ON THE SIMULATION OF FINANCIAL TRANSACTIONS FOR FRAUD DETECTION RESEARCH

Edgar Alonso Lopez-Rojas

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2014:02
Department of Computer Science and Engineering
On the Simulation of Financial Transactions for Fraud Detection Research

Edgar Alonso Lopez-Rojas
On the Simulation of Financial Transactions for Fraud Detection Research

Edgar Alonso Lopez-Rojas

Licentiate Dissertation in Computer Science

Department of Computer Science and Engineering
Blekinge Institute of Technology
SWEDEN
“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
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 cover more complex types of fraud, such as money laundering, that comprises multiple organisations and domains. Fraud is an important problem that impact the whole economy. Currently, there is a general 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 Simulator (PaySim) and Retail Store Simulator (RetSim), which allow us to generate synthetic transactional data. These simulations are based on real transaction data.

These simulations are multi agent based simulations. Hence, we developed agents that represent the clients 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, 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 results. 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.

Using statistics and social network analysis (SNA) on real data we could calibrate the relations between staff and customers and generate realistic synthetic data sets that were validated statistically against the original.

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

Helena, Isabella and Linnea
Acknowledgements

Many people have contributed to my work and it is 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 supervisor Dr. Stefan Axelsson, for being directly involved in my work and co-author of all my publications. I would like to thank Prof. Bengt Carlsson for his guidance as a senior researcher and immense support for my work. I have to extend my thanks to all doctoral students and researchers at our former research group DISL whom have actively participated in reviewing and contributing to my work with their comments to increase the quality of it.

Blekinge Institute of Technology have been economically supporting my research all over these years as a PhD student. I have only words of gratitude for them. This journey is not yet over and it would had not being started without the initial financial support of Fundacion Colfuturo and Universidad EAFIT in Colombia. I appreciate and thank that they believed in my project and helped to achieve my goals.

Finally, I would like to thank my family. My wife Helena, for her unconditional support and my two daughters, Isabella and Linnea. All together have been an important source of inspiration and motivation to continue my academic journey. I have to extend my gratitude to my parents (Jesus y Soledad), sister (Paula) and all relatives that came to visit me all the way from Colombia and brought part of my home country to my new home country Sweden.

Edgar Lopez
Karlskrona, April 2014

v
Preface

This thesis is based on the work presented in the following four papers. The first three papers are published in peer-reviewed conference proceedings. Paper III obtained the Best Paper Award during the EMSS 2013 conference and was invited in an extended version to a special issue of the International Journal of Simulation and Process Modelling. Therefore, Paper IV is been peer-reviewed and is currently in press. The included papers have been modified to fit this format, but the content is unchanged.

Paper I

Paper II

Paper III
Paper IV

# Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
</tbody>
</table>

## 1 Introduction
  1.1 Background  
  1.2 Aim and scope  
  1.3 Our Simulators  
  1.4 Research Questions  
  1.5 Research Method  
  1.6 Contributions  
  1.7 Related Work  
  1.8 Conclusions  
  1.9 Future Work

## 2 Money Laundering Detection using Synthetic Data  
  2.1 Introduction  
  2.2 Background  
  2.3 Related work  
  2.4 AML for mobile money  
  2.5 Mobile Money simulation  
  2.6 Discussion
# List of Tables

1.1 Contribution of Papers to Research Questions . . . . . . . . . . 17

3.1 Results for the class *money laundering* (suspicious) . . . . . . 52
3.2 Confusion Matrix . . . . . . . . . . . . . . . . . . . . . . . . . 52

4.1 Statistical analysis of five stores during one year . . . . . . . . 60
4.2 Sales clerk frequency . . . . . . . . . . . . . . . . . . . . . . . 62
4.3 Article categories . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.4 Network Analysis . . . . . . . . . . . . . . . . . . . . . . . . . 63
4.5 Statistical Analysis Store one vs RetSim Simulations . . . . . . 73
4.6 Network Simulated . . . . . . . . . . . . . . . . . . . . . . . . . 74

5.1 Network Analysis . . . . . . . . . . . . . . . . . . . . . . . . . 89
5.2 Article categories . . . . . . . . . . . . . . . . . . . . . . . . . 93
5.3 Statistical Analysis of *Store One* vs RetSim Simulations . . . . 96
5.4 Network Simulated . . . . . . . . . . . . . . . . . . . . . . . . . 100
5.5 Fraud Detection Results . . . . . . . . . . . . . . . . . . . . . . 107
5.6 Threshold Fraud Detection . . . . . . . . . . . . . . . . . . . . . 107
List of Figures

2.1 Scenario 1 - 2000 clients distributed across 7 cities and multiple edges connecting clients that produces legal transactions ........ 37
2.2 Scenario 2 - 20 accounts distributed across 3 cities generating suspicious transactions ........................................... 38

3.1 Simplified Class Diagram for the Mobile Money Simulation .. 48
3.2 Simplified State Diagram for the Mobile Money Simulation .. 49

4.1 Store one - sales distribution ........................................ 61
4.2 Store one - number of customers per day ............................. 62
4.3 Store one - Network of customers and sales clerks ............... 64
4.4 Store one - Customers per sales clerks .............................. 65
4.5 RetSim Use Case Diagram ............................................. 68
4.6 Screenshot of RetSim during a step ................................ 72
4.7 Small Simulated network .............................................. 75
4.8 Comparison of simulated vs real data ................................ 76
4.9 Comparison of distribution of simulated vs real data ............. 77
4.10 Box plot of simulated vs real data .................................. 78

5.1 Store One - Network of Customers and Salesmen ............... 90
5.2 Use Case Diagram for the Interaction of Agents in RetSim .... 94
5.3 Visualization of Simulated Network ................................. 95
5.4 Overlap of Two Runs of RetSim vs Real Data .................... 97
5.5 Boxplot of Simulated vs Real Data ................................. 98
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 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 financial transactions 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. 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.

The work presented in this thesis is an effort to address the lack of public available financial data, with the aim that: if we can not get 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
1. Introduction

new payment system that uses mobile phones to easy the payments called PaySim [31]. This system was under development at the moment of our research. Thus, 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 [32]. 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.

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 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. For example, we can simulate the introduction of new supervisors, security cameras, auditing routines, etc. Testing these scenarios in a real world situation introduces risks for the business, for instance, unhappiness amongst the staff, due to trials of an ill advised policy, which leads to even greater expense and unwanted problems.

Our approach is a method to generate anonymous synthetic data of the transactions in a “typical” financial chain, 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. Furthermore, the data set generated could be the basis for research in other fields, such as demand prediction, logistics and demand/supply 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
1.1. Background

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, the aim is to address this goal subsequently.

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 [38].

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 are making computer simulation a hot topic nowadays [38]. Specifically, because almost everybody have access to a computer with enough capacity to run quite large simulations. Large organisations and researchers benefit from the power of supercomputers to run simulations that demands huge amount of computer resources (disk, memory, speedy processing in parallel).

There are many different types of computer simulation: Discrete event simulations, Continuous system dynamics, Agent-Based Simulations and a combination or Hybrid simulation [48]. In this thesis we make use of Agent-Based Simulation-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 [4, 22, 44, 48].

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 Money Payment and customers and salesmen in our Retail Store Simulator. The behaviour is given by the rules of transactions and interaction between clients, or customers and salesmen. Some of these agents are intentionally designed to behave fraudulently by following known or assumed patterns of fraudulent behaviour. Their goals and final behaviour generate a log of synthetic data of all transactions and purchases that are the output of the simulation, and represent the emerging behaviour. These logs are then analysed with the purpose of developing fraud detection methods.

1.1.2.3 Working with Synthetic Data

When using synthetic data for research there are several benefits and threats [30]. The many benefits to be had by working with synthetic data can be summarised by:

- The data that represent realistic scenarios are readily available.
- The privacy of the customer is not impacted.
- The disclosure of results is not affected by policies or legal issues.
- The data set is available for other researchers to reproduce experiments.
- Different scenarios can be modelled with parameters controlled by the researcher.
- Injection of enough abnormal data to address the class unbalance problem.
1. Introduction

- Simulation of abnormal behaviour prevent the problem of mislabelled classes.

- Much more data, and much more varied types of data can be produced at will, than can be collected in the field.

Unfortunately other issues arise that are important to consider when using synthetic data. Some of these issues or disadvantages are:

- The data generated might be neither representative or realistic.
- Data can be biased.
- Data can be unconsciously designed to fit a specific model.
- It is difficult to build a realistic model due to the complexity of the variables and parameters.
- The simulated suspicious data cannot be investigated further. In a real scenario these results could be used for improving the accuracy of the existing classification algorithms.
- It is unknown whether we can transfer the learning from a simulated data set to a real-world data set.

Some of these disadvantages have been addressed in our research by building a simulation that is a realistic real-world representation of the financial domains. It is important to understand that the purpose of the simulation is not to perfectly reproduce the real world, but to provide an alternative simplified model or abstraction.

1.1.3 Domain

We cover two domains in this thesis: The Mobile Money Service Domain and the Retail Store Domain. Both domains are examples of financial transaction systems, since there are transactions between either the customers or between a merchant and a customer.
1.1. Background

1.1.3.1 Mobile Money Service Domain

The service Mobile Money\(^1\) is a platform for transferring money between users, using their mobile phones. This is accomplished by the use of codes sent through text messages or the Internet.

Mobile money brings several benefits for users, including the simplicity of transferring money between themselves and others. One user only needs to know the mobile phone number of the receiving user in order to send money. If the receiving users are registered in the system then the money can be deposited right away in their account. Otherwise, the users receive a code via SMS that enables the recipient to collect the money in cash at one of the nearby local concerns that are affiliated with the mobile money operator. It does not require a user to have a bank account, which is beneficial for many people in the (third) world who do not have sufficient assets to warrant a bank account. However, if the user wants to refill their account or withdraw money, then an existing bank account can be connected to the mobile money account, and used in conjunction with it. There are also other alternatives such as top-up card or credit cards connected to the service, that can be used to deposit or withdraw money from the mobile money system.

1.1.3.2 Retail Store Domain

The retail store domain is a general domain where commercial activity takes place between a store and a customer. In our project we specifically used the case of a shoe retail chain in Scandinavia. The short description given here corresponds with the design of our simulator and it is an abstraction that is used in our model to simplify the complexity of this domain.

In general, a customer can purchase any goods or articles from a store at a specific price. After this commercial transaction the customer has the benefit of returning an unwanted article during a limited period after the purchase to obtain a full or partial refund of the money paid. The return

\(^1\) Mobile Money is a generic name that we use in this study, and it is not the real name of the service provided by our partner in this research.
1. Introduction

can also take place due to guarantee issues. Customers also benefit from special offers that reduce the original price of the article, this is called a discount.

1.1.3.3 Fraud with Refunds and Discounts

The *Refunds* fraud includes cases where the salesman creates fraudulent refund slips, keeping the cash refund for him- or herself. *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 [39].

1.1.4 Fraud and Fraud Detection

Fraud is known as the wrongful or criminal act with the intention of obtaining financial or personal gain. In the domains that we cover there are several threats of fraud that we can analyse. The time available has allowed us to cover only few of them in this project. From the Mobile Money Service we are interested specifically in Money Laundering. From the retail store domain we will include only fraud in the form of criminal activities perpetrated by the staff. As an example we model fraud cases that occur when customers return articles to obtain a refund or when the sales staff applies a partial discounts to the price of the articles.

1.1.4.1 Money Laundering

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. 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 [51]. Thus, it is not only the 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 address money laundering detection with procedures and personnel specifically dedicated to perform tasks and follow the controls set by the Anti-Money Laundering (AML) regulations. The cost of implementing such controls for AML is quite high, mainly because of the amount of manual labour required. In Sweden alone it is estimated to be around 400 million SEK annually [37].

1.1.4.2 Fraud Detection

The most common method used today for detecting anomalous financial transactions consist of establishing thresholds for various quantities, statistical and otherwise, for different types of transactions. Transactions that exceed these thresholds require extra scrutiny, whereby the client needs to declare the precedence of the funds. These thresholds are set by internal policies or law, rarely with distinction between different economic sectors or actors. However, this of course leads to fraudsters changing their behaviour in order to avoid this kind of control, by e.g. making many smaller transactions that fall just below a set threshold [24], instead of fewer larger ones. Several machine learning techniques have been used for detecting fraud, and more specifically complex fraud cases such as credit fraud and money laundering [50].

1.2 Aim and scope

Our aim is to find suitable methods and techniques to simulate realistic financial scenarios and generate synthetic data sets. Standard simulators and data sets allow researchers and organisations to develop, test, experiment and compare diverse fraud detection methods. Our previous aim was to detect money laundering in a mobile money payment system. However two main constrains changed this aim: first, the lack of available data for this domain made us direct our research towards the generation of
synthetic data; second, the complexity of detecting fraudulent schemes in the money laundering domain often requires the securing of data from diverse sources such as banks, payments and retailers, pertaining to the same (set of) transactions.

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 part of the financial system that could support money laundering. The last domain which is covered with a bank simulator remains out of the scope of this thesis. But our intention is to develop and include it in future work to complete our financial model.

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 1.4. Both simulators are described in detail in the papers that make up 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 money laundering.

Unfortunately, this service has only been running in a demo mode. This prevents us from collecting any data that can be used for analysis of possible detection methods.

We modelled and implemented a MABS that uses the schema of the real mobile money service and generates synthetic data following scenarios that are based on predictions of what could be possible when the real system starts operating.
1.4. Research Questions

From this we learnt a lot about the importance of accessing and sharing real data for fraud detection. Hopefully in a near future, we will be able to obtain real data from this system that will help to improve the accuracy of the simulator.

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 Retail shoe store Simulation, built on the concept of 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. 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 [49]. 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 1.1.3.3 and includes: Refunds and Coupon Reductions/Discounts.

1.4 Research Questions

Through our research and having dealing with the lack of available data we formulated the five main research questions that were addressed by the four papers that this thesis is based on. The research questions that were formulated during our research are:
1. Introduction

RQ1

Is it possible to do reliable fraud detection using a Synthetic Dataset?

RQ2

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

RQ3

How could we model and simulate a retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?

RQ4

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

RQ5

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

A list of all four contributing papers is available in section 1.6.

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 research question (RQ1): Is it possible to do fraud detection using a Synthetic Dataset?. RQ1 is the initial step of this research and address the idea and the possibility of using synthetic data for fraud detection. We address this question in Paper I.

Once we positively believed that we could use synthetic data for this research we continued by formulating RQ2 How could we generate a realistic synthetic data set for financial transactions for the purpose of fraud detection?, RQ2 seeks to find a method/technique to properly generate a synthetic dataset for the specific domain of Mobile Money Payments. RQ2 is covered briefly in Paper I and it is the main focus of Paper II.
With the introduction of a new partner and the benefit of having access to real data, we formulated RQ3, *How could we model and simulate a retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?*. RQ3 directly questions how to create a model and simulate a retail store which main output is a synthetic dataset for applying fraud detection methods. Paper III covers mainly RQ3. Paper IV extends the work of Paper III in relation to RQ3.

The identification of a problem with our research raised RQ4. Our concern here is with respect to the generated data set and specifically to the privacy and anonymity of the customers. Even if we briefly cover this topic in Paper II, Paper IV address RQ4 more in detail.

A simulation is always constructed with a goal in mind, our next natural step after having a verified and validated simulator is to perform experiments that aim to address the question presented in RQ5. With RQ5, we are interested in answering whether a widely used, and simple fraud detection technique such as threshold detection is sufficient to keep the losses of fraud at a manageable level. Paper IV extends the work of Paper III and address directly RQ5.

## 1.5 Research Method

We started with an exploratory literature review, and preliminary research into the possibility of using synthetic data for fraud detection, covered in [30]. We then developed two different simulators based on two case studies of systems with many financial transactions. The first, consisted of a payment system that uses mobile phones *PaySim*, introduced in [31]. The second was *RetSim*, a simulation tool that generates realistic scenarios based on transactional data from a shoe retail store [32, 33]. All simulators use the same Multi-Agent Based Simulation toolkit, called MASON, which is implemented in Java [34].

*PaySim* is only based on the schema of the database, and the described behaviour of the customers for the simulated system. During the develop-
1. **Introduction**

The implementation of *PaySim*, the mobile payment system, was in a testing phase, which made it impossible for us to obtain real data from actual use of the system with which to calibrate the behaviour of the agents. However, we 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 example of fraud detection, using labelled data, and machine learning techniques to classify the injected malicious behaviour.

*PaySim* is still awaiting real data in order for us to continue with the calibration of the simulation and experimentation on different fraud scenarios. 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 is 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 (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 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: Refunds and 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.

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 can not 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 our first simulation (PaySim), validation was difficult since we did not have any real data to compare to the output. 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.

1.6 Contributions

This thesis is based on the work presented in the following four papers:

Paper I

Paper II

Paper III

Paper IV
These four papers addressed all five research questions (see section 1.4) as shown in table 1.1.

Table 1.1: Contribution of Papers to Research Questions

<table>
<thead>
<tr>
<th>Paper-RQ</th>
<th>RQ1</th>
<th>RQ2</th>
<th>RQ3</th>
<th>RQ4</th>
<th>RQ5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper I</td>
<td>Main</td>
<td>Brief</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper II</td>
<td>Main</td>
<td>Brief</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper III</td>
<td>Improved</td>
<td>Main</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper IV</td>
<td>Improved</td>
<td>Main</td>
<td>Main</td>
<td>Main</td>
<td>Main</td>
</tr>
</tbody>
</table>

Paper I addresses RQ1: *Is it possible to do reliable fraud detection using a Synthetic Dataset?*, to answer this question we present 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 Paper I we discuss the pros and cons of using synthetic data, and problems and advantages inherent in the generation of such a data set. We illustrate using a case study based on a Mobile Money Payment system and suggest an approach based on Multi-Agent Based Simulations (MABS). To positively answer RQ1 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 II addressed RQ2: *How could we generate a realistic synthetic data set for financial transactions for the purpose of fraud detection?*, Paper II also briefly covers RQ4 regarding the privacy preserving and anonymization of the generated data. We continued the analysis and work of Paper I and implement the approach suggested there. In Paper II we present an approach based on a Multi-Agent Based Simulation (MABS) for the generation of synthetic financial data. This paper 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) as part of our strategy to continue and extend our research.

Paper III improved the model presented in Paper II to contribute to answer RQ2 and directly address RQ3: 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 got invited to special issue of a journal were we were given the opportunity to extend our work, which resulted in Paper IV. Hence, Paper IV becomes an extension of Paper III and introduces a new research question.

Paper IV complemented RQ3 and RQ4 by extending the work in Paper III. In order to answer RQ4: 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 IV used the RetSim simulator to model malicious behaviour to answer RQ5: Is threshold detection sufficient to keep the losses from fraud at a manageable level?, RQ5 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 simulator 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 RQ5 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.

In summary our contributions begin with the introduction of two new simulators: PaySim and RetSim. These two simulators are important in that for the first time synthetic data of financial transactions are used 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 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.

1.7 Related Work

After we published Paper II [31], Gaber [19] made use of our idea of generating logs for fraud detection also in the mobile money domain. His works quotes and shares our disadvantage of lacking public real financial data, and differs from us mainly in the benefit of having a model that uses few instances of real data to calibrate the simulation. Similarly to our work he used a Multi-Agent Based Framework as the platform for their simulation.

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 [10, 13, 49]. At the time of writing this thesis we have found no other research that reports on simulations for generating fraud data to be used for fraud
detection in retail stores.

There are also 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. MABS are used to model complex social behaviour based on agent micro behaviour rather, than a fixed statistical distribution of macro parameters.

Other methods to generate the necessary fraud data have been proposed [2, 17, 27, 35, 53]. The work by Yannikos [53] 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 [35] 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 than those of the original real data [53]. The work by Kargupta [27] 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.

A number of basic countermeasures against money laundering have been proposed, including basic statistical analysis which constrains the amount of the transactions as well as restricting their frequency [9]. Other methods that complement these basic security measures are based on checking every customer against a black list originating from previous investigated cases and a white list to e.g. avoid mistakes when faced with persons with the same name. Unfortunately, these and other methods have
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 [45]. 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, 35]. 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.

1.8 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. 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
1. Introduction

over real data that we can quantify and measure the amount of loses committed by our malicious agents. 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 more costly to implement.

We intend to make PaySim and RetSim available to the research community together with standard data sets.

1.9 Future Work

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 build 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 will cover 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.

We aim to improve the accuracy of our payment simulator PaySim with the help of real data. We have successfully achieve a realistic simulation for a retail store with RetSim. 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.

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 are currently in a preliminary phase working on our third simulator called \textit{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 perfect \textit{BankSim}.

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.
Criminals use money laundering to make the proceeds from their illegal activities look legitimate in the eyes of the rest of society. Current countermeasures taken by financial organizations are based on legal requirements and very basic statistical analysis. Machine Learning offers a number of ways to detect anomalous transactions. These methods can be based on supervised and unsupervised learning algorithms that improve the performance of detection of such criminal activity.

In this study we present an analysis of the difficulties and considerations of applying machine learning techniques to this problem. We discuss the pros and cons of using synthetic data and problems and advantages inherent in the generation of such a data set. We do this using a case study and suggest an approach based on Multi-Agent Based Simulations (MABS).

**Keywords:** Machine Learning, Anti-Money Laundering, Money Laundering, Anomaly Detection, Synthetic Data, Multi-Agent Based Simulation
2. Money Laundering Detection using Synthetic Data

2.1 Introduction

Money laundering threatens the economic and social development of countries. The threat is due to the injection of illegal proceeds into the legitimate financial system. 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 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.

This paper aims to analyze the implications of using machine learning techniques for money laundering detection (also known as Anti-Money Laundering, AML) in a data set consisting of synthetic financial transactions.

Our case study is based on the company AB. Company AB has developed a mobile money implementation that provides users with the ability to transfer money between mobile phone users, by using the phone as a sort of electronic wallet. The task at hand is to provide a tool that detects suspicious money laundering activities.

The mobile money service is currently running in a demo phase. Hence, real data from this system is not available at this stage and therefore the system does not produce representative data that can be used e.g. for the training of the machine learning algorithm. Thus, due to the lack of real data we turn to the generation of synthetic data as an alternative.

The use of synthetic data for machine learning has implications. In this paper we present our ideas about how to address some of the difficulties raised by the lack of real data.

Outline The rest of this paper is organized as follows: sections 2.2 and 2.3 introduce the topic of money laundering and present previous work. Sec-

\footnote{The identity of the Company AB unfortunately cannot be disclosed due to a Non-Disclosure Agreement}
tions 2.4 and 2.5 address the main topic, which is the use of synthetic data for Anti-Money Laundering. We finish with a discussion and conclusions including future work in sections 2.6 and 2.7.

2.2 Background

Money Laundering affects the finances of nations and it may contribute to an increase in the funding of criminal activities [6]. Due to issues such as the high amount of transactions taking place in any financial service, it is not a trivial task to find specific anomalous transactions that should be marked as suspicious. The reported suspicious activity needs to be supported by tangible evidence that allows specialized government agencies to investigate further.

In Sweden and other countries, most companies in the financial sector are required by law to address money laundering detection. The cost of implementing such controls for AML is quite high, mainly because of the amount of manual labor required. In Sweden alone it is estimated to be around 400 million SEK annually [37].

The most common method today used for preventing anomalous financial transactions consist in establishing thresholds for all transactions. Transactions that exceed these thresholds require extra scrutiny, whereby the client needs to declare the precedence of the funds. These thresholds are set by law without distinction made between different economic sectors or actors. However, this of course leads to fraudsters changing their behavior in order to avoid this kind of control, by e.g. making many smaller transactions that fall just below the threshold [24].

The specific domain covered here is the service *Mobile Money*\(^2\), which is offered by Company AB. *Mobile Money* is a platform for transferring money between users, using their mobile phones. This is accomplished by the use of codes sent through text messages or the Internet.

\(^2\) *Mobile Money* is a generic name that we use in this study and it is not the true name of the service provided by Company AB
Mobile money brings several benefits for users, including the simplicity of transferring money between themselves and others. One user only needs to know the mobile phone number of the receiving user in order to send money. If the receiving users are registered in the system then the money can be deposited right away in their account. Otherwise, the users receive a code via SMS that enables the recipient to collect the money in cash at one of the nearby local stores that are affiliated with the mobile money operator. It does not require a user to own a bank account, which is beneficial for many people in the world who do not have sufficient assets to warrant a bank account. However, if the user wants to refill their account or withdraw money, then an existing bank account can be connected to the mobile money account, and used in conjunction with it. There are also other alternatives such as top-up card or credit cards connected with the service, that can be used to deposit or withdraw money from the mobile money system.

2.3 Related work

A number of basic countermeasures against money laundering have been proposed, including basic statistical analysis which constrains the amount of the transactions as well as restricting their frequency [9]. Other methods that complement these basic security measures are based on checking every customer against a black list originating from previous investigated cases and a white list to e.g. avoid mistakes when faced with persons with the same name. Unfortunately, these and other methods have proved to be insufficient [37].

Several machine learning techniques have been used for detecting fraud, and more specifically money laundering, [50]. From the point of view of machine learning, the application is interesting, due to the successful classification rate (high True Positives and low False Positives) that the classification model can achieve compared to other methods such as simple rule based detection that compares transactions against fixed thresholds.

Data mining based methods have also been used to detect fraud [45,
This leads to the observation that machine learning algorithms can identify novel methods of fraud by detecting those transactions that are different (suspicious) in comparison with the benign transactions. This problem in machine learning is known as novelty detection. Supervised learning algorithms have been used on a synthetic data set to prove the performance of outliers detection [1].

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 is been used to test fraud detection systems.

Gao [20] proposed one of the frameworks used for AML introduces the terms legal transaction, usual transaction, unusual transaction, suspicious transaction and illegal transaction for describing different possible categories of transactions. This framework aims to rank the likelihood that a transaction would be illegal on a scale from 0 to 100, which enables prioritization.

We wish to stress that a detector cannot be completely certain that a transaction corresponds to money laundering. This task is delegated to the legal authorities. Instead of doing that, we intend to flag customers and transactions with a label of suspicion that focus the attention of the operator for further investigations.

Despite the possible bias injected in the data set during the simulation, synthetic data has been previously used with similar reasons as the ones presented by Barse [5]. As in that work, the lack of real data and the low probability of real instances of fraud in the real world data obtained, are some of the reasons discussed further in this paper.

In general the availability of financial information for research is very restricted by the corporate policies and even the law. Customers are usually protected by the financial organizations and the disclosure of their private information is limited by internal and government policies.
access to such data, anonymization techniques should be used on the data set in order to allow for the preservation of privacy [26, 52].

2.4 AML for mobile money

The detection of money laundering in the mobile money service is not trivial due to the difficulty of classifying transactions that are intended to appear as normal and legal. In this paper we address this problem with the approach of learning from the experiences of past detected patterns of illegal behavior in order to gain more knowledge about the possible rules or new patterns of fraud that could emerge in a mobile money system.

The first step to address our problem is to start by clearly formulating the learning problem.

2.4.1 Problem definition

Regarding the mobile money AML domain, we formulate the learning problem as:

**Task T** Classification of transactions as normal or suspicious based on the known pattern of legal transactions. The aim is to find anomalies or outliers inside a data set of mobile money financial transactions.

**Performance Measure P** Percentage of transactions correctly classified as anomalous, also known as *True Positives* (TP) and the percentage of *False Positives* (FP) i.e. transactions that are not anomalies and are misclassified as anomalies.

**Experience E** Synthetic data generated with transactions labeled as legal (normal) and/or illegal (suspicious).

The main weakness is that the experience gained by using a synthetic data set can be biased and in some cases it may not match a realistic situation that would occur in a real-world data set. However in the
2.4. AML for mobile money

following sections we present our analysis of how this can also be used to our advantage by allowing us to gain information about unseen but expected situations in a real-world data set.

2.4.2 Data Preprocessing

One of the tasks that need to be addressed is data preprocessing. This task includes the selection of the attributes, discretization, noise removal and, in certain domains, data fusion.

Company AB has a design with a database that aims to store all the log information about the users interactions with the service. For this study we need to select the attributes that best contribute to the correct classification of suspicious transactions.

The customers are originally differentiated in the system by a profile. The profile for each customer is specified at the opening of the account by internal criteria. Customers with certain profiles are limited e.g. in the amount and the frequency of the transactions that they can perform. Additionally, there are specific methods that detect anomalies based on profiling customers which can be applied when faced with prior profiles such as these [8].

In addition to the generation of profiles, we added the following attributes to our simulation: Customer ID, Profile, Date of the Transaction, type of transaction (e.g. deposit, withdraw, transfer), Amount of the Transaction, Location (city), Account Age (months since the creation of the account) and Customer Age (years).

For each transaction of type transfer there is also a deposit transaction with the same value for a different customer. This transfer transaction describes a social network between customers. The rest of the fields are generated according to the given parameters of the simulation and random operations with range validation to guarantee consistent data that follows a realistic model.
Data labeled as anomalous should be added in the transaction database in order to run supervised algorithms. These anomalous records are created with the intention to replicate some of the common patterns used by fraudsters. One example of these patterns is an unverified user profile which makes either large deposits or large withdraws in comparison to a predefined threshold. Some other known problematic patterns of usage include: several withdraws from the same profile above the average normal value for transactions by young customers, a verified customer that performs a single large withdraw, and finally a chain of transactions that deposits money in a single account followed by consecutive withdrawals from that same account.

In a realistic situation we would be handling millions of transactions in a data set and in most of the cases only a small testing samples of the whole data set can be processed.

Although computational scaling performance is a topic that is not addressed here, the learning algorithms selected are profoundly affected by the amount of data provided for the training and the cross validation phase.

2.4.3 Learning with Synthetic Data

For a real world data set the selected algorithms should produce the best accuracy, i.e. TP rate, in comparison with the other algorithms. This tells us directly that our classifier can detect a significant number of suspicious transactions. However, the FP rate which is represented by the misclassified number of instances from the normal data, is also an important indicator of performance because we do not want a situation where the high number of FP will consume the time of an investigator and leads to possible missed cases and lack of trust by the investigators in the detector.

Thus, we are interested in providing an accurate method to improve the detection rate (TP) and reduce the misclassification rate of the benign data (FP) counted on the data collected from the simulation. A synthetic data set can be used to train the classifier and test different scenarios.
One example of such a scenario is when the customer population have low income and only a few customers have large assets.

The results of this classification algorithm with a synthetic data set should be interpreted in a broader perspective than results with a real-world data set. In a real scenario the results are used for prosecuting and reporting individuals. But when doing research using a synthetic data set the purpose is different. One of the goals is to identify the measures of detection and control that could be added to the system, given a set of conditions, with the clients.

We studied possible algorithms for our detection research using the same data set. The algorithms analyzed here are based on Decision Tree learning and clustering techniques. Other methods such as Support Vector Machine (SVM), Neural Networks, Link Analysis and Bayesian networks are not addressed here, but we expect to improve our approach in a future work by including these methods.

Decision Tree algorithms construct a tree that contains branches with rules that correctly classifies the most of the data set [11, 46]. The main advantage of using these algorithms (in comparison with other machine learning algorithms) for the domain of mobile money AML is the possibility for an investigator to determine common rules that classify suspicious behavior.

There could be situations where fraudsters start to behave according to a new pattern based on e.g. a specific location or city, combined with other attributes such as age, profile and others that inspected singly seem to be normal, but in combination could lead to the detection of a new fraud trend. However some of the requirements of these algorithms are that the data set used for the training should be representative enough of the whole data set in order to get rules that sufficiently generalize to the real scenario. These rules should be refreshed frequently in order to detect new possible fraud trends.

Clustering techniques such as distance based clustering and density
based clusters can be useful to classify natural clusters that appear in
the data set. The disadvantage with these techniques is the hard task
of finding the parameters that expose abnormal behavior clusters due to
the class unbalance problem of the distribution of the classes [14]. In a
normal situation most of the records in a data set are instances of the
normal behavior of a customer, with only a few representing anomalous
(i.e. interesting) behavior.

In addition, the complexity of the patterns used by fraudsters represent
a challenge, due to the fact that the fraudsters’ patterns intend to mimic
normal behavior in order to pass undetected by law enforcement.

Besides, from the perspective of our detection tool, we cannot be 100%
certain of the illegal precedence of the funds in a transaction, that is why
our detector should include a suspicious rank that allows an investigator
to prioritize more relevant and important cases.

2.5 Mobile Money simulation

During this research, we found a number of difficulties that affect working
with this domain. We found the lack of access to real data, the poor quality
of samples in the real-world data we could access, and the many possible
scenarios that we would like to explore, are some of the problems found
in our case study. Such a problems make the process of research more
difficult. This is the reason behind the discussion of using synthetic data
as an alternative to further research in this area.

Important issues arise when analyzing the use of machine learning for
money laundering such as: volume and complexity of data, class imbalance,
concept drift, class overlap and class mislabeling [50].

However, many other considerations also play a role in the simulation.
One of the most important challenges that we need to address is: How can
we make our model realistic and as close as possible to a desired scenario?

In order to answer this question, we present the following benefits and
disadvantages of using synthetic data for our research.

2.5.1 Benefits of using synthetic data

When using synthetic data one of the benefits we identified is the possibility of selecting attributes that reduce considerably the complexity of the data structures involved. Furthermore, this simplifies the tasks of data preparation and extraction from real sources. The volume of the data can be tuned to comply with different experimental setups.

The class imbalance problem can be reduced by setting up a simulation that produces enough records of each of the interesting classification classes. Class overlap is still a remaining issue with simulations. However, simulations that properly represent fraud behavior can avoid class mislabeling.

We summarize our findings as:

- The data that represent realistic scenarios are readily available.
- The privacy of the customer is not impacted.
- The disclosure of results is not affected by policies or legal issues.
- The data set is available for other researchers to reproduce experiments.
- Different scenarios can be modeled with parameters controlled by the researcher.
- Injection of enough abnormal data to address the class unbalance problem.
- Simulation of abnormal behavior prevent the problem of mislabeled classes.

2.5.2 Disadvantages of using synthetic data

Unfortunately other issues arise that are important to consider when using synthetic data. Some of these issues are:
2. Money Laundering Detection using Synthetic Data

- The data generated might be nor representative or realistic.
- Data can have biased information.
- It is difficult to build a realistic model due to the complexity of variables and parameters.
- The simulated suspicious data cannot be investigated further by the government agency. In a real scenario these results could be used for improving the accuracy of the existing classification algorithms.
- It is unknown if we can transfer the learning from a simulated data set to a real-world data set.

Some of the disadvantages presented can be minimized if we can build a simulation with records that can represent a real-world situation. It is important to understand that the purpose of the simulation is not to reproduce a view of the real world, but to provide an alternative simplified scenario that is designed according to a model as it is presented in the following section.

2.5.3 Multi-Agent Based Simulation

Multi-Agent Based Simulation is an approach that involves the use of autonomous and interactive agents and it has been used to model complex systems. These agents are described by their state, behavior and their interaction with other agents, which generates complex global behavior usually found in different domains [36].

Previous work has shown the use of Multi-Agent based simulation in the task of simulating social networks and analyzing social behavior [43]. Mobile Money resembles a social network of connected clients where the connections are represented by the transactions (money sent or received) and the nodes are represented by the clients.

The synthetic data from a simulation aims to represent the interactions of the customers of a mobile money system. The graph shown in Figure 2.1 is used to represent a desired scenario that can be used to study a certain phenomenon nor existing in a real world data set. This graph is an
2.5. Mobile Money simulation

hypothetical situation where 2000 clients from 7 different cities perform legal transactions with customers inside or outside their cities. The simulation allows the researcher to follow one agent and keep track of its behavior and also store all the transactions for further analysis.

Figure 2.1: Scenario 1 - 2000 clients distributed across 7 cities and multiple edges connecting clients that produces legal transactions

Figure 2.2 represent a small simulation of 20 malicious agents distributed across 3 cities. The behavior of these agents can be modeled as a cooperative
network of agents which aim to move a certain amount of money from the red nodes to the blue nodes. By doing this, the malicious agents avoid the threshold controls present in the system.

![Image of network graph]

Figure 2.2: Scenario 2 - 20 accounts distributed across 3 cities generating suspicious transactions

There are several agent-based frameworks that incorporate toolkits to aid the development of these kind of systems. Some of them are freely available and are widely used in academic simulations (e.g. MASON\(^3\), Repast\(^4\), Swarm\(^5\)). We used MASON for the network graphs and simulations presented in Figure 2.1 and 2.2.

### 2.6 Discussion

One of the difficulties with this domain is the lack of available data sets to use for training the machine learning algorithms and compare results with other researchers. This is the reason why a synthetic data set is proposed

---

\(^3\)MASON http://cs.gmu.edu/~eclab/projects/mason/

\(^4\)Repast http://repast.sourceforge.net/

\(^5\)Swarm http://www.swarm.org/
2.6. Discussion

as an alternative. We are unable to obtain any data set for the mobile money system at the moment.

Before, we have done some preliminary work based on a simplified statistical simulation. Because of the elementary of this simulation we are not going to discuss any result here. This simulation was a biased representation of the domain that was used to run different machine learning algorithms in order to experiment with different outputs. But from this experience, we gained interest in producing a better simulation that shows real behavior and characteristics of clients from this domain.

The concept of MABS presented in section 2.5.3 make us consider that we can obtain a model based on multi-agents that would be realistic enough for our purpose of analyzing the possible scenarios and build a detector for money laundering. This led us to work on a simulation based on the concept of Multi-Agents Systems.

The problem of finding anomalies within the domain of money laundering is really a challenge. Every time a new pattern of money laundering is detected by the authorities and a new control mechanism is implemented, the fraudsters change their modus operandi and create a new method that is undetectable by the current rules of a rule based AML system.

When doing research in the domain of mobile money a number of difficulties arise, including the lack of access to real data that can be used to evaluate the learning algorithms. Even with real data, the lack of anomalous transactions can be a problem. This is why the injection of synthetic anomalies in a real-world data set is an alternative to overcome the problem of e.g. an unbalanced number of classes.

We do not expect to be able to identify all anomalies but we intend to identify abnormal behavior from the customers that can lead to the detection of these new mutated methods.
2. Money Laundering Detection using Synthetic Data

2.7 Conclusions

We have presented an analysis of the use of a synthetic data set from the domain of mobile money for experimentation with machine learning algorithms. Through the use of simulation of different scenarios we can discover flaws in the current system. This can also lead to the finding of new policies and legislation that could detect the appearance of previous detected patterns of money laundering in the future.

We pretend to illustrate the methods that can be used to evaluate the accuracy of different algorithms, without going into specific details. Our analysis covers Decision Trees, Clustering techniques and Decision Rules that are more understandable by human operators than other machine learning algorithms.

When working with synthetic data there is always a risk of generating a data set that does not represent the real world data set. This can lead to results that are biased by the way the data was generated. On the other hand a synthetic data set can also simulate different scenarios that are not available for experimentation and analysis as they are unusual, catastrophic etc.

As shown before, the benefits presented in section 2.5.1 make us conclude that using synthetic data for machine learning experimentation is a good alternative in domains where the lack of real data is a problem.

Further work will focus on building a model for the simulation of mobile money transactions. Multi-Agent Based Simulation (MABS) is an interesting technique that can be used to improve the results of the generation of realistic synthetic data sets for this domain. We aim to test in the future the performance of several machine learning algorithms such as Support Vector Machine (SVM), Neural Networks, Link Analysis and Bayesian Networks. These algorithms have been used successfully in previous studies and it is of our interest to evaluate them in future research.
Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)

Edgar Alonso Lopez-Rojas and Stefan Axelsson

Abstract

Mobile money is a service for performing financial transactions using a mobile phone. By law it has to have protection against money laundering and other types of fraud. Research into fraud detection methods is not as advanced as in other similar fields. However, getting access to real world data is difficult, due to the sensitive nature of financial transactions, and this makes research into detection methods difficult.

Thus, we propose an approach based on a Multi-Agent Based Simulation (MABS) for the generation of synthetic transaction data. We present 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.

Keywords: Multi-Agent Based Simulation, Retail Store, Fraud Detection, Synthetic Data.
3. Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)

3.1 Introduction

Money Laundering affects the finances of nations and it may contribute to an increase in the funding of criminal activities [6].

Due to the large amount of transactions and the variety of money laundering techniques, it is difficult for the authorities to detect money laundering and prosecute the wrongdoers. Thus, it is not only the 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.

After analyzing the implications of using machine learning techniques for money laundering detection [30] (also known as Anti-Money Laundering, AML) in a synthetic data set, we propose an approach based on Multi-Agent Based Simulation (MABS).

The main goal and contribution of this paper is to study the generation and use of synthetic data as an approach for developing methods for money laundering detection. A case study containing different scenarios was used as a scientific methodological approach. This leads to identify measures of detection and control that could be applied in similar circumstances.

The case study is based on the company AB. Company AB 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.

The mobile money service is currently running in a demo phase. Hence, real data from this system is not available at this stage, and therefore the system does not produce representative data that can be used e.g. for the training of a machine learning detection algorithm. Thus, we have turned to the generation of synthetic data as an alternative.

1The identity of the Company AB unfortunately cannot be disclosed
3.2 Background and Related work

Outline: The rest of this paper is organized as follows: Section 3.2 introduce the topic of money laundering and present related work. Sections 3.3 describes the problem, which is the generation of synthetic data for Anti-Money Laundering. Section 3.4 presents an implementation of a MABS for our domain. We present our results in section 3.5 and finish with a discussion and conclusions, including future work in section 3.6.

3.2 Background and Related work

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 [12].

Due to issues such as the large amount of transactions typically taking place in a financial service, it is a nontrivial task to find specific transactions that should be marked as suspicious. The reported suspicious activity needs to be supported by tangible evidence that allows relevant government agencies to investigate further.

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 labor required. In Sweden alone the cost is estimated to be around 400 million SEK annually [37]. The most recently 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 [9]. Transactions that exceed these thresholds require extra scrutiny whereby the client needs to declare the precedence of the funds. These thresholds are usually set by
law without distinction made between different economic sectors or actors. This of course leads to fraudsters adapting their behavior in order to avoid this kind of control, by e.g. making many smaller transactions that fall just below the threshold. Hence, these and other similar methods have proven insufficient [37].

Several machine learning techniques have been used for the detection of fraud, and more specifically money laundering [50]. The application of machine learning to the problem is advantageous, due to the successful classification rate (high True Positives and low False Positives) that can be obtained in comparison to simple threshold methods [54, 55].

Data mining based methods have also been used to detect fraud [45]. 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. This problem in machine learning is known as novelty detection. Supervised learning algorithms have been used on synthetic data to prove the performance of outliers detection in a different domain [1].

One of the frameworks used for AML, presented by Gao(2007) [20], introduces the terms legal transaction, usual transaction, unusual transaction, suspicious transaction and illegal transaction for describing different possible categories of transactions.

Synthetic data has previously been used with similar reasons to the ones presented here [5]. The protection of the clients privacy is an advantage over using real data.

Multi-Agent Based Simulation is an approach that involves the use of autonomous and interactive agents and it is been used to model complex systems. The agents and their interaction with other agents, are described by simple rules, which generates complex emergent behavior usually found in different domains [48].

Previous work have shown the use of Multi-Agent based simulation
3.3 AML for Mobile Money

The specific domain covered here is the service Mobile Money. Mobile Money is a platform for transferring money between users by mobile phone. This is accomplished by the use of codes sent through text messages or the Internet. Mobile money brings several benefits for users, including the simplicity of transferring small sums of money between users. One user only needs to know the mobile phone number of the receiving user in order to send money.

3.3.1 Problem definition

The detection of money laundering in the mobile money service is non-trivial. Illegal transactions are intended to appear as normal and legal. In this paper we address this problem by learning from the experiences of past detected patterns of illegal behavior in order to hopefully gain knowledge about the possible rules or new patterns of fraud that could emerge in a mobile money system.

In the mobile money AML domain, we formulate the learning problem as [40]: Task (T), Classification of transactions as normal or suspicious based on the known pattern of legal transactions. Performance Measure

---

2MASON http://cs.gmu.edu/~eclab/projects/mason/
3Repast http://repast.sourceforge.net/
4Swarm http://www.swarm.org/
3. Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)

(P), Percentage of transactions correctly classified as suspicious, also known as True Positives (TP), and the percentage of False Positives (FP). Experience (E), Synthetic data generated with transactions labeled as legal (normal) and/or illegal (suspicious).

3.3.2 Data Preprocessing

Data preprocessing includes the selection of attributes, discretization, noise removal and in certain domains, data fusion.

The mobile money product stores all information about the users’ interactions with the service in a database. For this study we need to select the database attributes that contribute the most to the correct classification of suspicious transactions. Customers are associated with a specific profile at the opening of the account based on outside information about economic factors. High risk customers are limited e.g. in the amount and the frequency of the transactions that they can perform.

We selected the following attributes for our simulation: Customer ID, Profile, Date of Transaction (steps), Type of Transaction (e.g. deposit, withdraw, transfer), Amount Transferred, Location (e.g. city) and Customer Age (e.g. 1=Young, 2=Adult or 3=Senior).

For each transaction of type transfer there is also a deposit transaction of the same value in a different customer account. These transfer transactions, describe a social network between customers. The rest of the fields are generated according to the given parameters of the simulation and random operations with range validation to guarantee consistent data that follows a realistic model.

Data labelled as suspicious were also added to the transaction database in order to train supervised learning algorithms. These anomalous records were created with the intention to replicate some of the common money laundering patterns used by fraudsters [25].

Although performance is also a topic that we are concerned with here,
3.3.3 Machine Learning Training from Synthetic Data

We are interested in providing an accurate method to improve the detection rate (TP) and reduce the misclassification rate of the benign data (FP) counted on the data collected from the simulation.

Having the same data set, we studied possible algorithms for our detection research. The algorithms analyzed here are based on supervised learning with Decision Tree learning and Decision Rules techniques [11, 46]. The main advantage of using these algorithms (in comparison with other machine learning algorithms) for mobile money AML, is to enable an investigator to be able to determine common rules that classify suspicious behavior.

We cannot be completely certain of the illegal precedence of the funds in a transaction, that is why our detector only raises a suspicion flag that allows an investigator to perform further analysis of the evidence.

3.4 A Multi-Agent Based Simulation for Mobile Money

In order to illustrate our idea, we developed a simple MABS simulation for the Mobile Money domain. We used MASON for the simulation [34]. The main reason was that it has important extensions that facilitate the implementation of social networks.

3.4.1 Model

The implementation of a Multi-Agent Based Simulation was based on simulating the behavior of several clients interacting in a Mobile Money environment. Figures 3.1 and 3.2 show the basic design we used. Our aim
was to produce a log of transactions, represented by the class *Transaction*. This log was built to generate the attributes specified in sect. 3.3.2.

![Simplified Class Diagram for the Mobile Money Simulation](image)

The simulation is managed by the class *Clients* which initialize the environment and creates the agents. The agents are represented by the class *Client*. This class has two child classes (*ClientSimA* and *Fraudster*) which inherits all the behavior added to the parent class *Client*. This allow us to create different types of agents and instantiate them together in the class *Clients*. The states of an agent are handled by a Markov transition matrix of probabilities. This tells the system when to change from Active to Inactive and from Profile P1 to Profile P2, which allows higher limits for transactions.

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"
3.4. A Multi-Agent Based Simulation for Mobile Money

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 of each client’s profile.

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 with 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 to make a deposit when the balance is low. When the balance is high the agent has a higher probability to make a withdrawal or a transfer, rather than a deposit.
3. Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)

3.4.2 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 are 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 a 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. 3.5.

To reflect a realistic scenario we conserved the thresholds imposed by the original money laundering system. Simplifying the model, all the values in the simulation are given in Swedish Kronor (SEK). For profile P1 there are limits of 2500 SEK (approx. 370 USD) for all transactions per day and a maximum balance of 16000 SEK. For profile P2, which are the validated users, both thresholds are increased to 35000 SEK.

3.4.3 Synthetic Data generated

In total we simulated 486977 transactions after running 5 simulations, each one with 200 agents running 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),
3.5. Results and Analysis of performance for different classifiers

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 data is suspicious. For the experiment we ran different supervised algorithms that were selected for the purpose of classifying the class labeled as suspicious transactions.

3.4.4 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [42]. 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.

3.5 Results and Analysis of performance for different classifiers

For the experiment we used the tool Weka [21]. The selected algorithms were based on Decision Trees and Decision Rules. From the decision tree category we selected Random-Tree, Random-Forest and J48graft (C4.5). From the decision rules based classifiers we selected JRip, due to its capacity to describe the minority class, and Decision-Table. We added Naive-Bayes as a performance base-line to compare the other algorithms against.

The results can be seen in Table 3.1. We can see that JRip produces the best accuracy in TP (True Positive) rate and FP (False Positives) rate
3. Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML)

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 3.1: Results for the class money laundering (suspicious)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>TP</th>
<th>FP</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive-Bayes</td>
<td>0.988</td>
<td>0.479</td>
<td>8543</td>
</tr>
<tr>
<td>Decision-Table</td>
<td>0.999</td>
<td>0.029</td>
<td>200</td>
</tr>
<tr>
<td>Jrip</td>
<td>0.999</td>
<td>0.012</td>
<td>115</td>
</tr>
<tr>
<td>Random-Forest</td>
<td>0.999</td>
<td>0.009</td>
<td>66</td>
</tr>
<tr>
<td>Random-Tree</td>
<td>0.999</td>
<td>0.015</td>
<td>173</td>
</tr>
<tr>
<td>J48graft</td>
<td>0.999</td>
<td>0.014</td>
<td>118</td>
</tr>
</tbody>
</table>

Table 3.2: Confusion Matrix

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>JRip</th>
<th>Random-Forest</th>
<th>J48graft</th>
</tr>
</thead>
<tbody>
<tr>
<td>class*</td>
<td>a b</td>
<td>a b</td>
<td>a b</td>
</tr>
<tr>
<td>a</td>
<td>5934</td>
<td>72</td>
<td>5954</td>
</tr>
<tr>
<td>b</td>
<td>43</td>
<td>480928</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>34</td>
</tr>
</tbody>
</table>

* a=Normal b=Suspicious

The tree generated by Random-Tree is relatively bigger than the one generated by J48graft which makes it easier to use by an inspector. However if we intend to add controls to detect money laundering in suspicious transactions we prefer to use Random-Forest or the JRip algorithm over others due to the higher detection rate.

We prefer to use accuracy indicators such as (TP and FP) over ROC curve (Receiver Operating Characteristic) to compare the different algorithms, because we are more interested in providing a method to improve the detection rate (TP) and reduce the misclassification of the normal data (FP).

JRip pin points the behavior of our malicious agent with high accuracy.
3.6 Conclusions

We notice that some of the rules are very strict regarding the balance, since malicious agents are more likely to have a balance that is approaching the threshold of the system. These rules are easily understandable by a human operator and can be straightforwardly incorporated into a money laundering detector.

In Table 3.2 we present the Confusion Matrix for the best three performing classifiers which are J48graft, Random-Forest and JRip. The intersection of class 'a' shows the number of TP, the intersection of class 'a' and 'b' shows the records misclassified. The worst classifier result was expected to be Naive-Bayes according to the TP rate and the high number of misclassified instances. We aim to find a classifier that output a high number of TPs for the class suspicious and reduces the number of FP for the class normal.

3.6 Conclusions

The problem of finding anomalies to detect instances of money laundering presents a challenge. Every time a new pattern of fraud is detected by the authorities, and the control mechanism changed, the fraudsters change their modus operandi and create a new method that is undetectable by the current threshold-based methods.

We have presented an example of the use of a synthetic data set representing an a simulated scenario in the mobile money domain, for experimentation with machine learning algorithms due to the lack of real data. By doing this we also avoid any possible issue related to privacy and identity protection of the customers of the service.

Through the use of our simulation we can discover flaws in the current system. This can also lead to the finding of new policies and legislation that could prevent the appearance of different patterns of money laundering in the future.

Our analysis employs some of the machine learning algorithms from
the categories Decision Trees and Decision Rules. We think that these algorithms produce an output more understandable by human operators than other machine learning algorithms.

When working with synthetic data, there is always the risk of generating a data set that does not realistically represent the real world. This can lead to results that are biased by the way the data was generated. However, a synthetic data set can also simulate different scenarios (Sect. 3.4.2) that are not available for experimentation and analysis as they are unusual, catastrophic etc.

Further work will focus on building an improved model with increased fidelity to the real world, for the simulation of mobile money transactions and other examples of fraud. We expect to implement real-world geographical locations with the extension for MASON called GEOMASON.

The generation of a realistic synthetic data set for this domain, that can be validated and verified, is another planned task. We also aim to test the performance of several machine learning algorithms such as Support Vector Machine (SVM), neural networks, Link Analysis and Bayesian networks. These algorithms have been used successfully in previous studies and it is of interest to evaluate them in this domain as well.
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.
4. RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection

4.1 Introduction

In this paper we introduce RetSim, a Retail shoe store Simulation, 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 4.2 introduce the topic of fraud detection for retail stores and present related work. Sections 4.3 describes the problem, which is the generation of synthetic data of a retail store. Section 4.4 shows a data analysis of the current data. Section 4.5 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 4.7 and finish with a discussion and conclusions, including future work in section 4.8.

### 4.2 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 [10, 13, 49].

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 [30]. 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 [31]. 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 [45]. 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 [3]. 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.

### 4.3 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:

\textbf{RQ} \hspace{1em} \textit{How could we model and simulate a retail shoe store and obtaining a realistic synthetic data set for the purpose of fraud detection?}

\section*{4.4 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 4.1. 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.
4. RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection

Table 4.1: Statistical analysis of five stores during one year

<table>
<thead>
<tr>
<th>Stat-Store</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactions</td>
<td>147037</td>
<td>180626</td>
<td>44446</td>
<td>37776</td>
<td>28456</td>
</tr>
<tr>
<td>Receipts</td>
<td>43406</td>
<td>38376</td>
<td>10094</td>
<td>8595</td>
<td>7619</td>
</tr>
<tr>
<td>Returns</td>
<td>9.25%</td>
<td>9.67%</td>
<td>11.43%</td>
<td>9.89%</td>
<td>9.33%</td>
</tr>
<tr>
<td>Members</td>
<td>5509</td>
<td>6381</td>
<td>1375</td>
<td>1152</td>
<td>16</td>
</tr>
<tr>
<td>Mem. Rec</td>
<td>16.02%</td>
<td>14.14%</td>
<td>18.12%</td>
<td>22.33%</td>
<td>0.56%</td>
</tr>
<tr>
<td>Avg. Price</td>
<td>762.49</td>
<td>772.32</td>
<td>665.2</td>
<td>575.93</td>
<td>409.62</td>
</tr>
<tr>
<td>Std. Price</td>
<td>494.52</td>
<td>514.51</td>
<td>459.05</td>
<td>616.74</td>
<td>416.36</td>
</tr>
</tbody>
</table>

4.4.1 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 5,509 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 4.1 and plotted in figure 4.1 which represents the sales summary per day and figure 4.2 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.
4.4. Data Analysis

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 4.2 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.

4.4.2 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
4. RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection

![Store 1 - Sales customers](image)

Figure 4.2: Store one - number of customers per day

<table>
<thead>
<tr>
<th>Type</th>
<th>Avg. Days</th>
<th>Avg. Cust</th>
<th>Std. Cust</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>155,75</td>
<td>45,43</td>
<td>28,17</td>
<td>22,22%</td>
</tr>
<tr>
<td>Med</td>
<td>63,20</td>
<td>38,97</td>
<td>23,83</td>
<td>11,11%</td>
</tr>
<tr>
<td>Low</td>
<td>13,57</td>
<td>33,93</td>
<td>16,68</td>
<td>66,67%</td>
</tr>
</tbody>
</table>

Table 4.2: Sales clerk frequency

A network analysis is the ability to visualize the network by using different layout algorithms such as Force Atlas or Yifan Hu [23]. 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 [7].

We can create a network based on the interactions between each of the
4.4. Data Analysis

Table 4.3: **Article categories**

<table>
<thead>
<tr>
<th>Category</th>
<th>Probability</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>0.2705</td>
<td>+1000</td>
</tr>
<tr>
<td>High</td>
<td>0.2122</td>
<td>100-999</td>
</tr>
<tr>
<td>Medium</td>
<td>0.1109</td>
<td>20-99</td>
</tr>
<tr>
<td>Low</td>
<td>0.3495</td>
<td>3-19</td>
</tr>
<tr>
<td>Unfreq</td>
<td>0.0569</td>
<td>1-2</td>
</tr>
</tbody>
</table>

sales clerks and their respective customers. For the weight of the edges we
e use the total sales price with respect to each customer. Figure 4.3 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 4.4 show more information
about the network used for calibrating the simulation.

Table 4.4: **Network Analysis**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Store one</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>5,545</td>
</tr>
<tr>
<td>Sales Clerks</td>
<td>36</td>
</tr>
<tr>
<td>Customers</td>
<td>5,509</td>
</tr>
<tr>
<td>Avg. Degree</td>
<td>1.104</td>
</tr>
<tr>
<td>Diameter Undirected</td>
<td>10</td>
</tr>
<tr>
<td>Avg. Path Undirected</td>
<td>3.98</td>
</tr>
</tbody>
</table>

Figure 4.3 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 4.4.

Figure 4.3: 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.
4.5 Model and Method

The design of RetSim was based on the ODD model introduced by [22].
ODD contains 3 main parts: Overview, Design Concepts and Details.

4.5.1 Overview

4.5.1.1 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.

**4.5.1.2 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.

**4.5.1.3 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 4.2).

**4.5.2 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.

4.5.3 Details

4.5.3.1 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.

4.5.3.2 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.

4.5.3.3 Submodels

Figure 4.5 shows the different use cases of the agents. This model represents the different actions that an agent can take inside the system.

![RetSim Use Case Diagram](image)

**Figure 4.5: 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 4.3). 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.

### 4.6 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 [39] report. As can be seen in section 4.5, 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.

4.6.1 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 ones. 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.

4.6.2 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 ones. 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.

4.6.3 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.

4.7 Results

RetSim uses the Multi-Agent Based Simulation toolkit MASON which is implemented in Java [34]. MASON offers several tools that aid the development of a MABS. We justified our choice mainly for the benefits of supporting multi-platform, parallelization, good execution speed in comparison with other agent frameworks; which is specially important for computationally intensive simulations such as RetSim [47]. 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 4.6.

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.

4.7.1 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
4. **RetSim**: A Shoe Store Agent-Based Simulation for Fraud Detection

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 4.5 compares three runs of RetSim against the original data. Since this is a randomised simulation the values are of course not identical.
4.7. Results

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Store 1</th>
<th>RetSim1</th>
<th>RetSim2</th>
<th>RetSim3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Articles sold</td>
<td>81441</td>
<td>103716</td>
<td>95847</td>
<td>96492</td>
</tr>
<tr>
<td>Avg. Sales Price</td>
<td>372.3</td>
<td>405.5</td>
<td>405.2</td>
<td>407.1</td>
</tr>
<tr>
<td>Std. Sales Price</td>
<td>510.9</td>
<td>555.1</td>
<td>550.7</td>
<td>552.2</td>
</tr>
</tbody>
</table>

### 4.7.2 Social Network Calibration

We experimented with calibrating our results and aim to simulate the network presented in section 4.4.2. 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 4.6.

### 4.7.3 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [42]. The verification ensures that the simulation correspond to the described model presented by the chosen scenarios. We described RetSim in section 4.5. 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 4.6: Network Simulated

<table>
<thead>
<tr>
<th>Statistic</th>
<th>RetSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>4948</td>
</tr>
<tr>
<td>Edges</td>
<td>5339</td>
</tr>
<tr>
<td>Sales Clerks</td>
<td>36</td>
</tr>
<tr>
<td>Customers</td>
<td>5303</td>
</tr>
<tr>
<td>Avg. Degree</td>
<td>1.079</td>
</tr>
<tr>
<td>Avg. Weighted Degree</td>
<td>499.1</td>
</tr>
<tr>
<td>Modularity Undirected</td>
<td>0.845</td>
</tr>
<tr>
<td>Diameter Undirected</td>
<td>8</td>
</tr>
<tr>
<td>Avg. Path Undirected</td>
<td>4.19</td>
</tr>
</tbody>
</table>

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 4.8, 4.9 and 4.10.

Figure 4.8 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 4.9 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 4.10 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
section 4.7.2. 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 4.3 and 4.7.
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,
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.

### 4.8 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-
4. RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection

Figure 4.10: 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 4.3 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 4.5 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 4.7) and we evaluated and verified our model in section 4.7.3.

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
4.8. Conclusions

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 4.7 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.
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

### 5.1 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 [18]. 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 [15, 16].

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 losses 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 Retail shoe store Simulator, 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.
5. USING THE RETSIM SIMULATOR FOR FRAUD DETECTION RESEARCH

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
5.2 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 [10, 13, 49]. 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 [32].

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 [41]. 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 [3]. 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
5. Using the RetSim Simulator for Fraud Detection Research

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, 17, 27, 35, 53]. The work by [53] 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 [35] 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 than those of the original real data [53]. The work by [27] 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 [45]. 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] [35]. 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.

5.3 Research Questions

For clarification we summarise our research questions thus:

RQ How can we model and simulate a retail shoe store to obtain realistic
5.4. Analysis of the Retail Data

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?

5.4 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.

5.4.1 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.
5. Using the RetSim Simulator for Fraud Detection Research

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.

5.4.2 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 [7].

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 5.1 shows one way to visualize the sample data extracted from the database using Yifan Hu layout [23]. 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 5.1
shows additional information about the network used for the subsequent calibration of the simulation.

Table 5.1: *Network Analysis*

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Store one</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>5545</td>
</tr>
<tr>
<td>Edges</td>
<td>6120</td>
</tr>
<tr>
<td>Salesmen</td>
<td>36</td>
</tr>
<tr>
<td>Customers</td>
<td>5509</td>
</tr>
<tr>
<td>Avg. Degree</td>
<td>1.104</td>
</tr>
<tr>
<td>Avg. Weighted Degree</td>
<td>829.3</td>
</tr>
<tr>
<td>Modularity Undirected</td>
<td>0.822</td>
</tr>
<tr>
<td>Diameter Undirected</td>
<td>10</td>
</tr>
<tr>
<td>Avg. Path Undirected</td>
<td>3.98</td>
</tr>
</tbody>
</table>

Figure 5.1 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.

### 5.5 The Model and Simulator

RetSim uses the MABS toolkit MASON which is implemented in Java [34]. 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 [47].
5. Using the RetSim Simulator for Fraud Detection Research

Figure 5.1: Store One - Network of Customers and Salesmen

Our aim was to produce a simulation that produces synthetic transactions 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.
5.5. The Model and Simulator

5.5.1 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 [44].

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 5.2. This table was used during calibration to estimate the article selected by each of the customers during the sales operation.

<table>
<thead>
<tr>
<th>Category</th>
<th>Probability</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td>0.2705</td>
<td>+1000</td>
</tr>
<tr>
<td>High</td>
<td>0.2122</td>
<td>100-999</td>
</tr>
<tr>
<td>Medium</td>
<td>0.1109</td>
<td>20-99</td>
</tr>
<tr>
<td>Low</td>
<td>0.3495</td>
<td>3-19</td>
</tr>
<tr>
<td>Unfreq.</td>
<td>0.0569</td>
<td>1-2</td>
</tr>
</tbody>
</table>

Figure 5.2 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.
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.

5.5.2 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 5.3 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
5.5. The Model and Simulator

From this, we selected the top two simulations that scored better for the selected statistical test (see section 5.6.1).

In the following figures, the labels $rs3658$ and $rs5125$ correspond to each simulation. Table 5.3 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.
5. Using the RetSim Simulator for Fraud Detection Research

Table 5.3: Statistical Analysis of Store One vs RetSim Simulations

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Store one</th>
<th>rs5125</th>
<th>rs3658</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receipts</td>
<td>43406</td>
<td>43610</td>
<td>46881</td>
</tr>
<tr>
<td>Items</td>
<td>77186</td>
<td>82358</td>
<td>88668</td>
</tr>
<tr>
<td>Returns</td>
<td>4267</td>
<td>8385</td>
<td>9005</td>
</tr>
<tr>
<td>Avg Sales Price</td>
<td>372.3</td>
<td>369.8</td>
<td>371.0</td>
</tr>
<tr>
<td>Std. Sales Price</td>
<td>510.9</td>
<td>519.7</td>
<td>514.8</td>
</tr>
</tbody>
</table>

5.6 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data [42]. 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.

5.6.1 Statistical Tests

Generated distributions of sales are presented in figures 5.4 and 5.5 for visual comparison with the original data. Figure 5.4 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 5.4: 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 5.5 is a box plot comparison of store one with the two top
simulations generated by RetSim. We visually corroborate that the five statistical measures provided by the box plot are similar but not identical.

Figure 5.6 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.
5.6. Evaluation of the model

5.6.2 Social Network Comparison

We calibrated RetSim to simulate the network presented in section 5.4.2. 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 5.4.

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
5. Using the RetSim Simulator for Fraud Detection Research

Table 5.4: Network Simulated

<table>
<thead>
<tr>
<th>Statistic</th>
<th>RetSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>4948</td>
</tr>
<tr>
<td>Edges</td>
<td>5339</td>
</tr>
<tr>
<td>Salesmen</td>
<td>36</td>
</tr>
<tr>
<td>Customers</td>
<td>5303</td>
</tr>
<tr>
<td>Avg. Degree</td>
<td>1.079</td>
</tr>
<tr>
<td>Avg. Weighted Degree</td>
<td>499.1</td>
</tr>
<tr>
<td>Modularity Undirected</td>
<td>0.845</td>
</tr>
<tr>
<td>Diameter Undirected</td>
<td>8</td>
</tr>
<tr>
<td>Avg. Path Undirected</td>
<td>4.19</td>
</tr>
</tbody>
</table>

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 5.1 and 5.7.

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.
5.6. Evaluation of the model

Figure 5.7: 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.

5.6.3 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-
5. Using the RetSim Simulator for Fraud Detection Research

information 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.

5.7 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.

5.7.1 Fraud Scenarios in a Retail Store

The following retail fraud scenarios are based on selected cases from the Grant Thornton report [39]. 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
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.

5.7.2 Injection of Fraudulent Refunds

To model the first scenario we need information about the relevant parameters describing the normal behaviour: figure 5.8 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.

5.7.3 Injection of fraudulent discounts

Figure 5.9 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
5.7. Fraud and Fraud Detection

![Percentage Return per Salesmen](image)

**Figure 5.8: 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.

### 5.7.4 Detection

We will use a rule-based fraud detection approach similar to the “exception audit technique” presented in the Grant Thornton report [39]. 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
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. 5.8 and 5.9). The results for the returns are shown in tables 5.5 and 5.6.

In table 5.5, we see that when the fraud is more aggressive the threshold
5.7. Fraud and Fraud Detection

Table 5.5: Fraud Detection Results

<table>
<thead>
<tr>
<th>Statistic</th>
<th>rs3302</th>
<th>rs3712</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positives</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>False Positives</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>False Negatives</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Precision</td>
<td>71.42%</td>
<td>50%</td>
</tr>
<tr>
<td>Recall</td>
<td>83.33%</td>
<td>33.33%</td>
</tr>
</tbody>
</table>

The detection method at 95% cut-off is more effective and produces less false negatives than with moderate fraud. From Table 5.6 we observe that a non-trivial amount 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 5.6: Threshold Fraud Detection

<table>
<thead>
<tr>
<th>Item-Data</th>
<th>rs3302</th>
<th>rs3712</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>36,584,976</td>
<td>39,085,401</td>
</tr>
<tr>
<td>Fraud</td>
<td>400,452</td>
<td>161,631</td>
</tr>
<tr>
<td>Detected</td>
<td>371,463</td>
<td>11,577</td>
</tr>
<tr>
<td>NOT Detect.</td>
<td>28,989</td>
<td>150,054</td>
</tr>
</tbody>
</table>

107
Visual methods to identify fraud can also be applied as shown as in figure 5.10, 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.

![Network Filtering Only Fraudulent Transactions rs3712](image)

Figure 5.10: Network Filtering Only Fraudulent Transactions rs3712

### 5.8 Discussion

In section 5.3 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 5.6.3. 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 5.7.4 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.
5.9 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 5.5. We also hope to make analysis of stores in other domains and extend the fraud model to make data sets for fraud
5.9. Conclusions

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.


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


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 cover more complex types of fraud, such as money laundering, that comprises multiple organisations and domains. Fraud is an important problem that impact the whole economy. Currently, there is a general 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 Simulator (PaySim) and Retail Store Simulator (RetSim), which allow us to generate synthetic transactional data. These simulations are based on real transaction data.

These simulations are multi agent based simulations. Hence, we developed agents that represent the clients 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, 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 results. 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.

Using statistics and social network analysis (SNA) on real data we could calibrate the relations between staff and customers and generate realistic synthetic data sets that were validated statistically against the original.

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