Community Detection for Mobile Money Fraud Detection

Safa El Ayeb, Baptiste Hemery, Fabrice Jeanne, Estelle Pawlowski Cherrier

Community Detection for Mobile Money Fraud Detection

Safa El Ayeb*†, Baptiste Hemery*, Fabrice Jeanne* and Estelle Cherrier†
*Orange, Caen, France
Email: {safa.elayeb, baptiste.hemery, fabrice.jeanne}@orange.com
†Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
Email: estelle.cherrier@ensicaen.fr

***Abstract*—This paper presents an overview of a first-year PhD thesis. The main idea can be expressed as a community detection issue in a real network containing data from a telecommunications operator. More precisely, mobile money transactions will be studied, with a view to detecting fraudulent ones. For this purpose, we propose to address the problem by studying different community detection approaches. Indeed, in a network, communities are constituted between users sharing common habits, therefore fraudulent transactions may appear as unusual behaviors. The originality of this work is to process real data, gathered by a telecom operator. Complex network organization is expected, making this thesis quite promising.**

***Index Terms*—Mobile Money, Social Network Analysis, Community detection, Fraud detection.**

I. Introduction

Mobile money services permit users to use their money with their mobile phone. These services are mainly deployed in developing countries, as an alternative to traditional bank accounts. Users can push money into their mobile money account or take it back via some retailers. They can also use these services to send money to other users.

All these operations can be done with a mobile phone. Users can use a dedicated application on a smartphone, or use the USSD (Unstructured Supplementary Service Data) which is available on smartphone and feature phones as well. Most users access the mobile money service through USSD, as it is cheaper and feature phones are more widely spread in developing countries than smartphones.

However, USSD is vulnerable to an attack called *SMShing*, a specific form of phishing attack performed via a SMS. In this kind of attack, the fraudster sends to the victim a SMS containing a USSD link. The victim clicks on the link, which automatically (and unwittingly) initiates a transfer from his mobile money account to the fraudster one. It is important for mobile money service providers to prevent this kind of fraud.

From the service provider point of view, it is possible to collect all operations carried out by users. These operations are seen as a collection of transactions, from a sender to a recipient, with a certain amount, a type (cash-in, peer-to-peer, ...), a date, etc. ... These transactions can be considered as the edges of a social network of users. Community detection algorithms applied on that network might help to identify fraudulent transactions. Indeed, a transaction within a constituted community might be legitimate while a transaction that appears between two different communities might be fraudulent. Thus, we plan to research how the detection community techniques could help detect fraudulent transactions, in the specific case of SMShing.

The article further presents the context of a mobile money service as well as the specific use case of SMShing in section II. Section III presents existing approaches for community detection and fraud detection using social networks. The proposed research methodology including the dataset description, the methodology and the evaluation metrics are then discussed.

The authors would like to thanks Orange and the ANRT for funding this work.

II. Problem

A. Context

Mobile payment solutions first appeared in 2007 and since have developed considerably. While the leading country remains Kenya with the M-Pesa solution, Orange has been present in Africa since 2008 and is developing very rapidly, with a presence in more than 18 countries today. Mobile payment solutions meet a need, namely they offer financial services, based on a prepaid account, using almost exclusively cash, to a population with very few bank accounts. The initial service proposal was simple: cash in/out, transfer, merchant payment, airtime recharge... all accessible from basic feature phones via USSD.

Since then, the range of services has been greatly enriched by offering services similar to those offered by the banking sector: credit, savings, bank to wallet, web payment, international transfer, bill payment...

At the beginning of 2020, the Orange Money service already had more than 50 million customers. With the health crisis, central banks, governments and health authorities have largely encouraged the population to turn away from using cash and to massively move towards mobile payment solutions deemed more secure.

B. Motivation

As in any financial service, its development and its uses are unfortunately accompanied by an increase in fraud attempts, which can become very damaging in terms of the operator's

confidence and image. Technical measures are obviously implemented to guarantee the integrity of the service and the security of users.

Nevertheless, among the most common frauds and the most difficult to bypass, is a variant of SMS phishing (more rarely via a phone call): SMShing. Operators implement technical counter-measures and accompany customers who are victims of such fraud (refunds). However the impact in terms of image or loss of confidence is difficult to measure.

Among the counter-measures, the implementation of blacklists of numbers identified as *fraudulent* seems the simplest to implement, but is not necessarily the most reactive and effective. It is a common usage that a person owns several SIMs, accustomed to regularly changing operator and therefore to regularly opening/closing their mobile banking account. As these accounts are prepaid, they can easily be created temporally for a day or two, and be closed after cash is withdrawn. For mobile money provider, it is impossible to act after the fraudster has closed the account.

However, these fraud attempts could be thwarted with the help of solutions based on the observation and analysis of current usages.

Accounts corresponding to honest users are characterized by several features: an older existence, a longer life span, frequency, and a relatively limited number of recipients. These transactions progressively constitutes a real social network based on internal or external transactions within and between communities.

By nature, a community evolves over time and the strengthening of certain links (while other links disappear at the same time), allows to define stable communities, to qualify the internal or external links within or between the communities and thus to isolate weak links to ephemeral nodes that are indicators of a potential fraud attempt.

Given the large volume of data and the need for regular community updates to maintain a sufficient level of efficiency and avoid *false positives* it is necessary to find the most effective means of building the foundations of a graph-oriented data analysis tool.

III. Research challenges

A. Problem statement

Community detection has been widely studied during recent years in social network analysis. They define a portion of the society that is well connected. Among networks, connections can have different forms, varying from friendship, advisory, common interest, or even trust.

Therefore, the goal of our work is to study mobile money transactional data from a social network perspective. Given a set of transactions among users, how can we spot fraudsters? An inspection of the social network can unveil the way fraudsters behave, and the pattern of the ties they have with other users.

To understand how mobile money is flowing, communities across the network are going to be identified. After communities are formed, different features will be studied, and different patterns of activities are to be built. This step will enable the discovery of deviations from a common behavior. These deviations will be considered as anomalies. For example, an anomaly in this context can be represented by isolated actors performing transactions to unlikely destinations.

In order to handle the massive data volume recorded and exchanged across networks, big data frameworks and techniques will be used to manage different tasks like storage, visualisation, search, analysis, etc.

B. Existing approaches

Data we are concerned with in this paper can be represented as a social network. Taking into account the context and the problem formulation detailed in the previous sections, we are particularly interested in the community detection problem, which is an important issue when dealing with graphs and networks.

The main objective of social networks analysis is to understand individual behaviours, based on links and relationships within social networks. It is aimed at exploring both the underlying social structures and the influence of each feature characterizing different associations [1]. It also has the advantage of offering a common language shared by various research fields (biology, sociology, technology, information science, etc) [2].

There are many approaches to define social networks. From a mathematical point of view, graph theory provides a vocabulary which can be used to characterize social structural properties. Boccaletti *et al.* [3] describes graph theory as a natural framework for the exact mathematical analysis of complex networks. The visual representation of data as graphs also allows to uncover patterns that might otherwise remain undetected.

A network can be represented as a graph $G = (V, E)$, consisting of a set of vertices (or nodes) connected through a set of edges (or links). Matrices are an alternative way to summarize network data. Adjacency matrix in particular is widely used. It contains exactly the same information as a graph, but is more useful for computation purposes. Social graphs may be directed/undirected or weighted/unweighted or complex (having multiple edges)/simple.

For basic definitions and notions about graphs and networks, we refer the reader to the pioneering works of Girvan and Newman [4] or Fortunato [5].

Among the fundamental concepts for studying social graphs are centrality and modularity. Centrality gives a measure of the relative importance of a node in the network. There are three main measures of centrality: degree, closeness, and betweenness.

Degree centrality The degree is the number of direct ties an actor has. The higher degree an actor has, the more influential it may be in the network. For directed graphs, the degree of an node is divided into in-degree and out-degree.

Closeness centrality Closeness focuses on how close an actor is to all the other actors. It considers the geodesic distance

from each actor to all others. An actor is central if it can access all others more quickly than anyone else in the network.

Betweeness centrality Betweeness provides information on how important an actor is in the average pathways between other actors. An actor with high betweenness has a considerable influence over what flows in the network.

On the other hand, modularity estimates the fraction of incoming links in a community minus the expectation value of the same quantity in a network with the same community divisions but random connections between the nodes [6]. This measure provides insight on the strength of communities structures.

In this thesis work, we are concerned with real networks, based on mobile money transaction data and CDR (Call Data Record). In [5], the author notices that real networks exhibit big inhomogeneities, both globally and locally. Typically, real networks are dynamic, heterogeneous and multiplex. Nodes are usually qualified by a set of attributes and are related by different types of relations. This results in a high level of order and organization. Moreover, in real networks, edges can be concentrated within some groups of vertices (or nodes), while being less concentrated between these groups: this particular structure had led to issues related to community detection, or clustering. More precisely, a community, or cluster, represents a group of vertices either sharing some properties or playing similar roles inside the network. Namely, communities are subgraphs of a graph. These subgraphs contain densely connected vertices. This is an active research field, since social media (such as Facebook, Twitter...) are highly interested in identifying communities at different levels: at the family level, between friends or colleagues, neighbours etc. For companies like Facebook, Amazon, Google, Microsoft..., understanding how these groups evolve is of high value. Depending on the characteristics of the network, different types of communities can be identified. We can classify communities into three main categories:

Disjoint communities Communities that have no mutually common nodes.

Overlapping communities Communities that share membership or adjacency of one or more nodes with other communities.

Dynamic communities Communities that grow over time. They can expand or dissolve.

In the following, we summarize the existing community detection techniques.

1) Hierarchical clustering methods: Social networks often have hierarchical structure. Hierarchical methods illustrate the multilevel topology of networks. They aim at splitting the network into a finite number of groups based on a similarity metric. This similarity metric evaluates nodes closeness and determines the clusters to which they belong. The outcome of hierarchical methods is generally presented by a hierarchical tree also called dendrogramme. There are two types of hierarchical methods: divisive and agglomerative.

a) Divisive methods: Divisive algorithms start with the network as one community and at each step divide it into groups by deleting edges progressively. Girvan and Newman [4] proposed a divisive method where edges are eliminated based on edge-betweenness. They later generalized their algorithm considering both random-walk betweeness and current-flow betweenness [7].

In 2016, Moon *et al.* [8] adapted the Girvan-Newman method using a parallelization algorithm based on the MapReduce model and the vertex-centric model. These models were implemented on big data platforms such as *Hadoop* and *GraphChi*. Parallel Computing enabled the algorithm to discover communities in large networks.

b) Agglomerative methods: Agglomerative methods are bottom up approaches that start with leaf nodes and build clusters by merging nodes at every iteration, when their similarity is significant. In his work, Newman [9] proposed an agglomerative algorithm based on the modularity metric. Starting with nodes as individual clusters, the algorithm chooses at each step the join that leads to the greatest increase of modularity. The cut leading to the maximum value of the modularity is then selected.

In their paper, Wang *et al.* [10] has presented a novel agglomerative method founded upon link clustering. Instead of considering nodes, this method calculates the similarity between edges based on structural information. At every step, edges with high similarity are merged until the whole network becomes one community. Link communities are then converted to node communities. The advantage of the method is the accuracy of its partitions and the fact that it enables the creation of overlapping communities.

Hierarchical methods have been largely used for the detection of communities in networks so far. On the one hand, they have the advantage that they don't require to specify the number or the size of communities in advance. On the other hand, the hierarchical structure they produce is sometimes artificial, since it does not reflect the real structure of the original network.

Another weakness of hierarchical methods is that they depend fundamentally on a the similarity measure chosen for the clustering.

2) Modularity based methods: Newman [9] has shown that better network partitions are correlated with high modularity value. Modularity maximization methods are based on this theory to construct communities. This class of methods was widely used in the literature thanks to the versatility of modularity that can be extended to weighted and directed graphs [11]. Modularity optimization methods can be classified into three categories:

a) Greedy techniques: Clauset *et al.* [12] proposed an algorithm that greedily joins nodes to optimize modularity gain with an agglomerative hierarchical approach. When the CNM algorithm was proposed, it was one of the few algorithms capable of detecting communities on large networks in a relatively short time [13].

b) Simulated annealing: Simulated annealing is a probabilistic procedure that was first employed for modularity optimization by Guimerà *et al.* [14]. It is based on the search of a global optimum. The method combines two types of moves: a local move that replaces vertices from a cluster to another, and a global move that merges and splits communities if this action increases the modularity value.

c) Extremal optimization: Extremal optimization represents a heuristic search procedure proposed by Boettcher and Percus [15]. This method aims at reaching higher values of modularity by replacing nodes with low modularity from one cluster to another. In fact, modularity is expressed as a sum over vertices, where local modularity of a vertex is the value of its corresponding term in this sum [5]. The Extremal optimization method starts with a random partition of the network in groups with the same number of vertices. Vertices having the lowest level of fitness are then replaced with another vertex from the neighboring communities. The fitness value of a vertex is given by the corresponding local modularity divided by its degree. After every replacement, fitness values are updated and the process is iteratively implemented until the global modularity can not be improved.

Even though modularity optimization methods are easy to implement and have a good time complexity, they present a resolution limit, which implies that relatively small communities compared to the whole network are hard to detect [16].

3) Spectral methods: Spectral methods are inspired from linear algebra. They implement the projection of the networks' nodes on a k-dimensional space whose coordinates are eigenvectors of a similarity matrix. Generally, the Laplacian matrix is used. A partitional algorithm (such as k-means) is then applied on these eigenvectors. A very detailed tutorial of spectral algorithm can be found in [17]. Pothen *et al.* [18] have based their spectral algorithm on a parallel computing of the Fiedler vector [19]. This vector is the second eigenvector of the Laplacian matrix and contains information about the connected components in the network. The bisection of the network is accomplished using the median of this vector. Barnard and Simon [20] have proposed an alternative of this algorithm with a recursive multi-level approach. Instead of considering the Fiedler vector exclusively, Jianbo and Malik [21] have considered the k-first eigenvectors of the Laplacian matrix, in the context of image segmentation. A k-means is then applied in order to obtain k clusters, and then by adaptation to the graph, k communities. While the majority of spectral algorithms were applied on the eigenvectors of the Laplacian matrix, White and Smyth [22] and Newman [23] explored a modularity matrix, also called Q-Laplacian.

One of the drawbacks of spectral methods, is the fact they require to specify the number of communities requested beforehand. It has also been proved that spectral methods are incapable of discovering small communities within densely connected networks [24].

4) Label propagation methods: The label propagation method was introduced by Raghavan *et al.* [25]. It represents an iterative method built around the topological structure of networks. It forms communities based on the process of label propagation. The principle of this method is that a label can spread quickly and become dominant in a dense group of nodes, but it will be harder for it to cross to less connected components of the network [26].

The label propagation method starts with every node having an initial label. At every iteration, nodes update their labels to the maximum label of their neighborhoods using a voting mechanism and links are then broken. The algorithm converges when every node has the maximum label of its neighbors. Nodes sharing the same label are grouped into the same community.

Leung *et al.* [13] proposed a label propagation algorithm for the detection of communities in large dynamic networks. The novelty of this algorithm is a score function that decreases progressively when it moves from the source node. This "hop-attenuation" restrained the spread of the label from a random center as well as the formation of "monster clusters" at the expense of local communities.

In order to address the random assignation of labels, Zong-Wen *et al.* [27] introduced a label propagation algorithm with consensus weight *(LPAcw)*. This algorithm executes the basic label propagation approach repeatedly in order to obtain different sets of partitions. For each edge, a weight is computed. The weight of an edge is the proportion of the number of nodes assigned to the same group divided by the total number of groups. A weighted graph is then created. A consensus weight is proposed to specify the propagation direction of labels. Nodes update their label based on the mixing of consensus weight and label frequency.

Zhang *et al.* [28] have also worked on the randomness of the label propagation method. Thus, they developed algorithms based on node importance within the network. Since most influential nodes are more likely to impact other nodes, their labels is more likely to be spread.

Despite their efficiency for community detection and their wide use, label propagation methods have some problems. Among those shortcomings, Berahmand and Bouyer [29] have addressed the instability of these methods and monster community detection. Yamaguchi and Hayashi [30] have also specified that label propagation algorithms does not work well with disassortative node labels (ie connected nodes tending to have different labels) and communities having different edge densities.

Based on these applications, social network analysis can be seen as a tool for detecting fraud. This tool relies on analyzing activities and relationship on the network level, and identifying abnormal linkages between individuals [31]. The major domains where fraud is rapidly growing are e-commerce, telecommunication, online banking, insurance....

In this context, Noble and Cook [32] have introduced an algorithm for detecting anomalies in the network by detecting unusual structures. Suspicious structures are simply parts of the graph that produce low values on a heuristic they have selected.

Sun *et al.* [33] have considered a bipartite graph on which they computed the relevance of nodes using a random walk approach. Anomalies are then identified applying a graph partitioning algorithm and relevance scores. A node is considered abnormal if it has a low normality score.

In 2007, Pandit *et al.* [34] have worked on detecting fraud within the social network of online auctioneers. For their research, the authors combined the analysis of the fluctuation in sellers behaviors with their interaction graphs among immediate neighbors. Another algorithm based on assessing scores is proposed by [35]. Unusual groups are attributed suspicious scores leading to the identification of fraudsters.

In the context of credit card transactions, Van Vlasselaer *et al.* [36] have proposed an approach considering both intrinsic and network-based features. Intrinsic features are represented by customers' spending history. Network-based features displays the network of credit card holders along with merchants. Although considering these features separately offered good results in detecting fraud, combining them have led to a more performing model.

C. Proposed methodology

In this section, we are presenting the methodology to evaluate our proposed mobile money fraud detection algorithm, based on community detection in social network. First, a description of the dataset used for the analysis is provided. We then present our first experiment aiming to evaluate community detection algorithm, with regards to our use case. At last, we present a second experiment evaluating our fraud detection algorithm.

1) Dataset description: The used dataset in our experiments contains the financial transactions made by users of a mobile payment system. Each transactions is made of several fields such as sender, receiver, date, amount, and type. We will focus on four main types : Cash-in, Cash-out, merchant payments, and peer-to peer transfers. Cash-in is an increase of the customer's balance by paying cash. Cash-out, in the contrary, is a withdrawal of cash in exchange of a decrease in the balance. Payments are money used to acquire goods or services. Transfers are moving money from one account to another using the mobile money platform.

For the analysis, we are provided with a high volume of data. However, due to the sensitive nature of financial data, and the strict organisations' internal policies to protect users' confidentiality, we cannot disclose any information about users. That's why simulated data will be used in the first place for our experiments. Simulated data are "self-sufficient data with the goal of having similar statistical properties as the original ones" [37]. We plan on using such a simulator to create our simulated dataset, either PaySim [37] or one we are going to develop to fit our usecase. The synthetic data generated emulate normal customers' behavior as well as fraudsters' behavior. Simulation is also useful to assess the outcome of tested algorithms, before applying them on real data. Real data may be random and can't provide a ground truth needed for benchmarking different methods. Our goal is to have a large dataset representing real world data, with tens of million users and hundreds of million transactions.

2) Community detection evaluation:

The first step of the project consists of evaluating community detection algorithms on our transactions social network.

For this purpose, it will be necessary to define which features on the dataset will be used. It is also necessary to define upfront the granularity of the network and the nature of ties joining nodes, and their weight as well as the characteristics of the requested communities.

Different algorithms of community detection explored in the state of the art will be implemented. As mentioned above, the algorithms will be tested on simulated data. These simulated data will help test the efficiency of the algorithms before applying them on real data. The performance of different development environments such as *Neo4j* and *Spark*'s component *GraphX* will be compared as well.

To evaluate algorithms, two trends emerge: apply existing metrics or build a new metric. For the first approach, we can find in the literature some existing metrics to evaluate community detection algorithms [5], [38], [39]. Among these metrics, or scoring functions aimed at evaluating community detection algorithms, we can cite:

Modularity maximization The modularity evaluates the compactness of links *among* communities compared to links *between* communities

Normalized mutual information It requires the ground truth to measure the quality of a community detection algorithm. It may not be applicable in our case.

Others methods to detect overlapping communities

In the second approach, since we will work on real data network, we consider to develop our own metric, which would be well suited to the studied data. This metric can be considered as a quality function. This is an important issue, as mentioned in the reference [5]: it is very important for any testing framework to check for the mutual dependencies between: the benchmark, the quality function used to evaluate partitions, and the clustering algorithm to be tested.

Based on these metrics, the algorithm giving the best results or a new approach combining them will be then adopted.

3) Fraud detection evaluation:

The second step of the project is the development of a fraud detection method based on community detection. After identifying communities, we are using this information as a ground to analyse and classify transactions, and detect fraudulent ones. We suppose that transactions inside of a community are legitimate, while transactions between customers of different communities need further investigation.

We plan on developing an adequate algorithm and compare it to other fraud detection methods and machine learning algorithms. To achieve this goal, the various approaches will be tested using the same dataset. Concerning the evaluation of fraudulent transaction detection, classical metrics, like precision, recall, and weighted F_1 measure seem sound. It is particularly important to use weighted metrics as our data will

contain unbalanced classes: fraudulent transaction represents largely less than 1% of transactions.

IV. Conclusion

In this paper, we have presented our motivations and methodology to study social networks of mobile money transactions gathered by a telecom operator. Social networks are one of the most natural representation of any relational field, providing tools to the formulation of complex relations.

Our study will address the problem of detecting communities within social networks, and we aim to have an comparison of several community detection algorithm. Thereafter, we plan on developing an algorithm able to identify fraudulent actors, base on detected communities. Even though most experiment will be conducted on simulated data, we expect this Phd thesis to give effective ways to address real financial data using social networks and offering a solution to fraud detection.

References

[1] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, Nov. 1994.

[2] A.-L. Barabási and M. Pósfai, *Network Science*. Cambridge: Cambridge University Press, Jul. 2016.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4, pp. 175 – 308, 2006.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences, Section: Physical Sciences*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[5] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, Feb. 2010.

[6] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences, Section: Physical Sciences*, vol. 101, no. 9, pp. 2658–2663, Mar. 2004.

[7] M. E. J. Newman, "Detecting community structure in networks," *The European Physical Journal B*, vol. 38, no. 2, pp. 321–330, Mar. 2004.

[8] S. Moon, J.-G. Lee, M. Kang, M. Choy, and J.-w. Lee, "Parallel community detection on large graphs with MapReduce and GraphChi," *Data & Knowledge Engineering*, vol. 104, pp. 17–31, Jul. 2016.

[9] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066133, Jun. 2004.

[10] C. Wang, C. Hao, and X. Guan, "Hierarchical and overlapping social circle identification in ego networks based on link clustering," *Neurocomputing*, vol. 381, pp. 322–335, Mar. 2020.

[11] P. Wadhwa and M. P. S. Bhatia, "Social networks analysis: trends, techniques and future prospects," in *Fourth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom2012)*. IET Digital Library, jan 2012, pp. 1–6.

[12] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, p. 066111, Dec 2004.

[13] I. X. Y. Leung, P. Hui, P. Liò, and J. Crowcroft, "Towards real-time community detection in large networks," *Physical Review E*, vol. 79, no. 6, p. 066107, Jun. 2009.

[14] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, "Modularity from fluctuations in random graphs and complex networks," *Physical Review E*, vol. 70, no. 2, p. 025101, Aug 2004.

[15] S. Boettcher and A. G. Percus, "Optimization with extremal dynamics," *Physical Review Letters*, vol. 86, no. 23, p. 5211–5214, Jun 2001.

[16] A. Karatas and S. Sahin, "A comparative study of modularity-based community detection methods for online social networks," *CEUR Workshop Proceedings, Proceedings of the 12th Turkish National Software Engineering Symposium*, p. 12, Sept 2018.

[17] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[18] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning Sparse Matrices with Eigenvectors of Graphs," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 3, pp. 430–452, Jul. 1990.

[19] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[20] S. T. Barnard and H. D. Simon, "Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems," *Concurrency: Practice and Experience*, vol. 6, no. 2, pp. 101–117, 1994.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[22] S. White and P. Smyth, "A Spectral Clustering Approach To Finding Communities in Graphs," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, ser. Proceedings. Newport Beach, CA, USA: Society for Industrial and Applied Mathematics, Apr 2005, pp. 274–285.

[23] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, Sep. 2006.

[24] D. Shah and T. Zaman, "Community detection in networks: The leader-follower algorithm," *Computing Research Repository (CoRR)*, vol. abs/1011.0774, no. 2, 2010.

[25] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, Sep. 2007.

[26] R. Kanawati, "Détection de communautés dans les grands graphes d'interactions (multiplexes) : état de l'art," Laboratoire d'Informatique de Paris-Nord (LIPN), Tech. Rep., Nov. 2013.

[27] L. Zong-Wen, L. Jian-Ping, Y. Fan, and A. Petropulu, "Detecting community structure using label propagation with consensus weight in complex network," *Chinese Physics B*, vol. 23, no. 9, p. 098902, 2014.

[28] X.-K. Zhang, J. Ren, C. Song, J. Jia, and Q. Zhang, "Label propagation algorithm for community detection based on node importance and label influence," *Physics Letters A*, vol. 381, no. 33, pp. 2691–2698, 2017.

[29] K. Berahmand and A. Bouyer, "LP-LPA: A link influence-based label propagation algorithm for discovering community structures in networks," *International Journal of Modern Physics B*, vol. 32, no. 06, pp. 1 850 062–10, Mar. 2018.

[30] Y. Yamaguchi and K. Hayashi, "When Does Label Propagation Fail? A View from a Network Generative Model," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug 2017, pp. 3224–3230.

[31] N. Omar, I. Mohamed, Z. Mohd-Sanusi, and H. Prabowo, *Understanding Social Network Analysis (SNA) in fraud detection*. CRC Press, 2014, pp. 543–548.

[32] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '03. Washington, D.C.: Association for Computing Machinery, Aug 2003, pp. 631–636.

[33] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos, "Relevance search and anomaly detection in bipartite graphs," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 48–55, Dec. 2005.

[34] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos, "Netprobe: a fast and scalable system for fraud detection in online auction networks," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. Banff, Alberta, Canada: Association for Computing Machinery, May 2007, pp. 201–210.

[35] L. Šubelj, Š. Furlan, and M. Bajec, "An expert system for detecting automobile insurance fraud using social network analysis," *Expert Systems with Applications*, vol. 38, no. 1, pp. 1039–1052, Jan. 2011.

[36] V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens, "Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decision Support Systems*, vol. 75, pp. 38 – 48, Jul. 2015.

[37] E. Lopez-Rojas, A. Elmir, and S. Axelsson, "Paysim : A financial mobile money simulator for fraud detection," in *28th European Modeling and Simulation Symposium, EMSS*. Dime University of Genoa, Sept 2016, pp. 249–255.

[38] D. Delling, M. Gaertler, R. Görke, Z. Nikoloski, and D. Wagner, "How to evaluate clustering techniques," Tech. report, Universität Karlsruhe, Germany, 2007.

[39] R. Mittal and M. P. S. Bhatia, "Classification and Comparative Evaluation of Community Detection Algorithms," *Archives of Computational Methods in Engineering*, Apr. 2020.