BANKSIM: A BANK PAYMENTS SIMULATOR FOR FRAUD DETECTION RESEARCH

Edgar Alonso Lopez-Rojas[(a)] and **Stefan Axelsson**[(b)]

[(a),(b)]Blekinge Institute of Technology , School of Computing
[(a)]edgar.lopez@bth.se, [(b)]stefan.axelsson@bth.se

ABSTRACT

BankSim is an agent-based simulator of bank payments based on a sample of aggregated transactional data provided by a bank in Spain. The main purpose of BankSim is the generation of synthetic data that can be used for fraud detection research. Statistical and a Social Network Analysis (SNA) of relations between merchants and customers were used to develop and calibrate the model. Our ultimate goal is for BankSim to be usable to model relevant scenarios that combine normal payments and injected known fraud signatures. The data sets generated by BankSim contain no personal information or disclosure of legal and private customer transactions. Therefore, it can be shared by academia, and others, to develop and reason about fraud detection methods. Synthetic data has the added benefit of being easier to acquire, faster and at less cost, for experimentation even for those that *have* access to their own data. We argue that BankSim generates data that usefully approximates the relevant aspects of the real data. We intend to make the simulation and its results available to the research community.

Keywords: Multi-Agent Based Simulation, Bank Payments, Fraud Detection, Credit Card Fraud, Synthetic Data.

1. INTRODUCTION

In this paper we present *BankSim*, a **Bank** payment **Sim**ulation, built on the concept of Multi Agent-Based Simulation (MABS). *BankSim* is based on a sample of aggregated transaction data provided by one bank in Spain with the aim of promoting the development of applications for Big Data. This data contains several thousand records of transactional data covering six months, from November 2012 until April 2013 restricted by zip code location to Madrid and Barcelona. That is, this data is recent enough to reflect current conditions of payments, but aggregated to not pose a risk from a specific customer privacy standpoint.

The defence against fraud is an important topic that has seen some study. In a bank the cost of fraud are of course ultimately transferred to the consumer, and finally impacts the overall economy. Our aim with *BankSim* is to learn the relevant parameters that governs the behaviour of a bank payment system to simulate *normal* behaviour and inject specific fraud scenarios that are interesting to study.

The main contribution, and focus, of this paper is a method of generating anonymous synthetic data from aggregated transactional data of a bank payment system, that can then be used as part of the necessary data for the development and testing of fraud detection techniques. Even so, the data set generated could also be the basis for research in other fields, such as consumer behaviour, general economic study including social development and forecasting.

Later we plan to address the actual fraud and develop techniques to develop malicious agents to inject fraudulent and anomalous behaviour, and then develop and test different strategies for detecting these instances of fraud. Even though we do not address these issues in this paper, we describe some typical scenarios of credit card fraud that affects bank payments. As this is our ultimate goal, fraud heavily influenced the design of *BankSim*.

The main goal of developing this simulation is that it enables us to share realistic fraud data, without exposing potentially business or personally sensitive information about the actual source. As data relevant for computer security research often is sensitive, for a multitude of reasons, i.e. financial, privacy related, legal, contractual and other, research has historically been hampered by a lack of publicly available relevant data sets. Our aim with this work is to address that situation. However, simulation also have other benefits, it can be much faster and less expensive than trying different scenarios of fraud, detection algorithms, and personnel and security policy approaches in an actual store. The latter also risks incurring e.g. unhappiness amongst the staff, due to trying e.g. an ill advised policy, which leads to even greater expense and unwanted problems.

Outline: The rest of this paper is organised as follows: Section 2. introduce the topic of fraud detection for bank payments and present previous and related work. Sections 3. describes the problem, which is the generation of synthetic data of a bank payment system. Section 4. shows a data analysis of the current data. Section 6. presents an implementation of a MABS for our domain and shows the description of some credit card fraud scenarios. We present our results and verification of the simulation in section 7. and finish with a discussion and conclusions, including future work in section 8.

2. BACKGROUND AND RELATED WORK

Simulations in the domain of financial markets have traditionally been focused on finding answers to prediction problems such as economic growth, market growth, consumption patterns and so on.

There is currently a lack of research in the area of simulation of bank systems, more specifically for fraud detection.

We have previously analysed the implications of using machine learning techniques for fraud detection using a synthetic data set (Lopez-Rojas and Axelsson, 2012a). We then built a simple simulation of a financial transaction system based on these assumptions, in order to overcome our limitations and lack of real data (Lopez-Rojas and Axelsson, 2012b). However, this work was not based on any underlying data, but rather on assumptions of what such data could contain. We learn the principles of simulation and modelling and successfully applied them to *RetSim* (Lopez-Rojas et al., 2013). *RetSim* is the older brother of *BankSim* and uses data from a retail store to produce a realistic simulation that generates synthetic data.

Here we continued our work and built a realistic simulation based on a real aggregated payment data set that can be used to test diverse fraud detection techniques. All our simulators are part of a financial system chain. They have in common that all are built with the aim of modelling financial activity with the purpose of generating synthetic data sets for fraud detection research. We are continuing to build the needed components to integrate them into a complex financial chain and produce a virtual financial world that covers many domains. This is specifically useful to implement more complex fraud scenarios such as money laundering.

Data mining based methods have previously been used to detect fraud (Phua et al., 2010). This lead to the result that machine learning algorithms can identify novel methods of fraud by detecting those transactions that are different (anomalous) in comparison to benign transactions. This problem in machine learning is known as novelty detection. Supervised learning algorithms have previously been used on a synthetic data set to prove the performance of outliers detection (Abe et al., 2006), however this has not been performed on transactional data. There are tools such as IDSG (IDAS Data and Scenario Generator (Lin et al., 2006)) which was developed with the purpose of generating synthetic data based on the relationship between attributes and their statistical distributions. IDSG was created to support data mining systems during their test phase and it has been used to test fraud detection systems.

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 (Bolton and Hand, 2002). 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 legislation 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 control, by e.g. making many smaller transactions that fall just below the threshold. Hence, these and other similar methods have proven insufficient (Magnusson, 2009).

Nowadays with the popularity of social networks, such as *Facebook*, the topic of Social Network Analysis (SNA) has been given special interest in the research community (Alam and Geller, 2012). Social Network Analysis is a topic that is currently being combined with Social Simulation. Both topics support each other for the benefit of representing the interactions and behaviour of agents in the specific context of social networks.

Our approach aims to fill the gap between existing methods and provide researchers with a tool that generates reliable data to experiment with different fraud detection techniques and compare them with other approaches.

3. PROBLEM

Fraud and fraud detection is an important problem that has a number of applications in diverse domains. However, in order to investigate, develop, test and improve fraud detection techniques one needs detailed information about the domain and its specific problems.

There is a lack of data sets available for research in fields such as money laundering, financial fraud and illegal payments. Disclosure of personal or private information is only one of the many concerns that those that own relevant data have. This leads to in-house solutions that are not shared with the research community and hence there can be no mutual benefit from free exchange of ideas between the many worlds of the data owners and the research community.

After describing the problem we formulated the main research question that we address on this paper:

RQ *How could we model and simulate a bank payment system and generate a realistic and reliable synthetic data set for the purpose of fraud detection?*

4. Data Analysis

To better understand the problem we began by performing data analysis of the sample data provided by a bank in Spain. We are interested in finding the necessary and sufficient attributes to enable us to simulate a realistic scenario in which we could reason about and detect interesting cases of fraud.

The bank in Spain, which we will name *Bank Inc.*, provided a web service interface to query aggregated information about bank payments. The web service limited the query to transactions that occurred between November 2012 and April 2013 restricted to transactions that took place in Madrid and Barcelona. The service provided by *Bank Inc.* groups the data by month, week, day of the week and hour. The interface allows three types of queries: *consumption habits*, *customer classification* and *origin and source of transactions*. The basic information provided by the queries are mainly statistical information about payments such as: number, average, minimum and maximum values. It also provides information regarding zip code location of origin/source, merchant category and customer gender and age. There are 16 merchant categories that differentiate between payments made for example in a restaurant or payments performed while buying cars or other goods.

It was not possible to query information where less than 2 customers made payments. This means there is some missing information about the data provided, but fortunately we know exactly which data is missing, because the response from the web service is different depending on whether the data is missing or restricted.

We initially started by selecting a few zip codes that contain enough information to avoid missing fields. We selected two of the biggest zip codes by number of transactions and amount. We extracted statistical information, presented in table 1. Age Categories are given in table 2 and gender categories are given in table 3. All prices given are in euro.

Due to a lack of space we will focus our presentation of the analysis on one of the biggest zip code by payment volume that we will call Zip Code One (ZC1).

ZC1 is relatively richer in data than the smaller zip codes, it contains 731658 payments during a six month period. This is specially interesting, since we are more likely to find actual cases of fraud.

Table 1: Statistical Analysis Data

zipcode	gender	age	payments	avgAmountMonth	avgNumCardsMonth
ZC1	E	U	823	31.97	90.67
ZC1	F	6	12375	44.83	1002.33
ZC1	F	5	39461	35.81	3297.50
ZC1	F	4	72336	33.79	6514.83
ZC1	F	3	94536	31.87	9337.50
ZC1	F	2	128117	29.37	13457.33
ZC1	F	1	41299	30.13	5002.00
ZC1	F	0	1809	28.81	257.00
ZC1	M	6	18030	36.93	1676.33
ZC1	M	5	38097	33.29	3534.83
ZC1	M	4	62314	32.39	5871.67
ZC1	M	3	82222	30.38	8451.83
ZC1	M	2	106404	27.42	10969.33
ZC1	M	1	32031	27.70	3739.67
ZC1	M	0	1516	28.37	213.83
ZC1	U	6	193	17.56	13.83
ZC1	U	4	14	23.95	3.00
ZC1	U	3	54	12.03	4.00
ZC1	U	2	27	23.60	3.40
ZC2	E	U	23349	5.78	482.83
ZC2	F	6	13160	61.97	1373.00
ZC2	F	5	27250	55.58	2766.50
ZC2	F	4	50074	48.90	4508.00
ZC2	F	3	63122	43.59	5746.00
ZC2	F	2	91343	37.89	8026.67
ZC2	F	1	37303	30.17	3152.50
ZC2	F	0	1842	26.89	172.83
ZC2	M	6	11176	80.01	1203.00
ZC2	M	5	18854	74.22	1951.83
ZC2	M	4	29474	67.89	2990.83
ZC2	M	3	45850	53.18	4612.17
ZC2	M	2	63568	41.72	6048.00
ZC2	M	1	21538	32.88	2054.50
ZC2	M	0	977	28.16	92.83
ZC2	U	6	67	74.08	6.33
ZC2	U	5	8	103.15	3.00
ZC2	U	3	10	24.48	4.00

Table 2: Age Categories

idAge	Rank
0	<=18
1	19-25
2	26-35
3	36-45
4	46-55
5	56-65
6	>65
U	Unknown

Table 3: Gender Categories

idGender	Description
E	ENTERPRISE
F	FEMALE
M	MALE
U	UNKNOWN

Table 4: Categories ZC1

category	percentage	avg	std
Auto	0.0049	224.35916667	267.52611111
Bars and restaurants	0.0244	31.03238095	39.19238095
Books and press	0.0014	33.34714286	45.01428571
Fashion	0.0076	49.73190476	59.08452381
Food	0.0726	32.49333333	30.87285714
Health	0.0179	59.39119048	113.98619048
Home	0.0021	75.48317073	121.77292683
Accomodation	0.0016	97.41071429	86.85047619
Hypermarkets	0.0178	33.06547619	35.78166667
Leisure	0.0001	74.86357143	22.01107143
Other services	0.009	52.9897619	76.65309524
Sports and toys	0.0043	74.8047619	75.45452381
Technology	0.0031	67.28285714	108.68452381
Transport	0.8176	24.56047619	20.76928571
Travel	0.0004	577.46285714	518.41885714
Wellness and beauty	0.0155	44.20809524	55.29142857

5. Fraud Scenarios in a Bank Payment System

In this section we describe how three example of fraud that can be implemented in BankSim. These fraud scenarios are based on selected cases from the Grant Thornton report Member and Council (2009). As can be seen in section 6., the different scenarios can be implemented in almost the same way. Furthermore, a fraudster will probably use several different methods of fraud, which means that BankSim 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 BankSim.

We will focus on card related frauds. This kind of fraud usually begins when the the important data on the card is compromised: Account name, credit card number, expiration date and verification code. This data can be acquired by a fraudster either by theft of the physical card or by gaining knowledge of the important data associated with the account.

5.1. Theft

This scenario includes cases where the customer loses physical possession of her card and a fraudster impersonate the customer purchasing goods or service with the stolen card. In terms of the object model used in BankSim the Theft scenario can be implemented by the following setting: Include in the fraudster the behaviour of sensing customer proximity, then execute the theft and later purchase goods from another merchant with the information from the customer. The volume of fraudulent activity can be modelled changing the specific parameter of number of theft, zip code and frequency. A ``red flag" for detection in this case could be a high number of unusual transactions with high value in a short period.

5.2. Cloned Card/Skimming

This scenario includes cases where the fraudster creates a clone of the card, letting the user keep the original card but without knowledge of the loss of security. In terms of the object model used in BankSim, the cloned card scenario can be implemented by the following setting: Include in the fraudster the behaviour of sensing customers proximity, then execute the acquisition or cloning of a card and later purchase goods from another merchant with the information from the customer. An alternative way to implement this scenario could be when a merchant is compromised in different ways (e.g. by hacking) and allow a fraudster to steal information from all customers that have been served there on a massive scale. The volume of fraudulent activity can be modelled changing the specific parameter of number of theft and merchant affected, zip code and frequency of use for purchasing. A ``red flag" for detection in this case could be similar as previous case, a high number of unusual transactions with high value in a short period. Other methods such as simultaneous payments in different physical locations, or using the card far from previously known locations, could also be flagged.

5.3. Internet purchases

This scenario includes cases where the fraudster uses a method called *Carding* to purchase immaterial goods, e.g. music files, redeemable coupons, tickets etc. on the Internet using websites that check the validity of the card instantly. This is to ascertain whether the card data is still valid without having to run the risk of getting caught when using the card while physically present. Similar to cloned cards the customer keeps the original card but without knowledge of the situation. In terms of the object model used in BankSim the cloned card scenario can be implemented by the following setting: Include in the fraudster the behaviour of sense customers proximity, then execute the acquisition of the important information of a card and later on proceed with the method of Carding, to check for validity. A ``red flag" for detection in this case could be to have a black list of Carding websites and proceed to cross this information with current user activity to detect any unusual purchases after the Carding was executed.

6. MODEL AND METHOD

The design of BankSim was based on the ODD model introduced by Grimm et al. (2006). ODD contains 3 main parts: *Overview*, *Design Concepts* and *Details*.

6.1. Overview

6.1.1. Purpose

We aim to produce a simulation that resembles a bank payment system. Our main purpose is to generate a synthetic data set of commercial transactions that can be used for the development and testing of different fraud detection techniques.

If we want to use the real original data for the development of fraud detection methods, it often happens that is difficulty to find diverse and enough cases of fraud. However this is not the case of a simulated environment, where fraud can be injected following known patterns of fraud and flagged for easy recognition and evaluation of the performance of the detectors.

6.1.2. Entities, state variables and scales

There are three agents in this simulation: *Merchant*, *Customer* and *Fraudster*.

Merchant This agent serves the customer with one category of merchandise specified by the original data. It offers products or services according to the statistics obtained from the specific zip code and time (week, day of the week and/or hour). They are waiting for customers to request products and register the payments.

Customer This agent's main objective is to satisfy a need for one of the 16 categories and purchase goods or services from merchants. They posses a payment method which in this case we will be generalised as a credit card.

Fraudster The behaviour is determined by the goal of defrauding the customers and/or merchants. The specific behaviour can be extended to fulfil different patterns and can mutate depending on the specific fraud behaviour we are interested in studying. Some of the known fraud behaviour is presented in section 5..

6.1.3. Process overview and scheduling

During a normal step of the simulation, a customer that enters the simulation can decide to purchase an item or service from one of the offered categories. Once the category has been selected, it senses nearby merchants that offer that category and listen to the offers from the merchant. If accepted (with a certain probability of rejection) the transaction takes place and the merchant registers the payment.

The time granularity of the simulation is that each step represents a day of commercial activity, but the original data is so rich that this can be modified to the specific hour of the day. So a normal week has 7 steps and a month will consist of around 30 steps. Notice that in the future we can chose to make the distinction between specific days of the week explicit, since the information from Bank Inc. is good enough to obtain statistics from it. But for now, we are not taking specific day of the week into account to feed the consumption pattern and we treat all days the same.

6.2. 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 merchants. Each of the customers have the *objective* of purchasing articles from the merchants. The merchants *objective* is to serve the customers and commit the payment that result into the generation of a synthetic data set. In our virtual environment the *interaction* between agents is always between merchant and customer. Purchasing articles from another customer or selling articles to another merchant is not included in our model.

Customers can scout for the merchants in any radial direction from their current position in the virtual world and search for a merchant that matches its category selection. If no merchant is found then the transaction can not take place, and the step for this customer ends.

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.

6.3. Details

6.3.1. Initialization

The simulation starts with a number of merchants that match the categories of what a specific zip code offers, an initial number of customers and fraudsters.

6.3.2. Input Data

BankSim has different inputs needed in order to run a simulation. The input data concerns the distributions of probabilities for each of the merchants, and the consumer pattern behaviour of the customers specified by gender and age. The items that can be purchased are all grouped into a category using the statistic measures for the payments.

For setting the parameters, we use a parameter file that is loaded as the simulation starts, it contains zip codes that we want to simulate and the malicious parameters. Some parameters can also be set manually in the GUI. The zip codes are queried against the API of the bank and we retrieve information corresponding to the customers: quantity, age and gender distribution. We also query the merchants and obtain sales distributions for each of the merchant categories.

6.3.3. Submodels

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

Find Merchant The first step in a simulation for a customer is to find a merchant, each agent decides which category of service they will want to find, so the next step is to sense the environment and find a merchant that provides the category selected. Next search by the customer starts here, i.e. the customers move from merchant to merchant.

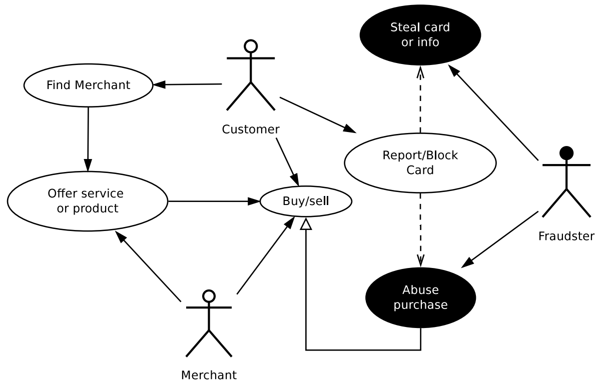

Figure 1: BankSim Use Case Diagram including misuse cases

Offer service or product Is performed by the merchant and once a merchant is approached by a customer, it offers a product or service according to the demand specified on the parameters for each category.

Buy/Sell Once a customer finds a merchant and after a merchant offers a product, a transaction takes place and it stores the required information for the generation of the synthetic data of transactions.

Steal card or info Fraudsters move around the environment of the simulation and find customers to steal the physical card or just the important information of the customer credit card. This information is stored for later use. In this misuse case we aim to emulate the behaviour of a criminal performing a cloning of a card or just stealing the card.

Abuse purchasing This misuse case is performed by Fraudsters, they make purchases of goods or services on physical merchants or internet merchants that hides their physical presence.

Report/Block Card This use case is performed by Customers, when they realise that abusive behaviour is committed on their accounts, they report the case to the bank and block the card for further abuse.

Log of transactions Each time an item or service is purchased from a merchant a transaction is created. A log contains the information about the customer, merchant, amount, location, date and fraud if any.

7. RESULTS

BankSim uses the Multi-Agent Based Simulation toolkit MASON which is implemented in Java (Luke, 2005). MASON offers several tools that aid the development of a MABS. We justified our choice mainly for the benefits of supporting multi-platform, parallelisation, good execution speed in comparison with other agent frameworks; which is specially important for computationally intensive simulations such as BankSim (Railsback et al., 2006). BankSim can be run with a GUI, that helps the user see the states and balance of the customers (purple dots) and easier identify the merchants (green circles) and fraudsters (red dots), as can be seen in the example in figure 2.

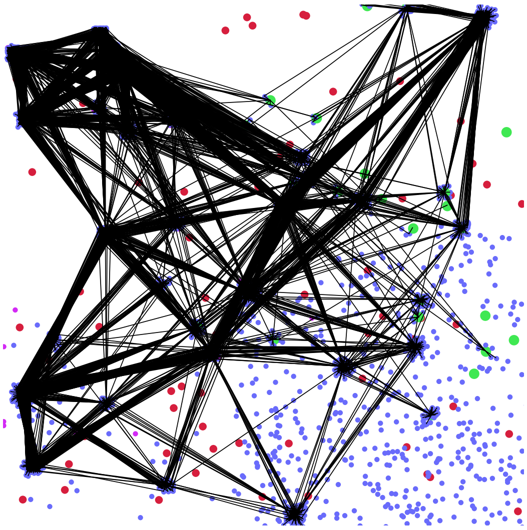

Figure 2: Screenshot of BankSim during a step

The output of BankSim is a CSV file that contains the fields: *Step*, *CustomerId*, *Age*, *Gender*, *zipCodeOrigin*, *merchant*, *zipMerchant*, *category*, *amount* and a special field to flag fraudsters called *fraud*.

7.1. Simulated scenarios

We aimed to perform a simulation that would produce a comparable data set to our sample data set which contained payments for over 6 months to match the original data. The simulation was loaded with information from ZC1 (see table 1), which was selected due to the highest amount of payments.

We ran BankSim for 180 steps (approx. six months), several times and calibrated the parameters in order to obtain a distribution that get close enough to be reliable for testing. We collected several log files and selected the most accurate. We injected thieves that aim to steal an average of three cards per step and perform about two fraudulent transactions per day. We produced 594643 records in total. Where 587443 are normal payments and 7200 fraudulent transactions. Since this is a randomised simulation the values are of course not identical to original data.

The result of the simulation for normal transactions is summarised in tables 5, 6 and 7. Remember that the codes for age categories are given in table 2 and gender codes are given in table 3. All prices given are in euro.

Table 5: Simulated ZC1

gender	age	payments	avgAmount
E	U	1171	34.02
F	6	13795	32.13
F	5	33574	31.47
F	4	57835	31.74
F	3	77333	31.97
F	2	103112	32.14
F	1	32340	32.09
F	0	1818	34.75
M	6	12718	31.57
M	5	28382	31.35
M	4	49780	31.99
M	3	67870	31.83
M	2	81690	31.48
M	1	24924	31.84
M	0	586	33.36
U	3	173	32.28
U	2	164	28.83
U	1	178	33.23

Table 6: Categories Simulated ZC1

category	payments	perc	avgAmount	std
Accomodation	1196	0.002	106.55	69.34
Bars and restaurants	6253	0.0105	41.15	29.55
Books and press	885	0.0015	44.55	33.14
Fashion	6338	0.0107	62.35	44.36
Food	26254	0.0442	37.07	25.00
Health	14437	0.0243	103.74	76.87
Home	1684	0.0028	113.34	83.23
Hypermarkets	5818	0.0098	40.04	27.96
Leisure	25	0	73.23	20.91
Other services	684	0.0012	75.69	54.59
Sports and toys	2020	0.0034	88.50	63.13
Technology	2212	0.0037	99.92	73.49
Transport	505119	0.8494	26.96	17.53
Travel	150	0.0003	669.03	494.90
Wellness and beauty	14368	0.0242	57.32	41.48

Table 7: Fraud Simulated ZC1

Fraud	payments	per	total	per	avg	std
0	587443	98.78	18708432.56	83.03	31.84	31.47
1	7200	1.21	3822671.17	16.96	530.92	835.52

7.2. Evaluation of the model

We begin the evaluation with the verification and validation of the generated simulation data (Ormerod and Rosewell, 2009). The verification ensures that the simulation correspond to the described model presented by the chosen scenarios. We described BankSim in section 6. In our model, we have included several characteristics from a real payment system, and successfully generated a distribution of payments that involved the interaction of merchants and customers.

The validation of the model answer the question: *Is the model a realistic model of the real problem we are addressing?* After several runs of the simulation to calibrate it, we are able to answer that question affirmatively. We present a table summarising the generated data in tables 5, 6 and 7.

Table 5 can be compared with table 1, both tables compare the distribution of payments by gender and age. Similar values are found in both tables because we created the agents based on gender and age distribution of the zip code. However, we did not programme the consumption behaviour of agents based on gender and age. This is because we did not have the statistic standard deviation for the consumption patterns per age and gender, we only have the average. This affects the results, despite that in the overall results we find similar data. But we think the missing information from the real system can be found with further calibration that is at the moment beyond the scope of our work. Figure 3 shows a distribution of gender and age from our simulated data.

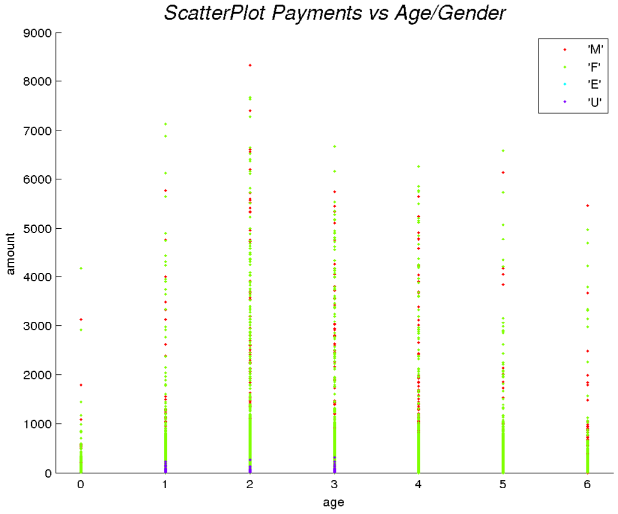

Figure 3: ScatterPlot Payments vs Age/Gender

Table 6 is comparable to table 4. We succeed in generating a distribution of categories that resembles the real data. We matched the percentage of categories and simulated similar average and standard deviation to the ones present in the original data. One thing to notice is that the category *auto* did not get any transaction during the simulation, this could be due to the location of the merchant in the environment being random and was perhaps far enough to be hidden from customers that wanted to purchase from this category. A box plot of the simulated categories is shown in figure 4. Since the values of travel are bigger than other categories, we decided to draw the box plot omitting this category in figure 5 to improve the visualization of the simulated data.

The simulated fraud behaviour is presented in table 7. The total amount stolen was around 3.8 million Euros which corresponds to a rather high crime rate of nearly 17% of the total amount of payments. We programmed an aggressive behaviour where few transactions (only 7200 and 1.2% of total)) could defraud 17% of the payments with an average of 530 Euros per fraud. For the purpose of fraud detection there is a benefit from the occurrence of enough cases of fraud that can help the investigators to gather the evidence needed to prosecute the criminals. In our case we benefit from the abundance of fraud cases because many detection methods need enough data to train better a classifier that can detect the fraud behaviour.

So in summary, our agent model with its programmed micro behaviour, produces the same type of overall interaction that we can observe in the original data, and furthermore, this interaction give rise to the same macro behaviour for the whole zip code as for a real situation as well.

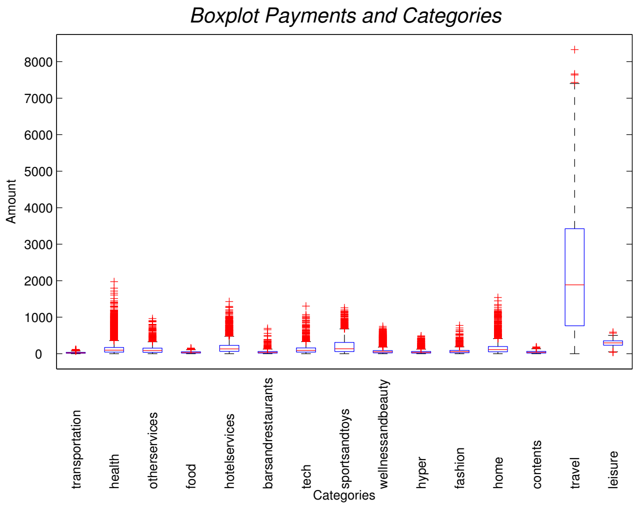

Figure 4: BoxPlot of a BankSim simulation

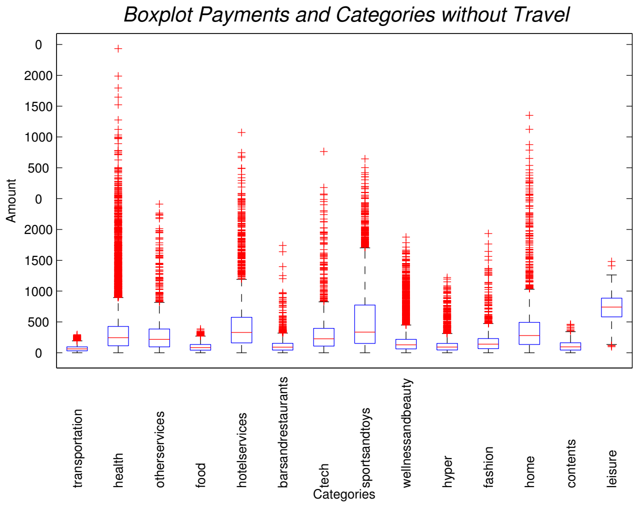

Figure 5: BoxPlot of a BankSim simulation without category Travel

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 payments, and simultaneously combine this with injected anomalies and known patterns of fraud.

8. CONCLUSIONS

BankSim is a simulation of bank payments with the objective to generate a synthetic transactional data set that can be used for research into fraud detection. The data sets generated with BankSim can aid academia, financial organisations and governmental agencies to test their fraud detection methods or to compare the performance of different methods under similar conditions using a common public available and standard synthetic data set for the test.

In section 3. we formulated our research question: *How could we model and simulate a bank payment system and generate a realistic and reliable synthetic data set for the purpose of fraud detection?*

In section 6. we presented the model for BankSim, which is based on the ODD methodology. In order to better support our claim and answer our research question we analysed the type of data needed to generate and output as a CVS file (see section 7.) and we evaluated and verified our model in section 7.2.

It is important to know how much information from the real data set is contained in the generated synthetic data. First we do not have access to any specific record of who is purchasing anything and neither the merchant involved in the transaction. We based our simulation purely on the aggregated statistical measures present in the original data that give us an approximate description of how the individual agents behave. This means that Bank Inc. can be sure that the privacy from the customers is preserved when using BankSim.

We argue that BankSim is ready to be used as a generator of synthetic data sets of financial activity of a payments. Data sets generated by BankSim can be used to implement fraud detection scenarios and malicious behaviour scenarios such as a stolen or cloned credit cards or unusual simultaneous activity of purchase in different physical locations. We will make a stable release of BankSim available to the research community together with standard data sets developed for this article and further research.

For the future we plan several improvements of and additions to the current model. BankSim can be calibrated to improve the results presented in section 7. and increase the granularity and the coverage of zip codes that enrich the synthetic data set and make it even more valuable as a realistic data set for fraud detection.

In order to generate records with malicious behaviour we plan to extend BankSim to also generate malicious activity that can come from the merchants, customers, different fraudsters or combinations of these.

Among the additions we consider are: increase the step granularity and add to the simulation more zip codes simultaneously. We intend to make BankSim a complete bank system by adding other bank transactions such as deposit, withdraws and transfers besides the current payments. Unfortunately for this addition there is a lack of real data that we can use for this purpose, but hopefully in the future we will find financial institutions interested in our project that are willing to share this data.

REFERENCES

Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 06*, page 504, 2006. doi: 10.1145/1150402.1150459.

SJ Alam and Armando Geller. Networks in agent-based social simulation. *Agent-based models of geographical systems*, pages 77--79, 2012.

R.J. Bolton and D.J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235--249, 2002.

Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K. Heinz, Geir Huse, Andreas Huth, Jane U. Jepsen, Christian Jø rgensen, Wolf M. Mooij, Birgit Müller, Guy Pe'er, Cyril Piou, Steven F. Railsback, Andrew M. Robbins, Martha M. Robbins, Eva Rossmanith, Nadja Rüger, Espen Strand, Sami Souissi, Richard a. Stillman, Rune Vabø, Ute Visser, and Donald L. DeAngelis. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198(1-2):115--126, September 2006. ISSN 03043800. doi: 10.1016/j.ecolmodel.2006.04.023.

P.J. Lin, B. Samadi, and Alan Cipolone. Development of a synthetic data set generator for building and testing information discovery systems. In *ITNG 2006.*, pages 707--712. IEEE, 2006. ISBN 0769524974.

Edgar Alonso Lopez-Rojas and Stefan Axelsson. Money Laundering Detection using Synthetic Data. *The 27th workshop of Swedish Artificial Intelligence Society (SAIS)*, pages 33--40, 2012a.

Edgar Alonso Lopez-Rojas and Stefan Axelsson. Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML). *The 17th Nordic Conference on Secure IT Systems*, pages 25--32, 2012b.

Edgar Alonso Lopez-Rojas, Stefan Axelsson, and Dan Gorton. RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection. *The 25th European Modeling and Simulation Symposium*, 2013.

S. Luke. MASON: A Multiagent Simulation Environment. *Simulation*, 81(7):517--527, July 2005. ISSN 0037-5497. doi: 10.1177/0037549705058073.

Dan Magnusson. The costs of implementing the anti-money laundering regulations in Sweden. *Journal of Money Laundering Control*, 12(2):101--112, 2009. ISSN 1368-5201. doi: 10.1108/13685200910951884.

Associate Member and Advisory Council. Reviving retail Strategies for growth in 2009 Executive summary, 2009. URL `http://www.grantthornton.com/staticfiles/GTCom/files/Industries/Consumer&industrialproducts/Whitepapers/Revivingretail_Strategiesforgrowthin2009.pdf`.

Paul Ormerod and Bridget Rosewell. Validation and Verification of Agent-Based Models in the Social Sciences. In Flaminio Squazzoni, editor, *LNCS*, pages 130--140. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-01108-5.

Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *Arxiv preprint arXiv:1009.6119*, 2010.

S. F. Railsback, S. L. Lytinen, and S. K. Jackson. Agent-based Simulation Platforms: Review and Development Recommendations. *Simulation*, 82(9):609--623, September 2006. ISSN 0037-5497. doi: 10.1177/0037549706073695.

AUTHORS BIOGRAPHY

MSc. Edgar A. Lopez-Rojas

Edgar Lopez is a PhD student in Computer Science and his research area is Multi-Agent Based Simulation, Machine Learning techniques with applied Visualization for fraud detection and Anti Money Laundering (AML) in the domains of retail stores, payment systems and financial transactions. He obtained a Bachelors degree in Computer Science from EAFIT University in Colombia (2004). After that he worked for 5 more years at EAFIT University as a System Analysis and Developer and partially as a lecturer. He obtained a Masters degree in Computer Science from Linköping University in Sweden in 2011 and a licentiate degree in computer science (a degree halfway between a Master's degree and a PhD) in 2014.

Dr. Stefan Axelsson

Stefan Axelsson is a senior lecturer at Blekinge Institute of Technology. He received his M.Sc in computer science and engineering in 1993, and his Ph.D. in computer science in 2005, both from Chalmers University of Technology, in Gothenburg, Sweden. His research interests revolve around computer security, especially the detection of anomalous behaviour in computer networks, financial transactions and ship/cargo movements to name a few. He is also interested in how to combine the application of machine learning and information visualization to better aid the operator in understanding how the system classifies a certain behaviour as anomalous. Stefan has ten years of industry experience, most of it working with systems security issues at Ericsson.