BOOTSTRAPPING THE PAYSIM FINANCIAL SIMULATOR FOR OPEN SOURCE

Edgar Alonso Lopez-Rojas[a] and **Katrin Franke**[b]

[a] ,[b] The Norwegian University of Science and Technology (NTNU in Gjovik)
[a] edgar.lopez@ntnu.no [b] katrin.franke@ntnu.no

ABSTRACT

PaySim Simulator is an approach to the the lack of legitimate datasets on mobile money transactions to perform and experiment with fraud detection techniques. In this paper we used a technique called bootstrapping which uses the synthetic dataset generated by PaySim to build the parameter files that were initially extracted as aggregated information from the original data which is sensitive to share to the public. This way, researchers will have the possibility to generate diverse dataset with sufficient quality to perform fraud detection research on it. This paper explains the bootstrapping approach and in addition a method to properly reuse the PaySim simulator in the same or similar other domains. This is a solution to ultimately yield the possibility to simulate and generate financial transactions in such a way that they become similar to the original dataset without any direct connection to the original source.

Keywords: Bootstrapping, Multi-Agent Based Simulation, Financial data, Fraud Detection,Mobile Money , Synthetic Data.

Available at:
http://edgarlopez.net/simulation-tools/paysim/

1. INTRODUCTION

PaySim Simulator is an approach to the the lack of legitimate datasets on mobile money transactions to perform and experiment with fraud detection techniques. 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 and 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.

The work shown in this paper aims to share with the research community our simulator. This is therefore the continuation of our work in this field and presents the development of a tool and a method to generate synthetic data that we previously named *PaySim* (Lopez-Rojas, 2016). 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 overcame the last barrier to share this simulator with the research community which is the sensitivity of the parameter file extracted from the original sample dataset. We used a technique called bootstrapping which uses the synthetic dataset generated by PaySim to build the parameter files that were initially extracted as aggregated information from the original data which is sensitive to share to the public.

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 a background of the simulators that lead to PaySim as well as the technique called bootstrap of the parameter files. We also present a method that consist on 7 steps to easy the adoption and use of the PaySim simulator.

2. BACKGROUND AND RELATED WORK

The use of Mobile Money Transfers have grown substantially in the last few years and have attracted greater attention from users, specifically in areas in which banking solutions may not be as procurable as in developed coun-

tries. Many providers of mobile money services have been working in several and similar solutions over the past years. There are existing mobile money services in more than 10 African countries which coverage of 14% of all mobile subscribers (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 countersolutions for the attacks.

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

With *PaySim*, we aim to address this problem by providing a simulation tool and a method to generate synthetic datasets of mobile transactions. The benefits of using a simulator to address fraud detection was first presented during our previews work in (Lopez-Rojas and Axelsson, 2012b; Lopez-Rojas et al., 2016). This research states the problem of obtaining access to financial datasets and propose using synthetic datasets based on simulations. The method proposed is based on the concept of MABS (Multi Agent Based Simulation). MABS has the benefits that allows the agents to incorporate similar financial behaviour to the one present in domains such as bank transactions and mobile payments.

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

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

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

(Rieke et al., 2013) 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 logs from a mobile money service over a period of 9 months. They use simulation due to the limitation and knowledge of existing fraud in the current logs. The main focus on PSA is to detect money laundering cases that are cause by the interaction of several users of the system in an attempt to disguise the fraud among the normal behaviour of the clients. As a result the paper shows that PSA is able to efficiently detect suspicious cases of money launder with the aim of automatically block the fraudulent transactions.

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

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 and 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. BankSim is based on the aggregated financial information of payments during 6 months of the two main cities of Spain that was provided by a bank in Spain with the purpose

of developing applications of different kinds that benefit from this sort of data. Our work differs from this work because the source of the data and the characteristics of bank payments and mobile transactions are different as presented later in the following sections.

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

3. PAYSIM BOOTSTRAPPING

We have finally passed the last barrier to share this simulator with the research community which is the sensitivity of the parameter file extracted from the original sample dataset. Aggregated information was originally calculated based on the measure of the STEP of the simulation. As we can see in fig. 1, this information was entirely based on the original dataset, therefore our NDA (Non Disclose Agreement) prevent us from publishing these files. For this reason, it was not possible for us to share the simulator. Without the parameter files the simulator is an empty shell that does not provide any insight from a mobile money domain.

We have previously shared generated synthetic datasets of PaySim in public website for the research community such as Kaggle. But these dataset represent just few instances of all possible scenarios that can be generated with PaySim. Researchers need to be able to answer several different questions regarding several different ”What If” possibilities in a future scenario.

We used a technique called bootstrapping which uses the synthetic dataset generated by PaySim to build the parameter files that were initially extracted as aggregated information from the original data which is sensitive to share to the public.

The process is rather simple. We used the original dataset to generate the needed parameters. Once the parameters are available to be used by the PaySim simulator, we generate a dataset that aims to reproduce a scenario similar to the original dataset. This synthetic dataset is used again to calculate the aggregated parameters that were initially used by the PaySim simulator. Once we have obtained these parameters from the synthetic files we are breaking the dependency of the simulator from the original data and therefore we are not violating any NDA.

Next time we run the PaySim instead of using the parameter files previously obtained from the real data, we used the calculated parameter files obtained from the generated synthetic dataset.

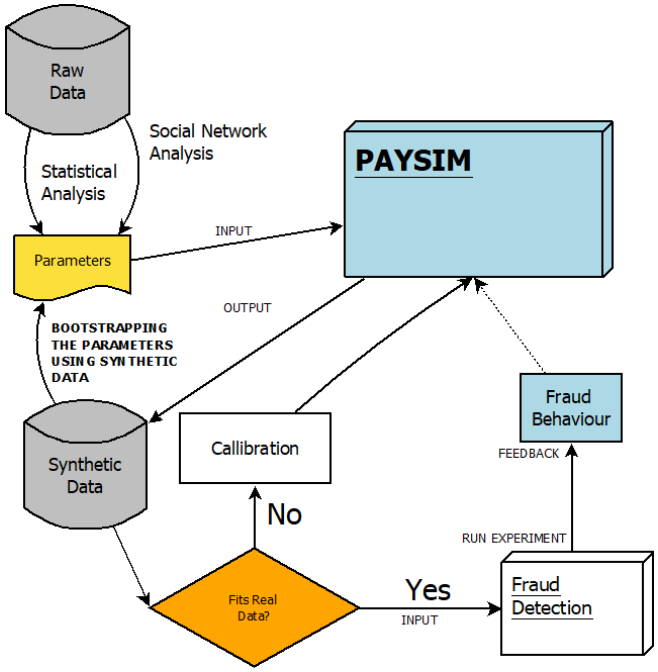

Figure 1: Bootstrapping the PaySim parameters from the Synthetic Dataset

4. MODEL

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

4.1. 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.

ISBN 978-88-97999-85-0; Affenzeller, Bruzzone, Jiménez, Longo and Piera Eds.

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.

4.2. Initiation Stage

In this stage, the PaySim simulator must load the necessary input data described in section 4.1.. 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.

4.3. 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.

4.4. 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.

5. METHODOLOGY FOR SIMULATING FINANCIAL TRANSACTIONS

It is important to follow our methodology in order to obtain the best benefit from the PaySim simulator. This section aims to explain how researchers can reuse our simulator to work in the same domain (mobile money) or similar domains that contain financial transactions. As a summary our method follows these steps in order to simulate financial transactions and perform experiments:

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

Obtaining a sample of real data is probably the hardest part of the whole method. Real data is important, because without it the simulation results can not be trusted. Fraud detection results are highly dependent on the dataset. For better results, a sample dataset that represents the financial service is required. This means that it covers enough (interesting) periods of time to learn more during the data analysis. If there is fraud, it should be properly labelled and identified with respect to which class of fraud it belongs in.

The real data sample can be obtained in several different ways: Full dump of a database (100% access), all of the data over a period of time, partial attributes of the data for some period of time, all data anonymised with respect to customer information, anonymisation by adding noise corruption (lowers the data quality) or simply aggregating information over a period of time.

The second step is to perform a data analysis to extract aggregated information for input to the simulator. Depending on the way the real data is provided, we need to perform several operations to convert the data into the format required. The simulator uses aggregations of information over a period of time, as input. The time granularity of the aggregation is specified on the simulation as a STEP. To accurately mimic the data distribution, we must extract aggregated information from the original data that matches each step in the simulation. There are also initial values and other input values extracted from the real data.

The information extracted is represented in terms of probabilities to ease the decision processes of the agents. Social Network Analysis (SNA) helps to recreate the topology of the customers' relations inside the simulation. The agent interacts with other agents within the environment and this interaction is specified by the information extracted during the SNA. The data analysis can also be done by employees of the financial institution that have access to the sample. Researchers only need the output of this step, to continue the process, this allows financial institutions to preserve the privacy of the customers.

In this paper we used the synthetic dataset generated by the PaySim simulator and applied the same data analysis that we performed over the original data. This prevents the problem of sharing the parameter files since they are directly connected to the original data and therefore are in most of the cases protected by NDA.

The third step is to add parametrization about expected fraud scenarios. The simulators are usually built to serve a purpose. Our simulators contain agents that, under certain conditions, act contrary to the law. The synthetic dataset has the benefit that can be generated according to the researcher's needs to study how certain fraud might affect a specific scenario. It can be a representation of the original dataset (sample). That is why we extract the aggregated information from the sample. Part of the simulator validation is to show that, given certain parameters, we can reproduce similar datasets.

The fourth step is to run the simulator several times using different random number seeds and/or different fraud configurations. In order to perform research in this field we need to be able to test different configurations. The Financial Simulator can also be used to answer all the ”WHAT IF” questions that are common during research. Researchers can run the simulator several times, using controlled variation on the parameters, to create new scenarios with normal and fraudulent data. This is specifically useful for answering questions such as WHAT IF: There is no fraud, there is little fraud, there is a lot of fraud, double the number of customers, and so on.

The fifth step is to apply the fraud detection methods on the generated synthetic dataset. This is one of the most important steps in the method. By changing the parameters in the previews step, we can generate diverse scenarios. These scenarios produce datasets with data that are labelled as fraudulent or not fraudulent (as appropriate). Once a dataset is generated, different methods for fraud detection can be tested and evaluated using the fraud label. A method for fraud detection can also be tested and evaluated with different scenarios that use the same fraud label. Fraud prevention methods can be also be added to the simulator to test and evaluate against fraud scenarios with the same flagged fraud.

The fifth step is to summarize the results and performance from the experiments. The biggest advantage of using a simulator over a real dataset is that we know with certainty how much fraud is present and where it is located. In a real dataset, it is impossible to guaranty that

there isn't any undetected hidden fraud. Since we control our malicious agents, we can flag all fraudulent behaviour, because we have prior knowledge about the level of fraud injected into the dataset. Measuring all the fraud present in a dataset is one of the biggest challenges when using real data, but not with synthetic data.

Finally the final step is to repeat from step three for different fraud scenarios. This methodology is an iterative process that does not end at the seventh step. The idea is that After analysing the results on the previous step there should be new question that aim to be answer. New questions may lead to new scenarios of fraud. Fraud detection methods can also be modified to improve the results. This way the PaySim simulator can be used in a loop to improve results and perform research in fraud detection.

Re-starting at step three is more effective for research because they can tweak the parameters to generate new scenarios. However, some researchers might chose to work on a previously generated dataset (starting at step 5) only to test different detection methods and compare results against previous research.

6. CONCLUSIONS

Fraud detection in financial transactions is affected by the availability of datasets for testing methods. PaySim is a simulation of mobile money transactions with the benefit that financial transactions can be generated at will following distributions obtained from real data.

Our approach presents an solution to the problem of sharing sensitive information in the parameters. We used a method called bootstrapping in order to generate the parameter files required from the original data using a synthetic dataset that resembles the original dataset. By doing this we avoid breaking any NDA and we are able to share the simulator and the parameter files with the research community.

In addition to this we present a method to reuse the PaySim simulator in similar or other domains that cover financial transactions.

We have available at: http://edgarlopez.net/simulation-tools/paysim/ the source code of the PaySim simulator together with a bootstrapped parameter files. We aim to give support to researchers interested in using our method or our simulator.

References

Chrystel Gaber, Baptiste Hemery, Mohammed Achemlal, Marc Pasquet, and Pascal Urien. Synthetic logs generator for fraud detection in mobile transfer services. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 174–179. IEEE, may 2013. ISBN 978-1-4673-6404-1. doi: 10.1109/CTS.2013.6567225.

Edgar Lopez-Rojas and Stefan Axelsson. Multi agent based simulation (mabs) of financial transactions for anti money laundering (aml). In Audun Josang and Bengt Carlsson, editors, *Nordic Conference on Secure IT Systems*, pages 25–32, Karlskrona, 2012a.

Edgar Lopez-Rojas and Stefan Axelsson. Using the RetSim simulator for fraud detection research. *International Journal of Simulation and Process Modelling*, 10(2):144, 2015a.

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

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

Edgar Alonso Lopez-Rojas and Stefan Axelsson. Money Laundering Detection using Synthetic Data. In Julien Karlsson, Lars ; Bidot, editor, *The 27th workshop of (SAIS)*, pages 33–40, Örebro, 2012b. Linköping University Electronic Press.

Edgar Alonso Lopez-Rojas and Stefan Axelsson. Social Simulation of Commercial and Financial Behaviour for Fraud Detection Research. In *Advances in Computational Social Science and Social Simulation*, Barcelona, 2014. ISBN 9789172952782.

Edgar Alonso Lopez-Rojas and Stefan Axelsson. Using the RetSim Fraud Simulation Tool to set Thresholds for Triage of Retail Fraud. In *20th Nordic Conference on Secure IT Systems, NordSec 2015*, pages 156–171, Stockholm, 2015b. Springer.

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

Edgar Alonso Lopez-Rojas, Ahmad Elmir, and Stefan Axelsson. A Review of Computer Simulation for Fraud Detection Research in Financial Datasets. In *Future Technologies Conference, San Francisco, USA*, 2016.

Roland Rieke, Maria Zhdanova, Jurgen Repp, Romain Giot, and Chrystel Gaber. Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis. In *2013 International Conference on Availability, Reliability and Security*, pages 662–669. IEEE, sep 2013. ISBN 978-0-7695-5008-4. doi: 10.1109/ARES.2013.87.

Maria Zhdanova, Jurgen Repp, Roland Rieke, Chrystel Gaber, and Baptiste Hemery. No Smurfs: Revealing Fraud Chains in Mobile Money Transfers. In *2014 Ninth International Conference on Availability, Reliability and Security*, pages 11–20. IEEE, sep 2014. ISBN 9781479942237. doi: 10.1109/ARES.2014.10.

AUTHORS BIOGRAPHY

MSc. Edgar A. Lopez-Rojas

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

Prof. Katrin Franke

Katrin Franke is a professor in computer science within the information security environment at NTNU in Gjovik. In 2007 she joined the Norwegian Information Security Lab (NISlab) with the mission to establish research and education in digital and computational forensics. In this context she was instrumental in setting up the partnership with the Norwegian police organisations as part of the Center for Cyber and information Security (CCIS). Dr. Franke is now leading the NTNU Digital Forensics group. Prof. Franke has 20+ years experiences in basic and applied research for financial services & law enforcement agencies (LEAs) working closely with banks and LEAs in Europe, North America and Asia.