This is the published version of a paper published in *International Journal of Simulation and Process Modelling*.

Citation for the original published paper (version of record):

Using the RetSim simulator for fraud detection research.
*International Journal of Simulation and Process Modelling*

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:bth-12930
Using the RetSim simulator for fraud detection research

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

Dan Gorton
Center for Safety Research, Department of Transport Science, KTH Royal Institute of Technology, Stockholm, Sweden
Email: dan.gorton@abe.kth.se

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

Abstract: Managing fraud is important for business, retail and financial alike. One method to manage fraud is by detection, where transactions, etc. are monitored and suspicious behaviour is flagged for further investigation. There is currently a lack of public research in this area. The main reason is the sensitive nature of the data. Publishing real financial transaction data would seriously compromise the privacy of both customers, and companies alike. We propose to address this problem by building RetSim, a multi-agent-based simulation (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; anonymisation; multi-agent-based simulation; MABS; ABS; retail store; fraud detection; synthetic data.


Biographical notes: Edgar A. Lopez-Rojas is a PhD student in Computer Science and his research area is related with multi-agent-based simulation, machine learning techniques with applied visualisation for fraud detection and anti money laundering (AML) in the domains of retail stores, payment systems and financial transactions.

Dan Gorton is a PhD candidate at KTH Royal Institute of Technology. His current research focuses on risk management of online financial services, including fraud detection. His previous research has focused on extending intrusion detection with alert correlation and intrusion tolerance. He has experience working with different security and risk issues primarily within the banking, defence, and telecom sectors.

Stefan Axelsson is a Senior Lecturer at Blekinge Institute of Technology. His research interests revolve around computer security, especially the detection of anomalous behaviour in computer networks, financial transactions and ship/cargo movements to name a few. He is also interested in how to combine the application of machine learning and information visualisation to better aid the operator in understanding how the system classifies a certain behaviour as anomalous.
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 (FBI, 2013). Return fraud, i.e., the defrauding of a retail merchant by abusing the return process, alone is estimated to cost US retailers about nine 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 (ECB, 2013; Council, 2011).

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. Anonymisation techniques are often not considered sufficiently effective, the risk of leakage is difficult to calculate, and furthermore, anonymisation is difficult to perform effectively on large datasets 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. 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 has other benefits. It can produce more data much faster and with less cost than, for instance, collecting data and trying different scenarios of fraud, detection algorithms, or personnel and security policy approaches, in an actual store. The latter also entails additional risks, e.g., incurring the wrath of angered staff, due to testing, an ill-advised policy, which may lead to even greater expense and unwanted problems.

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 such as Solis et al. (2013), Chaczko and Chiu (2008), Schweiger and Stahmer (2003), and Bovinet (1993). Other approaches have been used to simulate financial markets (Hedjazi et al., 2013). However, 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 by including agents that use known fraud behaviour patterns to test the performance of a simple threshold detection method in Lopez-Rojas et al. (2013).

Anonymisation techniques have been used to preserve the privacy of sensitive information present in datasets. But de-anonymising datasets is not an insurmountable task, far from it (Narayanan and Shmatikov, 2009). For this reason, we have decided to use simulation techniques to keep specific properties of the original dataset, such as statistical and social network properties, and at the same time providing an extra layer of insulation that pure anonymisation does not provide.

There are tools such as IDSG (IDAS data and scenario generator) (Lin et al., 2006) 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 (Alam and Geller, 2012). SNA is currently being combined with social simulation. Both topics support each other in the representation of interactions and behaviour of agents in the specific context of social networks. However, there is no work addressing the question of customer/salesman-interaction that we are aware of.

Other methods to generate the necessary fraud data have been proposed by Yannikos et al. (2010), Lundin (2002), Kargupta et al. (2003), Evfimievski et al. (2002), and Agrawal and Aggarwal (2001). The work by Yannikos et al. (2010) 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 (2002) makes use of a small sample of real data to generate synthetic data. This approach is similar to ours. However, the direct use of real data to prime the generation of synthetic data is limited in that it makes it harder to generate realistic data with other characteristics that those of the original real data (Yannikos et al., 2010). The work by Kargupta et al. (2003) 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 Phua et al. (2010). This problem in machine learning is known as novelty detection. Furthermore, supervised learning algorithms have been used on synthetic datasets to prove the performance of outlier detection (Abe et al., 2006; Lundin, 2002). However, none of these studies made use of synthetic data from retail stores. To our knowledge, there has been no investigation of what are the limits of effectiveness of, e.g., simple threshold-based monitoring.

3 Research questions

For clarification, we summarise our research questions thus:

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

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

RQ2 Is the generated dataset properly anonymised so that no sensitive information leaks?

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

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.

Owing to a lack of space we will focus our presentation of the analysis on one of the biggest stores by sales volume,
Using the RetSim simulator for fraud detection research

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.

4.1 Statistical analysis

The store one sample contains 147,037 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 5,509 unique members that had made at least one purchase during the period which accounted for 16% of the receipts. This means that the majority of receipts belong to unidentified customers. However, in all these records, we can still identify the item(s) sold, the sales price and the salesman.

We then investigated the performance of the staff. We divided the sales staff into three categories: top, medium and low. Top refers to staff who works regularly at the store. Medium refers to seasonal staff who usually works for a period between one and three months. Finally, low refers to staff who works 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.

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 visualise them using Gephi, a tool that can visualise networks using different layout algorithms (Bastian et al., 2009).

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 1 shows one-way to visualise the sample data extracted from the database using Yifan Hu layout (Hu, 2005). 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 5,545 nodes where 36 of them are sales staff, with the rest being customers. The network contains 6,120 edges that connect the sales staff and customers. Each edge weight represents the total number of purchases per customer. Table 1 shows additional information about the network used for the subsequent calibration of the simulation.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Store one</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>5,545</td>
</tr>
<tr>
<td>Edges</td>
<td>6,120</td>
</tr>
<tr>
<td>Salesmen</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>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 1 shows a visualisation 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 The model and simulator

RetSim uses the MABS toolkit MASON which is implemented in Java (Luke, 2005). 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 (Railsback et al., 2006).

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.

5.1 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 (Phan and Varene, 2010).

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 dataset. 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 dataset.

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 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>+1,000</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 2 shows the different use cases of the agents. This model represents the different actions that an agent can take.
Using the RetSim simulator for fraud detection research

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.

**Figure 2** Use case diagram for the interaction of agents in RetSim

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.

### 5.2 Simulated scenarios

Our aim was to produce a simulation that would result in data comparable to our real dataset. This contained 36 salesmen and around 45,000 receipts and 81,500 articles sold. The simulation was seeded with a subset of about 11,000 articles from the real store. Figure 3 shows a visualisation 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 ANOVA statistical test. From this, we selected the top two simulations that scored better for the selected statistical test (see Section 6.1).

**Figure 3** Visualisation of simulated network (see online version for colours)

In the following figures, the labels rs3658 and rs5125 correspond to each simulation. Table 3 compares the selected simulations against the real data from store one. Since this is a randomised simulation the values are of course not identical, nor should they be.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Store one</th>
<th>rs5125</th>
<th>rs3658</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receipts</td>
<td>43,406</td>
<td>43,610</td>
<td>46,881</td>
</tr>
<tr>
<td>Items</td>
<td>77,186</td>
<td>82,358</td>
<td>88,668</td>
</tr>
<tr>
<td>Returns</td>
<td>4,267</td>
<td>8,385</td>
<td>9,005</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>

### 6 Evaluation of the model

We start the evaluation of our model with the verification and validation of the generated simulation data (Ormerod and Rosewell, 2009). 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 dataset, 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.

6.1 Statistical tests

Generated distributions of sales are presented in Figures 4 and 5 for visual comparison with the original data. Figure 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 2,000 represent the most common prices for shoes. While there are several small visual differences between the distributions, the overall similarity is striking.

Figure 4 Overlap of two runs of RetSim vs. real data (see online version for colours)

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 ($F(2, 269,854) = 0.5, p = 0.61$).

Figure 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 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.

6.2 Social network comparison

We calibrated RetSim to simulate the network presented in Section 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 4.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>RetSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>4,948</td>
</tr>
<tr>
<td>Edges</td>
<td>5,339</td>
</tr>
<tr>
<td>Salesmen</td>
<td>36</td>
</tr>
<tr>
<td>Customers</td>
<td>5,303</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 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 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 visualisations. Notice the visual differences between Figures 1 and 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. The real and the simulated networks are very similar and build their communities around the salesmen. This can be clearly seen by studying the different colours used in all the visualisations.

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.

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 of verifying our approach, to build from obviously reasonable components, and show that the result resembles the total behaviour of the simulated enterprise.

6.3 Privacy issues

In order to answer the question: ‘Is the generated dataset properly anonymised with respect to the original dataset?’, we need to reason about what information from the real dataset 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.

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.

7.1 Fraud scenarios in a retail store

The following retail fraud scenarios are based on selected cases from the Grant Thornton report (Member and Council, 2009). As can be seen below, the different scenarios can be implemented in almost the same way in RetSim, and fit well within the framework given by the normal model. (A malicious salesman could use several different methods of fraud, which means that we needs to be able to model combinations of all fraud scenarios implemented, and we see no reason why that should not be the case.)

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

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

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

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

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

7.2 Injection of fraudulent refunds

To model the first scenario we need information about the relevant parameters describing the normal behaviour: Figure 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 161,630 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 400,451 units per year.

Figure 8 Return value over sales total per salesman (see online version for colours)

7.3 Injection of fraudulent discounts

Figure 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 18,423 units.

The second fraud simulation (-.-) shows an aggressive agent with a fraud threshold of 600 units and the frequency 10% of the sales. The total profit obtained by all agents is 80,600 units per year.

**Figure 9** Discount value over sales total before discount per salesman (see online version for colours)

7.4 Detection

We will use a rule-based fraud detection approach similar to the ‘exception audit technique’ presented in the Grant Thornton report (Member and Council, 2009). 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 20 (1/20) to need further investigation. This gives us the following cut-off points; 0.13 for returns, and 0.022 for discounts (see Figures 8 and 9). The results for the returns are shown in Tables 5 and 6.

In Table 5, we see that when the fraud is more aggressive the threshold detection method at 95% cut-off is more effective and produces fewer false negatives than with moderate fraud. From Table 6, we observe that a non-trivial amounts of fraud goes undetected when the fraud is moderate. However, threshold detection is very effective when applied to the simulation with the aggressive fraud behaviour, catching more than 90% of the returned fraud. However, in either case, the total amount of fraud is limited at a fixed percentage of turnover, and when fraud increases our method becomes relatively more effective. This seems to indicate that by adjusting the threshold, the business owner can trade off the level of ‘accepted’ fraud for the cost of performing further investigation into the flagged staff. Thus, being able to manage the risk to business due to fraud. We also performed a simulation with a handful of other agents that performed fraud with different percentages and different cut-off levels. However, as they did not add anything to the results presented here; they either behaved as the ones presented or differed in trivial ways (i.e., someone who does not 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** 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>

**Table 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>100.00%</td>
</tr>
<tr>
<td>Fraud</td>
<td>400,452</td>
<td>1.09%</td>
</tr>
<tr>
<td>Detected</td>
<td>371,463</td>
<td>1.02%</td>
</tr>
<tr>
<td>Not detect.</td>
<td>28,989</td>
<td>0.08%</td>
</tr>
</tbody>
</table>

**Figure 10** Network filtering only fraudulent transactions rs3712 (see online version for colours)
Visual methods to identify fraud can also be applied as shown as in Figure 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.

8 Discussion

In Section 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 dataset 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 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 losses from fraud and at the same time minimising the effort and cost of fraud detection. In Section 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.

9 Conclusions

RetSim is a simulator of a retail store that generates transaction dataset that can be used for research into fraud detection. Synthetic datasets 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 dataset, 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 datasets of commercial activity of a retail store. Datasets 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 datasets.

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. We also hope to make analysis of stores in other domains and extend the fraud model to make datasets for fraud detection in other domains available. We plan to make the simulator and data presented here available to the research community at large.

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


