APPLYING SIMULATION TO THE PROBLEM OF DETECTING FINANCIAL FRAUD

Edgar Alonso Lopez-Rojas

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2016:06

Department of Computer Science and Engineering

Applying Simulation to the Problem of Detecting Financial Fraud

Edgar Alonso Lopez-Rojas

Blekinge Institute of Technology Doctoral Dissertation Series
No 2016:06

Applying Simulation to the Problem of Detecting Financial Fraud

Edgar Alonso Lopez-Rojas

Doctoral Dissertation in
Computer Science

Department of Computer Science and Engineering
Blekinge Institute of Technology
SWEDEN

Department of Computer Science and Engineering
Publisher: Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
Printed by Lenanders Grafiska, Kalmar, 2016
ISBN: 978-91-7295-329-1
ISSN 1653-2090
urn:nbn:se:bth-12932

"Let's take flight simulation as an example. If you're trying to train a pilot, you can simulate almost the whole course. You don't have to get in an airplane until late in the process."

Roy Romer

Abstract

This thesis introduces a financial simulation model covering two related financial domains: Mobile Payments and Retail Stores systems.

The problem we address in these domains is different types of fraud. We limit ourselves to isolated cases of relatively straightforward fraud. However, in this thesis the ultimate aim is to introduce our approach towards the use of computer simulation for fraud detection and its applications in financial domains. Fraud is an important problem that impact the whole economy. Currently, there is a lack of public research into the detection of fraud. One important reason is the lack of transaction data which is often sensitive. To address this problem we present a mobile money Payment Simulator (*PaySim*) and Retail Store Simulator (*RetSim*), which allow us to generate synthetic transactional data that contains both: normal customer behaviour and fraudulent behaviour.

These simulations are Multi Agent-Based Simulations (MABS) and were calibrated using real data from financial transactions. We developed agents that represent the clients and merchants in *PaySim* and customers and salesmen in *RetSim*. The normal behaviour was based on behaviour observed in data from the field, and is codified in the agents as rules of transactions and interaction between clients and merchants, or customers and salesmen. Some of these agents were intentionally designed to act fraudulently, based on observed patterns of real fraud. We introduced known signatures of fraud in our model and simulations to test and evaluate our fraud detection methods. The resulting behaviour of the agents generate a synthetic log of all transactions as a result of the simulation. This

synthetic data can be used to further advance fraud detection research, without leaking sensitive information about the underlying data or breaking any non-disclose agreements.

Using statistics and social network analysis (SNA) on real data we calibrated the relations between our agents and generate realistic synthetic data sets that were verified against the domain and validated statistically against the original source.

We then used the simulation tools to model common fraud scenarios to ascertain exactly how effective are fraud techniques such as the simplest form of statistical threshold detection, which is perhaps the most common in use. The preliminary results show that threshold detection is effective enough at keeping fraud losses at a set level. This means that there seems to be little economic room for improved fraud detection techniques.

We also implemented other applications for the simulator tools such as the set up of a triage model and the measure of cost of fraud. This showed to be an important help for managers that aim to prioritise the fraud detection and want to know how much they should invest in fraud to keep the loses below a desired limit according to different experimented and expected scenarios of fraud.

to my beloved family:
Helena, Isabella and Linnea

Acknowledgements

Many people had contributed to this work. It's been an honour for me to gain such an incredible and valuable experience during this time with them. First, I would like to thank my main supervisor Dr. Stefan Axelsson, for being directly involved in my work and co-author of my publications. I would like to thank Prof. Bengt Carlsson for his guidance as a senior researcher and immense support to my work. I have to extend my thanks to all doctoral students, professors and researchers in our department (DIDD) at BTH for sharing this amazing journey with me over the last 5 years. Many of my colleagues have actively participated and contributing to my work with their comments to increase the quality of it.

Blekinge Institute of Technology has been economically supporting my research all over these 5 years as a PhD student. I have only words of gratitude for them. This thesis work was possible due to the research project "Scalable resource-efficient systems for big data analytics" funded by the Knowledge Foundation (grant: 20140032) in Sweden.

Finally, I would like to thank my family. They are an important source of inspiration, support and motivation to continue my academic journey. My wife Helena, for her unconditional support and my two daughters, Isabella and Linnea. I extend my thanks to Margareta, Tommy, Stefan and Karolina who are my Swedish family. I have the biggest gratitude to my parents Jesus and Soledad who are always there for me. My sister Paula and my other relatives that came to visit me all the way from Colombia and brought part of my native country to my new home country Sweden.

Edgar Lopez
September 2016, Karlskrona, Sweden.

Preface

This thesis is based on the work presented in the following six papers. The papers I, III, IV and V are published in peer-reviewed conference proceedings. Paper II is published in a journal and Paper VI is been submitted to a journal and is currently under peer-reviewing.

The included papers have been modified to fit this format, but the content is unchanged.

Paper I

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* (2013). (Best Paper Award)

Paper II

E. A. Lopez-Rojas, D. Gorton, and S. Axelsson. "Using the RetSim simulator for fraud detection research". In: *International Journal of Simulation and Process Modelling* 10.2 (2015), p. 144

Paper III

E. A. Lopez-Rojas and S. Axelsson. "Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research". In: *Advances in Computational Social Science and Social Simulation*. Barcelona, Spain, 2014

Paper IV

E. A. Lopez-Rojas. "Extending the RetSim Simulator for Estimating

the Cost of fraud in the Retail Store Domain". In: *The 27th European Modeling and Simulation Symposium-EMSS*. Bergeggi, Italy, 2015

Paper V

E. A. Lopez-Rojas and S. Axelsson. "Using the RetSim Fraud Simulation Tool to set Thresholds for Triage of Retail Fraud". In: *20th Nordic Conference on Secure IT Systems, NordSec 2015*. Stockholm: Springer, 2015, pp. 156–171

Paper VI

E. Lopez-Rojas and S. Axelsson. "Applications of the PaySim simulator for fraud detection research". In: *Submitted for Journal Publication* (2016). (Submitted)

There are other papers published that are not included in this thesis but are related to this research:

Paper VII

E. A. Lopez-Rojas and S. Axelsson. "Money Laundering Detection using Synthetic Data". In: *The 27th workshop of Swedish Artificial Intelligence Society (SAIS)* (2012), pp. 33–40

Paper VIII

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* (2012), pp. 25–32

Paper IX

E. A. Lopez-Rojas and S. Axelsson. "Banksim: A bank payments simulator for fraud detection research". In: *26th European Modeling and Simulation Symposium, EMSS 2014*. Bourdeaux,France, 2014, pp. 144–152

Paper X

E. A. Lopez-Rojas and S. Axelsson. "A Review of Computer Simulation for Fraud Detection Research in Financial Datasets". In: *Future Technologies Conference, San Francisco, USA*. 2016

Paper XI

E. A. Lopez-Rojas, A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: *The 28th European Modeling and Simulation Symposium-EMSS*. Larnaca, Cyprus, 2016

Contents

List of Tables

List of Figures

1 Introduction

Fraud is an important problem in a number of different fields. The economic impact can be substantial. The detection of fraud is therefore a worthwhile endeavour. However, in order to investigate, develop, test, evaluate, measure and improve fraud detection techniques there is a need for detailed information about the field the fraud targets and its peculiarities. All these needs can be satisfied if we could find publicly available data of diverse financial transactions scenarios so that different approaches can 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. Even with the access required to this kind of data it is often difficult to obtain data that represent diverse fraud scenarios where researchers can experiment different techniques and tune their parameters under a controlled environment.

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 this lack of publicly available relevant data sets. Our aim with this work is to address this situation with the use of computer simulation.

The work presented in this thesis is an effort to address the lack of public available financial data, with the aim that: if we cannot get full access to public financial records due the restrictions mentioned before, then one alternative is to generate such a data. However, simulating a

financial environment and generating synthetic data brings new challenges, specifically those related to characteristics of the generated data such as quality, privacy, fidelity and usefulness.

We present two different case studies regarding the simulation of financial transactions for fraud detection research. The first consists in a new payment system that uses mobile phones to easy the payments called *PaySim* [35]. This system was under development at the beginning of our research and was the inspiration for using this approach. We used the existing system to build a model but we lacked any real data to calibrate and evaluate the model. Which lead our research in the second case study called *RetSim* [40]. *RetSim* is a simulation tool that generates data from realistic scenarios of a retail store based on transactional data from one of the biggest shoe retailers in Scandinavia. We developed and used *RetSim* for research in the topic of staff fraud detection and later on we used it as a tool to measure the cost of fraud, which is an important application of this simulator. A couple of years after, we finally got hands on real data from the mobile payment system and improved the *PaySim* simulator using similar techniques that were developed in *RetSim*.

There is a third case study that we opted to leave out of the scope of this thesis, but it is also interesting to mention. We made use of aggregated shared information of payments from a bank in Spain to build a simulator called *BankSim* [37]. BankSim is implemented in a similar way as the *RetSim* and *PaySim* simulator.

The main goal of developing these simulators is that it enables us to produce and share realistic and diverse fraud data with the research community, without exposing potentially sensitive and private information about the actual source. The simulators use only aggregated information from the original database and based on that it generated synthetic individual behaviour. There is still a need of original data, but since it is only aggregated statistics, its disclosure prevents the exposure of personal or private customer information.

Simulation also have other benefits, it can produce more data much more

quickly and with less cost than for instance, collecting data, trying different scenarios of fraud, detection algorithms, and personnel and security policy approaches, in an actual store. Since we have control over the simulation, we can flag any malicious behaviour and measure the cost of fraud and perform supervised machine learning methods. Testing these scenarios in a real world situation introduces many uncertainties and cost for the business, for instance, a store can usually detect the lack of stolen articles only after performing a time consuming full inventory and comparing these values against the sales.

Our approach is a method to generate anonymous synthetic data of the transactions in a "typical" financial system, that can then be used as part of the necessary data for the research, development, evaluation, measure and testing of fraud and its countermeasures. Furthermore, the data set generated could be the basis for research in other fields, such as demand prediction, logistics and demand/supply research which is not covered in this thesis but a currently hot topic of research.

1.1 Background

This section explains terms that are pertinent to our research and give a brief overview of the topics that are covered in this thesis. We begin with some context explanation of the research, followed by general definitions of simulations and agent-based simulation. Then we explain the domain of the research, that is, financial transactions and we end with a small introduction into fraud and fraud detection in this domain.

1.1.1 Context of this research

It is very common to begin a research project with an ambitious goal. This research was no exception. We started with the goal to detect money laundering in a mobile money payment system. Even if we are not specifically covering this topic in this thesis, this is one of the capabilities of this approach.

The first issue we came up against was the current stage of development of the mobile money system, which made it impossible to collect any kind of data to analyse, test or produce any desirable scientific result. This lack of data made us think about alternatives to deal with this issue. Simulation of data started to sound as an attractive option. After all, many situations, scenarios and events that are expensive or hard to reproduce in our current world are being studied with the help of simulation. This brought the concept of Agent-Based Simulation (ABM) to the forefront. ABM is a modern and effective technique to deal with the complexity of the real world. Specifically, in our case, the simulation of the complex social behaviour of people performing financial transactions.

1.1.2 Simulation and related technologies

In this section we give a basic introduction to simulation and more specifically, Multi-Agent Based Simulation (MABS), which is the approach we are using to build our simulators. We also discuss the benefits and disadvantages of using synthetic data for fraud detection research.

1.1.2.1 Simulation and Computer Simulation

Simulation uses a model to infer conclusions about the behaviour of real-world phenomena. Computer simulation seeks to attain the same goal but requires the model to be implemented on a computer. Computer Simulation can be classified as a branch of applied mathematics [47].

Simulations with the aid of computers became very popular due to the impossibility to replicate or simulate certain complex phenomena using other techniques. The amount of processing needed for complex simulation makes the topic of computer simulation a hot research topic nowadays [47]. Specifically, because almost everybody has access to a computer with enough capacity to run quite large simulations. The scaling capabilities of simulation make possible for large organisations and researchers to benefit from the power of supercomputers to run simulations that demands huge amount of computer resources (disk, memory, speedy processing in parallel)

such as weather forecasting, astronomical phenomenons and large financial simulations.

There are many different types of computer simulation: *Discrete event simulations*, *Continuous system dynamics*, *Agent-Based Simulations* and a combination or *Hybrid simulation* [56]. In this thesis we make use of Agent-Based Simulation (ABS) and Multi-Agent-Based Simulation (MABS) approach.

1.1.2.2 Multi-Agent Based Simulation

Multi-Agent Based Simulations (MABS) are built from the bottom up. This means that the design does not need to know the complex structural behaviour of the system. It makes use of the knowledge of the individual behaviour of the components or agents. By programming the micro-behaviour of the agents, a macro-behaviour emerge and it is observed in the system [7, 23, 52, 56].

An Agent-Based Simulation (ABS) is centred on the *agent*. An agent is an autonomous self-directed unit. In our case agents are representations of people and entities with specific identifiable roles such as customers or salesmen. One important characteristic of an agent is that it comes with a specific *behaviour* that can be given, for instance, by a set of rules. Sometimes simple rules can result in an emerging behaviour that can hardly be foreseen. This important characteristic makes MABS a useful approach for modelling and simulating complex structures such as societies. All agents interact in an *environment*, which is designed to represent a real world scenario. Agents can sense the environment as well as other agents (usually in their proximity). Each agent may also include a memory that saves the possible different *states* or attributes of the agents. Finally, in each step of a simulation all agents behave with the aim of achieving a goal. The sum of all the agents' goals generates a system behaviour that sometimes result in an unpredicted, or rather, unpredictable system behaviour.

In our particular case, the agents represent the clients in our Mobile

1.2 Aim and scope

Our aim is to find suitable methods and techniques to simulate realistic financial scenarios and generate synthetic data sets that can be applied to fraud detection and related problem. Standard simulators and data sets allow researchers and organisations to develop, test, experiment, evaluate, measure and compare diverse fraud detection methods. During the first part of this study our initial aim was to generate a realistic synthetic dataset in a mobile money payment system [31]. The main reason was because in our research we had two main constrains: first, the lack of available data for this domain which made us direct our research towards the *generation of synthetic data*; and second, the impossibility to accurately measure the cost of fraud and estimate the loss of the business in real data sets lead as to one of the main applications of simulation and is the *measure of fraud*.

The initial scope of this thesis covered the domain of payments made through a mobile money payment system. We later extended the scope to cover the retail store domain, which is a related financial system. The other domain which was covered with a bank simulator remained out of the scope of this thesis.

1.3 Our Simulators

This section describes *PaySim* and *RetSim*. These two simulators were developed as part of this project to answer the research questions presented in section 1.5. Both simulators are described in detail in the papers that are included in the following chapters of this thesis.

1.3.1 PaySim, a Mobile Money Payment Simulator

The 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 fraud.

Unfortunately, during the initial part of our research this service was only been running in a demo mode. This prevented us from collecting any data that could had been used for analysis of possible detection methods.

The development of PaySim covers two phases. During the first phase, we modelled and implemented a MABS that used the schema of the real mobile money service and generated synthetic data following scenarios that were based on predictions of what could be possible when the real system starts operating [34, 35]. During the second phase we got access to transactional financial logs of the system and developed a new version of the simulator which uses aggregated transactional data to generate financial information more alike the original source [30, 41].

1.3.2 RetSim, a Retail Store Simulator

Since we have access to several years' worth of transaction data from one of the largest Scandinavian retail shoe store chains, we developed *RetSim*, a *Ret*ail shoe store *Sim*ulation, built on the concept of MABS. *RetSim* is designed to be used in developing and testing fraud scenarios at a retail store, while keeping business sensitive and private personal information about customer's consumption secret from competitors and others. Simulations in the domain of retail stores have traditionally been focused on finding answers to logistics problems such as inventory management, supply management, staff scheduling and for customer queue reductions [57]. To our knowledge, *RetSim* is the first simulator with the purpose of fraud detection on the retail store domain.

The defence against fraud is an important topic that has seen some study. In the retail store the cost of fraud if 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. However, we also model 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 are explained in section 1.1.3.2 and includes: *Refunds* and *Coupon Reductions/Discounts*.

In terms of the object model used in *RetSim* the refund fraud scenario was 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 ones. 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.

1.4 Research Method

We started with an exploratory literature review, and preliminary research into the possibility of using synthetic data for fraud detection, covered in [33]. Since our idea had no precedents we performed again a literature review and found similar work from 2012 until 2016 [39]. We present our findings in section 1.7.

Just as an astronomer uses a telescope to study the stars, we needed a proper tool to study our topic of fraud inside financial data. The best available way we found for performing our research was with the aid of computer simulation.

During the initial stage of our research we then developed two different simulators based on two case studies of systems within the domain of financial transactions. The first, consisted of a payment system that uses mobile phones *PaySim*, introduced in [35] and improved in [41]. The second was *RetSim*, a simulation tool that generates realistic scenarios based on transactional data from a shoe retail store [36, 40]. All simulators use the same Multi-Agent Based Simulation toolkit, called MASON, which is

implemented in Java [43].

PaySim was initially based on the schema of the database, and the described behaviour of the customers for the simulated system. During the development of *PaySim*, the mobile payment system was in a testing phase. This situation made it impossible for us to obtain real data at that time from actual use of the system which is needed to calibrate the behaviour of the agents.

PaySim got implemented and it is currently used in several countries around the word. But it took some time for this to happen. This lack of real data changed our focus towards our second case study.

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

To better understand the problem domain, specifically the normal operation of a store, 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 mimicking 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 used 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 in section 1.3.2: Refunds and Discounts.

A simulation can only be useful for a specific purpose if the model provides an abstraction of the real-world, capturing the essentials of the studied phenomenon. For our simulations we used a process of evaluation that consist of two main steps: verification and validation.

Verification is the process that consist of ensuring that the model follows the rules of the real-world scenario described. For verification we tested our model by checking that the behaviour of all agents reflects the real-world scenario and no other behaviour is present in the model that cannot possible happen in the reality.

Validation is more difficult than verification in our case. We need to evaluate if the output of the simulation satisfies requirements of similarity, i.e. if outputs of the real-world phenomenon given a specific set of input variable are sufficiently similar to our model. For the *RetSim* simulator we validated the output using graphic and statistical methods that allowed us to check if the output given by the simulator satisfied the distributions of sales present in the original data set. For our first simulation (PaySim), validation was difficult in the beginning since we did not have any real data to compare to the output, it was only until we improved the simulator with the help of real data that we could perform a proper validation in a similar way as the *RetSim* simulator.

Once the tools were built, we continued our research by finding different applications for fraud detection where we could use the simulators. Since we had more time to work with *RetSim*, we ran several simulations adjusting

different parameters of aggressiveness on the fraud behaviour while keeping the background data (or normal customers behaviour) the same. We did studies of threshold, triage model and cost of fraud in retail stores and presented our results in several of the contributions.

For *PaySim*, we initially used the synthetic generated data to illustrate the possibilities and usefulness of the model by first generating a synthetic data set and second by performing an exploratory evaluation of fraud detection, using labelled data, and machine learning techniques to classify the injected malicious behaviour. Once we calibrated the model with the help of real data set we performed similar experiments as with *RetSim* to study the cost of fraud and the effectiveness of threshold detection.

In summary we used similar research methodology for each of our two simulator. We first built a simulation tool according to a specific domain. We then calibrated it using real data obtained from our partners and evaluated the tool by performing verification and validation. Finally, we used the tool to experiment in fraud detection and estimated the cost of fraud in different scenarios.

1.5 Research Questions

Through our research and having dealing with the lack of available data, we performed a preliminary studied in order to find out whether it is possible to do reliable fraud detection using a synthetic dataset. After positively answering this, we initially formulated three research questions that were addressed by the contributing papers that this thesis is based on. The initial research questions that were formulated during our research are:

RQ1

How could we generate a realistic synthetic data set for financial transactions for the purpose of fraud detection?

RQ2

How could we model and simulate a retail shoe store and obtaining a

realistic synthetic data set for the purpose of fraud detection?

RQ3
Is the generated data set properly anonymized with respect to the original data set?

After having a proper method to simulate financial data, we formulated the next set of questions that where more oriented to address the application of the simulators as a tool and the benefits of synthetic data sets. Thus, we formulated four more research questions:

RQ4
Is threshold detection sufficient to keep the losses from fraud at manageable level?

RQ5
How to set up a triage model based on thresholds that detects the fraud that harms the most?

RQ6
How to estimate the cost of fraud in different expected scenarios?

RQ7
What are the applications of a simulator in the domain of mobile money transactions?

The publications presented in section 1.6 address all research questions as shown in table 1.1.

The lack of available data for the domain of mobile payments made us switch our research towards the generation of synthetic data. This is how we started to formulate our first preliminary study in Paper VII with the question: *Is it possible to do fraud detection using a Synthetic Dataset?*. This was the initial step of this research and addressed the idea and the possibility of using synthetic data for fraud detection. In Paper VII

Table 1.1: *Contribution of Papers to Research Questions*

Paper	RQ1	RQ2	RQ3	RQ4	RQ5	RQ6	RQ7
I	Main	Main					
II		Covered	Main	Main			
III	Covered	Covered	Covered	Covered			
IV						Main	
V					Main		
VI	Main		Covered	Main		Covered	Main
VII	Covered						
VIII	Covered		Covered				
IX	Covered						
X	Covered	Covered					Covered
XI	Covered		Covered				

we presented an analysis of the difficulties and consequences of applying machine learning techniques on a synthetic dataset for the purpose of detecting anomalous or suspicious transactions that are based on illegal activities. In this paper we also discuss the pros and cons of using synthetic data, and problems and advantages inherent in the generation of such a data set. We illustrated this idea using a case study based on a Mobile Money Payment system and suggest an approach based on Multi-Agent Based Simulations (MABS). We performed a literature review that lead to an analysis of the implications of using synthetic data, and we concluded that if we can build a realistic simulator that generates such data, the fraud detection techniques that we apply will be useful and applicable to the original data.

Paper I, III, VI, VII, VIII, IX and X addressed RQ1: *How could we generate a realistic synthetic data set for financial transactions for the purpose of fraud detection?*. RQ1 is our initial and generic research question that have been carried over several papers. In Paper I, we continued the analysis and work of Paper VII and VIII and implement the approach suggested there. In Paper I and II we present an approach based on a Multi-Agent Based Simulation (MABS) for the generation of synthetic

financial data. Paper VIII presents the generation of synthetic data logs of transactions and the use of such a data set for the study of different detection scenarios using machine learning techniques on labelled data with red flags for suspicious transactions that follows a fraud behaviour pattern. We later called this simulator *PaySim* (see section 1.3.1) and improved it in Paper VI as part of our strategy to continue and extend our research. Paper IX was a side project and include the introduction of the simulator called *BankSim*, which is based on aggregated transactions from a bank in Spain.

Paper I improved the model presented in Paper VIII to contribute to answer RQ2: *How could we model and simulate a retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?*, by introducing *RetSim*. *RetSim* (see section 1.3.2) is our proposal to generate realistic synthetic data, using an Agent-Based Simulator of a shoe store based on the transactional data of one of the largest retail shoe sellers in Sweden. Paper III is part of a special issue of a journal where we were given the opportunity to extend our work, which resulted in Paper II. Hence, Paper II becomes an extension of Paper I and introduces a new research question.

Paper II, III covers RQ2, RQ3 and RQ4 by describing the methods and the benefits of the simulators. Paper II complemented RQ2 and RQ3 by extending the work presented in Paper I. In order to answer RQ3: *Is the generated data set properly anonymized with respect to the original data set?*, we reasoned about what information from the real data set leaks to the generated synthetic data. Since we do not keep any record of who is purchasing what items in the store, we can ensure that no real customers are exposed. We then reasoned about the overall economic information about the store. Even though there is, of course, some leakage from a business perspective, the data owners consider that this data is old enough to not pose a risk for their business today but for our research is good enough to build our model from it.

Paper II and VI used the *RetSim* and the *Paysim* simulators to model

malicious behaviour and answer RQ4: *Is threshold detection sufficient to keep the losses from fraud at a manageable level?*, RQ4 directly concerns with the efficacy of current threshold detection techniques to keep financial losses low enough to not pose a risk for the business. Our findings using the *RetSim* and the *Paysim* simulators lead us to interesting experiments, that are not possible in a real scenario where the loses are mostly unknown. Even if our experiments are just a preliminary study, we can measure the efficiency of a threshold method detection by summing up the loses of our malicious agents and resting it from the total amount detected by a simple threshold detection method. The final answer to RQ4 is of course from the manager but we consider that in our research threshold detection performed well enough. We showed two simple scenarios where threshold control works to combat an aggressive and moderate fraud behaviour scenario. At the same time, we found that when the fraud is moderate, threshold control techniques are not that effective and the cost of false positives becomes higher, but still below our set level of acceptable fraud.

Paper V answers RQ5: *How to set up a triage model based on thresholds that detects the fraud that harms the most?*. Triage model applied to fraud detection is based on the principle of set up a priority bin for the most critical cases of fraud. It is a trial and error how the triage model is performing in most of the organizations. We made use of the *RetSim* simulator to show how can this process be tuned when we have an estimate of how much fraud can we detect given a certain set up.

Paper IV answers RQ6: *How to estimate the cost of fraud in different expected scenarios?*. We made use again of the *RetSim* simulator to generate more than 500 possible scenarios of fraud. The scenarios contained different parameters for fraud including number of staff compromised and the aggressiveness of the defraud. It turns out that the cost of fraud is an important value that it is hard to calculate without the help of our simulator. The cost of fraud gives the managers an idea of how much should they invest in fraud detection resources to minimize the loses.

Paper VI answer RQ7: *What are the applications of a simulator in the*

domain of mobile money transactions?. We improved the PaySim simulator with the logs from a running service in Africa, we injected fraud committed by taking over control on the accounts and transferring funds to third mule accounts just to empty them in cash. With this setup we were able to measure the cost of fraud given different fraud parameters and setup a fraud detection method based on the rate of how fast the victim account is getting empty. This shows that PaySim can be used for testing fraud detection methods and to estimate the cost of fraud.

1.6 List of Publications

This thesis is mainly based on the work presented in the following six papers that contributed to answer the research questions presented in section 1.5:

Paper I

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* (2013). (Best Paper Award)

Paper II

E. A. Lopez-Rojas, D. Gorton, and S. Axelsson. "Using the RetSim simulator for fraud detection research". In: *International Journal of Simulation and Process Modelling* 10.2 (2015), p. 144

Paper III

E. A. Lopez-Rojas and S. Axelsson. "Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research". In: *Advances in Computational Social Science and Social Simulation*. Barcelona, Spain, 2014

Paper IV

E. A. Lopez-Rojas. "Extending the RetSim Simulator for Estimating the Cost of fraud in the Retail Store Domain". In: *The 27th European Modeling and Simulation Symposium-EMSS*. Bergeggi, Italy, 2015

Paper V

E. A. Lopez-Rojas and S. Axelsson. "Using the RetSim Fraud Simulation Tool to set Thresholds for Triage of Retail Fraud". In: *20th Nordic Conference on Secure IT Systems, NordSec 2015*. Stockholm: Springer, 2015, pp. 156–171

Paper VI

E. Lopez-Rojas and S. Axelsson. "Applications of the PaySim simulator for fraud detection research". In: *Submitted for Journal Publication* (2016). (Submitted)

There are other papers published that are not included in this thesis but are also part of the work performed during this research:

Paper VII

E. A. Lopez-Rojas and S. Axelsson. "Money Laundering Detection using Synthetic Data". In: *The 27th workshop of Swedish Artificial Intelligence Society (SAIS)* (2012), pp. 33–40

Paper VIII

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* (2012), pp. 25–32

Paper IX

E. A. Lopez-Rojas and S. Axelsson. "Banksim: A bank payments simulator for fraud detection research". In: *26th European Modeling and Simulation Symposium, EMSS 2014*. Bourdeaux,France, 2014, pp. 144–152

Paper X

E. A. Lopez-Rojas and S. Axelsson. "A Review of Computer Simulation for Fraud Detection Research in Financial Datasets". In: *Future Technologies Conference, San Francisco, USA*. 2016

Paper XI

E. A. Lopez-Rojas, A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: *The 28th European Modeling and Simulation Symposium-EMSS*. Larnaca, Cyprus, 2016

1.7 State of Art

Financial fraud is being addressed by many different techniques including simple controls such as thresholds or statistical limits and more advanced and elaborated techniques such as data mining-based detection.

This section lends heavily from PAPER X. We did a literature study review of the state of art in the current field including previews surveys in fraud for financial transactions and related work in the same field [39]. In this section we present our findings.

1.7.1 Previews surveys

Some of the first surveys in this field were written by Bolton et al., Wang, Yue, Wu, and Wang [13, 61, 64].

Bolton et al. [13] reviews research papers in several fraud detection domains. It covers fraud relates with credit cards, money laundering, telecommunications, computer intrusion and medical and scientific fraud. In general, this survey shows that similar statistical methods and techniques have been applied in research in the detection of fraud in different domains. Some key points of this survey are that the speed of detection matters in certain domains such as banking fraud or in telecommunications to lesser the losses. Another interesting finding is the lack of flagged fraud data activity in many of these domains.

Yue, Wu, and Wang [64] introduces a classification framework for financial fraud detection which expands the typical data mining process and adds into consideration the specific characteristics of financial data for fraud detection. This survey covers several research on data mining

detection which includes work on regression, neural networks and statistical tests.

Wang [61] surveys the field of accounting fraud detection. This survey focuses more specifically on data mining-based techniques for financial statements. This surveys points to the two main difficulties in this field: lack of access to data and lack of mature methods to discover fraud. Besides this, another factor that hampers a proper comparison of these methods is that there is a notorious difference between the data sets, methods and the evaluation techniques.

There are many other surveys in this field that cover similar topics of fraud detection using data mining techniques as the main approach [11, 27, 50, 59, 60, 62].

Most recently Ahmed, Mahmood, and Islam [3] presented a survey on anomaly detection techniques in financial domains that covers credit card fraud, mobile phone fraud, insurance claim and insider trading in market investments. Some of the methods covered are partition based clustering, hierarchical clustering and others. A remarkable comment in this survey is that the scarcity of public available data is a problem in this domain. Obtaining appropriate access to financial data to perform research in this area is extremely difficult due to privacy and competitive reasons and finally points out that synthetic data sets are a possible solution for this problem. However, generating synthetic data has its challenges. Data should reflect the normal and fraudulent behaviour. Therefore, it requires expertise in the domain to design probable fraud scenarios. It is probably not straight forward to move the fraud detection techniques from the synthetic data to the real data set, it is expected to behave similar but this might require further refinements. Finally, both normal and fraud behaviour evolve over time, therefore there is a need to evolve the simulator as well.

1.7.2 Related work on simulating transactions

There is a novel approach in fraud detection which involves the use of simulators to produce enough financial data which contains both the normal

behaviour and the fraud behaviour. We found 15 research papers from 16 authors on this field during a period of time of 5 years that correspond of the years between 2012 and 2016. Our work represents 9 out of the 15 papers. We have the intention to show that there are some other researchers besides us that are currently working in this important topic.

The work by Gaber et al. [21] presents another similar technique to generate synthetic logs for fraud detection. The main difference here was that this time there was available real data to calibrate the results and compare the quality of the result of the simulator. The purpose of this study was to generate testing data that researchers can use to evaluate different approaches.

IncidentResponseSim by Gorton [22] is a simulation tool to support the assessment of risk of online banking services. This simulator uses the power of simulation to estimate the economic consequences of current and emerging threats modelled with the aid of an incident response tree in combination with a qualitative model.

The work by Rieke et al., Zhdanova et al. [55, 66] on fraud detection in mobile payments is done in a similar domain as our work and other authors reviewed [21, 33].

Rieke et al. uses a tool named Predictive Security Analyzer (PSA) with the purpose of identifying cases of fraud in a stream of events from a mobile money transfer service [55]. PSA is based on a dataset of 4.5 million logs from a mobile money service over a period of 9 months. They use simulation due to the limitation and knowledge of existing fraud in the current logs. The main focus on PSA is to detect money laundering cases that are cause by the interaction of several users of the system in an attempt to disguise the fraud among the normal behaviour of the clients. As a result, the paper shows that PSA is able to efficiently detect suspicious cases of money launder with the aim of automatically block the fraudulent transactions.

Zhdanova et al. [66] is a continuation of the work done by Rieke et

al. [55] and uses the simulator developed by Gaber et al. [21] to evaluate the results. Semi-supervised and unsupervised detection methods are applied to a mobile money dataset due to the advantage over supervised methods in this type of data where there is a difficulty in having a training data with known cases of fraud.

Malekian and Hashemi [46] worked on a fraud detection method that handles the concept drift on e-payments. The author used simulation due to lack of real credit card data for testing. This simulator aims to build customers profiles that contains seasonal and weekly patterns. The idea behind is to detect when a customer behaviour differs from the historic pattern. With the introduction of concept drift detection on e-payments the results showed a substantial reduction in the false positives cases.

Alexandre and Balsa [5, 6] present a method to detect fraud using intelligent agents that perform the tasks that manually a security officer should do on its own over a limited amount of data. The advantages are that the logic included in the agents allows them to perform with excel this tasks. They also used simulated data to evaluate the performance of their methods.

1.8 Contributions

Our contributions begin with the introduction of two new simulators: *PaySim* and *RetSim*. These two simulators are important because, to the best of our knowledge, it was the first time that synthetic data of financial transactions was proposed to develop fraud detection methods and techniques.

Our simulators contain a modern simulation framework based on the concept of Multi-Agent Based Simulation that allows the implementation of complex micro behaviour from normal customers and fraudulent customers to generate an aggregate macro behaviour on the generated data. Furthermore, by using our simulators we created experiments to test our generated questions concerning fraud detection methods such as simple

threshold detection.

In summary our contributions are:

PaySim simulator: A mobile payments computer simulation tool and a method to generate synthetic datasets. This is the first simulator of its kind and can generate different realistic scenarios. It contains a model of the normal and malicious behaviour of fraud that can be easily expanded by the introduction of new patterns (Papers III and VI).

PaySim dataset: A realistic dataset based on the sample obtained from our research partner. This dataset was calibrated to match the behaviour of customers using aggregated transactions from a mobile money service in Africa (Paper VI).

PaySim fraud dataset: A realistic dataset based on the sample obtained from the source which contains injected fraud behaviour. This fraud behaviour was modelled after the study of interesting fraud behaviours patterns. The importance of this dataset is that the fraud instances are labelled, allowing researchers to measure the fraud (Paper VI).

PaySim applications: A study of threshold detection and cost of fraud using PaySim and the generated datasets. In this study we argue that threshold detection is widely used due to its performance and simplicity that considerable reduces the loss due to fraud (Papers III and VI).

RetSim simulator: A retail store computer simulation tool and a method to generate synthetic datasets. Similar to PaySim, this is the first simulator of its kind and can generate different realistic scenarios. It contains a model of the normal and malicious behaviour of fraud that can be easily expanded by the introduction of new patterns (Paper I).

RetSim dataset: A realistic dataset based on the sample obtained from our research partner. This dataset was calibrated to match the

behaviour of staff and customers using aggregated transactions from a store of one of the biggest shoe retailers in Scandinavia (Paper I).

RetSim fraud dataset: A realistic dataset based on the sample obtained from the source which contains injected fraud from the staff. Once again the importance of this dataset is that the fraud instances are labelled, allowing researchers to measure the fraud (Paper II).

RetSim applications: A study of threshold detection, triage model and cost of fraud using RetSim. With the help of the RetSim simulator properly calibrated to match the normal customers and the injection of fraud behaviour from the staff we generated diverse scenarios of fraud where we could measure the cost of fraud. This quantification of fraud is of particular importance to managers who need to support their investments in fraud detection technology and personal (Papers III, IV and V).

1.9 Conclusions

In summary, we present two case studies that implement a Multi-Agent Based Simulation model to address the problem of simulating 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 emerging macro behaviour as found on the real dataset.

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. We later improved it and experimented with fraud detection and measure the cost of fraud. 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*. 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 simulators give us the benefit over real data that we can quantify and measure the amount of loses committed by our malicious agents, this is especially helpful for measuring the cost of fraud in different scenarios. Using this advantage, we evaluated if threshold based detection could keep the risk of fraud at a predetermined set level (threshold). 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 costlier to implement.

However, setting the right threshold does not seem to be an easy task without knowing the cost of fraud. This is where simulators come in handy and help the managers in answering several questions regarding fraud such as: Where should I set up a threshold to minimize the cost?, How much should we invest in fraud detection?, What is an acceptable level of fraud that does not harm our business?.

We argued that our simulators are ready to be used as tools for the generation of synthetic data sets of financial activity. We intend to make *PaySim* and *RetSim* available to the research community together with standard data sets.

1.10 Future Work

We aim to improve the accuracy of our payment simulator *PaySim* with the help of more real data that covers more time, therefore we could study more phenomenon's due to seasons or peak days for financial transactions

such as Black Friday or Christmas. We have successfully achieved a realistic simulation for a retail store with *RetSim* and presented different applications of this simulator. Despite that, our work is not complete in this area. We would like to extend to different kinds of retail stores, types of fraud and detection techniques.

It is on our interest to extend the number of different applications for both of our simulators in areas related to fraud detection.

Furthermore, detecting complex types of fraud such as money laundering often requires a simulator that contains diverse interconnected financial information from sources such as banks, payments and retailers. We are not yet ready to implement fraud detection methods for money laundering until we can complete such a financial model.

We would like to work more on our third simulator called *BankSim*. This simulator uses aggregated data from credit card payments that were made publicly available by a bank in Spain. We are also seeking partners in the bank industry to be able to extend our research in this domain and get access to real data sets to model and improve *BankSim*.

Our initial goal of addressing complex types of fraud such as Money Laundering is still present. In order to address such a complex problem, we aim to incorporate three different kind of simulators. The first one covers the Retail Stores (RetSim) the second one covers a payment system (PaySim) and the third one covers the bank transactions (deposits, withdraw and transfers). Our future work will then focus on the development, improvement and integration of different domain simulators as the key to research in the area of Money Laundering.

One of the biggest challenges of this development phase is to integrate all three simulators into one single Multi-Simulator that shares a common reference to customers and can keep track of the transactions of a single agent across all simulators.

2

RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection

Edgar Alonso Lopez-Rojas, Stefan Axelsson and Dan Gorton

Abstract

RetSim is an agent-based simulator of a shoe store based on the transactional data of one of the largest retail shoe sellers in Sweden. The aim of RetSim is the generation of synthetic data that can be used for fraud detection research. Statistical and a Social Network Analysis (SNA) of relations between staff and customers was used to develop and calibrate the model. Our ultimate goal is for RetSim to be usable to model relevant scenarios to generate realistic data sets that can be used by academia, and others, to develop and reason about fraud detection methods without leaking any sensitive information about the underlying data. 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 RetSim generates data that usefully approximates the relevant aspects of the real data.

Keywords: Multi-Agent Based Simulation, Retail Store, Fraud Detection, Synthetic Data.

2.1 Introduction

In this paper we introduce *RetSim*, a **Ret**ail shoe store **Sim**ulation, built on the concept of Multi Agent-Based Simulation (MABS). RetSim is based on the historical transaction data provided by one of the largest Nordic shoe retailers. This data contains several hundred million records of diverse transactional data from a few years ago, and covering several years. That is, this data is recent enough to reflect current conditions, but old enough to not pose a risk from a competitor analysis standpoint.

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 RetSim is to learn the relevant parameters that governs the behaviour in a retail store to simulate *normal* behaviour, which is our focus in this paper.

The main contribution and focus of this paper is a method to generate anonymous synthetic data of a retail store, that can then be used as part of the necessary data for the development of fraud detection techniques. Even so, the data set generated could also be the basis for research in other fields, such as demand prediction, logistics and demand/supply research.

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 fraud in a retail store. As this is our ultimate goal, fraud heavily influenced the design of RetSim.

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 due to 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 organized as follows: Section 2.2 introduce the topic of fraud detection for retail stores and present related work. Sections 2.3 describes the problem, which is the generation of synthetic data of a retail store. Section 2.4 shows a data analysis of the current data. Section 2.5 presents an implementation of a MABS for our domain and shows the description of some retail fraud scenarios. We present our results and verification of the simulation in section 2.7 and finish with a discussion and conclusions, including future work in section 2.8.

2.2 Background and Related Work

Simulations in the domain of retail stores have traditionally been focused on finding answers to logistics problems such as inventory management, supply management, staff scheduling and for customer queue reductions [14, 15, 57].

There is currently a lack of research in the area of simulation of the retail environment for fraud detection and here is where we focus in this work.

We have previously analysed the implications of using machine learning techniques for fraud detection using a synthetic dataset [33]. 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 [35]. However, this work was not based on any underlying data, but rather on assumptions of what such data could contain. Here we continue and build a realistic simulation based on a real data set that in the future can be used to test diverse fraud detection techniques.

Data mining based methods have been used to detect fraud [53]. 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 with the 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 [1], however this has not been done over transactional data. There are tools such as IDSG (IDAS Data and Scenario Generator [29]) 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.

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 [4]. 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.

2.3 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 retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?*

2.4 Data Analysis

To better understand the problem domain we began by performing a data analysis over the historical data provided by the retailer. 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.

We initially started by selecting five stores that represent different sizes of store in the company. We selected two big stores, one medium and two small. We extracted statistical information from the data set, presented in table 2.1. All prices given are in a fictitious currency.

Due to a lack of space we will focus our presentation of the analysis on one of the big stores by sales volume, store one. Store one is relatively richer in data than the smaller stores. This is specially interesting, since we are more likely to find actual cases of fraud in a big store. We took a sample that comprises the sales during a year. We selected the transaction tables that detail cash flow and the articles inventory, which give us a good idea of how many transactions a big store can produce in a year, and how many different types of articles and their quantities that are sold in a year.

Table 2.1: *Statistical analysis of five stores during one year*

Stat-Store	**1**	**2**	**3**	**4**	**5**
Transactions	147037	180626	44446	37776	28456
Receipts	43406	38376	10094	8595	7619
Returns	9,25%	9,67%	11,43%	9,89%	9,33%
Members	5509	6381	1375	1152	16
Mem. Rec	16,02%	14,14%	18.12%	22,33%	0,56%
Avg. Price	762,49	772,32	665,2	575,93	409,62
Std. Price	494,52	514,51	459,05	616,74	416,36

2.4.1 Statistical Analysis

The store one sample contains 147 037 records of transactions. Note that this does not mean receipts, as a single receipt can produce several records. The retailer runs a fidelity program that allows customers to register their purchases. From this one store we identified 5509 unique members that made at least one purchase during the period resulting in 16,02% of the receipts. This means that the majority of receipts belongs to unidentified customers. However in all these records we can identify the item(s), sales price and the sales clerk.

We extracted statistical information, presented in table 2.1 and plotted in figure 2.1 which represents the sales summary per day and figure 2.2 which shows the number of customers per day.

Some observations that stand out in the data set:

- There were 67 receipts where the customer did not pay anything for the item, it means that the discount was 100% without returning any other article to the store. This could possible be due to a fraud, and when investigated could be used for injecting malicious behaviour.
- It was very rare for a customer to buy the same article more than once in the same purchase, this happened only three times during the year.

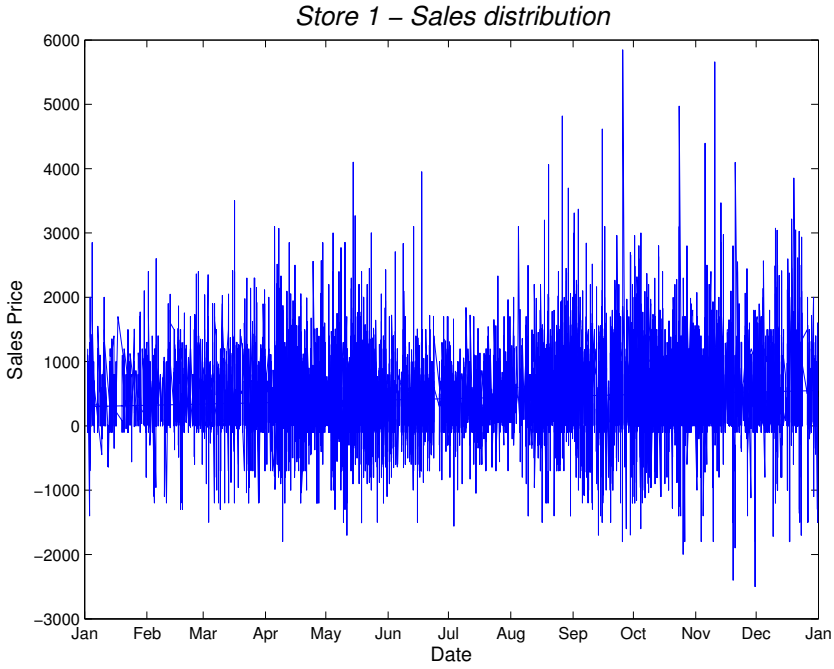


Figure 2.1: *Store one - sales distribution*

We then investigated the performance of the staff. We divided the sales staff into three categories: *top*, *medium* and *low*. *Top* refers to staff that works regularly at the store. *Medium* refers to seasonal staff that works usually for a period between one and three months. Finally *Low* refers to staff that worked for less than one month. Table 2.2 shows the distribution of frequencies found in the data. Top sale clerks work an average of 66% of the time at the store, and they are only 22% of the total number of sales staff.

2.4.2 Network Analysis

Fraud has traditionally had a strong association to network analysis. Due to the possibility of several actors participating in a specific fraud in order to confuse the investigators and dilute the evidence. Another advantage of

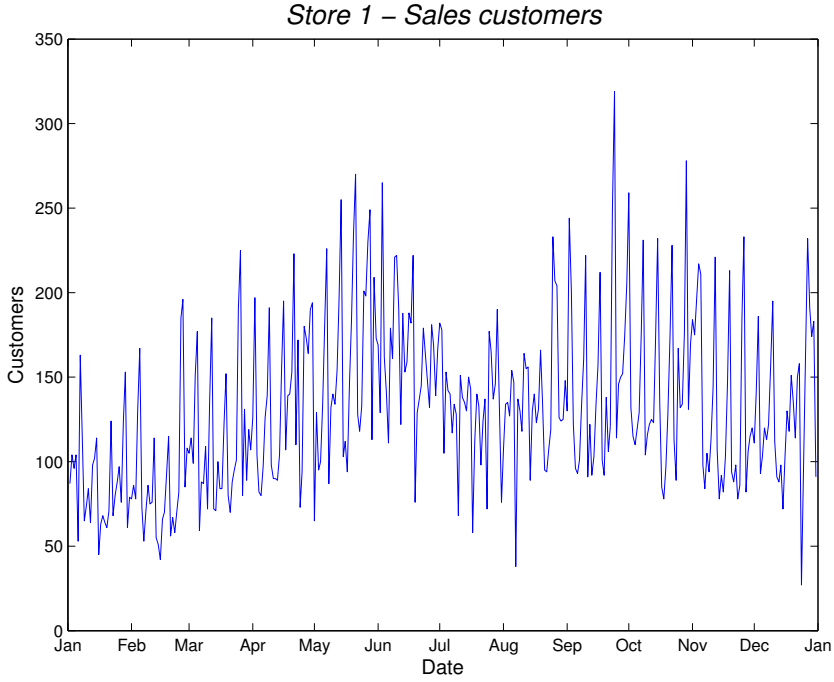


Figure 2.2: *Store one - number of customers per day*

Table 2.2: *Sales clerk frequency*

Type	**Avg. Days**	**Avg. Cust**	**Std. Cust**	**Quantiy**
Top	155,75	45,43	28,17	22,22%
Med	63,20	38,97	23,83	11,11%
Low	13,57	33,93	16,68	66,67%

a network analysis is the ability to visualize the network by using different layout algorithms such as *Force Atlas* or *Yifan Hu* [24]. In this project we used the *Gephi* software, that does network analysis and allows the use of different layout algorithms for the visualization of the network [10].

We can create a network based on the interactions between each of the

Table 2.3: *Article categories*

Category	**Probability**	**Rank**
Top	0,2705	+1000
High	0,2122	100-999
Medium	0,1109	20-99
Low	0,3495	3-19
Unfreq	0,0569	1-2

sales clerks and their respective customers. For the weight of the edges we use the total sales price with respect to each customer. Figure 2.3 shows one way to visualize the sample data extracted from the database using *Yifan Hu* layout.

The network topology resembles a hub topology, where the sales clerks are the central nodes of the hubs, and a few customers that have been helped by more than one sales clerk act as bridges between the hubs.

The store one sample contains 5545 nodes where 36 of them are sales staff, with the rest being customers. The network contains 6120 edges that connects the sales staff and customers. Each edge weight represents the total amount of purchases per customer. Table 2.4 show more information about the network used for calibrating the simulation.

Table 2.4: *Network Analysis*

Statistic	**Store one**
Nodes	5.545
Sales Clerks	36
Customers	5.509
Avg. Degree	1.104
Diameter Undirected	10
Avg. Path Undirected	3.98

Figure 2.3 shows a visualization of the network for the store, the size of the nodes is determined by the out-degree of the sales clerks. The number

inside the nodes also represent the number of customers that were helped by the sales clerk. The In-degree distribution can be better visualized in figure 2.4.

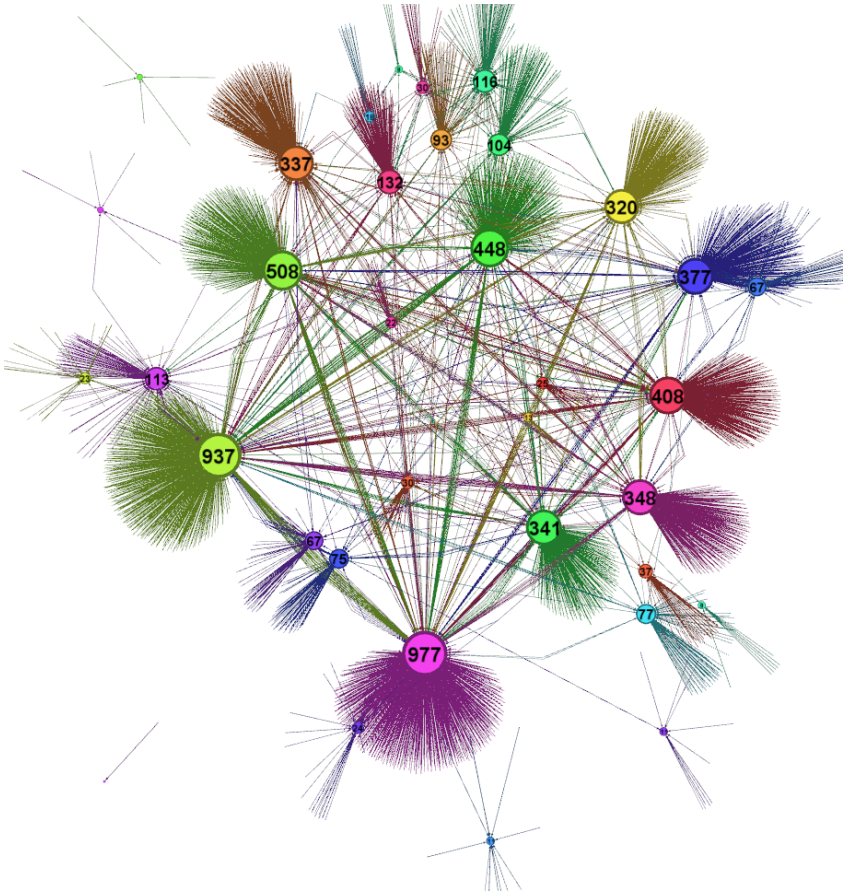


Figure 2.3: *Store one - Network of customers and sales clerks*

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

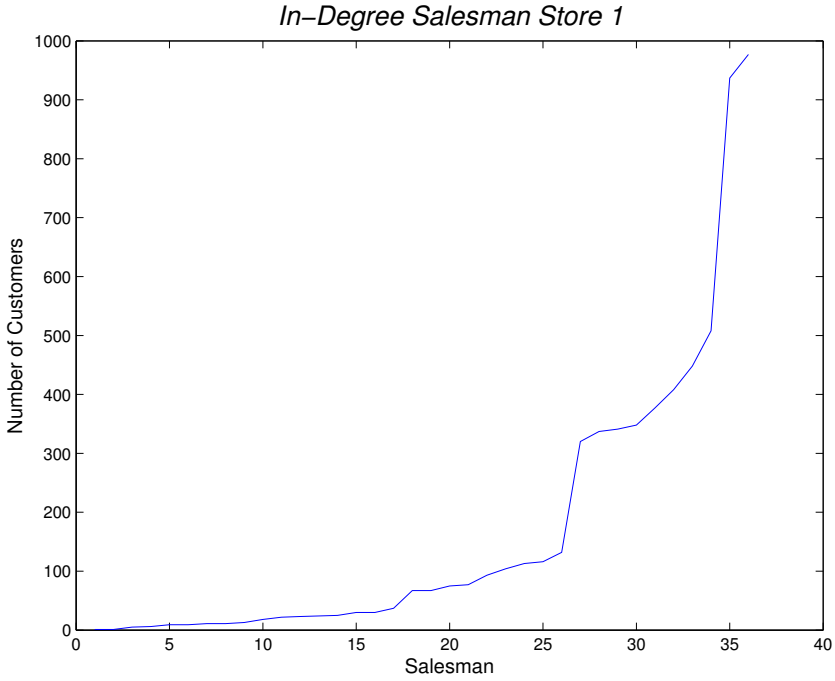


Figure 2.4: *Store one - Customers per sales clerks*

2.5 Model and Method

The design of RetSim was based on the ODD model introduced by [23]. ODD contains 3 main parts: *Overview*, *Design Concepts* and *Details*.

2.5.1 Overview

2.5.1.1 Purpose

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 to find diverse and enough cases of fraud in a real data set. However this is not the case of a simulated

environment, where fraud can be injected following known patterns of fraud.

2.5.1.2 Entities, state variables and scales

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 with shopping.

2.5.1.3 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 that each step represents 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 (see figure 2.2).

2.5.2 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.

2.5.3 Details

2.5.3.1 Initialization

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.

2.5.3.2 Input Data

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.

2.5.3.3 Submodels

Figure 2.5 shows the different use cases of the agents. This model represent the different actions that an agent can take inside the system.

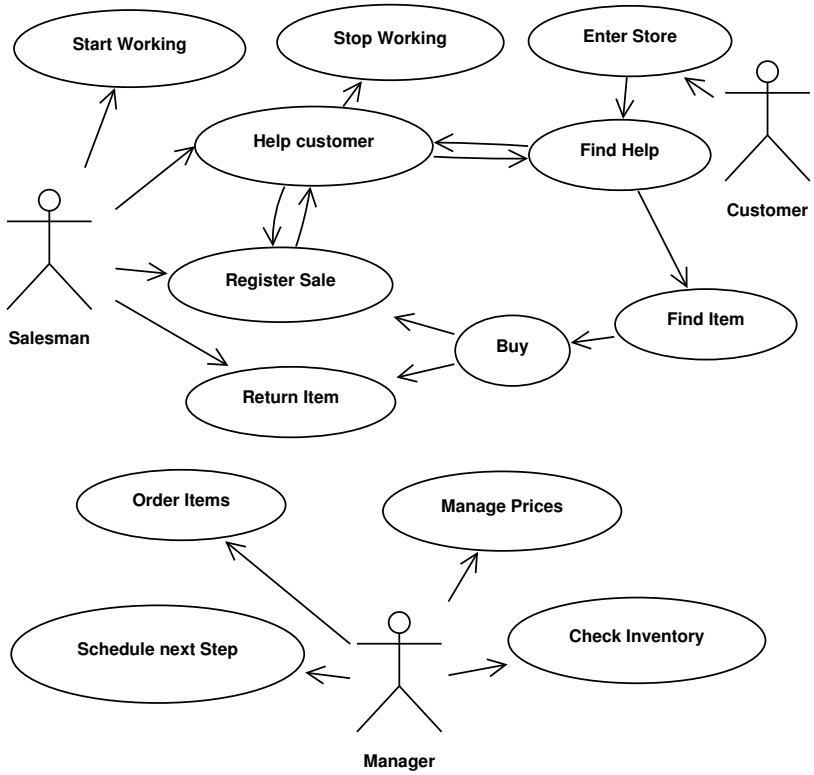


Figure 2.5: *RetSim Use Case Diagram*

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 (see table 2.3). 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.

2.6 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 [48] report. As can be seen in section 2.5, 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.

2.6.1 Sales cancellations

This scenario includes cases where the sales clerk cancels some of the items in the sale without telling the customer, i.e., the customer pays the full sales price, and the sales clerk keeps the difference. In terms of the object model used in RetSim the sales cancellation 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 normal cancellations in the RetSim model. Fraudulent sales clerks will perform normal cancellations, as well as fraudulent once. The volume of fraudulent cancellations can be modelled using a sales clerk specific parameter. The "red flag" for detection will in this case be a high number of cancellations for a sales clerk with a low number of average sales.

2.6.2 Refunds

This scenario includes cases where the sales clerk creates fraudulent refund slips, keeping the cash refund for him- or herself. 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.

2.6.3 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 keeps 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. Fraudulent sales clerks will perform 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.

2.7 Results

RetSim uses the Multi-Agent Based Simulation toolkit MASON which is implemented in Java [43]. MASON offers several tools that aid the development of a MABS. We justified our choice mainly for the benefits of supporting multi-platform, parallellization, good execution speed in comparison with other agent frameworks; which is specially important for computationally intensive simulations such as RetSim [54]. RetSim can be run with GUI, that helps the user see the states and relations between the sales clerks (bigger circles) and customers, as can be seen in the example in figure 2.6.

In RetSim we do not make any distinction between customers that are part of the membership programme or not. RetSim assumes that all the customers are members. This give us a way to track individual behaviours of all customers, which is beneficial.

The output of RetSim is a CSV file that contains the fields: *Step*, *Type* of *Transaction* (e.g. one sale, three returns), *Customer Id*, *Sales Clerk Id*, *Sales Price*, *Item Id* and *Item Description*.

2.7.1 Scenarios simulated

We aimed to perform a simulation that would produce a comparable data set to our sample data set which contained 36 sales clerks and around

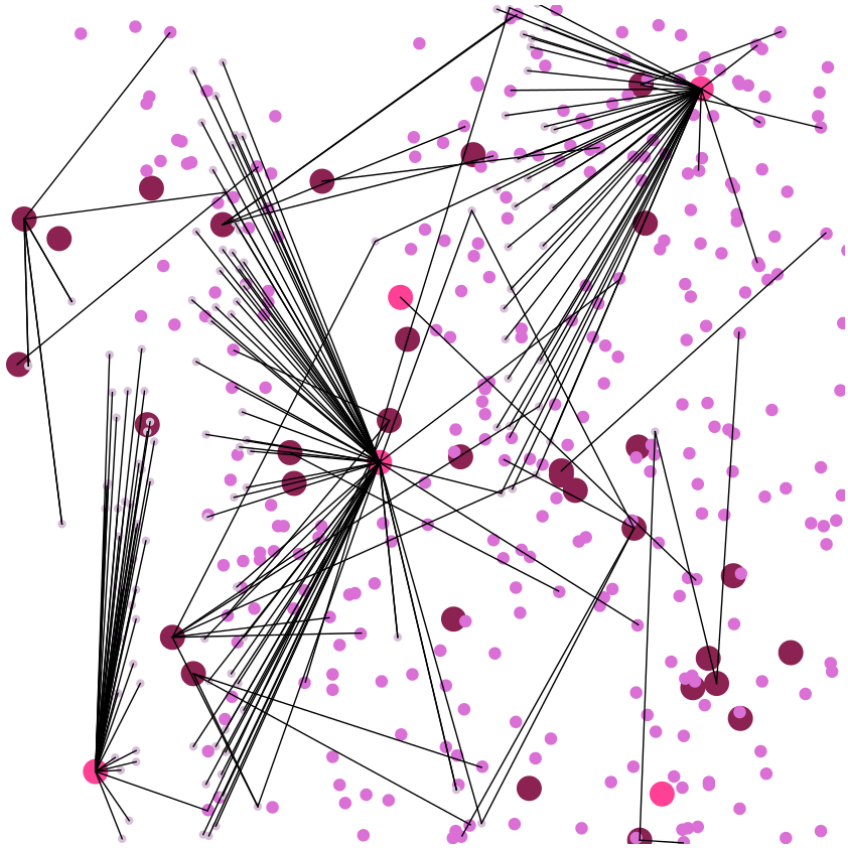


Figure 2.6: *Screenshot of RetSim during a step*

45000 receipts and 81500 articles sold. The simulation was loaded with a subset of about 11000 articles from the real store.

We ran RetSim for 361 steps (working days of the store), several times and calibrated the parameters given in order to obtain a distribution that get closer enough to be reliable for testing. We collected several log files and selected three from the latest runs. Table 2.5 compares three runs of RetSim against the original data. Since this is a randomised simulation the values are of course not identical.

Table 2.5: *Statistical Analysis Store one vs RetSim Simulations*

Statistic	**Store 1**	**RetSim1**	**RetSim2**	**RetSim3**
Articles sold	81441	103716	95847	96492
Avg. Sales Price	372.3	405.5	405.2	407.1
Std. Sales Price	510.9	555.1	550.7	552.2

2.7.2 Social Network Calibration

We experimented with calibrating our results and aim to simulate the network presented in section 2.4.2. Our aim was to obtain approximately the same amount of nodes and edges. We used the out-degree distribution to associate sales clerks with customers. So each sales clerk is capable to handle more or less customers during each step of the simulation and this creates the difference between nodes. This difference is interpreted in the real world by two parameters. The first is how many days a sales clerk work and the second is how good sales clerks they are. Accordingly, we only allow sales clerks with a high *in-degree* to be active during most of the steps. It means that we deactivate some sales clerks during any one specific step.

After several experimental runs and around 180 steps, keeping the most of the parameters from the original simulation, we selected one of the simulation runs to show in table 2.6.

2.7.3 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [51]. The verification ensures that the simulation correspond to the described model presented by the chosen scenarios. We described RetSim in section 2.5. In our model, we have included several characteristics from a real store, and successfully generated a distribution of sales that involved the interaction of sales clerks and customers. However, there are a few characteristics left from the real model such as discounts.

Table 2.6: *Network Simulated*

Statistic	**RetSim**
Nodes	4948
Edges	5339
Sales Clerks	36
Customers	5303
Avg. Degree	1.079
Avg. Weighted Degree	499.1
Modularity Undirected	0.845
Diameter Undirected	8
Avg. Path Undirected	4.19

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 some generated distributions of sales that are comparable visually in figure 2.8, 2.9 and 2.10.

Figure 2.8 shows a comparison of RetSim and the real sample data extracted from store one. We note several things: first the shape of the distributions look similar. Before zero are all the returns with a shape of a flat normal distribution. Between zero and 100 are the most frequently sold items such as shoe laces or accessories, which produces a peak. After 100 and before 2000 is the most common rank for shoes, so it presents another part of the distribution that contains the mean.

Figure 2.9 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.

In figure 2.10 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.

Now we will focus on evaluating the simulated network presented in

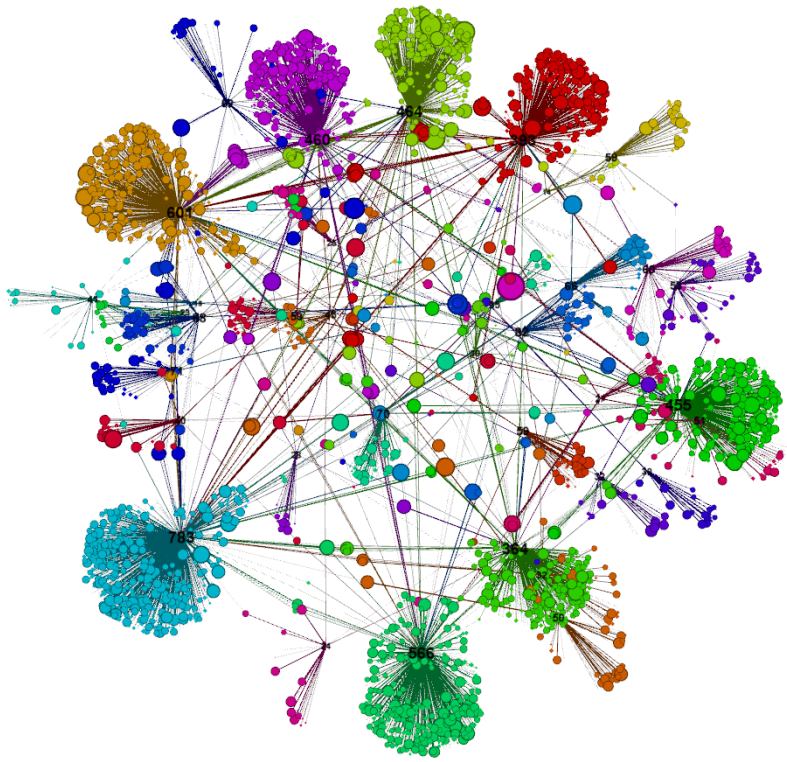


Figure 2.7: *Small Simulated network*

section 2.7.2. The simulation in comparison with the original data seems visually very similar. There are similarities between the hub topology, number of nodes, and sales clerks. However we also find some dissimilarities between the weighted average degree, which in the simulation was below the original data.

There is more homogeneity between the purchases of the customers in the original data than in the simulated data. This could be due to the random nature of the selection of items in the simulation. Notice the visual differences between figure 2.3 and 2.7.

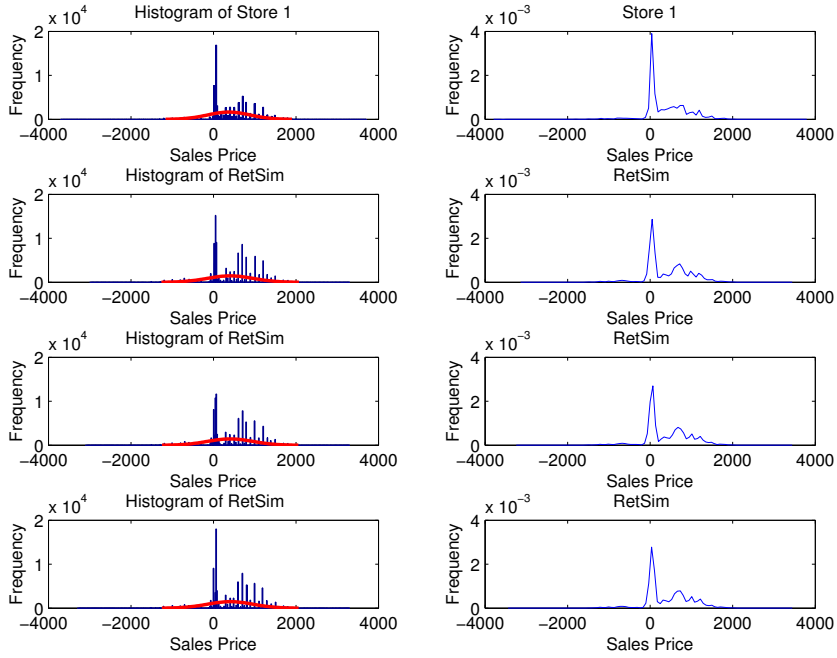


Figure 2.8: *Comparison of simulated vs real data*

Another difference that we found is that the simulated network generates one single giant component. In the original data we could perceive a few sales clerks that perhaps just worked there for a single/few days and only served few customers. Those sales clerks are identified as islands and separated components. The analysis of these islands might be of interest for fraud detection.

We can also look at the modularity of the simulated network as an emerging behaviour of the customers. Both, the original and the simulated network are very similar and build their communities around the sales clerks. This can be clearly visualized by the different colours used in all the visualizations.

So in summary, our agent model with its programmed micro behaviour,

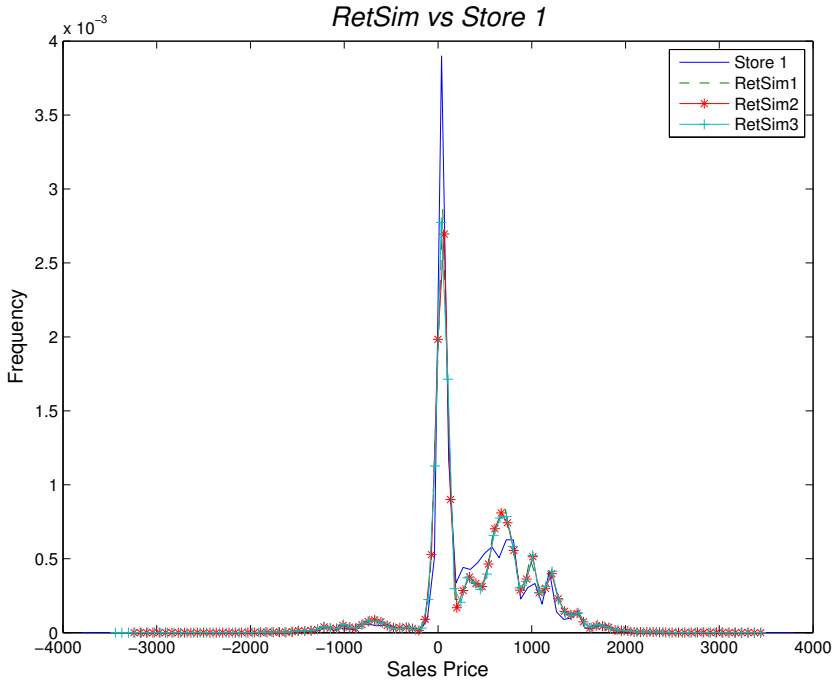


Figure 2.9: *Comparison of distribution of simulated vs real data*

produces the same 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.

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.

2.8 Conclusions

RetSim is a simulation of a retail shoe store with the objective to generate a sales data set that can be used for research into fraud detection. Syn-

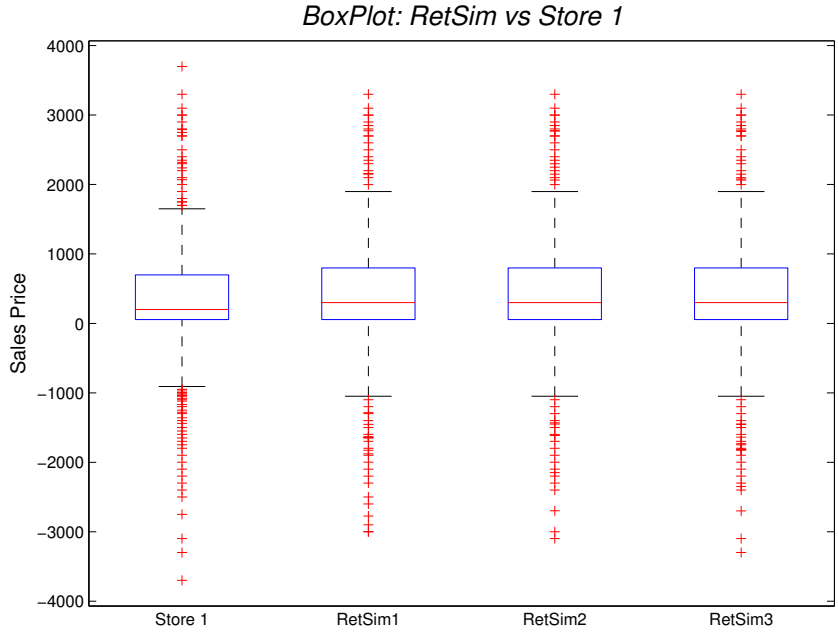


Figure 2.10: *Box plot of simulated vs real data*

thetic data sets generated with RetSim can aid academia, companies and governmental agencies to test their methods or to compare the performance of different methods under similar conditions on the same test data set.

In section 2.3 we formulated our research question for this paper: *How could we model and simulate a retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?* In section 2.5 we presented the RetSim model, 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 2.7) and we evaluated and verified our model in section 2.7.3.

It is important to know how much information from the real data set is contained in the generated synthetic data. First we do not keep any record of who is purchasing anything in the store, we based our simulation purely

on statistical measures and network measures that give us an approximate description of how the individual agents behave. This means that the retail store can be sure that the privacy from the customers is preserved when using RetSim.

We argue that RetSim 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 sales clerk returning stolen shoes or unusually low productivity of a sales clerk during a specific day which could mean that the clerk is not entering some of the receipts into the system. We will make a stable released of RetSim available to the research community together with standard data sets developed for this article and further research.

For future work we plan several improvements of and additions to the current model. RetSim can be calibrated to improve the results presented in section 2.7 and make the data set more realistic.

In order to generate records with malicious behaviour we plan to extend RetSim to also generate malicious activity that can come from the sales clerk, customer or even the managers, or combinations of these.

Among the additions we consider are: inventory control, discounts and promotions that affect the demand of certain products. We can also add hidden parameters to sales clerks such as skills in sales, which will increase the number of customers and the average cost of items purchased. Another possible inclusion in future versions is an interesting behaviour, the self transaction, where a sales clerk can play the role of a customer and a sales clerk at the same time. This behaviour can play a key role in order to find cases of fraud.

3

Using the RetSim Simulator for Fraud Detection Research

Edgar Alonso Lopez-Rojas, Dan Gorton and Stefan Axelsson

Abstract

Managing fraud is important for business, retail and financial alike. One method to manage fraud is by *detection*, where transactions etc. are monitored and suspicious behaviour is flagged for further investigation. There is currently a lack of public research in this area. The main reason is the sensitive nature of the data. Publishing real financial transaction data would seriously compromise the privacy of both customers, and companies alike. We propose to address this problem by building RetSim, a multi-agent based simulator (MABS) calibrated with real transaction data from one of the largest shoe retailers in Scandinavia. RetSim allows us to generate synthetic transactional data that can be publicly shared and studied without leaking business sensitive information, and still preserve the important characteristics of the data.

We then use RetSim to model two common retail fraud scenarios to ascertain exactly how effective the simplest form of statistical threshold detection could be. The preliminary results of our tested fraud detection method show that the threshold detection is effective enough at keeping fraud losses at a set level, that there is little economic room for improved techniques.

Keywords: Privacy; Anonymization; Multi-Agent-Based Simulation; MABS; ABS; Retail Store; Fraud Detection; Synthetic Data

3.1 Introduction

Fraud is an important problem in a number of different situations. The economic impact can be substantial. For example, in one recent case the major US home improvement chain *Home Depot* was the target of a fraudulent return scam where two perpetrators netted several thousand dollars before being caught [19]. Return fraud, i.e. the defrauding of a retail merchant by abusing the return process, alone is estimated to cost US retailers about 9 billion dollars yearly. To further illustrate the seriousness of the problem and try and combat it both EU and US recently started to mandate the use of fraud detection as one part of the minimum security requirements for financial services [16, 17].

However, in order to investigate, develop, test, and improve fraud detection techniques, there is a need for detailed information about the domain, its peculiarities and especially publicly available transaction data so that different approaches can be compared and contrasted.

For a multitude of reasons (e.g., privacy related, legal, financial, or contractual) the state of practice in research is to work with sensitive and hence secret data. Anonymization techniques are often not considered sufficiently effective, the risk of leakage is difficult to calculate, and furthermore, anonymization is difficult to perform effectively on large data sets with a high degree of certainty of coverage.

In this article we present a novel way of creating realistic fraud research data by developing a simulation, primed by real data, which enable us to share data with the research community, without exposing potentially sensitive information. Fraudulent behaviour is added and the resulting model is used to test if a simple threshold based detection technique is sufficient to keep fraud loses below a set threshold. This is often sufficient in a business setting. If the risk of fraud can be managed (i.e. a fraud

detection system can guarantee that fraud will stay below a reasonable level), the resources and efforts that would have gone to insure against the fraud risk can be put to better, more productive use, elsewhere in the organisation.

We base our model on historical transaction data provided by one of the largest Scandinavian shoe retailers. This data contains several hundred million records of diverse transactional data that is sufficiently recent to reflect current conditions, but sufficiently old to not pose a serious risk from a competitor analysis standpoint. (A risk our retail data providers tell us is exaggerated anyway, at least in regards to their business).

Since we have access to transaction data pertaining to shoe retailing, we developed a simulator called *RetSim*, a **Ret**ail shoe store **Sim**ulator, built on the concept of Multi-Agent-Based Simulation (MABS) that simulates the normal operation of a shoe store. We then extended RetSim to include simulation of fraud scenarios. RetSim is intended to be used for developing and testing fraud scenarios in a shoe retail store, while keeping business sensitive and private personal information about customers secret from competitors and others. However, as the model is focusing on the *salesman*, *customer* relation, we expect that it should be applicable to other retail settings. Our aim was to make the model sufficiently general to be applicable to other domains like online financial services, i.e. any number of systems dominated by handling many small transactions. (It should be noted that we would prefer to use the gender neutral term *sales clerk*, but as literature in the field use *salesman* exclusively, we have decided to follow that usage.)

The defence against fraud is an important topic that has seen some study. In the retail store setting the cost of fraud is of course ultimately transferred to the consumer, which ultimately impacts the overall economy. Our aim with the research leading to RetSim is to learn the relevant parameters that govern the behaviour in, and of, a retail store in order to simulate *normal* behaviour. We then add simulation of malicious behaviour and detection. However, our models of fraud are not yet as advanced as our

normal models. As fraud in the retail setting is usually perpetrated by the staff that is our focus. Examples of such fraud are, e.g. *sales cancellations* where the salesman does not tell the customer, pocketing the difference, or *refunds* where the salesman creates fraudulent refund slips and keeps the cash refund. *Coupon reductions/discounts* can also be applied to the sale without telling the customer. In many of these cases the fraud is simplified if the customer is in cahoots with the fraudster, as the risk of being detected by an alert customer is eliminated.

One of the main contributions of this article is a method to generate anonymous synthetic data of a "typical" retail store, that can then be used as part of the necessary data for the research, development and testing of fraud detection techniques, both research prototypes and commercially available systems. Our approach aims to provide researchers with a tool that generates reliable data with which to experiment with different fraud detection techniques and enable later comparison with other approaches, something that is not possible today. Another contribution is the result that threshold based detection seem to be sufficient. This is interesting in that it might be used to explain why the majority of fraud detection systems and procedures that are in actual use are based on this simple principle. It also give us a limit of how much money can be spent on more advanced, and more expensive, techniques, given the diminishing returns of these as the majority of fraud can be detected using much simpler and cheaper techniques.

In addition, simulation also have other benefits. It can produce more data much faster and with less cost than for instance; collecting data, and trying different scenarios of fraud, detection algorithms, or personnel and security policy approaches, in an actual store. The latter also entails additional risks, e.g., incurring the wrath of angered staff, due to testing, an ill-advised policy, which may lead to even greater expense and unwanted problems.

3.2 Related Work

Simulations in the domain of retail stores have traditionally been focused on finding answers to logistics problems such as inventory management, supply management, staff scheduling and customer queue reductions [14, 15, 57]. We find no research focusing on simulations generating fraud data to be used for fraud detection in retail stores. Therefore, we recently introduced RetSim with the purpose of fraud detection research. In this article we extend RetSim to study specific fraud scenarios, including agents using known fraud behaviour patterns [40].

Anonymization techniques have been used to preserve the privacy of sensitive information present in data sets. But de-anonymizing data sets is not an insurmountable task, far from it [49]. For this reason we have decided to use simulation techniques to keep specific properties of the original data set, such as statistical and social network properties, and at the same time providing an extra layer of insulation that pure anonymization does not provide.

There are tools such as IDSG (IDAS Data and Scenario Generator [29]) that were developed for 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 the testing phase, and it has been used to test fraud detection systems. Our approach differs in that we are implementing an agent-based model which is based on agent micro behaviour rather, than a fixed statistical distribution of macro parameters.

With the current popularity of social networks, such as *Facebook*, the topic of Social Network Analysis (SNA) has seen interest in the research community [4]. Social Network Analysis is currently being combined with *Social Simulation*. 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 addressing the question of customer/salesman-interaction, that we are aware of.

Other methods to generate the necessary fraud data have been proposed by [2, 18, 26, 44, 63]. The work by [63] lets the user specify the assumptions about the environment at hand; i.e., there is no need for access to real data. However, this will certainly affect the quality of the synthetic data. The work by [44] makes use of a small sample of real data to generate synthetic data. This approach is similar to ours. However, the direct use of real data to prime the generation of synthetic data is limited in that it makes it harder to generate realistic data with other characteristics that those of the original real data [63]. The work by [26] focused on privacy-preserving methods for data mining. However, that method also does not have the possibility of generating realistic data with other characteristics than those of the original data. In our work, we use social simulation, which makes it possible to change the parameters of the agents in the model to create realistic synthetic data, potentially producing emergent behaviour in the logs which is hard to produce in other ways.

Previous research on fraud detection algorithms has showed that data mining and machine learning algorithms can identify novel methods of fraud by detecting those records that are different (anomalous) in comparison with benign records, e.g., the work by [53]. This problem in machine learning is known as *novelty detection*. Furthermore, supervised learning algorithms have been used on synthetic data sets to prove the performance of outlier detection [1] [44]. However none of these studies made use of synthetic data from retail stores. To our knowledge, there has been no investigation of what are the limits of effectiveness of e.g. simple threshold based monitoring.

3.3 Research Questions

For clarification we summarise our research questions thus:

RQ How can we model and simulate a retail shoe store to obtain realistic synthetic data set for the purpose of fraud detection? Specifically:

RQ1 How do we evaluate, verify and validate our simulation model?

RQ2 Is the generated data set properly anonymized so that no sensitive information leaks?

RQ3 Is threshold detection sufficient to keep the losses from fraud at manageable level?

3.4 Analysis of the Retail Data

To better understand the problem domain, especially the normal operation of a store, we performed a data analysis of the historical data provided by the retailer. We were interested in finding necessary and sufficient attributes that enable us to simulate a realistic scenario in which we could reason about and detect interesting cases of fraud.

Due to a lack of space we will focus our presentation of the analysis on one of the biggest stores by sales volume, that we named *store one*. *Store one* is relatively richer in data than smaller stores.

We took a sample comprising the sales for one year. From this sample we selected the transaction tables that detail cash flows and the article inventory, which gave us a good idea of how many transactions a big store produces in a year and how many different types of articles and their quantities that are sold in a year.

3.4.1 Statistical Analysis

The *store one* sample contains 147037 records of transactions. Note that this does not necessarily mean receipts, as a single receipt can produce several transaction records. The retailer runs a fidelity program that allows customers to register their purchases. From this one store we identified 5509 unique members that had made at least one purchase during the period which accounted for 16% of the receipts. This means that the majority of receipts belong to unidentified customers. However in all these records we can still identify the item(s) sold, the sales price and the salesman.

We then investigated the performance of the staff. We divided the sales

staff into three categories: *top*, *medium* and *low*. *Top* refers to staff who work regularly at the store. *Medium* refers to seasonal staff who usually work for a period between one and three months. Finally, *Low* refers to staff who work for less than one month.

Top salesmen work an average of 66% of the time at the store, making up only 22% of the total number of sales staff.

3.4.2 Network Analysis

Fraud analysis has traditionally been heavily associated with network analysis. This is because of the possibility of several actors colluding in a specific fraud in order to confuse the investigators and scatter the evidence.

In this paper we develop a multi-agent simulation where the micro behaviour of the different agents together give rise to a macro behaviour that is close to the real observed behaviour at the store. Hence, to verify that our model is realistic, we need to study the behaviour of the real actors in the store. To show the networks occurring in the real data, we visualize them using *Gephi*, a tool that can visualize networks using different layout algorithms [10].

In our case, the interactions between each of the salesmen and their respective identifiable customers (members) describe a network. We use the total sales price with respect to each customer as the weight of the edges.

Figure 3.1 shows one way to visualize the sample data extracted from the database using *Yifan Hu* layout [24]. The network topology resembles a hub topology, where the salesmen are the central nodes of the hubs, and a few customers that have been helped by more than one salesman act as bridges between the hubs. The *store one* sample contains 5545 nodes where 36 of them are sales staff, with the rest being customers. The network contains 6120 edges that connect the sales staff and customers. Each edge weight represents the total number of purchases per customer. Table 3.1 shows additional information about the network used for the subsequent

calibration of the simulation.

Table 3.1: *Network Analysis*

Statistic	**Store one**
Nodes	5545
Edges	6120
Salesmen	36
Customers	5509
Avg. Degree	1.104
Avg. Weighted Degree	829.3
Modularity Undirected	0.822
Diameter Undirected	10
Avg. Path Undirected	3.98

Figure 3.1 shows a visualization of the network for the store, the size of the nodes is determined by the weighted out-degree of the customers. The number inside the salesman nodes represent the number of customers that were helped by each salesman. The in-degree distribution is used in the simulation to reflect the number of customers that a certain type of salesman usually serves.

The network analysis generates many useful statistics for our modelling. One interesting observation is that 90.26% of the members have been helped by only one salesman, as calculated by the out-degree distribution.

3.5 The Model and Simulator

RetSim uses the MABS toolkit MASON which is implemented in Java [43]. MASON offers several tools that aid the development of a MABS. We selected MASON because it is: multi-platform, supports parallelisation, and fast execution speed in comparison with other agent frameworks. This is especially important for computationally intensive simulations such as RetSim [54].

Our aim was to produce a simulation that produces synthetic trans-

Figure 3.1: Store One - *Network of Customers and Salesmen*

actions that is statistically similar to transactions from a real retail store. However, as in all simulations, we have to select a subset of the real world, which captures the aspects that we are interested in modelling.

3.5.1 Model

In the retail scenario, we have many different actors that interact and this interaction produces the emergent transaction behaviour that we can observe in the transaction history. Thus, a multi-agent-based model (MABS) seemed the natural choice. Additionally, MABS have been successfully used to represent complex social interactions in other scenarios which adds to its attractiveness [52].

Our model contains the following entities and behaviours. The *Store* is the main entity of the simulation, it contains all the variables and states required to run the simulation such as: *Salesmen*, *Customers*, *Products*, *Frequencies* and other parameters used to calibrate the model.

In this work, we chose to model three main actors, that is agents, who we argue capture the important interaction patterns. These agents are: *Manager*, *Salesman* and *Customer*.

Manager This agent reads parameters from the *Store* to decide about next step of the simulation by predicting the demand for products and customers, and scheduling the working days of salesmen.

Salesman The salesman agent is in charge of promoting items for sale, and issues the receipt after each sale. A salesman is in state *busy* when it is serving the maximum number of customers it can handle.

Customer The behaviour of a customer agent is determined by a goal function that tells it to purchase one or several items. A customer is in an active *need-help* state when no salesman is assisting with shopping.

During a single step of the simulation a customer is instantiated and a salesman 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 probability $1 - p$. Each step represents a day of sales. Hence, 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 number of customers per day.

The *basic principle* of this model is the concept of a commercial transaction. There is an *emergent* social network from the relation between the customers and the salesmen. Each of the customers has the *objective* of purchasing articles from the store. The *objective* of the salesman 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 generates a supply of customers for the next day and activates or deactivates specific salesmen in the store. In our virtual environment the *interaction* between agents is always between salesman–customer. The purchase of articles from another customer or selling articles to a salesman is not permitted. Customers and salesmen can scan the store surface in any direction for salesmen, or customers, and seek 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 estimated from the real data set.

The in-degree distribution is used as an indication of how good a salesman performs. Each salesman is assigned an in-degree value that affects each step of the simulation when the salesman searches for customers in need of assistance. The larger their in-degree, the more customers they can help and it also increases their scope of search.

RetSim is parametrised by the probability distributions for scheduling salesmen, the items that can be purchased, and for different statistical measures concerning the customers. A CSV-file which contains an identifier, description, price, quantity sold, and total sales specify these inputs. We use a parameter file, which is loaded when the simulation starts, for setting the parameters, including the name of the CSV-file. The parameters can also be set manually in the GUI.

We initially load the complete article list from the store and generated categories according to their sales frequency as shown in table 3.2. This table was used during calibration to estimate the article selected by each of the customers during the sales operation.

Table 3.2: *Article categories*

Category	**Probability**	**Rank**
Top	0.2705	+1000
High	0.2122	100-999
Medium	0.1109	20-99
Low	0.3495	3-19
Unfreq.	0.0569	1-2

Figure 3.2 shows the different use cases of the agents. This model represents the different actions that an agent can take in our simulated retail store. Agent *Salesman* is activated after the *start working* use case. A *customer* can either be offered help by a *salesman* or find some available *salesman* to *purchase an item*. The Customers *find available items* in the inventory and the salesman *apply any discounts* if applicable. After the transaction takes place (*register sale*) a customer can decide to *return an item*.

RetSim does not make any distinction between customers that are part of the membership program and customers that are not. RetSim assumes that all the customers are members. This enables us to track individual behaviour of all customers, which is useful as it is not possible in the raw data. RetSim flags all the transactions that involves malicious behaviour, i.e. that involves an agent that we have assigned to perform malicious behaviour and label them as fraudulent.

The output of RetSim is a CSV-file that contains the fields: *step*,*receipt*,*type* of *Transaction* (e.g., 1=sale, 3=returns, 6=discount), *customer Id*, *salesman Id*, *sales price*,*sales price before discount Item Id*, *Item Description* and *Fraud Flag*. RetSim can also generate an ARFF-file that can be used as input for the Weka machine learning system.

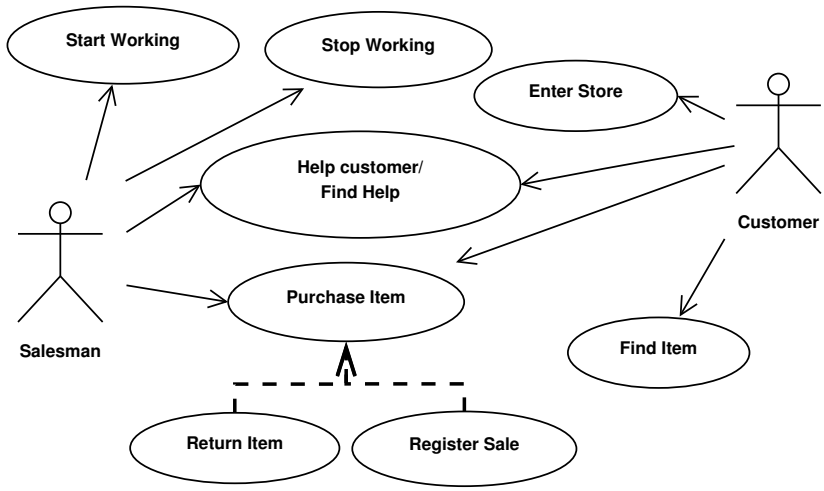


Figure 3.2: *Use Case Diagram for the Interaction of Agents in RetSim*

We are also interested in studying the social network interactions between the customers and the salesmen. For this, we produce another CSV-file that represents the edges of the social network described by the customers and the salesmen, with the weight of the edges given by the *sales price*. We also add labels for *Type of Transaction* and *Fraud Flag* which is used to identify fraudulent transactions.

3.5.2 Simulated Scenarios

Our aim was to produce a simulation that would result in data comparable to our real data set. This contained 36 salesmen and around 45000 receipts and 81500 articles sold. The simulation was seeded with a subset of about 11000 articles from the real store. Figure 3.3 shows a visualization of the generated social network between customers and salesmen.

To obtain a simulation that was sufficiently close to the real data, we ran multiple runs of RetSim for 361 steps (one per working day) and calibrated the simulation by performing adjustments to the parameters. Each simulation was then compared to the real data using the one-way

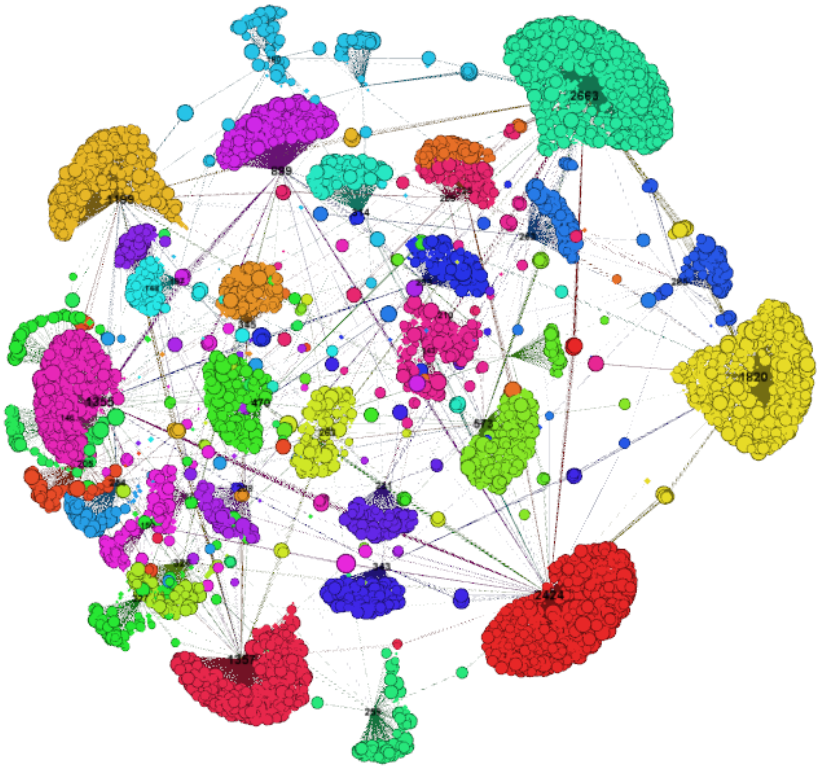


Figure 3.3: *Visualization of Simulated Network*

ANOVA statistical test. From this, we selected the top two simulations that scored better for the selected statistical test (see section 3.6.1).

In the following figures, the labels *rs3658* and *rs5125* correspond to each simulation. Table 3.3 compares the selected simulations against the real data from *store one*. Since this is a randomized simulation the values are of course not identical, nor should they be.

Table 3.3: *Statistical Analysis of* Store One *vs RetSim Simulations*

Statistic	**Store one**	*rs5125*	*rs3658*
Receipts	43406	43610	46881
Items	77186	82358	88668
Returns	4267	8385	9005
Avg Sales Price	372.3	369.8	371.0
Std. Sales Price	510.9	519.7	514.8

3.6 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [51]. 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 statistical tests and evaluated the network topology and parameters to deduce that our simulation is sufficiently similar to perform fraud detection testing.

3.6.1 Statistical Tests

Generated distributions of sales are presented in figures 3.4 and 3.5 for visual comparison with the original data. Figure 3.4 shows *store one* overlaid with the data from the two top simulations generated by RetSim. Visually the distributions do look similar. The shapes of the distributions look similar to the naked eye. The sales prices below 0 represent all the refunds, with a shape of a flat normal distribution. The sales prices between 0 and 100 represent the most frequently sold items, such as shoe laces or

accessories, which produce a peak. The sales prices above 100 and below 2000 represent the most common prices for shoes. While there are several small visual differences between the distributions, the overall similarity is striking.

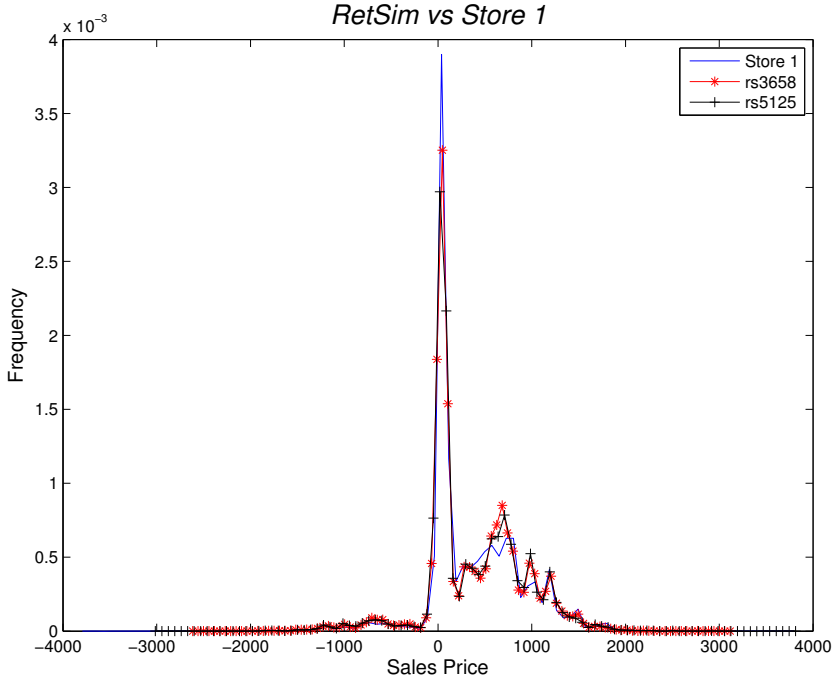


Figure 3.4: *Overlap of Two Runs of RetSim vs Real Data*

However, to sufficiently determine if these visual differences are significant, we performed a one-way ANOVA test to assess the differences between the real and the simulated data. The one-way ANOVA is considered to be robust in this case as it tolerates violations of the normality assumption well. We found that there were no statistically significant differences between group means as determined by the one-way ANOVA test ($F(2, 269854) = 0.5, p = 0.61$).

Figure 3.5 is a box plot comparison of *store one* with the two top

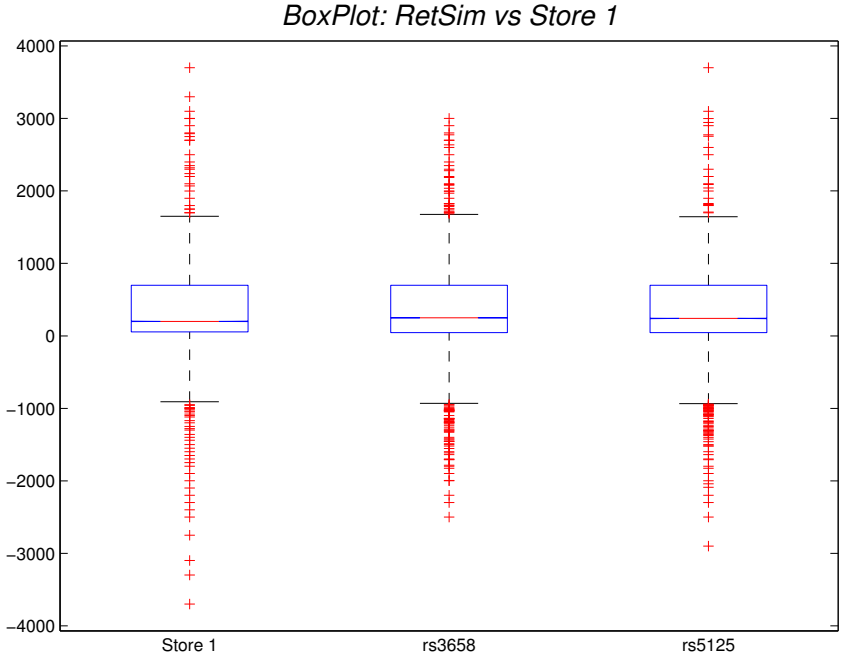


Figure 3.5: *Boxplot of Simulated vs Real Data*

simulations generated by RetSim. We visually corroborate that the five statistical measures provided by the box plot are similar but not identical.

Figure 3.6 shows a Q-Q plot comparison of *store one* with the two RetSim runs. We can see that the central parts of the simulations compare well with the distribution of *store one*. However, as is manifested by the deviating tails, the two simulations lack some of the extreme outliers.

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.

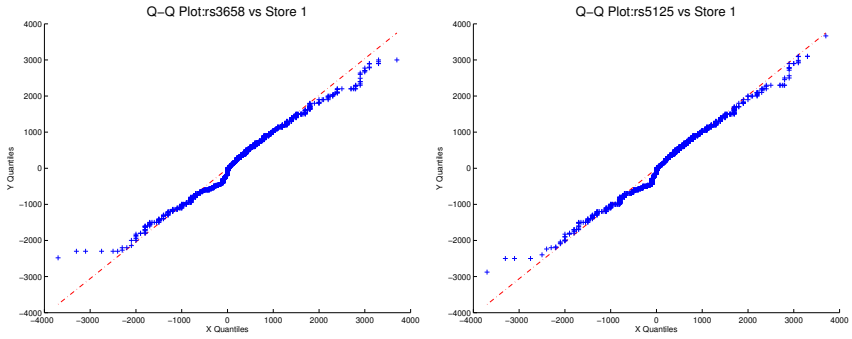


Figure 3.6: *Q-Q plot of Simulated vs Real Data*

3.6.2 Social Network Comparison

We calibrated RetSim to simulate the network presented in section 3.4.2. Our aim was to obtain approximately the same number of nodes and edges. We used the out-degree distribution to associate salesmen with customers. Each salesman is capable of handling more or less customers during each step of the simulation, and this creates the difference between nodes. This difference is measured in the real world by two parameters. The first is how many days a salesman works and second is how good the salesman is. Accordingly, we only allow salesmen with a high *in-degree* to be active during most steps. It means that we deactivate some salesmen during any one specific step.

After several experimental runs and around 180 steps, keeping most of the parameters from the original simulation, we selected one of the simulation runs to show in table 3.4.

The simulation with the real data seems visually very similar compared to the real data. There are similarities between the hub topology, number of nodes, and salesmen. However we also find some dissimilarities between the weighted average degree, which in the simulation was below the real data.

There is more homogeneity between the purchases of the customers in

Table 3.4: *Network Simulated*

Statistic	RetSim
Nodes	4948
Edges	5339
Salesmen	36
Customers	5303
Avg. Degree	1.079
Avg. Weighted Degree	499.1
Modularity Undirected	0.845
Diameter Undirected	8
Avg. Path Undirected	4.19

the real data than in the simulated data. This could be due to the random nature of the selection of items in the simulation. This can be specially seen when comparing both visualizations. Notice the visual differences between figure 3.1 and 3.7.

Some other differences that we found are that the simulated network generates one single giant component. In the original data we could identify a few salesmen that perhaps just worked for a single/few days and only served a handful of customers. Those salesmen are identified as islands and separated components. The analysis of these islands might be of interest for fraud detection in the future.

We can also look at the modularity of the simulated network as an emerging behaviour of the customers. Both, the real and the simulated network are very similar and build their communities around the salesmen. This can be clearly seen by studying the different colours used in all the visualizations.

In summary, our agent model with its programmed micro-behaviour, produces the same type of overall interaction network that we observe in the real data, and furthermore, this interaction network also gives rise to the same macro-behaviour for the whole store as for the real store as well.

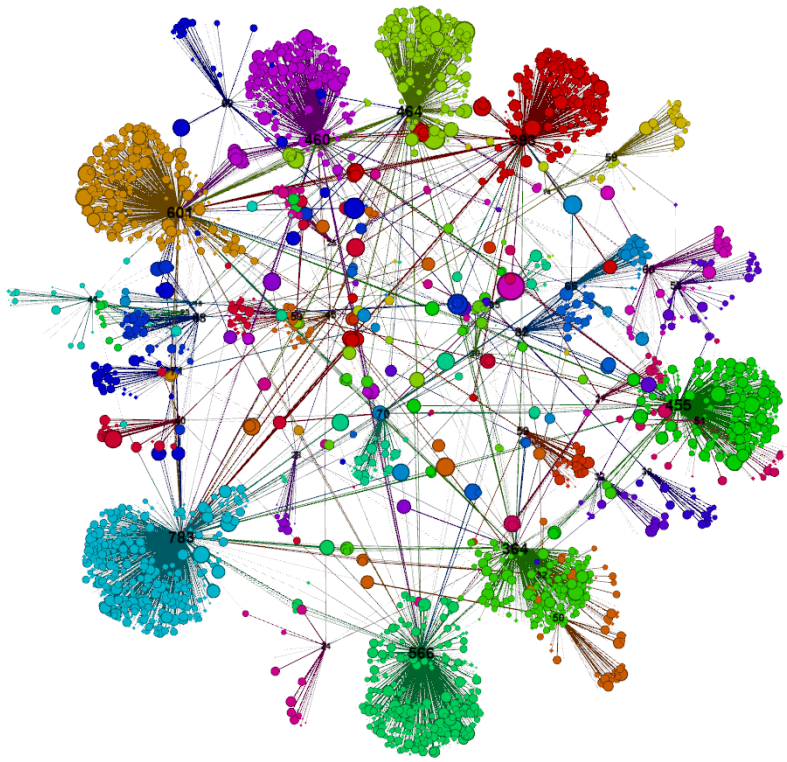


Figure 3.7: *Small Simulated network*

So given these results we declare our simulation a success. Building a reasonable micro model of the behaviour of the natural actors in the store leads to a model with similar emergent overall behaviour to the real store. These are the two fundamental ways to verifying our approach, to build from obviously reasonable components, and show that the result resembles the total behaviour of the simulated enterprise.

3.6.3 Privacy Issues

In order to answer the question: *Is the generated data set properly anonymised with respect to the original data set?*, we need to reason about what infor-

mation from the real data set leaks to the generated synthetic data.

First we do not keep any record of who is purchasing what in the store, we base our simulation purely on statistical and network measures that give us an approximate description of how the individual agents behave. So no direct information about the customer leaks. The other actor to consider is the salesman. In our simulation we generate the salesmen based on a statistical approach of how a salesman performs on average at the real store. We conjecture that one would have to have access to the real data in order to identify a salesman based on these statistics, in which case the point is moot.

Finally, what about the overall economic information about the store? This is the underlying reason financial institutions are really reluctant to part with data describing their operation. Competitors might find distributions of sales and overall performance for certain retail stores interesting. However, when we voiced this concern with the owner of the data, they were of the opinion that competitors would already know this as most of it can be deduced from public financial statements such as quarterly reports, etc. They were also of the opinion that the actual operation of a retail store chain and the inherent problems were more or less the same for all competitors, and that the sensitive data from a competitor standpoint was rather fashion line-ups and strategies for upcoming seasons etc. Since our data is a few years old, and we do not simulate changes in inventory, the simulated data ought not to be sensitive from that perspective. However, even so, we still try to mitigate any risks from leakage of economic information by scaling the values of sales etc. so that particulars of profit margins etc. is more difficult to deduce.

3.7 Fraud and Fraud Detection

Now that we have described the data from the store and the simulation of the background data, we finally come to the question of fraud and fraud detection. There are no known instances of fraud in the real data (as certified by the data owner). So we will inject malicious behaviour, by

programming agents that behave according to some known or hypothesised retail fraud case.

3.7.1 Fraud Scenarios in a Retail Store

The following retail fraud scenarios are based on selected cases from the Grant Thornton report [48]. As can be seen below, the different scenarios can be implemented in almost the same way in RetSim, and fit well within the framework given by the normal model. (A malicious salesman could use several different methods of fraud, which means that we needs to be able to model combinations of all fraud scenarios implemented, and we see no reason why that should not be the case.)

The *Refunds scenario* includes cases where the salesman creates fraudulent refund slips, keeping the cash refund for him- or herself.

In terms of the object model used in RetSim, the refund scenario was simulated by estimating the average number of refunds per sale and the corresponding standard deviation. We used these statistics to simulate refunds in the RetSim model. Fraudulent salesmen will perform normal refunds, as well as fraudulent ones. The volume of fraudulent refunds was modelled using a salesman specific parameter. The "red flag" for detection would in this case be a high number of refunds for a salesman.

Coupon reductions/discounts scenario includes cases where the salesman registers a discount on the sale without telling the customer; i.e., the customer pays the full sales price, and the salesman keeps the difference.

In terms of the object model used in RetSim, the coupon reduction/discounts scenario was implemented by estimating the average number of cancellations per sale and the corresponding standard deviation. Using these statistics we simulated discounts in the RetSim model. Fraudulent salesmen performed normal discounts, as well as fraudulent ones. The volume of fraudulent discounts was modelled using a salesman-specific parameter. The "red flag" for detection would in this case be a high number of discounts for a salesman with a relatively low number of average

sales.

There are other possible scenarios, but as mentioned in the introduction return fraud (both by customers and sales staff alike) is a major problem, so we have chosen to focus on return fraud and the structurally similar discount fraud, as these are common and serious.

3.7.2 Injection of Fraudulent Refunds

To model the first scenario we need information about the relevant parameters describing the normal behaviour: figure 3.8 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.

3.7.3 Injection of fraudulent discounts

Figure 3.9 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

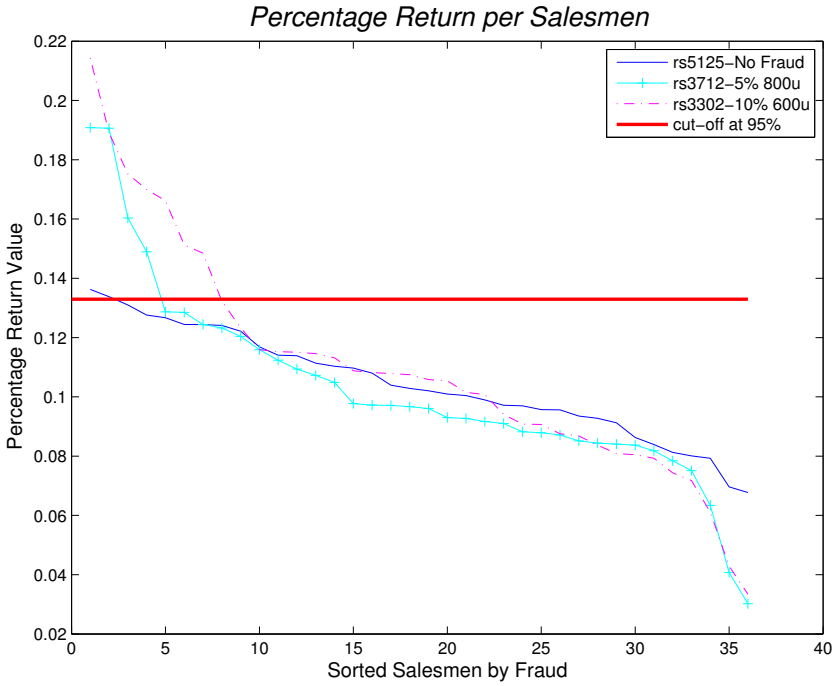


Figure 3.8: *Return Value Over Sales Total per Salesman*

fraud threshold of 600 units and the frequency 10% of the sales. The total profit obtained by all agents is 80600 units per year.

3.7.4 Detection

We will use a rule-based fraud detection approach similar to the "exception audit technique" presented in the Grant Thornton report [48]. The rule-based approach is usually acceptable when there are few parameters to model, and when we do not expect any larger variations between the agents "normal" behaviour. For example, it may be reasonable to expect that each salesman on average will handle approximately the same number of returns and discounts.

Furthermore, almost all commercially available fraud detection systems

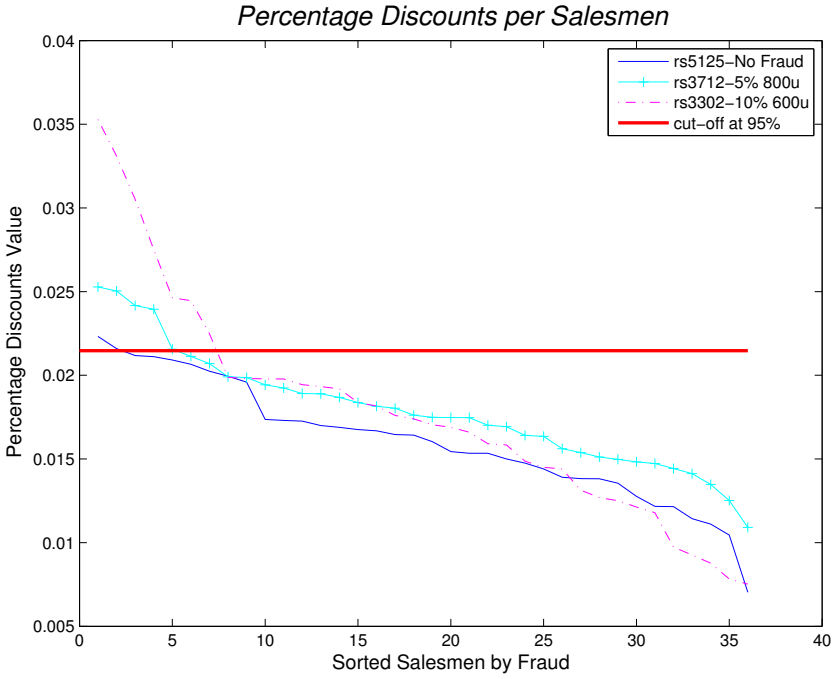


Figure 3.9: *Discount Value Over Sales Total before Discount per Salesman*

are based on the simple rule-based approach with more or less fixed thresholds of detection, more advanced systems based on machine learning are not as popular.

The cut-off points for the expected number of returns and discounts can be chosen in a number of ways; e.g., setting a limit based on the percentage of returns, or on the total value of the refunded items. We have chosen to set the limit at the 95% percentile of the distribution of percentage of refunds (and discounts). That is, on average we expect one salesman in twenty (1/20) to need further investigation. This gives us the following cut-off points; 0.13 for returns, and 0.022 for discounts (see fig. 3.8 and 3.9). The results for the returns are shown in tables 3.5 and 3.6.

In table 3.5, we see that when the fraud is more aggressive the threshold

Table 3.5: *Fraud Detection Results*

Statistic	**rs3302**	**rs3712**
True Positives	5	2
False Positives	2	2
False Negatives	1	4
Precision	71.42%	50%
Recall	83.33%	33.33%

detection method at 95% cut-off is more effective and produces less false negatives than with moderate fraud. From table 3.6 we observe that a non trivial amounts of fraud goes undetected when the fraud is moderate. However, threshold detection is very effective when applied to the simulation with the aggressive fraud behaviour, catching more than 90% of the returned fraud. However, in either case the total amount of fraud is limited at a fixed percentage of turnover, and when fraud increases our method becomes relatively *more* effective. This seems to indicate that by adjusting the threshold, the business owner can trade off the level of "accepted" fraud for the cost of performing further investigation into the flagged staff. Thus being able to manage the risk to business due to fraud. We also performed a simulation with a handful of other agents that performed fraud with different percentages and different cut-off levels. However, as they didn't add anything to the results presented here; they either behaved as the ones presented or differed in trivial ways (i.e. someone who doesn't perform much fraud will be difficult to predict, but also not a great source of loss) we decided not to report on them further.

Table 3.6: *Threshold Fraud Detection*

Item-Data	**rs3302**		**rs3712**	
Sales	36,584,976	100.00%	39,085,401	100.00%
Fraud	400,452	1.09%	161,631	0.41%
Detected	371,463	1.02%	11,577	0.03%
NOT Detect.	28,989	0.08%	150,054	0.38%

Visual methods to identify fraud can also be applied as shown as in figure 3.10, where we can see a network perspective of the malicious agents. We filtered out all other agents and present only the malicious agents network. It is clear from the aggressive fraud behaviour data that only agents that work often at the store are detected by simple threshold rules. On the other hand they are the ones that have the most opportunity to defraud, and are more trusted than e.g. recent hires.

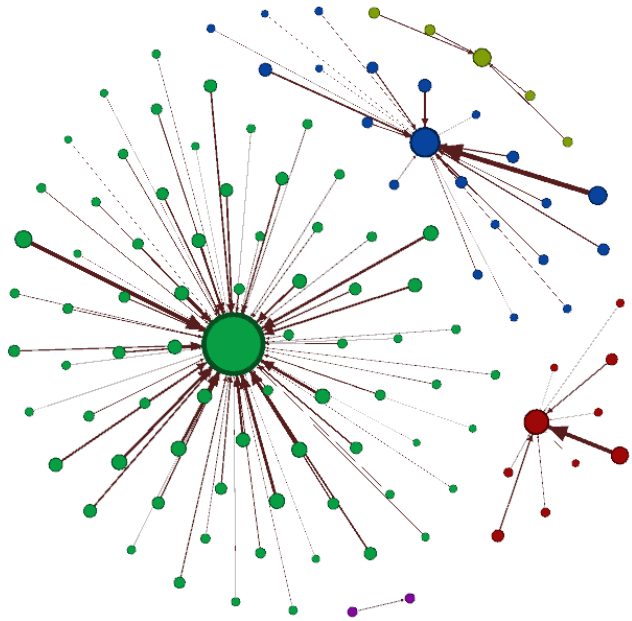


Figure 3.10: *Network Filtering Only Fraudulent Transactions* rs3712

3.8 Discussion

In section 3.3 we formulated our main research question for this paper: *How can we model and simulate a retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?*

To better support our claim and answer our main research question we

also formulated three more specific questions: RQ1, RQ2 and RQ3.

RQ1 discussed verification and validation. There are two main approaches to this when modelling; show that the parts of the model are reasonable and directly model the details of the real world, thus implying that the emergent behaviour will be realistic (in some sense the inductive argument). The second approach takes the other tack, by running the simulation and show that it produces a result that is (statistically) similar to real world measurements (the deductive argument). We make both types of arguments here, we first described an agent-based simulation that is analogue to the actors and actions in a retail store, and then we demonstrated that the simulation could produce behaviour patterns that mimic what we saw in the real data.

To address *RQ2*, we discussed some of the problems of sensitive data leakage and how we addressed them in section 3.6.3. The privacy and security problems incurred from performing a simulation based on real data seem manageable; even though there is, of course, some leakage from a business perspective, the data owners seem unfazed by it.

RQ3: "Is threshold detection sufficient to keep the losses from fraud at manageable level?" We make practical use of our simulation to answer a simple but important question for retail stores who aim to minimise the risk by managing the loses from fraud and at the same time minimising the effort and cost of fraud detection. In section 3.7.4 we show two simple scenarios where threshold control works to combat an aggressive fraud behaviour scenario. At the same time we found that when the fraud is moderate, threshold control techniques are not that effective and the cost of false positives becomes higher, but still below our set level of acceptable fraud.

However, it should be *stressed* that these results, while interesting, are preliminary. Much more simulation of differing scenarios, both from different business and types of fraud, and more detailed mathematical analysis is needed before this question can be answered conclusively.

3.9 Conclusions

RetSim is a simulator of a retail store that generates transaction data set that can be used for research into fraud detection. Synthetic data sets generated with RetSim can aid academia, companies and governmental agencies in testing their methods, in testing the performance of different methods under similar conditions on the same test data set, or in generally reasoning about the limits of effectiveness of fraud detection. We demonstrate this by performing simple rule-based detection and demonstrating what the performance would be if this were run at a real store with similar normal and fraudulent behaviour.

We used the 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 used 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 argue that RetSim 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 salesman returning stolen merchandise or unusually low productivity of a salesman during a specific day that may indicate that the salesman is not entering some of the receipts into the system. We intend to make RetSim available to the research community together with standard data sets.

For the future we plan several improvements and additions to the current model. RetSim can be calibrated for other stores to improve the results presented in section 3.5. We also hope to make analysis of stores in other domains and extend the fraud model to make data sets for fraud

detection in other domains available. We plan to make the simulator and data presented here available to the research community at large.

In order to generate more records with diverse malicious behaviour we will extend RetSim to generate malicious activity that can come from any number of different agents; the salesman, the customer or even the managers, or combinations of these agents. Another possible addition is an interesting scenario, the *self transaction*, where a salesman can play the role of both a customer and a salesman at the same time. This behaviour enables new types of fraud which is important to be able to detect and reason about.

4

Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research

Edgar Alonso Lopez-Rojas and Stefan Axelsson

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.

4.1 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* [35]. 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* [40]. *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 4.2 presents related work on simulation and fraud detection for the financial domain. Section 4.3 describes the methodology used for our research. Section 4.4 presents *PaySim*. Section 4.5 presents *RetSim*. Section 4.6 presents *BankSim*. We present a description of the model, evaluation and results for the simulators and finish with a discussion in section 4.7 and conclusions, including future work in section 4.8.

4.2 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 [57]. 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 [33]. 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 [2]. 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 [4]. 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 [45]. The most recent notorious case of money laundering is the HSBC Bank case [28], 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 [12]. 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 [45].

New promising research in the field of data mining based methods

have also been used to detect fraud [53]. 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 [1].

Several machine learning techniques have been used for the detection of fraud, and more specifically money laundering [60]. The application of machine learning to the problem is advantageous in many situations [64, 65]. 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.

4.3 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 [35]. 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. [36, 40]. 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 [43].

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*.

4.4 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.

4.4.1 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. 4.4.2.

4.4.2 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 4.1 and 4.2. 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 4.1: *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 4.2: *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				

4.4.3 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [51]. 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.

4.5 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

Retail 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.

4.5.1 Model

The design of *RetSim* was based on the ODD model introduced by Grimm et.al. [23]. 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.

4.5.1.1 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.

4.5.1.2 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.

4.5.1.3 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 4.1 shows the different use cases of the agents. This model represent the different actions that an agent can take inside the system.

Manager scheduler: This agent is in charge of scheduling the next

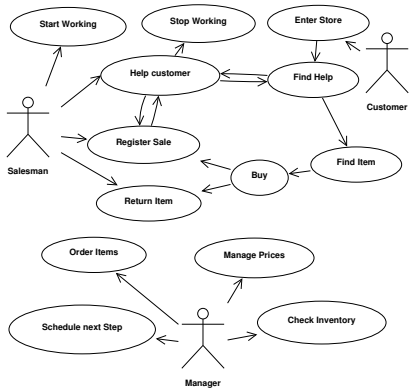


Figure 4.1: *RetSim Use Case Diagram*

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.

4.5.2 Validation and Verification

We start the evaluation of our model with the verification and validation of the simulator and the generated data [51]. 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 4.2 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.

In figure 4.3 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.

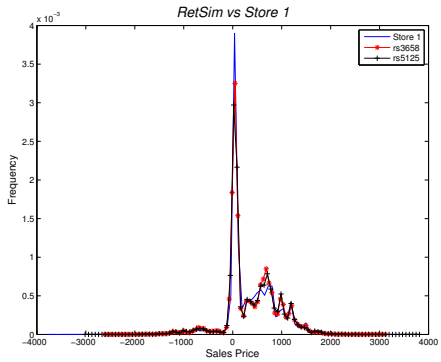


Figure 4.2: *Comparison of distribution of simulated vs real data*

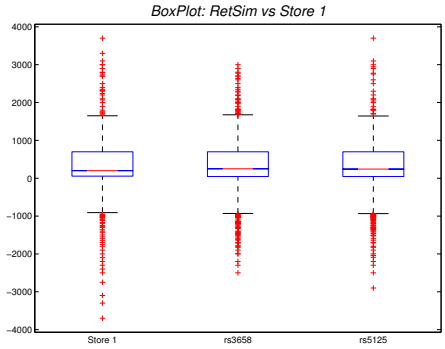


Figure 4.3: *Box plot of simulated vs real data*

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.

4.5.3 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 [48]. As can be seen in section 4.5.1, 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.

4.5.3.1 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 4.4 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.

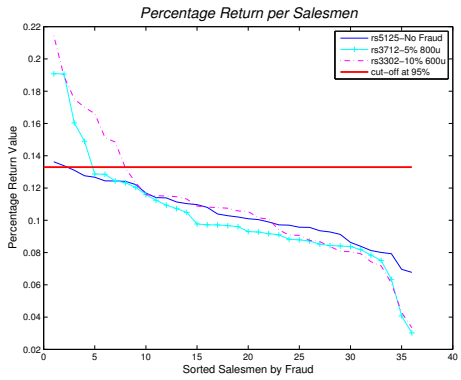


Figure 4.4: *Return Value Over Sales Total per Salesman*

4.5.3.2 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 4.5 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.

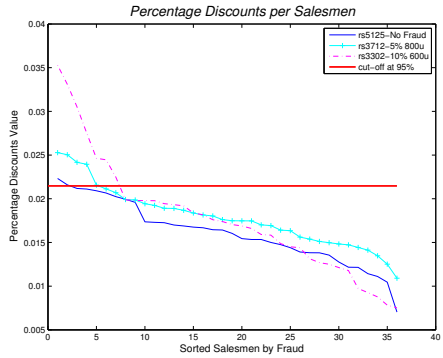


Figure 4.5: *Discount Value Over Sales Total before Discount per Salesman*

4.5.4 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.

4.6 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[17][16], 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.

4.6.1 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.

4.6.2 Model

The preliminary design of *BankSim* was again based on the ODD model introduced by [23].

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.

4.6.2.1 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.

4.6.2.2 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.

4.6.3 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 [51]. 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.

4.7 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.

4.8 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[43] 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.

5

Extending the RetSim Simulator for Estimating the Cost of fraud in the Retail Store Domain

Edgar Alonso Lopez-Rojas

Abstract

RetSim is a multi-agent based simulator (MABS) calibrated with real transaction data from one of the largest shoe retailers in Scandinavia. RetSim allows us to generate synthetic transactional data that can be publicly shared and studied without leaking business sensitive information, and still preserve the important characteristics of the data.

In this paper we extended the fraud model of RetSim to cover more cases of internal fraud perpetrated by the staff and allow inventory control to flag even more suspicious activity. We also generated sufficient number of runs using a range of fraud parameters to cover a vast number of fraud scenarios that can be studied.

We then use RetSim to simulate some of the more common retail fraud scenarios to ascertain exactly the cost of fraud using different fraud parameters for each case.

Keywords: Multi-Agent Based Simulation, Retail Store, Fraud Detection,Retail Fraud, Synthetic Data.

5.1 Introduction

Fraud is an important problem in a number of different situations, and more specifically in retail stores is a very common problem in all countries. The economic impact for the loses can be substantial, this is why many major store retailers invest in security. However, how much to invest could be a political decision since many times it is hard to calculate the cost of possible fraudulent behaviour due to the multiple possible fraud causes. Once the fraud problem and cost of it is identified, it is easier for managers to take the prevention measures to lower the loses. For example, in one recent case the major US home improvement chain *Home Depot* was the target of a fraudulent return scam where two perpetrators netted several thousand dollars before being caught [19]. Return fraud, i.e. the defrauding of a retail merchant by abusing the return process, alone is estimated to cost US retailers about 9 billion dollars yearly. To further illustrate the seriousness of the problem and try and combat it both EU and US recently started to mandate the use of fraud detection as one part of the minimum security requirements for financial services [16, 17].

Our approach to this problem makes use of the RetSim simulator as an strategy to measure and estimate the cost of loses and as a experimental laboratory for managers to apply measures to detect and discourage internal fraud. RetSim is a multi agent-based simulator (MABS) of a shoe store based on the transactional data of one of the largest retail shoe sellers in Sweden [40]. RetSim uses this real data to develop and calibrate the model through statistical and a social Network Analysis (SNA) of the relations between staff and customers and generates a synthetic data set similar to the original one but without any unwanted disclosure of private information about either the staff or the customers.

Therefore, the aim of RetSim is the generation of synthetic data that can be used for fraud detection research and prognosis of fraud scenarios. With the RetSim simulator researchers and managers can test and measure in different fraud scenarios the cost and performance of fraud detection methods as simple as thresholds or even more elaborated such as triage

models bases on thresholds, machine learning algorithms and others. The initial purpose of RetSim was to model a realistic data set that resembles the distribution of financial transactions found in an original given data set. Now that we have that we added fraudulent behaviour to study the cost of fraud in retail store.

In this paper we extended the fraud model of RetSim to cover more cases of internal fraud perpetrated by the staff and allow inventory control to flag even more suspicious activity. We also generated sufficient number of runs using a range of fraud parameters to cover a vast number of fraud scenarios that can be studied.

Our ultimate goal is for RetSim to be usable to model relevant scenarios to generate realistic data sets that can be used by academia, managers, security inspectors, students and others, to develop and reason about fraud detection methods without leaking any sensitive information about the underlying data. 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 RetSim generates data that usefully approximates the relevant aspects of the real data.

Outline: The rest of this paper is organised as follows: Section 5.2 introduce the topic of fraud detection for retail stores and present previous and related work on simulators. Section 5.3 describes the problem, which is the generation of synthetic data of a retail store system for estimating the cost of fraud. Sections 5.4 and 5.5 present our implementation and results of a MABS for our domain and shows the description of the extended fraud scenarios implemented on RetSim. We finish with a discussion and conclusions, including future work in section 5.6.

5.2 Background and Related Work

Simulations in the domain of retail stores have traditionally been focused on finding answers to logistics problems such as inventory management, supply management, staff scheduling and customer queue reductions [14,

15, 57]. We find no research focusing on simulations generating fraud data to be used for fraud detection in retail stores. Therefore, we recently introduced RetSim with the purpose of fraud detection research. In this article we built upon previous version of RetSim to study the cost of specific fraud scenarios, including agents using known fraud behaviour patterns [40].

Anonymization techniques have been used to preserve the privacy of sensitive information present in data sets. But de-anonymizing data sets is not an insurmountable task, far from it [49]. For this reason we have decided to use simulation techniques to keep specific properties of the original data set, such as statistical and social network properties, and at the same time providing an extra layer of insulation that pure anonymization does not provide.

There are tools such as IDSG (IDAS Data and Scenario Generator [29]) that were developed for 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 the testing phase, and it has been used to test fraud detection systems. Our approach differs in that we are implementing an agent-based model which is based on agent micro behaviour rather, than a fixed statistical distribution of macro parameters.

With the current popularity of social networks, such as *Facebook*, the topic of Social Network Analysis (SNA) has seen interest in the research community [4]. Social Network Analysis is currently being combined with *Social Simulation*. 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 addressing the question of customer/salesman-interaction, that we are aware of.

Other methods to generate the necessary fraud data have been proposed by [2, 18, 26, 44, 63]. The work by [63] lets the user specify the assumptions about the environment at hand; i.e., there is no need for access to real data. However, this will certainly affect the quality of the synthetic data. The

work by [44] makes use of a small sample of real data to generate synthetic data. This approach is similar to ours. However, the direct use of real data to prime the generation of synthetic data is limited in that it makes it harder to generate realistic data with other characteristics that those of the original real data [63]. The work by [26] focused on privacy-preserving methods for data mining. However, that method also does not have the possibility of generating realistic data with other characteristics than those of the original data. In our work, we use social simulation, which makes it possible to change the parameters of the agents in the model to create realistic synthetic data, potentially producing emergent behaviour in the logs which is hard to produce in other ways.

5.3 Problem

The RetSim simulator was previously used to solve the problem of finding a synthetic data set that realistically represent a given real data set without disclosing any specific information about customers or staff members. Now with this paper we aim to solve the question of how much are the loses due to fraud giving a specific scenario of fraud in a retail store. This problem is of particular interest for managers of stores, mainly because the investment on security is limited, therefore the impact of each of these fraud models and expected fraud scenarios will give an idea to a manager of the cost of fraud. Historically on a real store the cost is only estimated but can not be exactly calculated due to the disguise nature of the fraud.

By using simulators, we can be certain of the profit of each fraudulent agent over the time. We generate synthetic data that contains flagged fraud behaviour, therefore it simplifies the process of estimating the cost of loses due to specific fraud. These estimations can be used to take informed decisions because they are a good approximation of a real scenario.

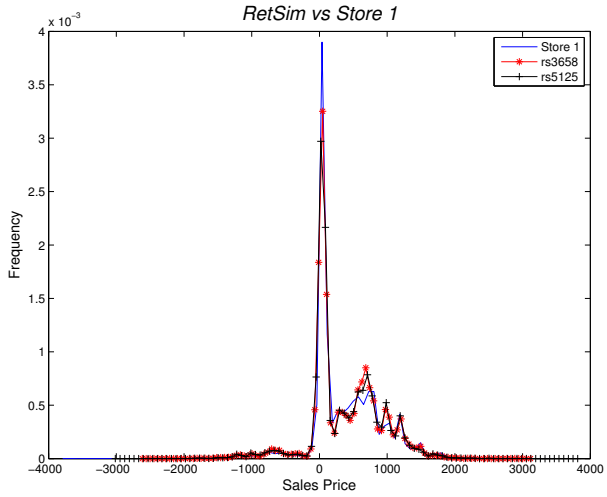


Figure 5.1: *Overlap of Two Runs of RetSim vs Real Data*

5.4 Model and Method

RetSim uses the MABS toolkit MASON version 17 which is implemented in Java [43]. We selected MASON because it is: multi-platform, supports parallelisation, and fast execution speed in comparison with other agent frameworks. This is especially important for multiple running and computationally expensive simulations such as RetSim [54].

Our model contains the following entities and behaviours. The *Store* is the main entity of the simulation, it contains all the variables and states required to run the simulation such as: *Salesmen*, *Customers*, *Products*, *Frequencies* and other parameters used to calibrate the model.

During the initialization of the simulation a store load all products, set up an inventory and creates the salesmen according to the parameters of the store. We calibrated RetSim to consider each day as a step in the simulation. During a single step of the simulation some of the salesmen became active and others inactive or not working that day. The salesmen are distributed in a virtual geographical space on the store. Once the day

starts an average of customers are instantiated and distributed around the geographical space.

The *basic principle* of this model is the concept of a commercial transaction. The process of creating financial transactions starts when a salesman sense nearby customers in the *need-help* state and offers help. There are two different outcomes: either a transaction takes place or not, similar to the real world.

There are no known instances of fraud in the real data (as certified by the data owner). So we modelled malicious agents with fraud behaviour, by programming agents that behave according to some known or hypothesised retail fraud case.

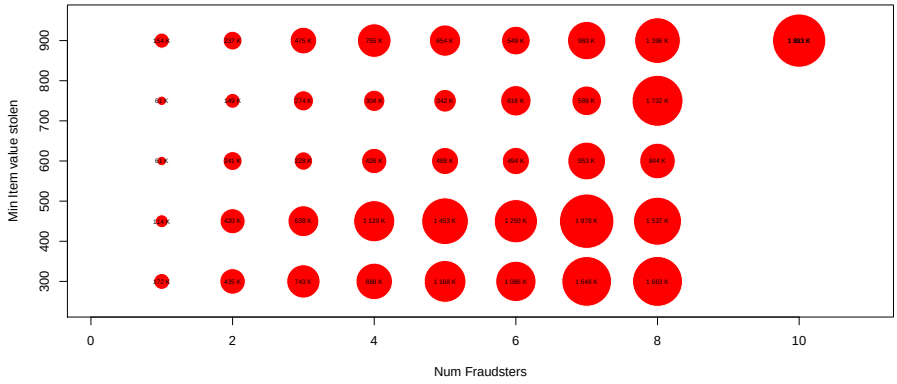
The following retail fraud scenarios are based on selected cases from the Grant Thornton report [48]. As can be seen below, we extended the RetSim model by adding different fraud scenarios. The initial purpose of RetSim was to model a realistic data set that resembles the distribution of financial transactions found in an original given data set. Now that we have that we added fraudulent behaviour to study the cost of fraud in retail store.

The *Refunds fraud scenario* is perhaps the most common cause of fraud and includes cases where a salesman creates fraudulent refund slips, keeping the cash refund for him- or herself.

In terms of the object model used in RetSim, the refund scenario was simulated by estimating the average number of refunds per sale and the corresponding standard deviation. We used these statistics to simulate refunds in the RetSim model. Fraudulent salesmen will perform normal refunds, as well as fraudulent ones. The volume of fraudulent refunds was modelled using a salesman specific parameter. The "red flag" for detection would in this case be a high number of refunds for a salesman.

The *Coupon reductions/discounts fraud scenario* is perhaps the second responsible for loses and includes cases where the salesman registers a

Figure 5.2: *Analysis of Cost of Fraud*



discount on the sale without telling the customer; i.e., the customer pays the full sales price, and the salesman keeps the difference.

In terms of the object model used in RetSim, the coupon reduction/discounts scenario was implemented by estimating the average number of cancellations per sale and the corresponding standard deviation. Using these statistics we simulated discounts in the RetSim model. Fraudulent salesmen performed normal discounts, as well as fraudulent ones. The volume of fraudulent discounts was modelled using a salesman-specific parameter. The "red flag" for detection would in this case be a high number of discounts for a salesman with a relatively low number of average sales.

We extended the original RetSim by adding inventory control over products. Our initial assumptions were that the replenish of the inventory was performed automatically and therefore we did not focus on this aspect. But in reality inventory control is one of the more effective ways to detect fraud. Unfortunately performing inventory control is an expensive procedure in most of the stores due to manual counting.

There are other possible scenarios, but as mentioned in the introduction return fraud (both by customers and sales staff alike) is a major problem, so we have chosen to focus on return fraud and the structurally similar discount fraud, as these are common and serious.

5.5 Results

We built upon previous successful simulations that were calibrated from a Store taken from the original data set of our retailer. Figure 5.1 shows the results from a comparison of sales frequency for items of 2 runs of RetSim versus the original store (named Store 1). We can clearly see that our simulator successfully models the sales of Store 1. We can analyse that items that cost less are the most wanted products by the customers. For convenience and protection of disclosure of the business we decided to use a fictitious monetary unit that we will name just *units*.

From an analysis of figure 5.1, a manager can design experiments to study the phenomenon of fraud performed by the staff in the previous scenarios explained (Refunds fraud and discounts fraud).

Our experiment aims to calculate the total cost of loses if one or many employees are responsible for fraud. One common behaviour is of fraudsters is to mentally set some limits about the min or max value of the item stolen. Many fraud controls nowadays are performed using simple thresholds around the prices. One way to avoid these controls is by carefully selecting items with value below. We will assume that the fraudster mentally set up a minimum value for stealing to make it worth the risk of being caught by the manager.

We setup our experiment to run 100 times under a specific setting. We iterated around minimum values starting from 300 *units* up to 900 *units*. We use a probability of 10% of the time that a salesmen should do the checks for performing fraud. In total we collected 500 runs of RetSim and summarize our cost analysis in figure 5.2.

Figure 5.2 shows one way to visualize the total cost of fraud performed by a different number of staff members. This figure allows a manager to study better the phenomenon of fraud in a store. One can notice for instance that the most of the items sold are around the 450 *units* price. So the profit of a fraudster that steals around this minimum value will be in many cases higher than other that decide to steal only high value articles. This information is particularly important when we require to implement controls over the refunds of items that cost more or less than 450 *units*. We can also see that below this value the amount of loses is considerable when then number of staff members involved in fraud increases.

Besides the controls and thresholds that can be used for fraud, a major contribution of the RetSim to the managers is to measure the total cost of fraud. With this information a manager can decide how much to invest in any required fraud detection control. In many cases it is *"acceptable"* for managers to deal with low loses if the investment in fraud is higher than the fraud that can be prevented.

5.6 Conclusions

RetSim is a simulator of a retail store that generates transaction data set of diverse fraud scenarios, in this paper we extend the fraud model and measure the cost of total loses of each scenario. The cost of fraud in different scenarios can be estimated by summing the profit from each malicious agent (usually known as fraudsters). These estimations can be used to take informed decisions about how much should be invested in fraud controls, because they are a good approximation of real fraud scenarios.

Synthetic data sets generated with RetSim can aid academia, managers of companies and governmental agencies in testing their methods, in exploring the cost of different fraud scenarios and the performance of different fraud detection methods while maintaining similar conditions by using the same test data set, or in generally reasoning about the limits of effectiveness of fraud detection.

Future work on RetSim may include the addition of more relevant fraud models as well as tools for setting up experiments with higher degree of parametrization that includes more variables such as variations (increase or decrease) of sales with respect to the original store.

6

Using the RetSim Fraud Simulation Tool to set Thresholds for Triage of Retail Fraud

Edgar Alonso Lopez-Rojas and Stefan Axelsson

Abstract

The investigation of fraud in business has been a staple for the digital forensics practitioner since the introduction of computers in business. Much of this fraud takes place in the retail industry. When trying to stop losses from insider retail fraud, triage, i.e. the quick identification of sufficiently suspicious behaviour to warrant further investigation, is crucial, given the amount of normal, or insignificant behaviour.

It has previously been demonstrated that simple statistical threshold classification is a very successful way to detect fraud [42]. However, in order to do triage successfully the thresholds have to be set correctly. Therefore, we present a method based on simulation to aid the user in accomplishing this, by simulating relevant fraud scenarios that are foreseeing as possible and expected, to calculate optimal threshold limits.

Our proposed method gives the advantage over arbitrary thresholds that it reduces the amount of labour needed on false positives and gives additional information, such as the total cost of a specific modelled fraud behaviour, to set up a proper triage process. With our method we argue that we contribute to the allocation of resources for further investigations

by optimizing the thresholds for triage and estimating the possible total cost of fraud. Using this method we manage to keep the losses below a desired percentage of sales, which the manager considers acceptable for keeping the business properly running.

6.1 Introduction

The economic impact of fraud by staff can be substantial in several types of business. Thus the detection and management of fraud is an important topic. In the retail store the cost of fraud is of course ultimately transferred to the consumer, and finally impacts the overall economy. For example; in one recent case the major US home improvement chain *Home Depot* was the target of a fraudulent return scam where two inside members of staff netted several thousand dollars before being caught. They perpetrated the fraud by abusing their knowledge of the processes of the shipment and return of products [19]. Retail fraud was estimated to cost US retailers about 42 billion dollars in 2013 from which 43% was committed by dishonest staff [8]. Due to the seriousness of this type of fraud, both EU and US recently started to mandate the use of fraud detection as one part of the minimum security requirements for financial services [16, 17].

The constant change of criminal behaviour and patterns, and the introduction of new fraud schemes makes this an important topic for researchers and practitioners alike. For a multitude of reasons (e.g., privacy related, legal, financial, or contractual) the state of practice in fraud research is to work with sensitive and hence secret data [33]. This difficulty hinders researchers to develop methods to detect, prioritize investigations (triage process), and finally share the results with other researchers without problem. We name this problem the *data secrecy problem.*

The *data secrecy problem* has previously been addressed by the use of synthetic data, generated by the RetSim simulator [40]. However, a simulator has other benefits aside from being able to share relevant data for research. One major advantage is that different scenarios can be tested. In these scenarios parameters such as the number of fraudulent staff, their

propensity to perpetrate fraud, cost of merchandise etc. This enables the testing of e.g. new fraud detection schemes. These detection schemes can later be applied to real data, so we can prioritize and allocate resources for performing further investigation of fraud.

In this paper we address the problem of how to apply statistical threshold detection to perform triage by proposing a technique which uses synthetic data, generated by the RetSim simulator [40], to test different fraud scenarios so that thresholds can be set and the resulting performance studied. These methods can later be applied on real data.

In our study we target fraud caused by fraudulent refunds performed by sales staff. This is a common type of fraud and accounts for around 28% globally of the total fraud in a retail store, with this situation being more critical in North America [8]. This fraud scheme takes advantage of the lack of security controls inside the store and the difficulty to perform inventory control often in most of the retail business settings. Once the missing inventory is noticed by the inventory control officers, a digital forensic investigation can be performed over the available evidence to identify the people responsible, which in this particular case happens to be associated in most of the cases with the salesperson staff.

However, it is difficult to prioritise which of the staff members should be investigated, especially when we are dealing with a chain of stores with multiple branches and the corresponding number of staff. When investigating losses from theft and fraud in the retail setting, we are most often not interested in finding every last instance of fraud but rather of limiting our overall loss to an acceptable level, often put as a set percentage of overall turnover. Spending time and resources on the investigation of the pettiest of thefts of office supplies is counter productive, as the investigative resources are both scarce and expensive. It can also negatively affect the workplace atmosphere. Thus, being able to focus the investigative effort on the cases that can affect the bottom line is vital.

Thus there is an evident need to prioritize and allocate personal for performing such investigations of staff fraud in the retail sector. We do

this by using a triage process model and categorising the fraud threat into critical, important and low impact. However, the financial and transaction data available is large and hence we need an effective way of performing triage, to focus efforts where they may make the most impact, and hence keeping the total loss to fraud at a set, acceptable, level.

6.2 Related Work

Simulations in the domain of retail stores have traditionally been focused on finding answers to logistics problems such as inventory management, supply management, staff scheduling and customer queue reductions [14, 15, 57]. We find no research focusing on simulations generating fraud data to be used for fraud detection in retail stores besides the RetSim simulator [40].

One of the reasons data is simulated instead of taken directly from the original source is the *data secrecy problem*. In our experience, the privacy of the customers has always been the main concern when disclosing any transactional data. This can be seen by the lack of any kind of public transactional data set that reflects financial statement of individual persons. Many anonymization techniques have been used to preserve the privacy of sensitive information present in data sets. But de-anonymizing data sets is not an insurmountable task, far from it [49]. This is one of the reasons why we have decided to use simulation techniques to keep specific properties of the original data set, such as statistical and social network properties, and at the same time providing an extra layer of insulation that pure anonymization does not provide.

However, using a simulator also has many other benefits, the main one being that the experimenter is in total control of the environment and can vary parameters to try different scenarios; increasing and decreasing the intensity and severity of fraud, for example.

There are tools such as IDSG (IDAS Data and Scenario Generator [29]) that were developed for 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 the testing phase, and it has been used to test fraud detection systems. The RetSim approach differs in that it is implementing an agent-based model which is based on agent micro behaviour rather, than a fixed statistical distribution of macro parameters.

Other methods to generate the necessary fraud data have been previously proposed [26, 44, 63]. The work by Yannikos et al. [63] lets the user specify the assumptions about the environment at hand; i.e., there is no need for access to real data. However, this will certainly affect the quality of the synthetic data. The work by Lundin et al. [44] makes use of a small sample of real data to generate synthetic data. This approach is similar to the one in RetSim. However, the direct use of real data to prime the generation of synthetic data is limited in that it makes it harder to generate realistic data with other characteristics than those of the original real data [63]. The work by Kargupta et al. [26] focused on privacy-preserving methods for data mining. However, that method also does not have the possibility of generating realistic data with other characteristics than those of the original data. RetSim, uses social simulation, which makes it possible to change the parameters of the agents in the model to create realistic synthetic data, potentially producing emergent behaviour in the logs which is hard to produce in other ways.

Previous research on fraud detection algorithms has showed that data mining and machine learning algorithms can identify novel methods of fraud by detecting those records that are different (anomalous) in comparison with benign records, e.g., the work by Phua et al. [53]. This problem in machine learning is known as *novelty detection*. Furthermore, supervised learning algorithms have been used on synthetic data sets to prove the performance of outlier detection [1, 44]. More particularly in retail stores the use of pattern discovery to address retail fraud has been used by Gabbur et al. [20] with many limitations to train a classifier due to the lack of reliable fraud data. However none of these studies made use of synthetic data from retail stores. To our knowledge, there has been no investigation

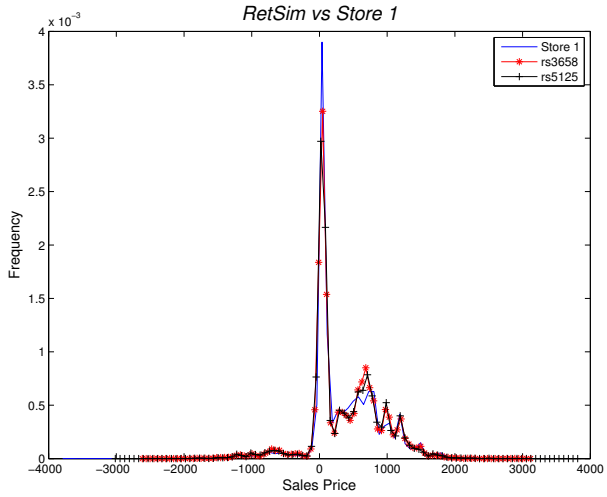


Figure 6.1: *Overlap of Two Runs of RetSim vs Real Data*

of what the limits of effectiveness of e.g. simple threshold based monitoring are over other complex techniques such as machine learning and pattern recognition.

6.3 RetSim: the Simulator for Retail Store Data and Fraud

Since we have access to several years worth of transaction data from one of the largest Scandinavian retail shoe store chains, we made use of *RetSim*[40], a *Ret*ail shoe store *Sim*ulation, built on the concept of Multi-Agents Based Simulation (MABS). *RetSim* is intended to be used in developing and testing fraud scenarios at a retail shoe store, while keeping business sensitive and private personal information about customers consumption secret from competitors and others.

RetSim uses the relevant parameters that govern the behaviour in and of a retail store to simulate *normal* behaviour. The output of this process is a synthetic data set that contains similar properties as the original data

set. We also model the malicious behaviour of staff and simulate this behaviour together with our normal behaviour to produce a rich data set useful for fraud research.

One of the main advantages of simulating data for fraud over real data is that it can quantify the loss due to the identification of malicious agents since the activities of these are known [33, 35]. Due to this capability, one of the main results of previous research with RetSim is that in many cases a proper setting of a threshold detection control can be enough to keep the loss of a business to fraud, at a desired level, and at the same time avoiding the cost and complexity of more advance methods, such as data mining and machine learning.

Fraud in the retail setting is in many cases perpetrated by the staff so we have decided to focus on that. A common example of such fraud is *Refunds due to Fraudulent Returns*. In this paper we make use of the *RetSim* tool to study this specific fraud scenario that includes agents defrauding the store and performing known fraud behaviour patterns.

With the help of *RetSim*, we produced a simulation that results in data comparable to our real data set. The generated synthetic data set contains 36 salesmen and around 45,000 receipts and 81,500 articles sold. The simulation was seeded with a subset of about 11,000 articles from the real store (that we named *Store 1*). One of the challenges when simulating data is to evaluate how realistic the data is in comparison with the source. In Figure 6.1 there is evidence of the similarities from an overlapping plot of the generated distribution of sales by price of both: original data set (Store 1) and simulated data sets (*rs3658* and *rs5125*). In this paper we make use of *rs5125* as a reference data set for a *No Fraud* scenario. This evidence is part of previous work using this tool to generate a realistic data set for research [40].

Now that we have a proper data set simulated that resemble the original data, the next step is to inject malicious behaviour that can be used for research. In the chosen scenario of fraudulent returns we include cases where a salesman creates fraudulent refund slips, keeping the cash refund

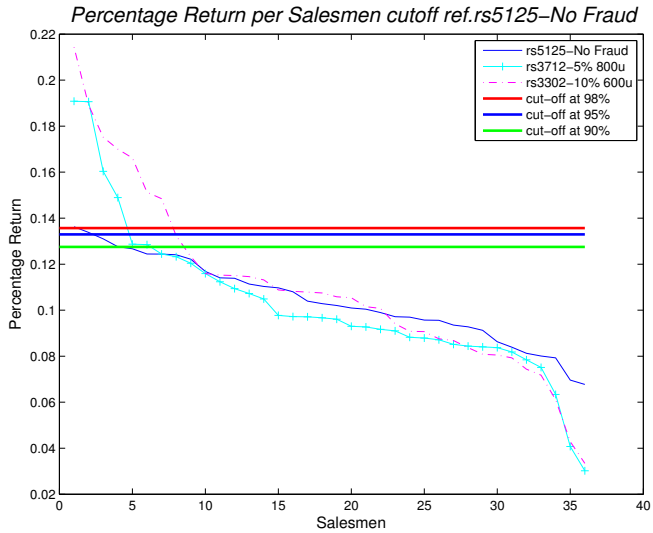


Figure 6.2: *Triage cut off using as reference no fraud behaviour*

Table 6.1: *Triage Threshold Limits*

Data Set-Triage	Red 98%	Blue 95%	Green 90%
rs5125-No Fraud	0.136	0.133	0.128
rs3712-Moderate Fraud	0.191	0.182	0.147
rs3302-Aggressive Fraud	0.209	0.185	0.170

for him- or herself. In terms of the object model used in RetSim the refund scenario can be implemented by: Estimating 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 one. The volume of fraudulent refunds can be modelled using specific parameters that determine the *aggressiveness* of the fraudulent behaviour. The "red flag" for detection will in this case be a high number and value of refunds in average divided by the total sales for a salesman.

6.4 Triage Process in a Retail Store Scenario

When investigating the instance of fraud in a retail store, it is important to quickly eliminate all the normal background behaviour that is not indicative of fraudulent behaviour. This is of course (hopefully) the overwhelming majority of the transactions. So we need to perform some form of triage, where we quickly identify the abnormal behaviour and single that out for further investigation.

So inspired by the original triage,[1] we have chosen to divide the studied behaviour into three categories, bins, to classify suspicious activity of staff members: the first category requires *critical* or urgent investigation, Category one - red line in figures), the second category is *important* to detect significant loss to the business (Category 2 - blue line in figures), the last category is the category where investigation would probably not be fruitful and have *low impact* on business (Category 3 - green line in figures). The idea being an investigator ought to focus on the red category (Critical), maybe keeping an open mind regarding people in the blue category (Important), and disregard the green (Low Impact) as a cost of doing business if indeed any problematic behaviour should lurk in that category.

To illustrate this triage process in a fictitious retail store we use the return fraud scheme as an example, and we wish to set statistical thresholds to identify the limits of the categories. Staff that process many refunds in comparison to their sales are subject to investigation, therefore our definition of suspicious behaviour is: *An unusually high fraction of the total value of refunds to the total value of sales for the individual salesman in question, in comparison with the average value for the sales staff as a whole.* I.e. the fraud score for the individual salesman is:

$$FraudScore = \frac{ValueOfRefunds}{ValueOfSales}$$

[1] From the French *trier* (v): to separate, sift, or select.

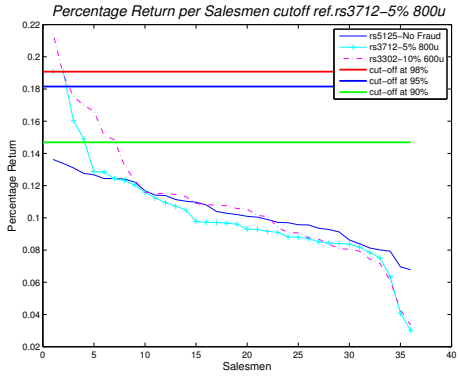


Figure 6.3: *Triage cut off using moderate fraud behaviour as reference*

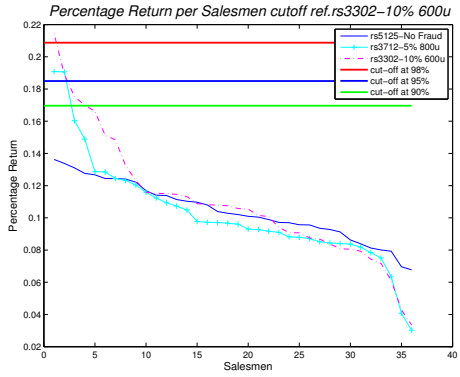


Figure 6.4: *Triage cut off using aggressive fraud behaviour as the reference*

However the task of finding the limits for each of the categories is not trivial. If we set the limits to high, much fraud will be unidentified. On the other hand if we set a lower limit we will experience many false positives. False positives are always a problem when doing any form of detection, and must be avoided [9].

To start formulating the problem we assume that all staff members have a similar probability of performing returns, and that returns of valuable articles are more interesting than lower priced articles from the criminal perspective. We chose, somewhat arbitrary, threshold cut-off limits of (90%, 95% and 98%) to cover at least 10% of the staff that are of particular interest due to suspicious fraud behaviour (higher amount of refunds). The calculated values for the thresholds limits are shown in table 6.1. We do this in order to show how a simulation tool can be used for setting these limits.

Figure 6.2 shows the values for both the normal behaviour, and two simulations with injected *return fraud*. This figure shows the total value of refunds divided by the total sales for each salesman as percentages, for the three simulations *rs5125, rs3712 and rs3302* explained below.

To model a normal sales scenario we used information about relevant parameters describing normal behaviour without any fraud and selected *rs5125* as the No Fraud reference data set. As we said before in section 3, this data set was one of the two evaluated against the original data set to verify that we are using a realistic synthetic data set that resembles the original, and maintains interesting properties of sales without revealing specific details of particular customers.

In a normal situation the only information available is the normal behaviour of the refund process (no fraud). If we set up our threshold limits for triage using this data as a reference (as shown in figure 6.2), we notice that the limits are perhaps too close to each other due to the assumption, as perceived by studying the real data from the store, that each salesman has a similar probability of processing refunds.

The resulting triage processes using an arbitrary threshold for each of the simulations is not optimal for detecting fraud in most cases due to the possible high number of false positives needed to have sufficient effectiveness. For instance, if we set up the thresholds of triage using the data set that contains no fraud (First case Triage 1 in tables 6.2 and 6.3), we have many salesmen to investigate that are flagged as the top priority (red). This could lead to an overwhelming effort to investigate all the fraud

Table 6.2: *Triage of moderate fraud data set with rs3712-5% 800u (Top Fraud Score)*

ID Salesman	**Fraction of Sales**	**Refunded**	**Total Sales**	**Total Stolen**	**Triage1**	**Triage2**	**Triage3**
S836051140	0.191	-20234	106026	3874	Red	Blue	Blue
S1068592722	0.191	-12774	67000	0	Red	Blue	Blue
S1408212765	0.160	-10168	63409	0	Red	Green	
S1948780723	0.149	-42865.86	287783.1	7702.857	Red	Green	
S1568033761	0.129	-406122.8	3155567	119662	Green		
S1434682851	0.128	-53809.7	418757.1	0	Green		
S193026137	0.124	-14805	118980.4	0			
S24105143	0.123	-35499.56	288014	0			
S705613182	0.120	-99021	822423.2	16449			

without the necessary resources.

6.5 Tuning the Parameters of the Triage set up

In this section we propose a method to set up a triage process that fits the business expectations and limitations. To start, in section 6.4 we analysed the consequences of setting arbitrary thresholds due to lack of information of possible fraud. Our proposed method works for either detecting new fraud as it takes place or for the processing of historical data of refunds in order to perform a forensic digital investigation.

To begin, we make use of the RetSim simulator to generate a synthetic data set that contains information about an expected fraud behaviour scenario. In this study we started modelling two possible scenarios, one with moderate fraud behaviour and another one with aggressive fraud behaviour.

The first fraud simulation *(rs3712)* shows a conservative fraud behaviour agent where each of the fraudsters will attempt to commit fraud only if the sales value is worth more than 800 units in the fictitious currency, and

Table 6.3: *Triage of aggressive fraud data set with rs3302-10% 600u (Top Fraud Score)*

ID Salesman	Fraction of Sales	Refunded	Total Sales	Total Stolen	Triage1	Triage2	Triage3
S836051140	0.214	-3597	16783	1199	Red	Red	Red
S1068592722	0.189	-11087	58619.38	0	Red	Blue	Red
S1568033761	0.175	-553671.7	3160122	267760.9	Red	Green	Green
S1948780723	0.170	-10965	64498.8	5343	Red	Green	Green
S1063661000	0.166	-41420.67	249302.5	0	Red	Green	
S705613182	0.151	-185363	1225409	68065	Red	Green	
S1884511064	0.148	-55632	374763	29095	Red	Green	
S944780329	0.132	-51983.12	392897.4	28989	Blue		
S1888626692	0.123	-66295.58	537627.2	0			

Table 6.4: *Fraud Detection Results for Triage of moderate fraud using rs3712*

Statistic	**Triage 1**	**Triage 2**	**Triage 3**
True Positives	3	2	1
False Positives	3	2	1
False Negatives	3	4	5
Detected	131239	11577	3874
Not Detected	30392	150054	157757
Precision	50%	50%	50%
Recall	50%	33%	17%

Table 6.5: *Fraud Detection Results for Triage of aggressive fraud using rs3302*

Statistic	**Triage 1**	**Triage 2**	**Triage 3**
True Positives	6	5	3
False Positives	2	2	1
False Negatives	0	1	3
Detected	399253	371463	274302
Not Detected	0	28989	126149
Precision	75%	71%	75%
Recall	100%	83%	50%

the frequency with which he/she commits this fraud is 5% of all sales. The total amount pilfered by all fraudulent agents in a year is 161,630 units in this scenario, which is around 0.43% of total revenue (39,085,000 units).

The second fraud simulation *(rs3302)* represents an aggressive fraud agent behaviour where the threshold to commit fraud is 600 units and the frequency is 10% of sales. The total amount defrauded by all agents is 400,451 units per year, which is around 1.09% of total revenue (36,584,000 units).

The percentage of returns per salesman for the three generated synthetic data sets are plotted together in figures 6.2, 6.3 and 6.4 for comparison purposes. One of the many benefits of using a simulator is that we can flag all sales refunds that are fraudulent. In a real data set this is unknown

unless someone has already vetted the entire data set, which is difficult both from a practical and theoretical standpoint. In table 6.2 and 6.3 we can partially see the information concerning the plots for those salesmen with higher percentage of refunds per the total sales, more specifically information about total value of sales, value of refunds and the total value stolen.

Now that we have all information required of the expected fraud scenarios, we can make use of these two simulations to set up the same arbitrary thresholds limits of (90%, 95% and 98%). The thresholds limits are shown in figures 6.3 and 6.4 and we will name them Triage 2 and 3.

After looking more in detail into the second case (Triage 2 in table 6.3), we see that inside the categories that are detected as red and blue, there are only 2 salesmen detected. Finally the last case (Triage 3 in table 6.3), does not flag any salesman in the blue category. But we notice that the amount of red flags for investigation is considerably lowered in comparison to Triage 1.

The evaluation of the fraud detection methods using different triage processes is presented in table 6.4 and 6.5. From these tables we see that when the fraud is more aggressive, the triage process has a higher precision and recall than in any of the other triage set ups. However if the goal is to minimize the false positives we should carefully chose a threshold limit that minimize this value.

If we aim to investigate moderate fraud we should carefully set up triage thresholds somewhere in between the settings for Triage 1 and 2. Triage 3 is very inefficient in this scenario as seen in table 6.4. From our results in table 6.5 for the aggressive fraud, we can see that the higher recall is of course with a very low threshold as in Triage 1, but the effort to investigate 8 members of the staff is higher than using Triage 3, where we are still able to detect most of the fraud but lower the number of staff to investigate by half and still detect about 68% of the fraud committed with just one false positive.

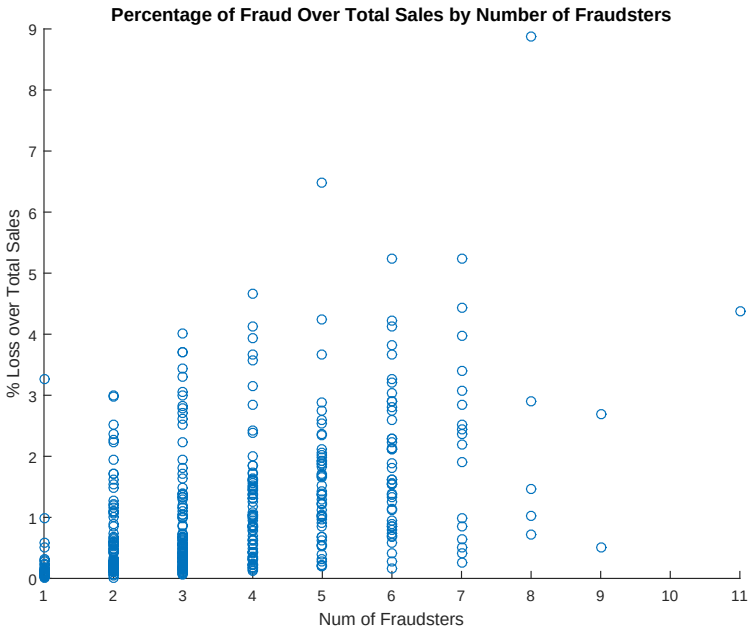


Figure 6.5: *Percentage of fraud divided by total sales grouped by number of fraudsters*

Finally, after having all the information available, a fraud investigator can decide on the goal of the investigation and calculate the effort needed to investigate each of the categories and establish new thresholds according to business strategy to detect and prevent fraud, with the certainty that the chosen triage fraud detection strategy will cover a specific fraud behaviour, and therefore minimise the risk of a big loss.

For example if the goal is to detect as much as possible without considering cost, we can set up very flexible thresholds to achieve this goal. If on the other hand the goal is to catch only substantial amounts of losses we can filter out the staff that has not had enough total refund value from the flagged staff to not spend any resources investigating them. However, if the goal is to deter thieves from stealing, then we can think about prosecuting even minor fraud that become evident and hence easy to flag due to a

Figure 6.6: *Percentage of loss divided by Total Sales vs Detected*

higher proportion of refunds versus sales.

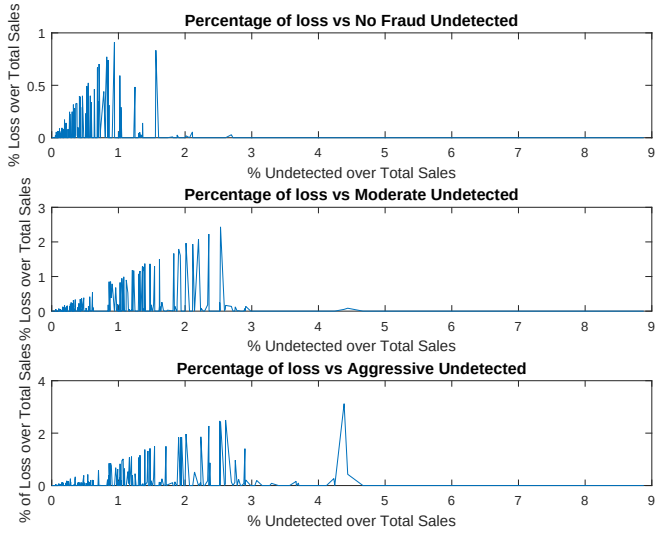


Figure 6.7: *Percentage of loss divided by Total Sales vs Undetected*

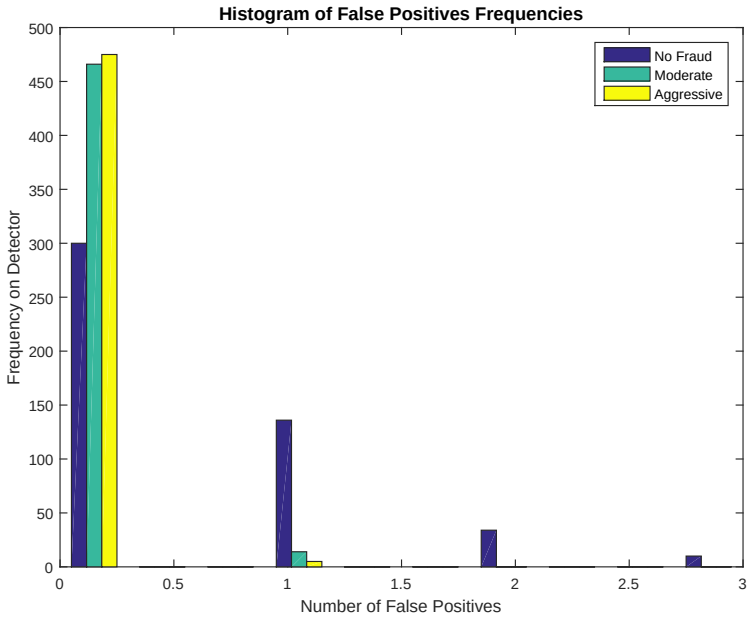


Figure 6.8: *False Positives Frequency on Different Triage Models*

6.6 Using the Triage Setup

Now we are ready to start using the threshold for triage. In the previous section we set the triage limits using 3 different ways (Table 6.1): using no fraud data (rs5125), moderate fraud behaviour (rs3712) and aggressive fraud behaviour (rs3302). In this section we make use of the RetSim simulator to simulate enough probable scenarios where the fraud can vary from none to aggressive fraud that rises up to nearly 9% of total sales as shown in fig. 6.5.

Using similar settings as the previous simulation we simulated scenarios for 48 different fraud behaviours each one 10 times randomly, which in total were 480 data sets. We did this by changing 3 different parameters of the fraud behaviour: the number of fraudsters, the minimum article value they are willing to defraud, and finally the frequency with which they steal and keeping the same parameters of sales for each store.

Each simulation is based on a store with 36 salesmen that according to different situations could work full time, part time or seasonally. The different circumstances vary the total amount of refunds considerably and the quantities sold by each salesman. This is the main reason for using *the fraction of total value of refunds divided by total value of sales per each staff* as a Fraud Score in our fraud detection model. We assumed that all the salesmen should perform at or below approximately the same value with this indicator. Any deviation is due to possible abnormal or fraudulent behaviour.

With these scenarios the goal is to study how much fraud we can catch on average, and if we can keep the loss of the business below a certain level.

We will primarily focus on the red flags identified by the triage process since in most of the cases this would be the main focus of investigations, and also where most of the fraud investigation resources are placed.

Figure 6.6 compares the effectiveness of the different methods. A perfect detection method should detect all possible cases of fraud, and as expected a low threshold such as the one for no fraud can detect almost all frauds. However there is a cost of detecting all fraud, which is the presence of false positives (see figure. 6.8).

A false positive in the context of a retail store might incur cost and difficulties to investigate all possible salesmen that matches the low suspicious criteria and perhaps if they notice it, an internal conflict due to the submission of staff to unwarranted suspicion.

One important fact about fig 6.6 is that the moderate and aggressive models used to calculate the triage limits are sufficient in most of the cases, to keep the loss to a maximum of 3% of total sales in the worst case, and most often below 2% of total sales, which are commonly accepted figures for fraud risk (see figure 6.7).

6.7 Discussion

The biggest threat to the validity of our research is concerning the synthetic data. Many would think that using a synthetic data set is not the same as using the original one. That is true to some extent. A synthetic data set is an abstraction of the original data set. The main goal in our case is to preserve the privacy of customers and business by keeping the original source secret but allowing third parties to interact with the essence of the data in order to provide access to the business through a simulation tool that aids to develop a layer of fraud protection.

Now moving to the retail store business, one of the biggest questions when inventory is missing is: where did it go? Missing inventory directly affects the revenue of many retail operations, especially in markets with lower margins. The causes can vary from customer theft, to staff fraud. In this paper we focus on the study of losses due to fraudulent refunds by the staff.

Gathering evidence of this fraud is a difficult task, specially when there is not a clear starting point. The triage process came about in wartime where medical first responders needed a way to prioritise how to allocate resources to wounded soldiers on the battle field. By analogy, using a similar triage process in the retail setting, when presented with evidence of loss due to fraud in the form of missing inventory, can be a useful way to allocate scarce investigative resources.

The goal of the investigation is an important variable when setting up the thresholds for triage. Often the investigation of small losses is costly for the business and we can reduce the number of investigations to those where the amount is substantial enough to affect the bottom line.

After the triage process is properly calibrated to fit the goals of the retailer, then a proper investigation can be carried out. There are many tools available for performing such an investigation, and discussing these go beyond the scope of this paper, but one could for example match the time stamps of refund receipts to video surveillance records of the cash

register, to see if the receipts match the expected transaction.

The main concern of the retail fraud executives is to keep the business loss as low as possible. Reducing the cost of fraud is a constant process that requires business resources. Since the final cost of the fraud and the investment in fraud detection ends with the customer, managers have an important role to play in this process. They can either go for the detection of the minimum case of fraud with big investments, or accept that part of the business is to keep running at a certain low level of fraud which does not deeply affect the business end customer. The total percentage defrauded by all agents is in our aggressive simulation around 1.09% of total revenue. Most businesses would consider this an acceptable loss rate because it will not severely affect the final customer.

6.8 Conclusions

The RetSim simulator is a useful tool to implement a triage process based on a suspected fraud detection scenario. Without knowing the loss for a specific fraud scenario, retail loss prevention executives are basically left to set up arbitrary thresholds for the triage process. When using the RetSim simulator, they can model the expected cost of fraud by simulating and analysing a synthetic data set with already identified instances of fraud.

Developing a proper and effective triage process without knowing all the underlying details of the expected fraud is difficult. By using the RetSim simulator we can gather enough information for starting a digital forensic investigation since we can model and investigate how different parameters affect the situation.

For threshold detection to be effective, the reference data set that is used to calculate the cut off should contain enough fraud data. Otherwise the range of the categories for a triage process are in risk of being so slim that they can not detect any fraud or perhaps the fraud is outside of the region delineated by the thresholds.

The generation of synthetic fraud is necessary in many cases, to properly set up the triage categories. Our triage set up method substantially reduce the scope of investigation by using triage optimization based on simulated data generated from expected fraud scenarios. It is a management decision to properly set up the threshold according to the resources available for investigating more or fewer cases of suspicious staff returns, given the accepted overall loss to the business. It is off course a task for the managers to decide on the different thresholds to correspond with their business goals, which could be i.e. to maximise the fraud detection while avoiding the cost of invested resources for investigating many false positives, or keeping the total losses below 2% of turnover annually etc. In our examples above, we meet that level.

Further research on this topic will be focused on simulating new types of fraud and at the same time apply these results on the real business to measure the effectiveness of detecting simulated fraud versus real fraud *in vivo*. Another direction of research is to identify other domains where similar methods as those presented in this paper can be applied to circumvent the *data secrecy problem* such as when studying bank transactions, mobile payments and similar financial services.

Acknowledgements

This work is part of the research project "Scalable resource-efficient systems for big data analytics" funded by the Knowledge Foundation (grant: 20140032) in Sweden.

7

Applications of the PaySim simulator for fraud detection research in a financial mobile money service

Edgar Alonso Lopez-Rojas and Stefan Axelsson

Abstract

There is a lack of public available datasets on financial services and specially in the emerging mobile money transactions domain. Financial datasets are important to many researchers and in particular to us to perform research on in the domain of fraud detection. Part of the problem is the intrinsic private nature of financial transactions, that leads to lack of public available datasets.

Fraud inspectors are drowning in real fraud data. They are losing the opportunity that qualified people from the research community contribute to their task due to the impossibility to share private datasets. This problem lead to a situation where only researchers inside organizations that provide financial services are able to perform actual research on fraud detection. This paper proposes an approach that consist in using aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behaviour to later evaluate the performance of fraud detection methods. We built a tool that we named the PaySim simulator.

PaySim is a financial simulator that simulates mobile money transactions

based on an original dataset. In this paper, we present a solution to ultimately yield the possibility to simulate mobile money transactions in such a way that they become similar to the original dataset. With technology frameworks such as Agent-Based simulation techniques, and the application of mathematical statistics, we show in this paper that the simulated data can be as prudent as the original dataset for research.

7.1 Introduction

Fraud is a common threat in financial services. Some of the more commons frauds are committed using stolen credit cards, online banking identity and social engineering for perpetrating elaborate scams that induce the victim into voluntarily sending money to the scammers. Financial companies are providing new ways to facilitate the commercial exchange between people every day. One of these financial services that are becoming popular is the Mobile Money Service.

For instance, in many parts of Africa the adoption of mobile money services as a means of sending & receiving funds using a mobile phone have improved the life of merchants and customers alike. In Tanzania for instance, which according to the world bank is one of the fastest growing economies in the world, the adoption of mobile money as a solution for creating payments has had a positive effect on the overall economy. During December 2013 alone, 100 million transactions were made in total netting a volume of $1.8 billion dollars [58].

Obtaining access to data sets of mobile transactions for research is a very hard task due to the intrinsic private nature of such transactions [37]. Scientists and researchers must today spend time and effort in obtaining clearance and access to relevant data sets before they can work on such data set. This is time consuming and distracts researchers from from focusing on the main problem which is developing and improving their methods, performing experiments on the data and finding novel ways to solve problems such as the problem that inspired this paper which is the fraud detection on financial data. Fraud inspectors on the other hand are

drowning in real fraud data. They are losing the opportunity that qualified people from the research community contribute to their task due to the impossibility to share private datasets.

The work shown in this paper is the continuation of our work in this field and presents the development of a tool and a method to generate synthetic data that we named *PaySim* [41]. PaySim generates synthetic datasets similar to real datasets from mobile money transactions. This is done by the means of computer simulation, in particular, agent based simulation. Agent based simulation is of great benefit in this particular context because the models created represent with accuracy the human behaviour during transactions and are flexible enough to easily be adapted to new constraints. In this paper we improved and extend the PaySim model to include fraud behaviour and a study of fraud by measuring the cost and the economical impact of different fraud detection methods.

PaySim simulates mobile money transactions based on a sample of real transactions extracted from the logs of a mobile money service implemented in an African country. The logs were provided by the multinational company Ericsson (ericsson.com), who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world.

With the help of a statistic analysis and a social network analysis PaySim is able to generate a realistic synthetic dataset similar to the original dataset. PaySim models not only the customers behaviour but the fraudulent behaviour using malicious agents that follow known criminal patterns. By doing this, the resulting dataset is a rich source of data for researchers to perform different sort of test and evaluate not only the performance of fraud detection algorithms, but to measure the cost of fraud, which is otherwise an estimation on the real dataset.

The scope of this paper covers the design and construction of the simulator as well as the evaluation of the quality of the data generated. We use the PaySim simulator and inject malicious fraud behaviour in order to show the different applications and uses of this tool for fraud detection research.

Outline This paper is structured as follows: Section 7.2 presents the background and previous work in simulating financial data. Section 7.3 states the problem. We introduce the fraud scenarios in section 7.4 and during sections 7.5 and 7.6 we present the implementation of PaySim and the results of the simulations. Finally section 7.7 present the conclusions and future work.

7.2 Background and Previous Work

The use of Mobile Money Transfers have grown substantially in the last few years and have attracted greater attention from users, specifically in areas in which banking solutions may not be as procurable as in developed countries. Many providers of mobile money services have been working in several and similar solutions over the past years. There are existing mobile money services in more than 10 African countries which coverage of 14% of all mobile subscribers [55].

The ever growing usage of mobile money has increased the chances and likelihood of criminals to perform fraudulent activities in an attempt to circumvent the security measures of mobile money transfers services for personal financial gain. There is therefore a great amount of pressure on researching the potential security pitfalls that can be exploited with the ultimate goal to develop counter-solutions for the attacks.

Due to the large amount of transactions and the ever changing characteristics on fraud. The most of the measures against fraud start when the customer issue a complain. Many current system still base their detection mechanism on simple thresholds assigned arbitrarily. Therefore there is a need to push forward and investigate the effect of fraud and stop the wrongdoers from profiting from their fraud.

With *PaySim*, we aim to address this problem by providing a simulation tool and a method to generate synthetic datasets of mobile transactions. The benefits of using a simulator to address fraud detection was first presented during our previews work in [33, 39]. This research states

the problem of obtaining access to financial datasets and propose using synthetic datasets based on simulations. The method proposed is based on the concept of MABS (Multi Agent Based Simulation). MABS has the benefits that allows the agents to incorporate similar financial behaviour to the one present in domains such as bank transactions and mobile payments.

Our first implementation of a simulator for financial transaction was introduced in 2012 with a mobile money transactions simulator [35]. This simulator was implemented due to the difficulties to implement a proper fraud detection control on a mobile money system that was under development and that did not produce at the moment any real data sets to use for this research. This was the first paper to present an alternative to the lack of real data problem. The synthetic dataset generated by the simulator was used to test the performance of different machine learning algorithms in finding patterns of money laundering. In this paper we continue with this work. After we obtained access to a real data set of transactions we built a better model and calibrated the model to evaluate the results against the original data set.

The work by Gaber et al. [21] introduced another similar technique to generate synthetic logs for fraud detection. The main difference here was that this time there was available real data to calibrate the results and compare the quality of the result of the simulator. The purpose of this study was to generate testing data that researchers can use to evaluate different approaches. This works differers significantly from our work because we present a different method for analysing the data place special attention on evaluating the quality of the resultant synthetic data set.

The work on fraud detection in mobile payments by Rieke et al., Zhdanova et al. [55, 66] is done in a similar domain as the work by Gaber et al., Lopez-Rojas and Axelsson [21, 33].

Rieke et al. uses a tool named Predictive Security Analyzer (PSA) with the purpose of identifying cases of fraud in a stream of events from a mobile money transfer service [55]. PSA is based on a dataset of 4.5 million logs from a mobile money service over a period of 9 months. They

use simulation due to the limitation and knowledge of existing fraud in the current logs. The main focus on PSA is to detect money laundering cases that are cause by the interaction of several users of the system in an attempt to disguise the fraud among the normal behaviour of the clients. As a result the paper shows that PSA is able to efficiently detect suspicious cases of money launder with the aim of automatically block the fraudulent transactions.

Zhdanova et al. [66] is a continuation of the work done by Rieke et al. [55] and uses the simulator developed by Gaber et al. [21] to evaluate the results. Semi-supervised and unsupervised detection methods are applied to a mobile money dataset due to the advantage over supervised methods in this type of data where there is a difficulty in having a training data with known cases of fraud.

There is a previews work done about simulations in the domain of financial transactions for retail stores with the purpose of fraud detection [40]. The work done in that paper is very similar to the work done in this paper. A large collection of data was gathered from one of the Sweden's biggest shoe-retailer. This data was used to produce a simulator called RetSim. RetSim was later used to model fraudulent behaviour from the staff and develop fraud detection techniques. There has been subsequent work on RetSim that produced among other results social network analysis (SNA) which described the relationship between the clients and the staff for each store, measuring the cost of fraud with the purpose of minimize the risk and properly estimate a security budget [32], threshold detection and methods to optimize the setup of thresholds [42] and finally using this thresholds to properly setup a triage model that prioritize fraud suspiciousness [38].

Public databases of financial transactions are almost non existent. However our previous work during the implementation of a simulator called BankSim presents a MABS of financial payments [37]. BankSim is implemented in a similar way as the RetSim simulator and our simulator using in addition to statistical analysis a social network analysis. BankSim is based on the aggregated financial information of payments during 6

months of the two main cities of Spain that was provided by a bank in Spain with the purpose of developing applications of different kinds that benefit from this sort of data. Our work differs from this work because the source of the data and the characteristics of bank payments and mobile transactions are different as presented later in the following sections.

The key common aspect on previous work is the use of the paradigm of "Multi Agent Based Simulation" approach which incorporates into the behaviour of the agents the main customer logic to reach similar results as the real world. It is important to recognize that a simulation is not an actual "replication" of the original data set. Rather, a simulation will with the aid of statistical methods generate a very similar data set of the original data set. The degree in variance will largely be dependent on how the data on the original data set is structured, hence, different simulations based on different seeds will generate different output data sets but consistent with the real world.

7.3 Problem and Method

The problem formulation for this paper tackles the issue of whether the generation of synthetic financial data is sufficient to supersede real financial data whilst simultaneously yield reliable results if the synthetic data is used as the source data set for any research. This is of primary concern for any researcher that wish to perform scientific experiments but does not have access (or only limited access) to a real financial data set.

The main focus and goal for the simulation is to create another completely self-sufficient data set with the goal of having similar statistical properties as the original data set with the advantage of containing known fraud data that behaves with similar pattern as some of the documented fraud instances found in the real system. To yield such results, the simulator must go through several steps to be able to complete.

In order to simulate the mobile money service, we need to properly simulate the different kind of transactions that the system supports. We

decided to cover 7 of the most important transaction types: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

CASH-IN is the process of increasing the balance of account by paying in cash to a merchant.

CASH-OUT is the opposite process of CASH-IN, it means to withdraw cash from a merchant which decreases the balance of the account.

DEBIT is similar process than CASH-OUT and involves sending the money from the mobile money service to a bank account.

PAYMENT is the process of paying for goods or services to merchants which decreases the balance of the account and increases the balance of the receiver.

TRANSFER is the process of sending money to another user of the service through the mobile money platform.

There are other types of transactions such as the creation of VOUCHERS and redemption of them. We decided to exclude from the scope of this paper due to the low percentage of instances found in the source logs.

Once the simulator is built containing all instances of normal customer behaviour, the next step is to model known fraud patterns that interact with the rest of the normal customers and affect their accounts through fraudulent methods.

The final step is to tweak the parameters for normal and fraud behaviour to generate different fraud scenarios that will produce synthetic datasets ready to use and perform the different experiments such as the evaluation of performance of different fraud detection methods.

As a summary our method follow these steps in order to use the simulator and perform the experiments:

1. Obtain a sample of the real data.
2. Perform a data analysis to extract aggregated information that feed the PaySim simulator.
3. Add parametrization about expected fraud scenarios.
4. Run the simulator several times using different seeds and/or different fraud configurations.
5. Apply the fraud detection methods on the generated synthetic dataset.
6. Summarize the results and performance of the experiments.
7. Repeat from step 3 for different fraud scenarios.

Since privacy is one of the concerns in many organizations, a researcher can start working on step 3 when the aggregated information has already been extracted from the data sample by someone inside the financial company.

Another place to start doing research is at step 5, some researchers do not need to use the simulator at all, but can benefit from the synthetic datasets generated. It is one of the aims of this project that researchers are able to compare results on the same dataset. If the simulation output is shared in a repository others can benefit and have a standard comparison of their methods.

7.4 Fraud scenarios

The mobile money service has many fraud threats, some of them come from the merchants, the customers, the insider in the organization, hackers and the common thief. In this paper we will discuss two of the main categories of threats that have been identified by the mobile money service security experts. Both fraud methods involve cashing out money out of the service through merchants. Cashing out money is the easiest way for the fraudsters

to obtain profit and avoid the risk of frozen accounts due to detection. In the first method the customers loses complete access and control to their account, the second method involves the scams to the customers. For the purpose of this paper we have modelled and implemented only the first method, which involves the customer to lose access of his/her account. However is is completely possible in a future work to extend the model and cover more of the fraud schemas presented in this section and the newly discovered.

7.4.1 Lost account

For this method to work, the criminal needs to use one of the different methods to obtain the access to the mobile money account of the customer. Methods such as SIM physhing swap, fake support calls to obtain pin reset and stolen phones are the most common. Once the fraudsters gain access to the account the next step is to empty the victim's account by either transferring money to mule accounts (that will subsequently cash out the profit) or directly using a merchant to cash out the maximum allowed. When the balance of the victim exceeds the maximum allowed, several mule accounts are used for collecting the money.

7.4.2 Scammed customers

Customers are the usual target for scams in services that involve money. There are many ways to scam the customers. Some of the methods take advantage of the "good will" of many customers and obtain credentials, voucher codes and other important information to perform operations in the system.

Merchants are the intermediaries between the customers and the mobile money services, they provide additional services to the customers such as cash in, cash out, vouchers expedition and claim. These operations are always at risk when the merchant is involved in the scam. Some of the common scams performed by the merchants consist in avoiding or faking receipts of transactions to pocket cash that otherwise should go into the customers account.

Vouchers are usually needed when a user of the mobile money service wants to transfer money to a person that is not using the service. Vouchers are specially prone to fraud due to the asynchronous nature of the transfer. This situation brings a "race condition" due to the window of opportunity for a third person to claim the money before the intended receiver can perform such operation. Some of the common scams are: when a fraudster read the voucher code from the customers phone and use a third person to claim the voucher; when a merchant pretend to create a voucher code but instead send an already claimed code to the victim; when the customer is giving a fake SMS voucher code that will not work for the intended receiver; when the merchant claims that the code is being used but instead is him who send the code to a third person to be claimed.

7.5 Modelling the system

PaySim uses the MABS toolkit called MASON version 19 which is implemented in Java [43]. We selected MASON because it is: multi-platform, supports parallelism, and fast execution speed in comparison with other agent frameworks. This is especially important for multiple running and computationally expensive simulations such as PaySim [54].

The design of PaySim was based on the ODD model introduced by [23]. ODD contains 3 main parts: *Overview*, *Design Concepts* and *Details*. The original design of PaySim has evolved over the time with different publications, we first designed a simulator without calibration due to lack of real data [35], we then designed a full calibrated simulator [41] and now in this paper we extend the model to include fraud behaviour.

7.5.1 ODD Overview

The purpose of this simulator is to simulate payments done in the realms of mobile transactions. The simulator should ultimately perform simulations in such a way that synthetic data in regards to mobile transactions can be generated. The simulator should generate synthetic data that is very similar to a batch of real transactional Data provided by Ericsson. The

goal is to have a generator that can produce data on the fly that can later be used by the scientific community in an attempt to research more about fraud detection.

The model has one primary type of Entity which is *Client*. Each client has a profile that describes the allowed behaviour for the client such as the limit on transactions daily/yearly, the transaction limit and the maximum balance for the client. Furthermore the number of transactions, withdrawals, transfers and deposits is stored for each client. Each client has a base currency in which the transactions are based upon. The client can perform transactions in the form of deposits, withdrawals and transfers. For every transaction that is made, it is stored and saved within the system.

The client has several processes that alter their internal states. For each step that is made by the simulator, based on a random variable that is contingent on calculated probabilities, a type of transaction that is to be performed by the client is chosen. A deposit transaction will increase the balance of the client, a withdrawal will decrease the balance of the client and a transfer transaction will withdraw money from the original client and then deposit them to the destination client in question.

Besides the client the are two other important entities. The first one is the merchants, who are in charge of delivering additional services to the clients such as cash in and cash out operations. The final entity that we used on this simulation is the fraudster. A fraudster is an agent that has as a main purpose to acquire control of the victims account to empty their balance. A fraudster uses one or several accounts as mule to temporarily receive the stolen money before it is cash out of the system.

As a summary we have three main entities or agents in the system: Clients, Merchants and Fraudsters. **Clients** are the normal customers of the system, **merchants** play a passive role during the simulation and only serve the clients in certain operations and finally the **fraudsters** are the threat to the system and the principal focus of our study in fraud detection.

7.5.2 Design Concepts

The basic design concepts is the emerging relation between customers-customers and customers-merchants. From this relation and the different nature of the financial operations we can observe several variation in an important variable which is the balance. The balance represent how much money a customer keeps into his mobile money account.

The concepts that are behind the model are extracted from a statistical analysis of a large batch of real data. From this batch of data, probabilities of each action were obtained and incorporated into the model to generate synthetic information as close as possible to the real data. The client agent has some adaptive behaviours that will alter their way of acting; for instance if the client has reached its daily limit it cannot withdraw money any more for that day. This adaptive behaviour is a direct result of the *transfer* process mentioned above. There is interaction between agents since there is a probability that at a particular step of the simulation, an agent might transfer money to another agent and thus alter its and the other agents state.

To study the phenomenon of fraud we use the information from section 7.4 to model the case scenario where the customer losses control of his/her account.

7.5.3 Details

In this section we describe and detail the parts that are required to build the PaySim simulator such as the inputs, the initialization, the execution and the outputs.

7.5.3.1 Inputs

There are multiple inputs required in order for the simulator to function smoothly. As initial input, the number of clients neighbours for each agent is assigned. The profile for each agent is then further attached based on a probability. Their location on the spatial space along with their neighbours

is also randomly initialized. Some of the inputs used by PaySim are listed here:

Parameter File: This is the file that contains all of the needed parameters that the simulator needs to initiate. Among these parameters we find the seed and perhaps the most relevant of which is the paths for where the input files and the output files are placed on the current machine.

Aggregated Transaction File: This file contains the distribution of the transactions from the original data set. More precisely, it contains how many transactions were made at any given day/hour combination (step). what is the average price for that, what type of transaction it was etc. This is of paramount importance for the accurate results of the simulator since the synthetic data is generated from the information gathered from this file.

Repetitions File: This file contains the frequency of transactions that the original clients had per type of transaction. This means that some of the agents are schedule more than others based on a social network analysis of the indegree and outdegree of the customers.

Fraud Parameters: These parameters specify the number of fraudulent agents as well as the different probabilities to perform fraud and max/min amount of money for attempting to perform a fraud.

Since the simulator is using MASON as the framework for performing the simulation, it is important to define how each step is going to map real world time. For this simulation we defined that each day/hour combination represents one step. At each step, a Client that represents the agent for the simulator is generated. The client will be placed in an environment in which it is to make decisions based on the information it perceives. The Client is created with the statistical distribution of the possibilities to perform each transaction type for a specific day/hour combination. The client then randomly perform (based on the distribution initiated) different transaction types in relation to the other clients on the simulator. Also, for each client generated, there is a probability **P** for the client to make

future transactions at later steps. This probability is gathered from the database of the original data set.

7.5.3.2 Initiation Stage

In this stage, the PaySim simulator must load the necessary input data described in section 7.5.3.1. The first and most important step is to load the values for each parameter in the parameter file. These will among other things contain the file paths for the source data inputs that the simulator needs to load.

Apart from the statistical distribution for each transaction type input to the client, there is another important input, which is the initial balance of the clients. Upon the generation of each client in the simulation, there must be an initial balance attached to that client. Besides the clients, the merchants and the fraudsters are also initialized based on the parameters.

7.5.3.3 Execution Stage

Upon completion of the Initiation Stage when all the parameters are successfully loaded, the simulator can now proceed to the execution stage. It is at this stage that the simulator will perform the actual simulation that lead to the simulated transaction results.

The agents are the founding blocks of the "Agent Based Simulator". The agent in this context, resembles the clients, the merchants and the fraudsters. Upon each step of the simulation, the PaySim simulator will convert each step to a "Day/hour" combination. This will then be used as an input to extract the statistical distributions from the original data set. Based on the *Aggregated Transaction File*, PaySim harness the probability **P** of performing each each transaction in the simulator and save it into the model of the client. With this information, the client now has gained more knowledge and will know the following important things:

Number Of Transactions: This is the total number of transactions that this generated client will do.

Make Future Steps: This is the information of whether the client is to participate in future steps. Which means scheduling the tasks of performing more transactions during further steps.

Statistical Distribution: This is the different probabilities that the client will have loaded into it which entails the probability **P** of performing each action.

Initial Balance: This will be the initial balance that the client will have once generated.

After each client is generated, the client will make the decision of what type of transaction it will ultimately make, again this is completely derived from the distribution loaded. The client is in an environment which allows it to freely interact with other clients in the simulation. There are some types of transaction types that is based on that, like "TRANSFER" for instance. The "TRANSFER" type is exchange of money from one client to another; hence, the client will have to interact with other clients to simulate the actual exchange of funds.

The merchants play a passive role during the simulation and the only functions they have is to serve the clients during cash in and cash out transactions and the fraudsters during the cash out operations to fraudulent profit from their victims.

A fraudster will sense nearby clients and perform attempts to take control of their accounts. Upon succeeding, a fraudster will start to empty their accounts either by using a merchant to directly cash out or transferring money to mule accounts which in a short period of time will be also emptied through merchants and the cash out operations.

7.5.3.4 Finalization Stage

After each of the agents have completed their role in the simulation and performed all of the actions the results must be saved. There are 4 outputs generated after each simulation. All of which serve a specific purpose which

allow a researcher to further test the quality of the generated data and save the configuration of the simulation with the in order to be able to repeat the simulation with the exact initial properties and results.

Logfile: Each transaction that is made will contain a record with the meta-data for that transaction. Data such as what client performed which action, to which other client, the sum of the transaction, and the delta in balance for all clients involved. Each such record will be saved in a logfile unique for the specific simulation.

Database: Apart from the logfile, the record for each transaction will also be saved into a database. The purpose of which is to allow for easier queries when the analysis of the resulsts is to be made.

Aggregated Dump An aggregated dump that is similar to the original aggregated dump from the original data set will also be generated. It is these two files that will be used to generate the plots and graphs resembling the results of the transactions.

Parameter File History This file will contain the exact properties needed for the simulation to be able to reproduce the exact same results again. This is important because each simulator must be able to be reproduced again, and without the original "seed" used, it will not be possible.

7.6 Results

The results are divided in two parts, the first part shows the results of the calibration which relay on our previous work [41]. The second part shows the results of the simulation after injecting the fraud scenarios and performing the fraud detection analysis.

For both parts we ran PaySim several times using random seeds for 744 steps, representing one month of real time data, which matches the time with the original logs. Each run took around 45 minutes on an i7 intel processor with 16GB of RAM. This time includes the interaction of the

agents, the writing of the results on a log text file and the loading of this file to a MySql database. The final result of a run contains approximately 24 million of transactional financial records divided into the 5 types of categories presented before.

7.6.1 Calibration and Evaluation

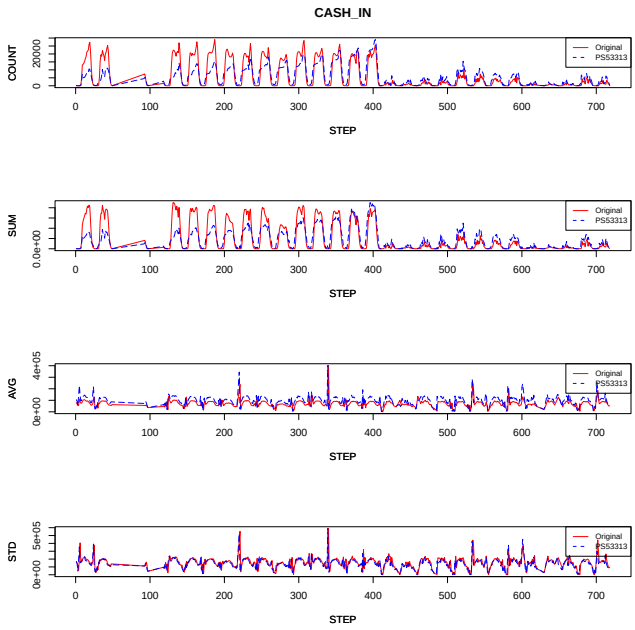
The first goal with PaySim is to produce a dataset that resembles the original one. For doing this, we selected the generated dataset that contained the lowest difference in values in comparison with the original dataset provided. The evaluation of the quality of the database was first calculated using the sum of square error (SSE) method on the quantities of the different datasets. Despite all simulations being fairly consistent, there are small differences due to the random seed selected, the one with the lowest error was *PS53313*. The selected synthetic dataset was named *PS53313* after an arbitrary random log name. Table 7.1 shows the types of transactions, count and average amount generated with the simulator. The amount values are given in an African currency that we can not disclose.

Table 7.1: *Simulated synthetic dataset PS53313*

Type	**Count**	**Total Amount**	**avg**
CASH_IN	4,941,188	821,047M	166,164
CASH_OUT	8,469,357	1,453,189M	171,582
DEBIT	117,365	612M	5,216
PAYMENT	8,889,664	114,267M	12,854
TRANSFER	2,148,905	1,875,323M	872,688

In order to verify that the simulation was working properly we plotted the distributions to visually identify significant differences between the original and the synthetic dataset. Figures 7.1, 7.2 and 7.3 show the visualization of three types of transactions (CASH IN, CASH OUT and TRANSFER). Each figure contains the output for each step regarding the count of transactions, the total sum of transaction, the average and the standard deviation. The red continuous line represent the original data distribution and the blue dashed line represent the synthetic dataset

Figure 7.1: *Visualization of transaction type CASH-IN*



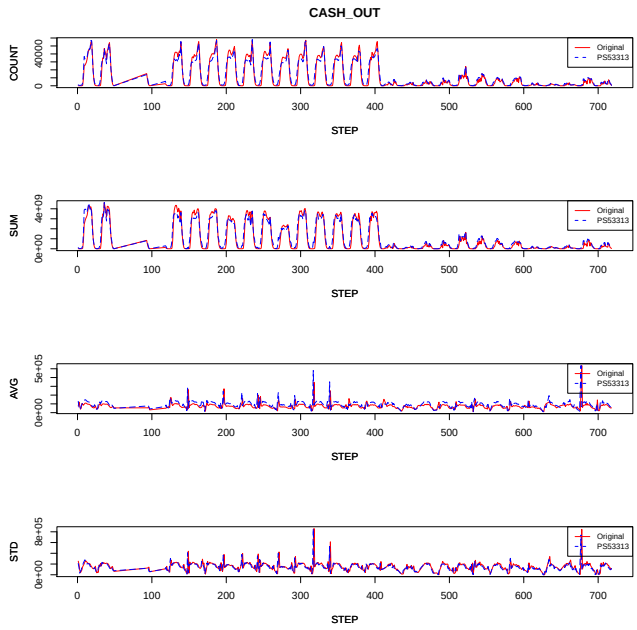
PS53313. We excluded the other types of transactions since they are presented in our previous work [41] and for our fraud detection study we used only CASH OUT and TRANSFER types.

Something we noted is that during the first 14 days of the simulation the activity in the system is higher compared to the remaining days. This is perhaps a phenomenon present due to the introduction of income during the first days of the month or in the worse case missing logs from the original data.

7.6.2 Fraud Scenarios

The scenario selected is based on the first fraud case presented in section 7.4 which happens when the client loses control and access to his/her account. The fraudster takes control and uses disposable mule accounts to transfer

Figure 7.2: *Visualization of transaction type CASH-OUT*

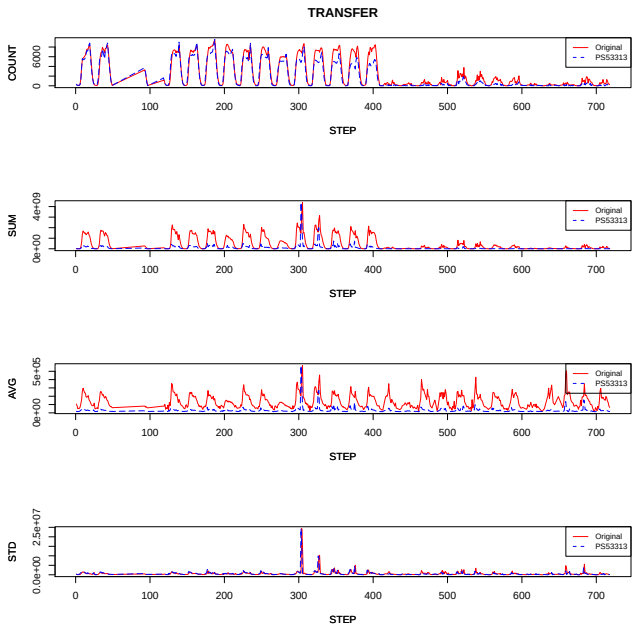


the money and later cash them out of the service. All of this can only happen in a very short time, because when the clients discovers that their accounts are compromised the first action they should do is to contact customer service to block any possible malicious transaction activity. Since the current fraud detection reacts only after a customer complain, we want to study how much money can we prevent our customers from losing after applying our fraud detection mechanism presented in the following section.

The experiment introduces a 3% probability for each of the 1000 fraudsters to perform fraud at any giving step of the simulation, which is perhaps an aggressive value (30 fraudulent activities per hour), but this helps to inject enough data to study the phenomenon.

In order to study this phenomenon, we run the system four times increasing the maximum amount of transaction possible in a single TRANSFER

Figure 7.3: *Visualization of transaction type TRANSFER*



each time. We selected four synthetic datasets that used the thresholds on transfer transactions of 300k (PS89745), 600k (PS80775), 900k (PS00273) and 1200k (PS98516). The first case obligate the fraudster to perform several operations to empty accounts that contains balance above this threshold. The last limit is very flexible and allows the fraudsters to perform a single TRANSFER operation in most of the cases. By implementing an extra control that will temporarily block accounts that exceed three consecutive transfers for the maximum amount in a short period of time we could effectively reduce the amount of fraud and measure the benefit of this control. With the help of PaySim, we can also measure how other users will be affected by this block in their accounts (False Positives).

We ran the PaySim simulator with the same parameter file that generated the dataset *PS53313* for the calibration. Table 7.2 shows the number of transactions type TRANSFER and the classification of fraud. The first

Table 7.2: *Fraud Detection Classification*

LogName	Class	Count	Amount	% count	% amount
PS89745 (300k)	FN	27,412	6,724M	1.005%	0.363%
	FP	982	214M	0.036%	0.012%
	TN	2,607,642	1,816,764M	95.579%	98.162%
	TP	92,211	27,076M	3.380%	1.463%
PS80775 (600k)	FN	24,400	11,291M	0.990%	0.581%
	FP	58	17M	0.002%	0.001%
	TN	2,396,684	1,907,409M	97.239%	98.126%
	TP	43,604	25,114M	1.769%	1.292%
PS00273 (900k)	FN	21,072	12,854M	1.024%	0.768%
	FP	8	1M	0.000%	0.000%
	TN	2,011,006	1,639,699M	97.712%	97.903%
	TP	26,006	22,264M	1.264%	1.329%
PS98516 (1200k)	FN	20,493	16,189M	0.921%	0.858%
	FP	1	0.168M	0.000%	0.000%
	TN	2,186,516	1,849,707M	98.215%	97.993%
	TP	19,248	21,686M	0.865%	1.149%

important and obvious thing to notice is that whenever there is a control, the effort for committing the fraud gets harder. Just by introducing a lower threshold on the maximum amount allowed for a transfer, the number of transactions needed for empty an account increases several times. However, the number of legitimate users that will be affected increases. This is the trade-off in fraud detection that a manager needs to setup in the system.

Table 7.2 also shows the loss due to fraud. If we focus attention on the False Negative (FN) row of each simulation, we can see the profit from fraud. The bigger the threshold the higher the profit. The task for a manager is to reduce this amount while minimising False Positives (FP) cases, which are the legitimate customers that get their account blocked by the controls implemented to prevent the fraud.

Table 7.3 show the fraud detection results of each of the datasets evaluated. We can see that the precision is higher when the threshold

is higher as in the dataset *PS98516* (1200k). This means that we will get lower customers affected. However, the recall is seriously affected, which means that the fraudsters will profit more using this control (16,189 millions).

On the other hand when we have a lower threshold as in *PS89745* (300k), the number of false positives (FP) increases to 982. But, we have a considerable higher recall which means that we lower the total value of fraud (6,724 millions).

Table 7.3: *Fraud Detection Results*

LogName	**Precision**	**Recall**
PS89745	98.946%	77.085%
PS80775	99.867%	64.120%
PS00273	99.969%	55.240%
PS98516	99.995%	48.434%

The data simulated did not contained many instances of false positives. We can think that the criminal behaviour that we modelled is not common among the customers. It is unlikely that we can find customers reducing their balance through consecutive transfer in the real data. This situation happens perhaps because the customers have other options (bank transfers) that are less risky than the method used by the fraudsters in this paper.

7.7 Conclusions

PaySim is a simulation of mobile money transactions with the objective to generate a synthetic transactional data set for research in fraud detection. The data sets generated with PaySim 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 their tests.

The results presented in the section 7.6 help to appreciate that the

generated dataset captures the process and the frequencies of the different transaction types of the mobile money service. We argue that PaySim is ready to be use as a tool to generate synthetic transactions that resemble the original and private data set supplied.

PaySim can generate diverse fraud scenarios and contribute to the elaboration of fraud detection mechanism due to the unique advantage which is the possibility to measure the total cost of fraud. By using PaySim we protect the privacy of the customers at the same time that interesting results are possible to share with other researchers without the constrains and legal boundaries of the original data.

Future work on the simulator is to improve the model of fraudulent agents and cover other different scenarios to test the efficacy and accuracy of diverse fraud detection methods. We also want to make a synthetic data set available to other researchers and be able to compare and share diverse results.

Bibliography

[1] N. Abe, B. Zadrozny, and J. Langford. "Outlier detection by active learning". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 06* (2006), p. 504.

[2] D. Agrawal and C. Aggarwal. "On the design and quantification of privacy preserving data mining algorithms". In: *PODS '01 Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (2001).

[3] M. Ahmed, A. N. Mahmood, and M. R. Islam. "A survey of anomaly detection techniques in financial domain". In: *Future Generation Computer Systems* 55 (2016), pp. 278–288.

[4] S. Alam and A. Geller. "Networks in agent-based social simulation". In: *Agent-based models of geographical systems* (2012), pp. 77–79.

[5] C. Alexandre and J. Balsa. "Integrating client profiling in an anti-money laundering multi-agent based system". In: *World Conference on Information Systems and Technologies*. Recife, Brazil, 2016, pp. 931–941.

[6] C. R. Alexandre and J. Balsa. "A multiagent based approach to money laundering detection and prevention". In: *International Conference on Agents and Artificial Intelligence*. April 2016. 2015, pp. 230–235.

[7] R. Allan. "Survey of agent based modelling and simulation tools". In: *Challenges* October (2010).

[8] R. Arora, A. Khan, and E. Deyle. “The Global Retail Theft Barometer 2014”. In: *Thorofare, NJ USA: Checkpoint Systems, Inc.(1-81). Acedido em* (2014), p. 60.

[9] S. Axelsson. “The Base-rate Fallacy and the Difficulty of Intrusion Detection”. In: *ACM Transactions on Information and System Security (TISSEC)* 3.3 (2000), pp. 186–205.

[10] M. Bastian, S. Heymann, and M Jacomy. “Gephi: An open source software for exploring and manipulating networks”. In: *International AAAI conference on weblogs and social media* 2 (2009).

[11] S. Benson Edwin Raj and A. Annie Portia. “Analysis on credit card fraud detection methods”. In: *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*. IEEE, 2011, pp. 152–156.

[12] R. Bolton and D. Hand. “Statistical fraud detection: A review”. In: *Statistical Science* 17.3 (2002), pp. 235–249.

[13] R. R. J. Bolton et al. “Statistical fraud detection: A review”. In: *Statistical Science* 17.3 (2002), pp. 235–249.

[14] J. Bovinet. *RETSIM: A Retail Simulation with a Small Business Perspective*. Minneapolis/St. Paul: West Pub. Co., 1993.

[15] Z. Chaczko and C. Chiu. “A smart-shop system - Multi-agent simulation system for monitoring retail activities”. In: *20th European Modelling and Simulation Symposium* (2008), pp. 20–26.

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

[17] E. C. B. ECB. *Recommendations for the Security of Internet Payments*. Tech. rep. January. 2013, pp. 1–16.

[18] A. Evfimievski et al. “Privacy preserving mining of association rules”. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’02* (2002), p. 217.

[19] FBI. *Ticket Switch Fraud Scheme at Home Deopt.* 2013.

[20] P. Gabbur et al. “A pattern discovery approach to retail fraud detection”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 11* (2011), p. 307.

[21] C. Gaber et al. “Synthetic logs generator for fraud detection in mobile transfer services”. In: *2013 International Conference on Collaboration Technologies and Systems (CTS)* (May 2013), pp. 174–179.

[22] D. Gorton. “IncidentResponseSim: An agent-based simulation tool for risk management of online Fraud”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Ed. by S. Buchegger and M. Dam. Vol. 9417. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 172–187.

[23] V. Grimm et al. “A standard protocol for describing individual-based and agent-based models”. In: *Ecological Modelling* 198.1-2 (Sept. 2006), pp. 115–126.

[24] Y. Hu. “Efficient and High Quality Force-Directed Graph”. In: *The Mathematical Journal* 10 (2005), pp. 37–71.

[25] J. Hunt. “How terrorist organizations use cyberlaundering to fund their activities, and how governments are trying to stop them”. In: *Information & Communications Technology Law* 20.2 (June 2011), pp. 133–152.

[26] H. Kargupta, S. Datta, and Q. Wang. “On the privacy preserving properties of random data perturbation techniques”. In: *Third IEEE International Conference on Data Mining* (2003), pp. 99–106.

[27] E Kirkos, C Spathis, and Y Manolopoulos. “Data Mining techniques for the detection of fraudulent financial statements”. In: *Expert Systems with Applications* 32.4 (2007), pp. 995–1003.

[28] 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, p. 340.

[29] P. Lin, B. Samadi, and A. Cipolone. “Development of a synthetic data set generator for building and testing information discovery systems”. In: *ITNG 2006.* IEEE, 2006, pp. 707–712.

[30] E. Lopez-Rojas and S. Axelsson. “Applications of the PaySim simulator for fraud detection research”. In: *Submitted for Journal Publication* (2016). (Submitted).

[31] E. A. Lopez-Rojas. “On the Simulation of Financial Transactions for Fraud Detection Research”. Licentiate’s Thesis Computer Science. Blekinge Institute of Technology, 2014.

[32] E. A. Lopez-Rojas. “Extending the RetSim Simulator for Estimating the Cost of fraud in the Retail Store Domain”. In: *The 27th European Modeling and Simulation Symposium-EMSS*. Bergeggi, Italy, 2015.

[33] E. A. Lopez-Rojas and S. Axelsson. “Money Laundering Detection using Synthetic Data”. In: *The 27th workshop of Swedish Artificial Intelligence Society (SAIS)* (2012), pp. 33–40.

[34] E. A. Lopez-Rojas and S. Axelsson. “Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)”. In: *Nordsec 2012*. 2012, pp. 1–8.

[35] 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* (2012), pp. 25–32.

[36] E. A. Lopez-Rojas and S. Axelsson. “Banksim: A bank payments simulator for fraud detection research”. In: *26th European Modeling and Simulation Symposium, EMSS 2014*. Bourdeaux,France, 2014, pp. 144–152.

[37] E. A. Lopez-Rojas and S. Axelsson. “Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research”. In: *Advances in Computational Social Science and Social Simulation*. Barcelona, Spain, 2014.

[38] E. A. Lopez-Rojas and S. Axelsson. "Using the RetSim Fraud Simulation Tool to set Thresholds for Triage of Retail Fraud". In: *20th Nordic Conference on Secure IT Systems, NordSec 2015*. Stockholm: Springer, 2015, pp. 156–171.

[39] E. A. Lopez-Rojas and S. Axelsson. "A Review of Computer Simulation for Fraud Detection Research in Financial Datasets". In: *Future Technologies Conference, San Francisco, USA*. 2016.

[40] 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* (2013). (Best Paper Award).

[41] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson. "PaySim: A financial mobile money simulator for fraud detection". In: *The 28th European Modeling and Simulation Symposium-EMSS*. Larnaca, Cyprus, 2016.

[42] E. A. Lopez-Rojas, D. Gorton, and S. Axelsson. "Using the RetSim simulator for fraud detection research". In: *International Journal of Simulation and Process Modelling* 10.2 (2015), p. 144.

[43] S. Luke. "MASON: A Multiagent Simulation Environment". In: *Simulation* 81.7 (July 2005), pp. 517–527.

[44] E. Lundin. "A synthetic fraud data generation methodology". In: *Information and Communications Security*. Berlin Heidelberg: Springer, 2002, pp. 265–277.

[45] D. Magnusson. "The costs of implementing the anti-money laundering regulations in Sweden". In: *Journal of Money Laundering Control* 12.2 (2009), pp. 101–112.

[46] D. Malekian and M. R. Hashemi. "An adaptive profile based fraud detection framework for handling concept drift". In: *2013 10th International ISC Conference on Information Security and Cryptology (ISCISC)*. IEEE, 2013, pp. 1–6.

[47] R. MCHaney. *Understanding Computer Simulation*. Zivonin: Roger McHaney-Ventus Publishing ApS, 2009, p. 172.

[48] A. Member and A. Council. *Reviving retail Strategies for growth in 2009 Executive summary*. 2009.

[49] A. Narayanan and V. Shmatikov. "De-anonymizing Social Networks". In: *2009 30th IEEE Symposium on Security and Privacy* (May 2009), pp. 173–187.

[50] E. E. Ngai et al. "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature". In: *Decision Support Systems* 50.3 (2011), pp. 559–569.

[51] P. Ormerod and B. Rosewell. "Validation and Verification of Agent-Based Models in the Social Sciences". In: *LNCS*. Ed. by F. Squazzoni. Springer Berlin / Heidelberg, 2009, pp. 130–140.

[52] D Phan and F Varenne. "Agent-Based Models and Simulations in Economics and Social Sciences: from conceptual exploration to distinct ways of experimenting". In: *Journal of Artificial Societies and Social Simulation* (2010).

[53] C. Phua et al. "A comprehensive survey of data mining-based fraud detection research". In: *Arxiv preprint arXiv:1009.6119* (2010).

[54] S. F. Railsback, S. L. Lytinen, and S. K. Jackson. "Agent-based Simulation Platforms: Review and Development Recommendations". In: *Simulation* 82.9 (Sept. 2006), pp. 609–623.

[55] R. Rieke et al. "Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis". In: *2013 International Conference on Availability, Reliability and Security*. IEEE, 2013, pp. 662–669.

[56] T. Salamon. *Design of Agent-Based Models*. Zivonin: Tomas Bruckner, 2011, p. 208.

[57] A. Schwaiger and B Stahmer. "SimMarket: Multiagent-based customer simulation and decision support for category management". In: *Multiagent System Technologies* (2003), pp. 74–84.

[58] B. Seetharam and D. Johnson. "Mobile Money's Impact on Tanzanian Agriculture". In: (2015).

[59] A. Sorin. “Survey of Clustering based Financial Fraud Detection Research”. In: *Informatica Economica* 16.1 (2012), pp. 110–123.

[60] A. Sudjianto et al. “Statistical Methods for Fighting Financial Crimes”. In: *Technometrics* 52.1 (Feb. 2010), pp. 5–19.

[61] S. Wang. “A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research”. In: *2010 International Conference on Intelligent Computation Technology and Automation*. Vol. 1. IEEE, 2010, pp. 50–53.

[62] J. West and M. Bhattacharya. “Intelligent financial fraud detection: a comprehensive review”. In: *Computers & Security* 57 (2015), pp. 47–66.

[63] Y. Yannikos et al. “3LSPG : Forensic Tool Evaluation by Three Layer Stochastic Process-Based Generation of Data”. In: *4th International Workshop in Computational Forensics*. Tokyo, Japan, 2010, pp. 200–211.

[64] 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, Sept. 2007, pp. 5514–5517.

[65] Z. Zhang and J. Salerno. “Applying data mining in investigating money laundering crimes”. In: *discovery and data mining* Mlc (2003), p. 747.

[66] M. Zhdanova et al. “No Smurfs: Revealing Fraud Chains in Mobile Money Transfers”. In: *2014 Ninth International Conference on Availability, Reliability and Security*. IEEE, 2014, pp. 11–20.

ABSTRACT

This thesis introduces a financial simulation model covering two related financial domains: Mobile Payments and Retail Stores systems.

The problem we address in these domains is different types of fraud. We limit ourselves to isolated cases of relatively straightforward fraud. However, in this thesis the ultimate aim is to introduce our approach towards the use of computer simulation for fraud detection and its applications in financial domains. Fraud is an important problem that impact the whole economy. Currently, there is a lack of public research into the detection of fraud. One important reason is the lack of transaction data which is often sensitive. To address this problem we present a mobile money Payment Simulator (PaySim) and Retail Store Simulator (RetSim), which allow us to generate synthetic transactional data that contains both: normal customer behaviour and fraudulent behaviour.

These simulations are Multi Agent-Based Simulations (MABS) and were calibrated using real data from financial transactions. We developed agents that represent the clients and merchants in PaySim and customers and salesmen in RetSim. The normal behaviour was based on behaviour observed in data from the field, and is codified in the agents as rules of transactions and interaction between clients and merchants, or customers and salesmen. Some of these agents were intentionally designed to act fraudulently, based on observed patterns of real fraud. We introduced known signatures of fraud in our model and simulations to test and evaluate our fraud detection methods.

The resulting behaviour of the agents generate a synthetic log of all transactions as a result of the simulation. This synthetic data can be used to further advance fraud detection research, without leaking sensitive information about the underlying data or breaking any non-disclose agreements.

Using statistics and social network analysis (SNA) on real data we calibrated the relations between our agents and generate realistic synthetic data sets that were verified against the domain and validated statistically against the original source.

We then used the simulation tools to model common fraud scenarios to ascertain exactly how effective are fraud techniques such as the simplest form of statistical threshold detection, which is perhaps the most common in use. The preliminary results show that threshold detection is effective enough at keeping fraud losses at a set level. This means that there seems to be little economic room for improved fraud detection techniques.

We also implemented other applications for the simulator tools such as the set up of a triage model and the measure of cost of fraud. This showed to be an important help for managers that aim to prioritise the fraud detection and want to know how much they should invest in fraud to keep the loses below a desired limit according to different experimented and expected scenarios of fraud.

ISSN: 11653-2090

2016:06

ISBN: 978-91-7295-329-1