Synthetic logs generator for fraud detection in mobile transfer services

Chrystel Gaber*, Baptiste Hemery†,‡,§, ¶, Mohammed Achemlal*, Marc Pasquet†,‡,§, ¶ and Pascal Urien‖
*Orange Labs
France Telecom 42 rue des coutures, F-14000Caen, France
Email: {chrystel.gaber,mohammed.achemlal}@orange.com
†Normandie Univ, France
‡UNICAEN, GREYC, F-14032Caen, France
§ENSICAEN, GREYC, F-143032, France
¶CNRS, UMR 6072, F-14032Caen, France
‖Telecom Paristech UMR 5141, F-75014Paris, France

***Abstract*—Mobile payments become more and more popular and thus are very attractive targets for fraudsters. As the latter always find new ways to commit crimes and avoid detection, research in the field of fraud is always evolving. However, transactional data and feedback from existing services are lacking. This article addresses this issue by proposing a synthetic data generator. Our idea is to model the behavior of various actors to generate testing data that researchers can use to evaluate approaches for identifying fraudulent transactions. This paper presents our approach and prototype. The logs generator was evaluated by comparing the generated synthetic logs with real ones.**

I. Introduction

Frauds in the field of electronic payment evolve continuously as new payment technologies and models are introduced. Fraudsters always imagine new ways to bypass security features and detections. Unfortunately, the research in fraud detection is limited because publicly available transactional databases containing frauds are scarce [1], [2]. Moreover, algorithm comparison is not reliable since the data's groundtruth may not be known with certainty and since there is no public reference dataset. This situation is mainly due to the fact that stakeholders like banks are very reluctant to publicly disclose information about frauds and their clients. All these difficulties are greatly increased for fraud detection in mobile payments. In this case, the data are not only confidential, but feedback about fraud is unsufficient. As a solution to this type of problem, the authors of [2] suggest that synthetic data could be created. This article describes how we modeled and simulated a specific mobile payment system to generate synthetic events and logs. As this work was done in the scope of the European FP7 project MASSIF, our model is based on the mobile-based transaction system and its users described by the MASSIF scenario providers in [3].

This paper breaks down into four parts. First, we examine existing works about the generation of synthetic transactional data. We also describe the benefits and drawbacks of using synthetic data to study fraud detection algorithms. Second, we present our model and its implementation. Third, we describe and discuss our preliminary evaluation of the logs generator and the preliminary validation of the underlying model with real data. Finally, we conclude and give the perspectives of this work.

II. Background

A. Related work

Synthetic data are not commonly used in the field of fraud detection although there is a lack of test data [1], [2]. To our knowledge, only two synthetic log generators exist in the field of fraud detection.

The first one [4] has been adapted to the field to the field of fraud detection in a Video-on-Demand system. Compared to this logs generator, ours can not only be set up with parameters driven from real data but it can also create logs with a pre-determined and not necessarily realistic shape. This simplifies the observation of specific characteristics of fraud detection algorithms. For example, we can create a set of users who change their behavior and others who do not in order to observe how algorithms react to this kind of problem. /textbfFinally, the authors of [4], [5] do not provide an evaluation of their model and the data they create.

The second one [6] models a mobile money system and money-laundering cases. While our simulator targets the same type of system, we do not model money-laundering cases but behavioral frauds [14]. This type of fraud correspond to a concurrent use of an account by a legitimate user and a fraudulent one and results in a shift of behavior. Our user behavior model is also more complex than [6]. While their user model is limited to random actions at a random time, we use the concept of habit to create a meaningful pattern of normal user behavior. Our implementation enables to model a user with multiple habits as well as random transactions. This results in a more realistic model.

Other methods are used to create synthetic data but do not specifically target the field of fraud detection. The closest approach [7], proposes to create synthetic data related to credit card transactions. However, this method does not generate data related to frauds and attacks as it aims at evaluating datamining methods in general.

B. Use of synthetic data for fraud detection

Synthetic data are commonly used in the fields of pattern recognition and machine learning. Although using real data is often preferred, using synthetic data enable to circumvent specific drawbacks of real data [5]. The properties required to study detection algorithms are not necessarily present in real data. For example, the training phase of some detection systems require large quantities of labelled data with overrepresented fraudulent events. Such kind of data are not always available for existing systems [5]. The quantity of data to stress test the detection systems are not necessarily available either [5]. Generally speaking, the major benefits of synthetic logs generator are :

- the possibility to generate as much data and as many different scenarios as neeeded,
- the control by researchers over parameters of the generated data,
- the possibility to create data with specific properties to test charactieritic features of the algorithms,
- the absence of privacy or disclosure issues which hinder the research in fraud detection,
- the possibility to test and chose detection algorithms for systems which are not yet deployed.

Some drawbacks balance the advantages mentioned above. There is no guarantee that the features observed on synthetic data are transfered to real situations nor that synthetic logs can give rise to anomaly detection systems that are effective against today's attacks as well as newly evolving ones. Moreover, the synthetic data may be biased or unrepresentative.

Given the characteristics of synthetic and real data, we consider that both types of data are complementary. We therefore consider that a comprehensive and totally reliable study of detection algorithms require the use of real and synthetic data.

III. Model and implementation

The system considered is the Mobile-based Money Transfer (MMT) service described in [3]. This service enables end-users to transfer money to other end-users or buy goods and services to merchants. These transactions are made with mMoney, which corresponds to electronic money emitted by the operator that manages the service. End-users can exchange cash for mMoney and vice versa at mMoney vendors by depositing or withdrawing cash from their MMT accounts.

The simulator is built according to the methodology proposed in [5] and which is depicted figure 1. As the methodology suggests, our simulator is composed of a simulated system, a user model and user profiles.

A. Simulated Mobile-based Money Transfer Platform

As the real platform described in [3], the simulated platform is made of (1) a front office which interacts with users and processes operation requests and connections to the service, (2) an account management system which controls accounts and processes financial operations, (3) a logs server and (4) a data warehouse which register the history of respectively the front

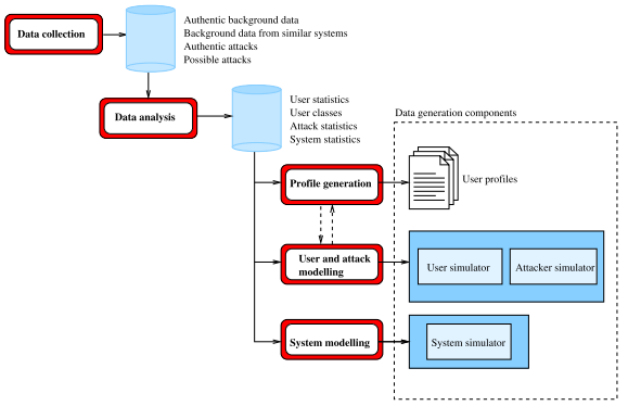


Figure 1: Synthetic log data generation method, source [5]

office and the account management. A security database was added to these components. It contains the user profiles which are necessary to authenticate users and to authorize a transaction. The payment sequence respects the following pattern [3]: (1) authentication, (2) transmission of sender's payment instructions and transaction details to the MMT platform, (3) authorization by the MMT platform, (4) credit and debit on the receiver and sender's accounts.

The logs are created when a simulated person carries out a transaction. To achieve this, the behavior of those who can interact with the platform has been modeled.

B. Users behavior model

At first sight, logs may look like noise. However, the events related to one user form a story and the events registered in the logs are the result of actions carried out in parallel by different actors. In order to recreate this overall complexity, we propose (A1) to model individual trajectories and to combine them. This first assumption (A1) is the basis of our logs generator. This approach corresponds to multi-agents models [8]. In our case, the agents are the various actors of the system: (1) the legitimate users who subscribe to the Mobile-based Money Transfer Service and thus own an account and (2) the fraudsters who attack the system.

Legitimate actors. Three categories of legitimate actors are involved in the MMT system. Each category is made of several roles which are associated to specific actions in the platform. *End-users* are individuals who use their mobile devices to access to the MMT platform through the network provided by their operator. They carry out transactions. *Service providers* sell services or goods to end-users. *Channel users* are in charge of the distribution of electronic money. Other actors of the system can either acquire or sell electronic money from or to them. Service providers and channel users can access to the MMT platform through their mobile devices.

Our model is based on the assumption that legitimate users transactions are mostly related to their habits. This means that legitimate users tend to carry out frequently and repeatedly a specific set of transactions. This assumption is also taken in the fraud detection methods which are based on anomaly detection [9], [10]. In this field, it is considered that normal transactions correspond to legitimate actors habits and that

fraudulent transactions inevitably differs from normal ones. Frauds are therefore searched among outliers.

Our definition of a habit is inspired from [10] which considers that "*a habit is an equivalence class on legitimate transactions*". We go further by defining that a habit is a repetition of a sequence of legitimate transactions which are characterized by (1) a type of transaction, (2) a normally distributed transaction amount, (3) a normally distributed period of time that separates two transactions of the considered habit, (4) an initial date and (5) a final date. In our first version model presented here, we consider only one-event sequences. We assume (A2) that a user's behavior is composed of a set of habits $H = \{H_1, H_2, ..., H_i\}$, where H_i is a habit for one specific type of transaction. Based on [11], we believe that the same applies to the actions carried out by service providers and channel users although this is not verified.

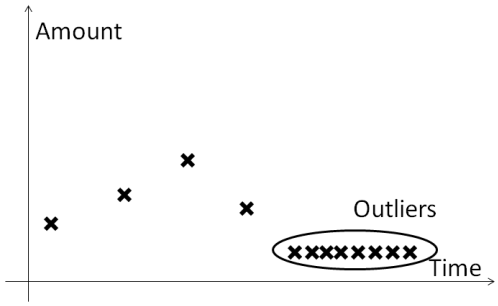
We also assume (A3) that the activity of each actor in the system is restricted to a consolidated set of end-users, service providers and channel users. This set of persons who interact on a regular basis with an actor compose his Community Of Interest (COI) [12]. This concept has also been used in the field of fraud detection in telecoms [13].

Fraudulent actors. The type of frauds considered here are behavioral frauds [14]. This type of frauds is realistic for mobile payment as fraud cases since they have been observed in the field of credit card payment [14] and telecommunications [15]. They correspond to scenarios where accounts are taken over after a device is stolen or corrupted. In this case, transactions are not made by legitimate users but by another entity. It is generally considered that when such a fraud occurs, shifts can be observed in the user's behaviors. Figure 2 shows typical examples of behavior shifts: change in the user's mean transaction amount(figure 2.a) and transaction frequency (figure 2.b).

We consider that this type of fraudsters do not need to hold an account in the MMT system and that their attacks follow a specific pattern which is modeled in our log generator. For the moment, two types of attacks have been modeled. The first one corresponds to a thief who attacks end-users and steals their mobile device. He then makes several attempts to guess the legitimate user's authentication code. When he succeeds, he carries out several transactions, purchases or money withdrawals, with several other actors. The second type of attack we modeled is a fraudster who deploys botnets among the population of end-users. The bots then carry out transactions without the legitimate user's consent or authorization. In the current model, the bot chooses a mule among the fraudster's mule COI and carries out a transaction at a random time with a random amount. Mules are legitimate end-users of the system who are used by fraudsters to retrieve stolen money [16]. The bots send money to mules whose role is then to withdraw the money or buy items that are then sent to the fraudster (The MMT platform only registers the mule's purchase or withdrawal). A mule is modeled as a legitimate user with a set of habits to which a specific behavior is added. We consider that the mule will withdraw money or make a purchase corresponding to the amount of money stolen a certain time after he receives it.

Discussion. Let us consider an example of habit. The

(a) Mean amount



(b) Mean frequency

Figure 2: Changes in spending habits

amount follows a normal distribution with a mean value of 6 and a standard deviation of 4. The period of time between two transactions follows a normal distribution with a mean value of 12 and a standard deviation of 1. The probability density function of this habit is represented in figure 3. As a result of the model defined previously for the legitimate and fraudulent actors, the simulator mostly generates the legitimate transactions according to this probability density function. Therefore, the events located within the ellipses have a very high probability to correspond to a legitimate transaction whereas any transaction located out of the ellipses are considered as outliers. They have a higher probability to be a fraud.

C. Implementation

The simulated MMT platform and the users behavior model have been implemented on the agent-based modeling and simulation platform Repast Simphony [17]. The combination of habits and behaviors were created with the decorator design pattern [18]. It is an alternative to subclassing where each decoration is a brick which can be combined to create a complex behavior as shown figure 4. If two users are implemented according to a specific user type, their behavior's parameters can differ. Fraudsters are implemented in the same way except that we do not consider habits but fraud patterns.

Two log files are generated by the simulation, one for all types of transaction events and another one for failed authentication events. The fields in the log files are the type of operation, the sender's and receiver's phone numbers and account numbers, the transaction's amount, date and groundtruth.

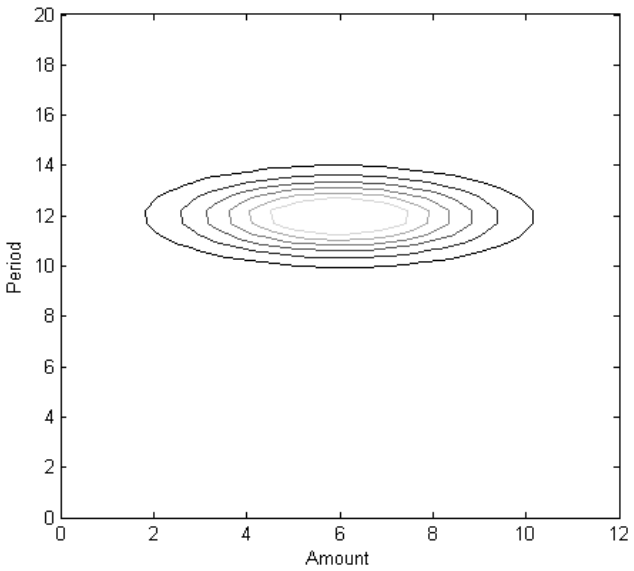
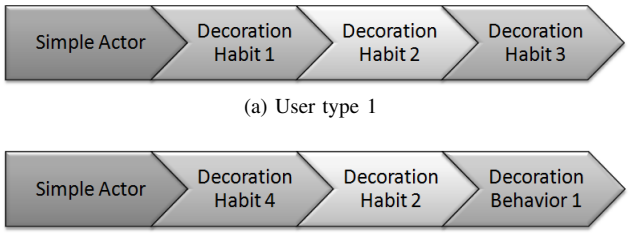


Figure 3: Probability density function of a habit



(a) User type 1

(b) User type 2

Figure 4: Decorator design pattern

IV. Preliminary evaluation of the logs generator

We wish to evaluate how the proposed prototype achieves to generate synthetic data related to the MMT System. The simulated MMT platform and the format of the generated logs have been validated by MASSIF's MMT scenario provider. We will now evaluate the users behavior model. At the current stage of development, we wish to validate the assumptions (A1), (A2) and (A3) with a real transactions dataset of three months which was provided by the MMT scenario provider. This dataset is confidential and we are not authorized to disclose detailed characteristics about the real logs. The dataset is composed of 58 725 user and 320 528 transactions after the data preparation phase.

A. Data preparation

Only the successful transactions related to end-users were kept because in this first prototype they are our main concern. Then, the end-users were separated in seven different sets as described in table I. The groups were created to highlight different behaviors which may affect how we interpret results and validate our assumptions. For example, an irregular or new end-user may not display habits as assumed in (A2) or have a COI as assumed in (A3). Five end-users were then chosen randomly in each set. Only end-users with more than 30 transactions were considered in order to have a sufficient number of transactions for the analysis. Finally, for each of those five users, the transactions were separated according to their type: Money deposit (MD), Money withdrawal (MW), Airtime recharge (AR), Merchant Payment (MP) or Client-to-Client transfer (C2C). We consider that for one user, each type of transaction corresponds one behavior.

Sets	Deposit and withdrawal	C2C transfer	Purchases
A	8.12	17	-
B, C	13.25	5.83	-
D, E, F	14.05	5.83	1.5
G	4.03	1.5	-
Total	11.54	5.42	1.25

Table III: Number of transactions per user's transactional partners

B. Results

The microscopic-level assumptions (A2) and (A3) are evaluated before the macroscopic assumption (A1) because they are the foundation of our approach and that if they are false, (A1) cannot be verified. For the validation of (A2), χ^2 tests with a significance level of 5% were used to check whether the amounts and periods are normally distributed. Only the five users selected from each set during data preparation were considered. The results are summarized in table II. The results are organized according to the user's category defined in table I and the category of transaction considered as a habit. The column users indicates the number of users of the category who display the considered habit. The columns amount and period show the proportion of the concerned users for which the amounts or periods of transaction are normally distributed according to the χ^2 test. For example, all the selected regular users display a deposit behavior and all five of them have a normally-distributed amount and period. As (A1) is confirmed for a majority of user categories and types of transaction, we conclude that users behavior can be modeled by habits as defined in III-B. We believe that the discrepancies are either due to multiple habits for one type of transaction or to unexpected events for the end-user. Both will be investigated in future works for a fine-grained model.

In order to verify (A3), the average ratio of transactions to the user's transactional partners was calculated for each of the thirty five selected users and type of partner. Only users who made more than one transaction were considered for each type. The results are gathered in table III. For example, we estimate that an end-user from set A makes 17 C2C transactions per partner. We consider that the data confirm (A3) as we observe that users carry out several transactions with one transactional partner whether the latter is another end-user, a mMoney vendor or a merchant.

Finally, (A1) was tested by setting up our prototype with the profile of the thirty five selected users. Three independent simulations were run. The accuracy of the logs generated was measured by calculating the symmetric mean absolute percentage error (SMAPE) for several macroscopic indicators: number of transactions, total amount of transactions and mean transaction amount. The SMAPE of an indicator I is equal

Set	Month 1	Month 2	Month 3	Possible category of user
A	x	x	x	Regular end user
B		x	x	Irregular or new end users
C			x	Irregular or new end users
D	x	x		Irregular or former end users
E	x			Irregular or former end user
F		x		Irregular or former end users
G	x		x	Irregular end users

Table I: Sets of users according to their registered activity

	MD			MW			C2C			MP			AR		
Sets	amount	period	users	amount	period	users	amount	period	users	amount	period	users	amount	period	users
A	100%	100%	5	100%	100%	2	100%	100%	5	-	-	0	60%	100%	5
B, C	70%	70%	10	100%	100%	3	78%	67%	9	-	-	0	50%	83%	6
D, E, F	87%	80%	15	100%	100%	5	100%	75%	12	100%	100%	2	67%	92%	12
G	100%	100%	5	100%	100%	5	100%	100%	2	100%	100%	1	100%	100%	5
Total	86%	83%	35	100%	100%	15	93%	79%	28	100%	100%	3	68%	93%	28

Table II: Results of the χ^2 test for each set of users

Transaction type	MD	MW	C2C	MP	AR	All
Number of transactions (%)	12	5	3	6	31	18
Total Transaction Amount (%)	3	3	3	3	3	5
Mean Transaction Amount (%)	14	11	19	2	21	23

Table IV: Evaluation of the log generator's average accuracy

to $\frac{1}{3}\sum_{j=1}^{3}\frac{A_{I,j}-S_{I,j}}{A_{I,j}+S_{I,j}}$, where $A_{I,j}$ and $S_{I,j}$ correspond respectively to the actual and simulated value of the indicator I in the run j. We chose this accuracy measure because it facilitates the interpretation of results. The closer the SMAPE value is to 0%, the more accurate the evaluation is and the closer it is to 100%, the less accurate it is. The various indicators were measured for all the transactions and for each type of transaction. Table IV gathers the evaluation of the log generator's accuracy for each indicator. We observe that the least accurately modeled property is the number of airtime recharges with a 31% average error and that the most accurately modeled indicator is the mean amount of merchant payments with a 2% average error. We also observe that among all three indicators, the total transaction amount is the most accurately simulated. It seems that the period and amount parameters tend to be overfitted which leads to less frequent transactions of higher value. We argue that, as the overall structure of the logs and the overall amount of money moved is respected, the logs generated are acceptable.

C. Discussion

The preliminary evaluation and validation were carried out with 35 users distributed among 7 sets of users. Only the 1 644 users who had done more than 30 transactions were considered because we believe they were the most challenging for the evaluation of our three assumptions. We limited our preliminary study to 35 users because the simulator is currently parametered manually and it seems unreasonable to parameter 1630 different users. The class which currently contains the profiles of the 35 users contains 915 lines. Building more users manually would be too time-consuming and the probability of mistakes would be high. Moreover, we wanted to have a feedback on our assumptions before developing our behavior model and simulator any further.

Such an evaluation was not carried out for the two similar generators [4], [6]. Barse *et.al.* [4] does not evaluate their results at all. Lopez-Rojas and Axelsson do not validate the statistical distribution of the logs they generate. They perform practical tests by running different fraud detection algorithms on their synthetic data but without comparision with real data. Therefore, there is no reference result about the validity of the existing models. These preliminary results were considered as sufficient at this stage of the MASSIF project. The simulator has been used as part of a demonstrator which shows the outcomes of the correlation and countermeasure modules. It was used for the MASSIF project review by the European Commission. It will also be used during the last year of the project for the evaluation of other modules of the MASSIF project.

As a result of this evaluation, we wish to continue to model users with the concept of habits and we wish to carry out an enhanced validation. This will be possible as we are currently developping a graphical user interface for the simulation.

V. Conclusion and future works

This article presents our approach to generate synthetic data of a mobile-based transfer service for the evaluation of fraud detection algorithms. Our logs generator consists of a model which simulates the payment platform and a user's behavior model which simulates the actions of the actors of the mobile-based transfer system. Both modules and their implementation were presented in this article as well as a preliminary evaluation.

In future works, we will complete our validation. We also wish to research and integrate enhanced user behavior models. We also plan to use the synthetic data to carry out our research in the field of fraud detection for mobile payments. Finally, our simulator has been linked to other MASSIF components (correlation and countermeasure modules) in order to demonstrate and evaluate MASSIF's results.

REFERENCES

[1] Richard J. Bolton and David J. Hand. Unsupervised profiling methods for fraud detection. In *Conference on credit scoring and credit control*, 2001.

[2] Clifton Phua, Vincent Lee, Kate Smith-Miles, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, 2005.

[3] Mohammed Achemlal, Saïd Gharout, Chrystel Gaber, Marc Llanes, Elsa Prieto, Rodrigo Diaz, Luigi Coppolino, Antonio Sergio, Rosario Cristaldi, Andrew Hutchison, and Keiran Dennie. Scenario requirements. http://www.massif-project.eu/, March 2011.

[4] E.L. Barse, H. Kvarnstrom, and E. Jonsson. Synthesizing test data for fraud detection systems. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 384 – 394. dec. 2003.

[5] Emilie Lundin, Hakan Kvarnström, and Erland Jonsson. A synthetic fraud data generation methodology. In Robert Deng, Feng Bao, Jianying Zhou, and Sihan Qing, editors, *Information and Communications Security*, volume 2513 of *Lecture Notes in Computer Science*, pages 265–277. Springer Berlin Heidelberg, 2002.

[6] E.A. Lopez-Rojas and S. Axelsson. Multi agent based simulation (mabs) of financial transactions for anti money laundering (aml). In *Nordic Conference on Secure IT Systems*. Blekinge Institute of Technology, 2012.

[7] D.R. Jeske, B. Samadi, P.J. Lin, L. Ye, S. Cox, R. Xiao, T. Younglove, M. Ly, D. Holt, and R. Rich. Generation of synthetic data sets for evaluating the accuracy of knowledge discovery systems. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 756–762. ACM, 2005.

[8] J. Ferber. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 248. Addison-Wesley, 1995.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[10] A.I. Kokkinaki. On atypical database transactions: identification of probable frauds using machine learning for user profiling. In *Knowledge and Data Engineering Exchange Workshop, 1997. Proceedings*, pages 107–113. IEEE, 1997.

[11] William Jack, Suri Tavneet, and Robert Townsend. Monetary theory and electronic money: Reflections on the kenyan experience. *Economic Quarterly*, 96-1(96):83–122, First Quarter 2010 2010.

[12] William Aiello, Charles Kalmanek, Patrick McDaniel, Subhabrata Sen, Oliver Spatscheck, and Jacobus Merwe. Analysis of communities of interest in data networks. In Constantinos Dovrolis, editor, *Passive and Active Network Measurement*, volume 3431 of *Lecture Notes in Computer Science*, pages 83–96. Springer Berlin Heidelberg, 2005.

[13] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Communities of interest. In *In Proceedings of the Fourth International Conference on Advances in Intelligent Data Analysis (IDA*, pages 105–114. 2001.

[14] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J. Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50:602–613, 2011.

[15] R.J. Bolton and D.J. Hand. Statistical fraud detection: A review. *Statistical Science*, pages 235–249, 2002.

[16] Matthew DeSantis, Chad Dougherty, and Mindi McDowell. Understanding and protecting yourself against money mule schemes. http://www.us-cert.gov/reading_room/money_mules.pdf, 2011. last visited on 08/10/2012.

[17] M.J. North, T.R. Howe, N.T. Collier, and J.R. Vos. A declarative model assembly infrastructure for verification and validation. In Shingo Takahashi, David Sallach, and Juliette Rouchier, editors, *Advancing Social Simulation: The First World Congress*, pages 129–140. Springer Japan, 2007.

[18] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design patterns: Abstraction and reuse of object-oriented design. In Oscar Nierstrasz, editor, *Object-Oriented Programming*, volume 707 of *Lecture Notes in Computer Science*, pages 406–431. Springer Berlin, 1993.