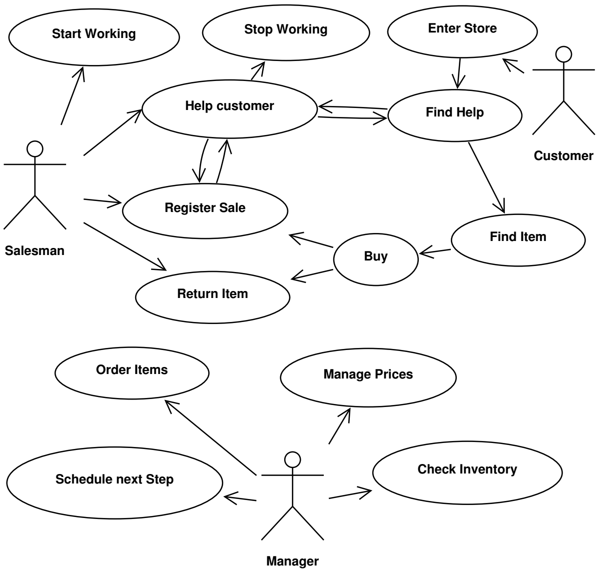

Fig. 1. RetSim Use Case Diagram

Manager scheduler: This agent is in charge of scheduling the next step of the simulation. There is only one manager per store. This agent creates the new customers that are going to arrive to the store according to a distribution function extracted from the original data set. The manager also allocate the sales clerks that are going to be active during the this step of the simulation.

Customer finder: Is performed by the sales clerk and it starts with the agent searching nearby for a customer that is not being helped by an other sales clerk. Once the contact is established a sale is likely to occur with a certain probability.

Sales clerk finder: Customers that are still in need for help can also look for nearby sales clerks. This again could lead to a sale.

Network generation: Every time a transaction is performed between a customer and a sales clerk, an edge is created in the network composed of the customers and the sales clerks in attendance. The weight of the edge represent the sales price. The network grows by the inclusion of new customers or sales clerks.

Item selection for purchasing: Items are classified into 5 different categories according to their quantity or units sold. From the original data we extracted the probabilities of each of the categories and quantities. A customer can also purchase more than one item.

Item return after purchasing: A customer can also decide to return a purchased item with a certain probability p.

Log of receipt transactions: Each time an item is purchased a receipt is created. A receipt contains the information about the customer, sales clerk, item(s), quantities, sales price, date and discount if any.

B. Validation and Verification

We start the evaluation of our model with the verification and validation of the simulator and the generated data [17]. Verification ensures that the simulation corresponds to the described model presented by the chosen scenarios. In our model, we have included several characteristics from a real store, and successfully generated a distribution of sales that involved the interaction of salesmen and customers.

The validation of the model answers the question: *Is the model a realistic model of the real problem we are addressing*? After the calibration of the model using the original data set, we can see that the descriptive statistics of both top simulations are close to the descriptive statistics of the real data. For the purpose of this presentation we performed visual, statistical tests and evaluated the network topology and parameters to verify that our simulation is sufficiently similar in behaviour to the original data to perform fraud detection testing.

Figure 2 shows an overlap of our sample store with different simulation runs by RetSim. Visually the distributions look similar. However there are several differences in the small shapes.

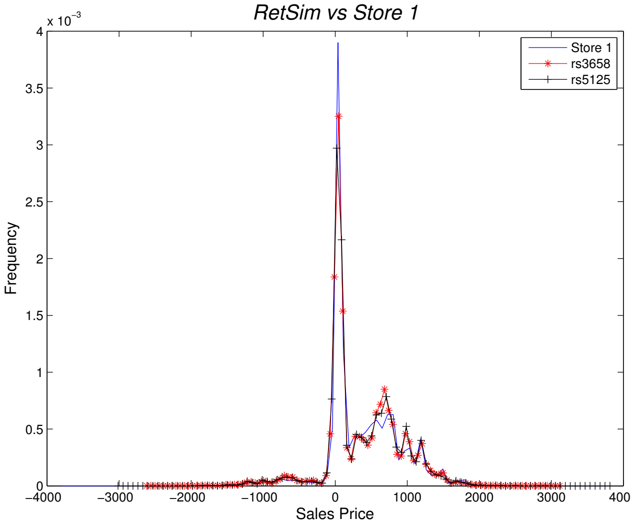

Fig. 2. Comparison of distribution of simulated vs real data

In figure 3 we can see a box plot comparison of store one with the RetSim runs. We can visually identify that the five statistical measures provided by the box plot are similar without being identical.

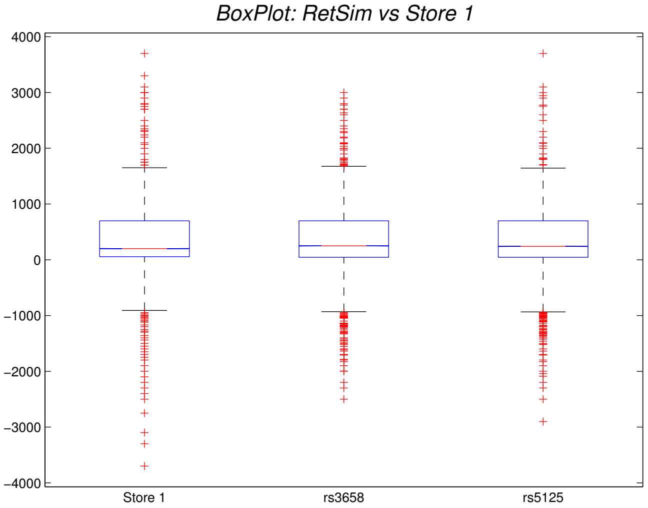

Fig. 3. Box plot of simulated vs real data

Since we are running a simulation, we argue that the differences are not significant for our purpose, which is to use this distribution to simulate the normal behaviour of a store, and later combine this with injected anomalies and known patterns of fraud.

C. Fraud Scenarios in a Retail Store

In this section we describe how three examples of retail fraud can be implemented in RetSim. These fraud scenarios are based on selected cases from the Grant Thornton report [19]. As can be seen in section V-A, the different scenarios can be implemented in almost the same way. Furthermore, a fraudulent sales clerk will probably use several different methods of fraud, which means that RetSim needs to be able to model combinations of all fraud scenarios implemented. Although the implementation of these scenarios are out of the scope of this paper, we include a description and explain how to implement them in RetSim.

1) Refunds: This scenario includes cases where the sales clerks creates fraudulent refund slips, keeping the cash refund for themselves. In terms of the object model used in RetSim, the refund scenario can be implemented by the following setting: Estimate the average number of refunds per sale and the corresponding standard deviation. Use these statistics for simulating refunds in the RetSim model. Fraudulent sales clerks will perform normal refunds, as well as fraudulent once. The volume of fraudulent refunds can be modelled using a sales clerk specific parameter. The "red flag" for detection will in this case be a high number of refunds for a sales clerk.

To model the first scenario we need information about the relevant parameters describing the normal behaviour: figure 4 shows the percentage of total value of refunds divided by the total sales for each salesman, for the simulation *rs5125*. The figure shows the values for both the normal behaviour, and two simulations with injected *return fraud*. The first fraud simulation (-+-) shows a conservative fraud behaviour agent where the agent will not attempt to commit fraud if the sales value is more than 800 units in the fictitious currency, and the frequency with which it commits this fraud is 5% of all sales. The total profit obtained by all fraudulent agents in a year is 161630 units in this scenario.

The second fraud simulation (-.-) shows an aggressive fraud agent behaviour where the threshold to commit fraud is 600 units and the frequency is 10% of sales. The total profit obtained by all agents is 400451 units per year.

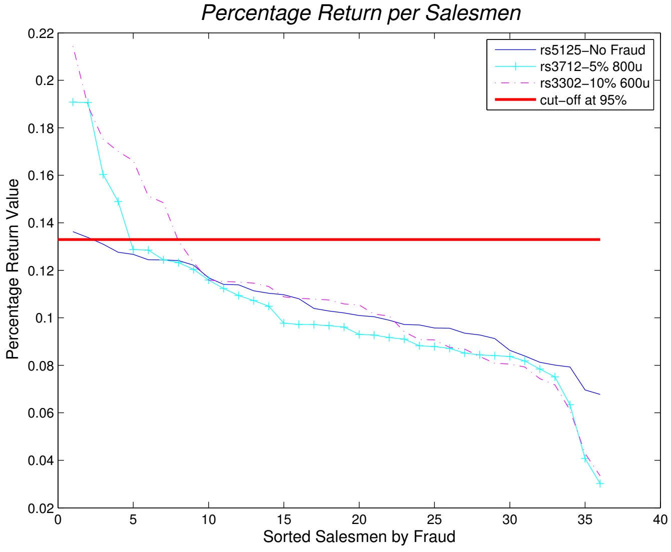

Fig. 4. Return Value Over Sales Total per Salesman

2) Coupon reductions/discounts: This scenario includes cases where the sales clerk registers a discount on the sale without telling the customer, i.e., the customer pays the full sales price, and the sales clerk pockets the difference. In terms of the object model used in RetSim the coupon reduction/discounts scenario can be implemented by the following setting: Estimate the average number of cancellations per sale and the corresponding standard deviation. Use these statistics for simulating discounts in the RetSim model. Sales clerks who perform fraud will make normal discounts, as well as fraudulent ones. The volume of fraudulent discounts can be modelled using a sales clerk specific parameter. The “red flag” for detection will in this case be a high number of discounts for a sales clerk with a low number of average sales.

Figure 5 shows the percentage of the total value of discounts over the total sales before discount for each salesman for the simulation *rs5125*. The figure shows the values for both normal behaviour together with two simulations with injected discount fraud. The first fraud simulation (-+-) shows a conservative fraud agent behaviour where the threshold to commit fraud is 800 units and the frequency is 5% of sales. The total profit per year, for by all agents is 18423 units.

The second fraud simulation (-.-) shows an aggressive agent with a fraud threshold of 600 units and the frequency 10% of the sales. The total profit obtained by all agents is 80600 units per year.

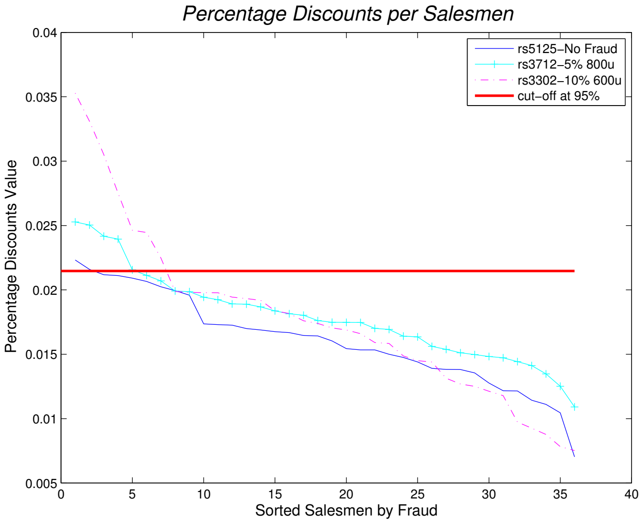

Fig. 5. Discount Value Over Sales Total before Discount per Salesman

D. Results

We extracted statistical information that comprises the sales from one store during one year. The *store one* sample contains 147037 records of transactions. The retailer runs a fidelity program that allows customers to register their purchases. This means that the majority of receipts belong to unidentified customers. However for all these records we can identify the item(s), sales price and the salesman.

Fraud analysis has traditionally been strongly associated with network analysis. This is because of the possibility of several actors participating in a specific fraud in order to confuse the investigators and dilute the evidence, hence describing a network of actors, companies, ownership etc. By doing this we aim to model the micro behaviour of the different agents that captures the observed macro behaviour and gives rise to a total picture of the store. We use the properties of the original social network generated from the customers and simulated a similar network with the aim of keeping the social network properties or the original such as topology, average in-degree and out-degree distribution of the salesmen and customers.

From the network analysis there is a lot of data we can use for our model. One of data point is that the 90.26% of the members have been helped by only one salesman, as described by the out-degree distribution.

We have no known instances of fraud in the real data (as certified by the data owner). So we will have to inject malicious behaviour, by programming agents that behave according to some known or hypothesised retail fraud case presented before: Refunds and Discounts.

In terms of the object model used in RetSim the refund scenario can be implemented by the following setting: Estimate the average number of refunds per sale and the corresponding standard deviation. Use these statistics for simulating refunds in the RetSim model. Fraudulent salesmen will perform normal refunds, as well as fraudulent once. The volume of fraudulent refunds can be modelled using a salesman specific parameter. The "red flag" for detection will in this case be a high number of refunds for a salesman.

Similar to refund scenario, RetSim generates malicious coupon reduction/discounts and the analysis can also be performed in similar way as with refund fraud.

VI. BankSim, a bank transactions simulator

Initial studies started on *BankSim* with the purpose of creating a MABS that can be used for studying fraud prevention pertaining to online financial services. The motivation for this is that authorities like the Federal Financial Institutions Examination Council (FFIEC) in the US and the European Central Bank (ECB) in Europe have stepped up their expected minimum security requirements for financial institutions[20][21], including requirements for risk management of online banking. Thus, access to proper risk management tools is becoming increasingly important, including tools for simulating and being prepared for emerging threats. However we had no access to this type of financial data until we participated in a contest presented by the BBVA bank in Spain. his contest had the aim of promoting the development of applications for the so called "big data challenge" using their aggregated financial information provided by a web service.

A. Data Analysis

The data exposed to the public in the Bank web service contained information on credit card payments during 6 months (November 2012 until April 2013) for the cities of Madrid and Barcelona.

The data was segregated by zip code, gender, and age, and was aggregated by week and month. The web service implemented by the bank provided rich statistical information useful to build an agent model that contains all the consumption patters specified in the data.

The payments were categorised in 14 different categories that allowed the differentiation between e.g. transactions made at a restaurant or in a car dealership. We could also identify consumptions patterns by gender and age, that allowed us to build different kind of agents and implement their consumption pattern according to their given initial characteristics.

A social network is also possible to implement due to the possibility to see the zip code origin of the card used for the payment. Therefore we could identify and build a social network of different agents making payments in different zones of origin.

B. Model

The preliminary design of *BankSim* was again based on the ODD model introduced by [18].

We aim to produce a simulation that resembles real bank transactions between customers and merchants. Our main purpose is to generate a synthetic data set of payment transactions that can be used for the development and testing of different fraud detection techniques. Our model so far only covers bank payments and withdrawals, but we aim to extend it to bank deposits as soon as we can get access to statistical information or real data to properly validate the outcome.

There are two agents in this simulation are: *Merchants* and *Customers*.

Merchant Is in charge of selling one of the categories of available products to the customers.

Customer The behaviour is determined by the goal of purchasing one or several items from the different categories. A customer searches for merchants in its surroundings and execute payments after obtaining the goods.

1) Process overview and scheduling: During a normal step of the simulation a customer can select a category to start a purchase. After selecting the desired category, it enters the simulation environment and sense any nearby merchants that matches the selected category. There are two different outcomes: Either a transaction takes place, with probability p, or no transaction takes place (with probability $1 - p$).

The web service provides detailed information about the time granularity of the transactions. This allows the simulator to set its time granularity between hours or days. So a normal week can either have seven steps or 24×7, if hour granularity is chosen. We do not make any explicit distinction between specific days of the week, information about each day of the week is provided by the web service of the bank.

2) Design Concepts: The *basic principle* of this model is the concept of a commercial transactions. We can observe an *emergent* social network from the relation between the customers and merchants of the same or different zip codes. Each of the customers have the *objective* of purchasing articles or services from the merchants. In our virtual environment the *interaction* between agents is always between merchants and customer. However we aim to extend the model later to allow customer/customer interaction (transfers).

The agents do not perform any specific learning activities. Their behaviour is given by probabilistic Markov models where the probabilities are extracted from the provided data set and specified per hour or day.

C. Evaluation and Results

Similar to the *RetSim* case, we start the evaluation of our model with the verification and validation of simulator and the generated data [17]. Verification ensures that the simulation corresponds to the described model presented by the chosen scenarios. In our model, we have included several characteristics from a real scenario where the interaction between merchants and customers is given by the commercial transaction. We successfully generated a data set of payments that involved the interaction of our agents under our virtual environment.

The validation of the model answers the question: *Is the model a realistic model of the real problem we are addressing*? We calibrate the model using the original data set values.

But since *BankSim* is currently in a development phase, the evaluation and results of this simulator are not yet available. Similar to what we previously did with *RetSim*, we aim to perform visual, statistical tests and evaluated the network topology and parameters to deduce whether our simulation is sufficiently similar to perform fraud detection testing.

VII. Discussion

We started with a rather trivial but meaningful simulation of a payment system (*PaySim*). The original goal of finding money laundering in financial transaction is an ambitious goal which lead us to the building of two more simulators *RetSim* and *BankSim*.

RetSim was our first attempt to simulate commercial transactions based on real data. The benefit of a deep data analysis allowed the simulator to accurately generate synthetic transactional logs of the store. Our evaluation showed that we obtained a data set that resembled the original data set. This without disclosing personal and private information of the customers. We succeed on using this simulator to seek answers about simple threshold detection and its effectiveness. In a real data set the cost of the fraud is most of the time unknown, and it is estimated by using a control mechanism such as inventory control and video surveillance of the store. This does not represent a problem for *RetSim* since we flag each transaction with the type of fraud committed.

RetSim has many improvements over *PaySim*. First, it uses the benefit of real data to calibrate and evaluate the model, second it uses the ODD methodology to describe and model the whole process and specify the agents. It finally uses its output to analyse a realistic fraud scenario and answer questions regarding fraud detection methods.

One piece was missing in the financial chain, and it was a bank simulator. We started to develop *BankSim*. *BankSim* is still in early development but we hope to follow the path of *RetSim* and prove its usefulness on developing and testing fraud detection methods.

All simulators share common log formats for compatibility with other software used to analyse the transactional logs. This is an important characteristic in this framework that will enable us in a future to make available standard data sets to the research community and the public in general.

Every time we build a simulator for financial transactions we aim to make it compatible with the previous simulators and also to avoid previous pitfalls in the design, model and implementation. *PaySim* for instance, required real data to calibrate the model. *RetSim* uses less detailed aggregated information as we are currently using on *BankSim*.

Modelling social financial behaviour of customers have its challenges. This paper present the way we addressed the problem of social simulation for financial transactions. One approach we considered was to implement social economical patterns of consumptions to build up an agent with preferences and choices. However, our goal here is to replace a data set that currently represent an detailed instance of a real world social situation. Using an statistical approach was a straight forward direction for simulation the "normal" behaviour of agents. However, the behavioural patterns known by fraudsters and criminals, allow the implementation a different model that makes the fraudster an agent that aims to maximise its profit and uses specifics patterns of action that aims to disguise the crime. This social behaviour was implemented in our simulators using known criminal behavioural patterns parameterised to fit different fraud scenarios.

We injected the most common known fraud behaviours, but we are aware that there are many other fraud behaviours that can have a significant economical impact on the criminal activities. We have only touched the surface of what is possible with the scenarios we have implemented.

VIII. Conclusions

This paper addressed the problem of a lack of public available data sets for fraud detection research. We experienced this difficulty and discovered that many other researchers in this field share this experience. The three simulators presented in this paper allow researchers to generate synthetic data sets that are useful for experiments in fraud detection.

In summary, we presented three case studies that implement a Multi-Agent Based Simulation model to address the problem of social simulation of financial transactions for fraud detection research. Our agent model with its programmed micro behaviour, produces a similar type of overall interaction network that we can observe in the original data, and furthermore, this interaction network give rise to the same macro behaviour for the whole store as for the real store as well. All three simulators use the same Multi-Agent Based Simulation toolkit called MASON[16] which is implemented in Java.

PaySim is our first attempt and a good example of the use of a synthetic data set representing a simulated scenario in the mobile money domain. We tested some machine learning algorithms to try to detect fraud using labelled data. While doing this we also avoided any possible issue related to privacy and identity protection of the customers of the service.

We also presented *RetSim*, and argued that it is ready to be used as a generator of synthetic data sets of commercial activity of a retail store. Data sets generated by *RetSim* can be used to implement fraud detection scenarios and malicious behaviour scenarios such as a salesmen returning stolen shoes or abusing discounts. We used the *RetSim* simulator to investigate these two fraud scenarios. Our simulator give us the benefit over real data that we can quantify and measure the amount of loses committed by our malicious agents.

We used the *RetSim* simulator to investigate two fraud scenarios to see if threshold based detection could keep the risk of fraud at a predetermined set level. While our results are preliminary, they seem to indicate that this is so. This is interesting in that it could act to explain why we have not observed more use of more advanced methods in industry even though research into more advanced techniques has been common for quite some time now. Another consequence could well be that given that simple threshold based detection is sufficient there is little economic room for other more advanced fraud detection methods that are more costly to implement.

We are currently in a preliminary phase of development with regards to *BankSim*. Our work with this simulator is just beginning with the hope to present interesting results in a future paper. We aim to rebuild our payment simulator based on real data. We have successfully achieved a realistic simulation for a retail store which we would like to extend to different kinds of retail stores. And finally we are negotiating with a Bank in Scandinavia to be able to extend the scope of *BankSim* and be able to access real data sets to model and develop deposits and enrich the *BankSim* simulator.

One of the biggest challenges for is to integrate all three simulators into one single Multi-Simulator that shares a common reference to the customers and can keep track of the transactions of a single agent across all simulators. Money Laundering exist somewhere in a complex chain that starts with *placement* of illegal funds into the legal financial systems, then a number of *layering* operations to hide the true origins and finally an *integration* stage that involves formal and legal economic activities. Our approach will focus on the integration of these different domain simulators as the key to research in the area of money laundering.

References

[1] E. A. Lopez-Rojas and S. Axelsson, “Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML),” in *The 17th Nordic Conference on Secure IT Systems*, Karlskrona, 2012, pp. 25–32.

[2] E. A. Lopez-Rojas, S. Axelsson, and D. Gorton, “RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection,” in *The 25th European Modeling and Simulation Symposium*, Athens, Greece, 2013.

[3] A. Schwaiger and B. Stahmer, “SimMarket: Multiagent-based customer simulation and decision support for category management,” *Multiagent System Technologies*, pp. 74–84, 2003. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-39869-1_7

[4] E. A. Lopez-Rojas and S. Axelsson, “Money Laundering Detection using Synthetic Data,” in *The 27th workshop of (SAIS)*. Örebro: Linköping University Electronic Press, 2012, pp. 33–40.

[5] D. Agrawal and C. Aggarwal, “On the design and quantification of privacy preserving data mining algorithms,” *PODS ’01 Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=375602

[6] S. Alam and A. Geller, “Networks in agent-based social simulation,” *Agent-based models of geographical systems*, pp. 77–79, 2012. [Online]. Available: http://www.springerlink.com/index/H328T536877UM556.pdf

[7] D. Magnusson, “The costs of implementing the anti-money laundering regulations in Sweden,” *Journal of Money Laundering Control*, vol. 12, no. 2, pp. 101–112, 2009. [Online]. Available: http://www.emeraldinsight.com/10.1108/13685200910951884

[8] C. Levin, E. J. Bean, and K. Martin-browne, “U.S. Vulnerabilities to Money Laundering , Drugs , and Terrorist Financing : HSBC Case History,” Tech. Rep., 2012.

[9] R. Bolton and D. Hand, “Statistical fraud detection: A review,” *Statistical Science*, vol. 17, no. 3, pp. 235–249, 2002. [Online]. Available: http://www.jstor.org/stable/3182781

[10] C. Phua, V. Lee, K. Smith, and R. Gayler, “A comprehensive survey of data mining-based fraud detection research,” *Arxiv preprint arXiv:1009.6119*, 2010. [Online]. Available: http://arxiv.org/abs/1009.6119

[11] N. Abe, B. Zadrozny, and J. Langford, “Outlier detection by active learning,” *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 06*, p. 504, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1150402.1150459

[12] A. Sudjianto, S. Nair, M. Yuan, A. Zhang, D. Kern, and F. Cela-Díaz, “Statistical Methods for Fighting Financial Crimes,” *Technometrics*, vol. 52, no. 1, pp. 5–19, Feb. 2010. [Online]. Available: http://pubs.amstat.org/doi/abs/10.1198/TECH.2010.07032

[13] Z. Zhang and J. Salerno, “Applying data mining in investigating money laundering crimes,” *discovery and data mining*, no. Mlc, p. 747, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=956750.956851 http://portal.acm.org/citation.cfm?id=956851

[14] D. Yue, X. Wu, and Y. Wang, “A Review of Data Mining-Based Financial Fraud Detection Research,” in *2007 Wireless Communications, Networking and Mobile Computing*. Ieee, Sep. 2007, pp. 5514–5517. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4341127

[15] E. A. Lopez-Rojas, S. Axelsson, and D. Gorton, “Using the RetSim Simulator for Fraud Detection Research,” *International Journal of Simulation and Process Modelling*, vol. 1, no. 1, pp. 1–16, 2014.

[16] S. Luke, “MASON: A Multiagent Simulation Environment,” *Simulation*, vol. 81, no. 7, pp. 517–527, Jul. 2005. [Online]. Available: http://sim.sagepub.com/cgi/doi/10.1177/0037549705058073

[17] P. Ormerod and B. Rosewell, “Validation and Verification of Agent-Based Models in the Social Sciences,” in *LNCS*, F. Squazzoni, Ed. Springer Berlin / Heidelberg, 2009, pp. 130–140. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01109-2_10

[18] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jø rgensen, W. M. Mooij, B. Müller, G. Peer, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. a. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis, “A standard protocol for describing individual-based and agent-based models,” *Ecological Modelling*, vol. 198, no. 1-2, pp. 115–126, Sep. 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0304380006002043

[19] A. Member and A. Council, “Reviving retail Strategies for growth in 2009 Executive summary,” Grant Thornton, Tech. Rep., 2009. [Online]. Available: http://www.grantthornton.com/staticfiles/GTCom/files/Industries/Consumer & industrial products/White papers/Reviving retail_Strategies for growth in 2009.pdf

[20] E. C. B. ECB, “Recommendations for the Security of Internet Payments,” Tech. Rep. January, 2013.

[21] F. Council, “Supplement to Authentication in an Internet Banking Environment. 2011,” *URl: http://www. ffiec. gov/pdf/Auth-ITS-Final*, pp. 206–222.