Analysis of Fraud Controls Using the PaySim Financial Simulator

Edgar Alonso Lopez-Rojas*

Department of Information Security and Communication Technology (IIK),
Norwegian University of Science and Technology (NTNU),
Gjøvik, Norway
E-mail: edgar.lopez@ntnu.no
* Corresponding author

Stefan Axelsson

Department of Information Security and Communication Technology (IIK),
Norwegian University of Science and Technology (NTNU),
Gjøvik, Norway
E-mail: stefan.axelsson@ntnu.no

Dejan Baca

M-Commerce
Ericsson AB,
Karlskrona, Sweden
E-mail: dejan.baca@ericsson.com

Abstract: Fraud controls for financial transactions are needed and required by law enforcement agencies to flag suspicious criminal activity. These controls however require deeper analysis of the effectiveness and the negative impact for the legal customers. Due to the intrinsically private nature of financial transactions this analysis is often performed after several months of actively using fraud controls.

In this paper, we present an analysis of different fraud prevention controls on a mobile money service based on thresholds using a simulator called PaySim. PaySim uses aggregated data from a sample dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behaviour.

With technology frameworks such as Agent-Based simulation techniques, and the application of mathematical statistics, we show in this paper that the simulated data can be as prudent as the original dataset for setting optimal controls for fraud detection.

Keywords: Multi-Agent-Based Simulation; MABS; Financial Fraud; Mobile Money; Fraud Detection; Synthetic Data

Reference to this paper should be made as follows: Lopez-Rojas, E.A., Axelsson S. & Baca, D. (2017) ‘Applications of the PaySim simulator for fraud detection research in a financial mobile money service’, *International Journal of Simulation and Process Modelling*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Edgar Lopez is a postdoctoral fellow in computer science at NTNU in Gjøvik and his research area is related with Multi-Agent Based Simulation, Machine Learning techniques with applied Visualization for fraud detection and Anti Money Laundering (AML) in the domains of retail stores, payment systems and financial transactions.

Stefan Axelsson is an associate professor at NTNU in Gjøvik. His research interests revolve around computer security, especially the detection of anomalous behaviour in computer networks, financial transactions and ship/cargo movements to name a few. He is also interested in how to combine the application of machine learning and information visualization to better aid the operator in understanding how the system classifies a certain behaviour as anomalous.

Dejan Baca received his PhD in 2009 after defending his thesis titled Developing Secure Software. He then continued to make a career in Ericsson AB as a senior security specialist, while still maintaining a research interest involving several studies.

This paper is a revised and expanded version of a paper entitled ‘PaySim: A financial mobile money simulator for fraud detection’ presented at the EMSS 2016, Larnaca, Cyprus, 26 September, 2016.

1 Introduction

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

For instance, in many parts of Africa, the adoption of mobile money services as a means of sending and receiving funds using a mobile phone have improved the life of merchants and customers alike. In Tanzania, for instance, which according to the world bank is one of the fastest growing economies in the world, the adoption of mobile money as a solution for creating payments has had a positive effect on the overall economy. During December 2013 alone, 100 million transactions were made, netting $1.8 billion dollars Seetharam & Johnson (2015) in total. Fraud controls for financial transactions are needed and required by law enforcement agencies to flag suspicious criminal activity. These controls however require deeper analysis of the effectiveness and the negative impact for the legal customers. Due to the intrinsically private nature of financial transactions this analysis is often performed after several months of actively using different fraud controls.

The work shown in this paper is the continuation of our work in this field and presents an analysis of the performance of different controls for a tipical fraud scheme in a mobile money service. To do this we use a tool that we named *PaySim* Lopez-Rojas et al. (2016) and a method to generate synthetic data that aims to reduce the time required to evaluate these controls. The PaySim simulator is available and can be used by researchers Lopez-Rojas & Franke (2017).

Obtaining access to data sets of mobile transactions for research is a very hard task due to the intrinsic private nature of such transactions Lopez-Rojas & Axelsson (2014). Scientists and researchers must today spend time and effort in obtaining clearance and access to relevant data sets before they can work on them. This is time consuming and distracts researchers from from focusing on the main problem, which is developing and improving their methods, performing experiments on the data, and finding novel ways to solve problems; such as the problem that inspired this paper, which is the fraud detection in financial data. Fraud inspectors on the other hand, are drowning in real fraud data. They are losing the opportunity that qualified people from the research community contribute to their task due to the impossibility to share private datasets.

PaySim generates synthetic datasets similar to real datasets from mobile money transactions. This is done by the means of computer simulation, in particular, agent based simulation. Agent based simulation is of great benefit in this particular context because the models created represent to some extent the human behaviour during transactions and are flexible enough to easily be adapted to new constraints. In this paper we improve and extend the PaySim model to include fraud behaviour and a study of fraud by measuring the cost and the economical impact of different fraud detection methods.

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

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

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

Outline The rest of this paper is structured as follows: Section 2 presents the background and previous work in simulating financial data. Section 3 states the problem. We introduce the fraud scenarios in section 4 and during sections 5 and 6 we present the implementation of PaySim and the results of the simulations. Finally section 7 present the conclusions and future work.

2 Background and Previous Work

The use of Mobile Money Transfers have grown substantially in the last few years and have attracted greater attention from users, specifically in areas in which banking services may not be as accessible as in more developed countries. Many providers of mobile money services have been working on several similar solutions over the past years. There are existing mobile money services in more than 10 African countries which coverage of 14% of all mobile subscribers Rieke et al. (2013).

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

Due to the large amount of transactions and the ever changing characteristics of fraud. Most of the measures against fraud start when the customer issue a complaint. Many current system still base their detection mechanism on simple thresholds assigned arbitrarily. Therefore there is a need to push detection forward to try and stop the wrongdoers from profiting from their fraud before it has taken place.

With *PaySim*, we aim to address this problem by providing a simulation tool and a method to generate synthetic datasets of mobile transactions. The benefits of using a simulator to address financial fraud detection was first presented in Lopez-Rojas & Axelsson (2012*a*). This research states the problem of obtaining access to financial datasets and propose using synthetic datasets based on simulations. The method proposed is based on the concept of MABS (Multi Agent Based Simulation). MABS has the benefits that allows the agents to incorporate similar financial behaviour to the one present in domains such as bank transactions and mobile payments.

The concept of financial simulators is been develop earlier to create models of financial markets and financial forecast Hedjazi et al. (2013), O'Loughlin et al. (1990), Takaishi (2014). Simulation has also been applied to solve cyber-security problems Legato & Mazza (2017). The first implementation of a financial simulator for fraud research was introduced in 2012 with a mobile money transactions simulator Lopez-Rojas & Axelsson (2012*b*). This simulator was implemented to address the difficulties of implementing proper fraud detection control on a mobile money system that was under development, and that had not produced any real data to use for research. This was the first paper to present an alternative approach to tackle the lack of real data. The synthetic dataset generated by the simulator was used to test the performance of different machine learning algorithms in finding patterns of money laundering. The work done by Lopez-Rojas & Axelsson (2016) introduced the PaySim simulator as an improvement over the previous simulator with a better model, and the use of real data to calibrate and evaluate the fit of the synthetic data versus the real world transactions distribution. In this paper we continue and extend the work done in Lopez-Rojas & Axelsson (2016) by modelling and implementing fraud scenarios into PaySim for fraud detection research.

The work by Gaber et al. (2013) introduced another similar technique to generate synthetic logs for fraud detection. The main difference here with previous work was that this time there was available real data to calibrate the results and compare the quality of the result of the simulator. The purpose of this study was to generate testing data that researchers can use to evaluate different approaches. This works differs significantly from our work because we present a different method for analysing the data that puts special attention on evaluating the quality of the resulting synthetic data set.

The work on fraud detection in mobile payments by Rieke et al. (2013), Zhdanova et al. (2014) is done in a similar vein as the work by Lopez-Rojas & Axelsson (2012*a*), Gaber et al. (2013).

Rieke et al. uses a tool named Predictive Security Analyzer (PSA) with the purpose of identifying cases of fraud in a stream of events from a mobile money transfer service Rieke et al. (2013). PSA is based on a dataset of 4.5 million log entires from a mobile money service over a period of 9 months. They use simulation due to the limitation and knowledge of existing fraud in the current logs. The main focus on PSA is to detect money laundering cases that are caused by the interaction of several users of the system in an attempt to disguise the fraud among the normal behaviour of the clients. As a result the paper shows that PSA is able to efficiently detect suspicious cases of money launder with the aim of automatically block the fraudulent transactions.

Zhdanova et al. (2014) is a continuation of the work done by Rieke et al. (2013) and uses the simulator developed by Gaber et al. (2013) to evaluate the results. Semi-supervised and unsupervised detection methods are applied to a mobile money dataset due to the advantage over supervised methods in this type of data where there is a difficulty in having a training data with known cases of fraud.

There is a previews work done about simulations in the domain of financial transactions in retail stores for the purpose of fraud detection Lopez-Rojas et al. (2013). A large collection of data was gathered from one of the Sweden's biggest shoe-retailer. This data was used to produce a simulator called RetSim. RetSim was later used to model fraudulent behaviour of the staff and develop fraud detection techniques. The work done in that paper is very similar to the work done in this paper but in a different financial domain. There has been subsequent work on RetSim that produced among other results social network analysis (SNA) which described the relationship between the clients and the staff for each store, measuring the cost of fraud with the purpose of minimize the risk and properly estimate a security budget Lopez-Rojas (2015), threshold detection and methods to optimize the setup of thresholds Lopez-Rojas et al. (2015) and finally using this thresholds to properly setup a triage model that prioritize fraud suspiciousness Lopez-Rojas & Axelsson (2015).

Public databases of financial transactions are almost non existent. However our previous work during the implementation of a simulator called BankSim presents a MABS of financial payments Lopez-Rojas & Axelsson (2014). BankSim is implemented in a similar way as the RetSim simulator and our simulator using in addition to statistical analysis a social network analysis. MultiMAuS is another example of a payment system developed with the purpose of study the cost of additional authentication to prevent fraud in a payment system Zintgraf et al. (2017). Our work differs from this work because the source of the data and the characteristics of payments and mobile transactions are different as presented later in the following sections.

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

3 Problem and Method

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

The main focus and goal for the simulation is to create another completely self-sufficient data set with the goal of having similar statistical properties as the original data set with the advantage of containing known fraud data that behaves with similar pattern as some of the documented fraud instances found in the real system.

In order to simulate the mobile money service, we need to properly simulate the different kind of transactions that the system supports. We decided to cover 5 of the most important transaction types: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

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

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

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

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

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

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

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

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

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

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

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

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

4 Fraud scenarios

The mobile money service has many fraud threats from the merchants, the customers, the insiders in the organization, hackers, and the common thief. In this paper we will discuss two of the main categories of threats that have been identified by the mobile money service security experts. Both fraud methods involve cashing money out of the service through merchants. Cashing out money is the easiest way for the fraudsters to obtain profit and avoid the risk of frozen accounts due to detection. In the first method the customers loses complete access and control of their accounts, the second method involves scamming the customers. For the purpose of this paper we have modelled and implemented only the first method, which involves the customer losing access to their account. However it is of course possible in the future to extend the model to cover more of the fraud schemata presented in this section, as well as newly discovered ones.

4.1 Lost account

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

4.2 Scammed customers

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

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

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

5 Modelling the system

PaySim uses the MABS toolkit called MASON version 19 which is implemented in Java Luke (2005). We selected MASON because it is: multi-platform, supports parallelism, and its fast execution speed in comparison with other agent frameworks. This is especially important for multiple running and computationally expensive simulations such as PaySim Railsback et al. (2006).

The design of PaySim was based on the ODD model introduced by Grimm et al. (2006). ODD contains 3 main parts: *Overview*, *Design Concepts* and *Details*. The original design of PaySim has evolved over time with different publications detailing the progress, we first designed a simulator without calibration due to a lack of real data Lopez-Rojas & Axelsson (2012*b*), we then designed a fully calibrated simulator Lopez-Rojas et al. (2016) and now, in this paper, we extend the model to include fraud behaviour.

5.1 ODD Overview

The purpose of this simulator is to simulate payments done in the realm of mobile transactions. The simulator should ultimately perform simulations in such a way that synthetic data of mobile transactions can be generated. The simulator should generate synthetic data that is very similar to a batch of real transactional data provided by Ericsson. The goal is to have a generator that can produce data on the fly that can later be used by the scientific community to perform research on fraud detection.

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

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

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

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

5.2 Design Concepts

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

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

To study the phenomenon of fraud we use the information from section 4 to model the case scenario where the customer losses control of his/her account.

5.3 Details

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

5.3.1 Inputs

There are multiple inputs required in order for the simulator to function smoothly. As initial input, the number of client neighbours for each agent is assigned. The profile for each agent is then assigned based on probability. Their location in space in relation to their neighbours is also randomly initialized. Some of the inputs used by PaySim are listed here:

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

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

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

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

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

5.3.2 Initiation Stage

In this stage, the PaySim simulator must load the necessary input data described in section 5.3.1. The first and most important step is to load the values for each parameter in the parameter file. These will among other things contain the file paths for the source data inputs that the simulator needs to load.

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

5.3.3 Execution Stage

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

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

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

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

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

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

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

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

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

5.3.4 Finalization Stage

After each of the agents have completed their role in the simulation and performed all of the actions the results must be saved. There are 4 outputs generated after each simulation. All of which serve a specific purpose which allow a researcher to further test the quality of the generated data and save the configuration of the simulation with the in order to be able to repeat the simulation with the exact initial properties and results.

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

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

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

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

6 Results

The results are divided in two parts, the first part shows the evaluation and calibration of the simulator. These results rely on our previous work Lopez-Rojas et al. (2016) but using a different dataset. The second part covers the main contribution of this paper which contains the results of the simulation after injecting the fraud scenarios and performing the fraud detection analysis.

For both parts we ran PaySim several times using random seeds for 744 steps, representing one month of real time, which matches the original logs. Each run took around 45 minutes on an i7 intel processor with 16GB of RAM. This time includes the interaction of the agents, the writing of the results on a text log file and the loading of this file to a MySql database. The output of a run contains approximately 24 million of transactional financial records divided into the 5 types of categories presented before.

6.1 Calibration and Evaluation

The first goal of PaySim is to produce a dataset that resembles the original one. To accomplish this, we selected the generated dataset that contained the lowest difference in values in comparison with the original dataset provided. The evaluation of the quality of the database was first calculated using the sum of square error (SSE) method on the quantities of the different datasets. Despite all simulations being fairly consistent, there are small differences due to the random seed selected, the one with the lowest error was *PS53313*. The selected synthetic dataset was named *PS53313* after an arbitrary random log name. Table 1 shows the types of transactions, count and average amount generated with the simulator. The amount values are given in an African currency that we can not disclose.

In order to verify that the simulation was working properly we plotted the distributions to visually identify significant differences between the original and the synthetic dataset. Figures 1 and 2 show the visualization of two types of transactions (CASH OUT and TRANSFER). Each figure contains the output for each step; the count of transactions, the total sum of transaction, the averages, and the standard deviations. Figure 1 is presented in this paper bigger to allow a better visualization. The red continuous line represent

Table 1 Simulated synthetic dataset PS53313

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

Figure 1 Visualization of transaction type CASH-OUT

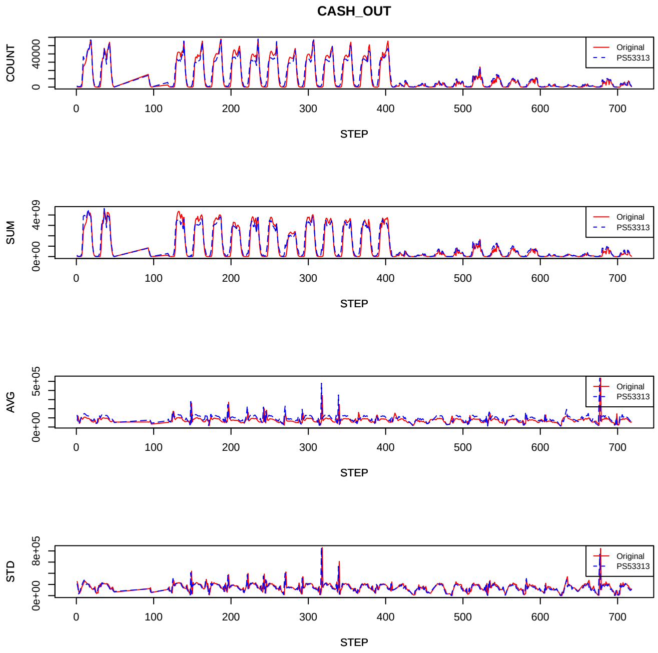
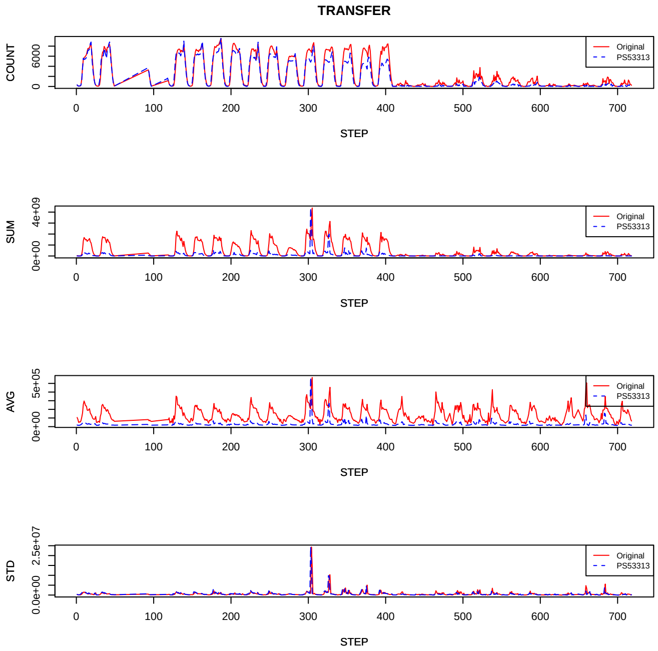


Figure 2 Visualization of transaction type TRANSFER



the original data distribution and the blue dashed line represent the synthetic dataset *PS53313*. We excluded the other types of transactions since they are presented in our previous work Lopez-Rojas et al. (2016) and for our fraud detection study we used only the CASH OUT and TRANSFER types.

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

6.2 Fraud Scenarios

The scenario selected is based on the first fraud case presented in section 4 which happens when the client loses control and access to his/her account. The fraudster takes control and uses disposable mule accounts to transfer the money, and later cash them out. All of this can only happen in a very short time, because when the clients discover that their accounts are compromised the first action they should take is to contact customer service to block any possible further malicious transactions. Since the current fraud detection reacts only after a customer complaint, we want to study how much money can we prevent our customers from losing after applying the fraud detection mechanism presented in the following section.

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

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

We ran the PaySim simulator with the same parameter file that generated the dataset *PS53313* for calibration. Table 2 shows the number of transactions type TRANSFER and the classification of fraud. The first obvious and important thing to notice is that whenever there is a control, the effort required for

Table 2 Fraud Detection Classification

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

committing fraud gets higher. Just by introducing a lower threshold on the maximum amount allowed for a transfer, the number of transactions needed to empty an account increases several times. However, the number of legitimate users that will be affected increases. This is the trade-off in fraud detection that a manager needs to address in enacting the fraud controls.

Table 2 also shows the loss due to fraud. If we focus attention on the False Negative (FN) row of each simulation, we can see the profit from fraud. The bigger the threshold the higher the profit. The task for a manager is to reduce this amount while minimising False Positives (FP) cases, which are legitimate customers that have their account blocked by the controls implemented to prevent fraud.

Table 3 show the fraud detection results of each of the datasets evaluated here. We see that the precision is higher when the threshold is higher as in dataset *PS98516* (1200k). This means that we will have fewer customers affected. However, the recall is seriously affected, which means that the fraudsters will profit more using this control (16 189 millions).

On the other hand when we have a lower threshold as in *PS89745* (300k), the number of false positives (FP) increases to 982. But, we have a considerable higher recall which means that we lower the total value of fraud (6724 millions).

Table 3 Fraud Detection Results

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

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

It is on our interest to extend this fraud model to cover other possible cases of fraud and even more important is to be able to measure the impact of fraud by quantifying the loss.

7 Conclusions

We used the PaySim simulator as a tool to analyse the cost of fraud in a mobile money service. We do this by generating diverse synthetic transactional data set for research in fraud detection. The data sets generated with PaySim can aid academia, financial organisations and governmental agencies in testing their fraud detection methods, or to compare the performance of different methods under similar conditions using a common public available and standard synthetic data set.

After we showed that the generated dataset captures the process and the frequencies of the different transaction types of the mobile money service, we argue that PaySim is ready to be used as a tool to evaluate the performance of different controls on a generated synthetic data set that resemble the original and private data set supplied.

PaySim can generate diverse fraud scenarios and contribute to the elaboration of fraud detection mechanisms due to the unique advantage that is the possibility to measure the total cost of fraud. By using PaySim we protect the privacy of the customers at the same time that interesting results are possible to share with other researchers without the constraints and legal boundaries associated with sharing the original data. We also made available a synthetic data set sample and the code to other researchers.

Future work on the simulator will improve the model of fraudulent agents, and cover other different scenarios to test the efficacy and accuracy of diverse fraud detection methods.

Acknowledgement

The research leading to these results was part of the research project "Scalable resource-efficient systems for big data analytics" funded by the Knowledge Foundation (grant: 20140032) in Sweden and has also received funding from the Norwegian Ministry of Justice and Public Security (JD), through the NTNU Center for Cyber and Information Security (CCIS)

References

Gaber, C., Hemery, B., Achemlal, M., Pasquet, M. & Urien, P. (2013), Synthetic logs generator for fraud detection in mobile transfer services, *in* '2013 International Conference on Collaboration Technologies and Systems (CTS)', IEEE, pp. 174–179.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jø rgensen, C., Mooij, W. M., Müller, B., Peer, G., Piou, C., Railsback, S. F., Robbins, A. M., Robbins, M. M., Rossmanith, E., Rüger, N., Strand, E., Souissi, S., Stillman, R. a., Vabø, R., Visser, U. & DeAngelis, D. L. (2006), 'A standard protocol for describing individual-based and agent-based models', *Ecological Modelling* **198**(1-2), 115–126.

Hedjazi, B., Ahmed-Nacer, M., Aknine, S. & Benatchba, K. (2013), 'Multi-agent financial market simulation: Evolutionist approach', *International Journal of Simulation and Process Modelling* **8**(2-3), 185–199.

Legato, P. & Mazza, R. (2017), 'A simulation optimisation-based approach for team building in cyber security', *International Journal of Simulation and Process Modelling* **11**(6), 430–442.

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

Lopez-Rojas, E. A. & Axelsson, S. (2012*a*), Money Laundering Detection using Synthetic Data, *in* J. Karlsson, Lars ; Bidot, ed., 'The 27th workshop of (SAIS)', Linköping University Electronic Press, Örebro, pp. 33–40.

Lopez-Rojas, E. A. & Axelsson, S. (2014), Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research, *in* 'Advances in Computational Social Science and Social Simulation', Barcelona.

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

Lopez-Rojas, E. A. & Axelsson, S. (2016), A Review of Computer Simulation for Fraud Detection Research in Financial Datasets, *in* 'Future Technologies Conference, San Francisco, USA'.

Lopez-Rojas, E. A., Axelsson, S. & Gorton, D. (2013), 'RetSim: A Shoe Store Agent-Based Simulation for Fraud Detection', *The 25th European Modeling and Simulation Symposium* . (Best Paper Award).

Lopez-Rojas, E. A., El-mir, A. & Axelsson, S. (2016), PaySim: A financial mobile money simulator for fraud detection, *in* 'The 28th European Modeling and Simulation Symposium-EMSS, Larnaca, Cyprus'.

Lopez-Rojas, E. A. & Franke, K. (2017), Bootstrapping the paysim financial simulator for open source, *in* 'The 29th European Modeling and Simulation Symposium-EMSS, Barcelona, Spain'.

Lopez-Rojas, E. & Axelsson, S. (2012*b*), Multi agent based simulation (mabs) of financial transactions for anti money laundering (aml), *in* A. Josang & B. Carlsson, eds, 'Nordic Conference on Secure IT Systems', Karlskrona, pp. 25–32.

Lopez-Rojas, E., Gorton, D. & Axelsson, S. (2015), 'Using the RetSim simulator for fraud detection research', *International Journal of Simulation and Process Modelling* **10**(2), 144.

Luke, S. (2005), 'MASON: A Multiagent Simulation Environment', *Simulation* **81**(7), 517–527.

O'Loughlin, M. J., Driskell, M. K. & Diehl, G. (1990), Financial simulation: Combining cost information in systems analysis, *in* 'Winter Simulation Conference Proceedings', pp. 578–581.

Railsback, S. F., Lytinen, S. L. & Jackson, S. K. (2006), 'Agent-based Simulation Platforms: Review and Development Recommendations', *Simulation* **82**(9), 609–623.

Rieke, R., Zhdanova, M., Repp, J., Giot, R. & Gaber, C. (2013), Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis, *in* '2013 International Conference on Availability, Reliability and Security', IEEE, pp. 662–669.

Seetharam, B. & Johnson, D. (2015), 'Mobile money's impact on tanzanian agriculture.', *IEEE Software* **32**(1), 29 – 34.

Takaishi, T. (2014), 'Multiple time series ising model for financial market simulations', *Journal of Physics: Conference Series* **574**(1).

Zhdanova, M., Repp, J., Rieke, R., Gaber, C. & Hemery, B. (2014), No Smurfs: Revealing Fraud Chains in Mobile Money Transfers, *in* '2014 Ninth International Conference on Availability, Reliability and Security', IEEE, pp. 11–20.

Zintgraf, L. M., Lopez-Rojas, E. A., Roijers, D. & Nowe, A. (2017), Multimaus: A multi-modal authentication simulator for fraud detection research, *in* 'The 29th European Modeling and Simulation Symposium-EMSS, Barcelona, Spain'.